

debian

Debian-Referenz

Osamu Aoki

Copyright © 2013-2026 Osamu Aoki

Diese Debian-Referenz (version 2.153) (2026-11-29 05:38:00 UTC) wurde geschrieben, um für die Zeit nach der Installation einen groben Überblick über das Debian-System in Form eines Benutzerhandbuchs zu bieten. Sie behandelt viele Aspekte der Systemadministration mittels Shell-Befehlsbeispielen für einfache Benutzer.

Inhaltsverzeichnis

1 GNU/Linux-Lehrstunde 1

1.1 Grundlagen für die Konsole . . . 1

1.1.1 Die Shell-Eingabeaufforderung (Shell-Prompt) . . . 1

1.1.2 Die Shell-Eingabeaufforderung auf einem GUI-System . . . 2

1.1.3 Das root-Benutzerkonto . . . 2

1.1.4 Die root-shell-Eingabeaufforderung . . . 3

1.1.5 Systemadministrations-Werkzeuge mit grafischer Oberfläche (GUI) . . . 3

1.1.6 Virtuelle Konsolen . . . 3

1.1.7 Wie Sie die Eingabeaufforderung wieder verlassen . . . 4

1.1.8 Wie Sie das System herunterfahren . . . 4

1.1.9 Eine Konsole wiederherstellen . . . 4

1.1.10 Zusätzliche Paketempfehlungen für Neulinge . . . 4

1.1.11 Ein zusätzliches Benutzerkonto . . . 5

1.1.12 sudo-Konfiguration . . . 5

1.1.13 Zeit zum Spielen . . . 6

1.2 Unix-ähnliches Dateisystem . . . 6

1.2.1 Unix-Dateigrundlagen . . . 7

1.2.2 Dateisystem-Internas . . . 8

1.2.3 Dateisystem-Berechtigungen . . . 8

1.2.4 Steuerung der Berechtigungen für neu erzeugte Dateien: umask . . . 11

1.2.5 Berechtigungen für Gruppen von Benutzern (group) . . . 12

1.2.6 Zeitstempel . . . 13

1.2.7 Links . . . 14

1.2.8 Benannte Pipes (FIFOs) . . . 15

1.2.9 Sockets . . . 16

1.2.10 Gerätedateien . . . 16

1.2.11 Spezielle Gerätedateien . . . 17

1.2.12 procfs und sysfs . . . 18

1.2.13 tmpfs . . . 18

1.3 Midnight Commander (MC) . . . 18

1.3.1 Anpassen des MC . . . 19
1.3.2 Starten von MC . . . 19
1.3.3 Dateimanager in MC . . . 19
1.3.4 Befehlszeilentricks in MC . . . 20
1.3.5 Der interne Texteditor in MC . . . 20
1.3.6 Der interne Dateibetrachter in MC . . . 20
1.3.7 Autostart-Funktionalitäten von MC . . . 21
1.3.8 Virtuelles Dateisystem von MC . . . 21
1.4 Die grundlegende Unix-ähnliche Arbeitsumgebung . . . 21
1.4.1 Die Login-Shell . . . 21
1.4.2 Anpassen der bash . . . 21
1.4.3 Spezielle Tastendrücke . . . 23
1.4.4 Mausoperationen . . . 23
1.4.5 Der Pager . . . 24
1.4.6 Der Texteditor . . . 24
1.4.7 Einen Standard-Texteditor einstellen . . . 25
1.4.8 Verwenden von vim . . . 25
1.4.9 Aufzeichnen der Shell-Aktivitäten . . . 25
1.4.10 Grundlegende Unix-Befehle . . . 27
1.5 Der einfache Shell-Befehl . . . 29
1.5.1 Befehlsausführung und Umgebungsvariablen . . . 29
1.5.2 Die ”$LANG”-Variable . . . 29
1.5.3 Die ”$PATH”-Variable . . . 31
1.5.4 Die ”$HOME”-Variable . . . 31
1.5.5 Befehlszeilen-Optionen . . . 31
1.5.6 Shell-Glob . . . 32
1.5.7 Rückgabewert eines Befehls . . . 32
1.5.8 Typische Befehlssequenzen und Shell-Weiterleitungen . . . 33
1.5.9 Befehls-Alias . . . 35
1.6 Unix-ähnliche Textverarbeitung . . . 35
1.6.1 Unix-Textverarbeitungswerkzeuge . . . 35
1.6.2 Reguläre Ausdrücke . . . 36
1.6.3 Ersetzungsausdrücke . . . 38
1.6.4 Globale Ersetzungen mit regulären Ausdrücken . . . 39
1.6.5 Extrahieren von Daten aus einer Textdatei-Tabelle . . . 39
1.6.6 Skript-Schnipsel für die Befehlsweiterleitung . . . 41

2 Debian-Paketmanagement 42

2.1 Grundvoraussetzungen für das Debian-Paketmanagement . . . 42
2.1.1 Debian-Paketmanagement . . . 42
2.1.2 Paketkonfiguration . . . 43
2.1.3 Grundsätzliche Vorsichtsmaßnahmen . . . 43
2.1.4 Leben mit den ewigen Aktualisierungen . . . 44
2.1.5 Grundlagen über das Debian-Archiv . . . 45
2.1.6 Debian ist zu 100% freie Software . . . 49
2.1.7 Paketabhängigkeiten . . . 51
2.1.8 Die Ereignisabfolge für das Paketmanagement . . . 52
2.1.9 Erste Hilfe bei Paketmanagement-Problemen . . . 53
2.1.10 Wie Sie Debian-Pakete auswählen . . . 53
2.1.11 Wie Sie mit Konflikten umgehen . . . 54
2.2 Grundlegende Paketmanagement-Operationen . . . 54
2.2.1 `apt` contra `apt-get` / `apt-cache` contra `aptitude` . . . 54
2.2.2 Grundlegende Paketmanagement-Operationen auf der Befehlszeile . . . 57
2.2.3 Interaktive Nutzung von aptitude . . . 57
2.2.4 Tastaturkürzel von aptitude . . . 58
2.2.5 Paketansichten in aptitude . . . 58
2.2.6 Optionen für Suchmethoden mit aptitude . . . 59
2.2.7 Aptitudes Regex-Formel . . . 60
2.2.8 Abhängigkeitsauflösung bei aptitude . . . 62
2.2.9 Protokollierung der Paketaktivitäten . . . 62
2.3 Beispiele für aptitude-Operationen . . . 62
2.3.1 Suchen nach interessanten Paketen . . . 62
2.3.2 Auflisten von Paketen mit Regex-Suche auf den Paketnamen . . . 62
2.3.3 Durchsuchen mit der Regex-Suche . . . 62
2.3.4 Entfernte Pakete endgültig löschen . . . 63
2.3.5 Automatisch/manuell-Installationsstatus bereinigen . . . 63
2.3.6 Systemweite Hochrüstung . . . 64
2.4 Erweiterte Paketmanagement-Operationen . . . 65
2.4.1 Erweiterte Paketmanagement-Operationen auf der Befehlszeile . . . 65
2.4.2 Verifizierung von installierten Paketdateien . . . 67
2.4.3 Absicherungen für den Fall von Paketproblemen . . . 67
2.4.4 Durchsuchen der Paket-Metadaten . . . 67
2.5 Internas des Debian-Paketmanagements . . . 68
2.5.1 Archiv-Metadaten . . . 68
2.5.2 ”Release”-Datei im Wurzelverzeichnis und Authentizität . . . 68
2.5.3 ”Release”-Dateien im Archivverzeichnis . . . 69

2.5.4 Empfangen der Metadaten für ein Paket . . . 70
2.5.5 Der Paketstatus für APT . . . 70
2.5.6 Der Paketstatus von aptitude . . . 71
2.5.7 Lokale Kopien der empfangenen Pakete . . . 71
2.5.8 Debian-Paketdateinamen . . . 71
2.5.9 Der dpkg-Befehl . . . 72
2.5.10 Der update-alternatives-Befehl . . . 73
2.5.11 Der dpkg-statoverride-Befehl . . . 73
2.5.12 Der dpkg-divert-Befehl . . . 73
2.6 Wiederherstellung eines beschädigten Systems . . . 74
2.6.1 Inkompatibilität mit alter Benutzerkonfiguration . . . 74
2.6.2 Fehler beim Zwischenspeichern der Paketdaten . . . 74
2.6.3 Systemrettung mit dem dpkg-Befehl . . . 74
2.6.4 Fehlgeschlagene Installation aufgrund von fehlenden Abhängigkeiten . . . 75
2.6.5 Mehrere Pakete mit überlappenden Dateien . . . 75
2.6.6 Behebung von Problemen aufgrund von beschädigtem Paketskript . . . 75
2.6.7 Paketauswahldaten wiederherstellen . . . 76
2.7 Tipps für das Paketmanagement . . . 76
2.7.1 Wer hat das Paket hochgeladen? . . . 76
2.7.2 Die Download-Bandbreite für APT einschränken . . . 77
2.7.3 Automatisches Herunterladen und Aktualisieren von Paketen . . . 77
2.7.4 Aktualisierungen und Backports . . . 77
2.7.5 Externe Paketarchive . . . 78
2.7.6 Pakete aus verschiedenen Paketarchiven ohne **apt-pinning** . . . 78
2.7.7 Die Installationskandidat-Version anpassen mit **apt-pinning** . . . 79
2.7.8 Über ”Recommends” installierte Pakete blockieren . . . 81
2.7.9 Nutzen von `Testing` mit einigen Paketen aus `Unstable` . . . 81
2.7.10 Nutzen von `Unstable` mit einigen Paketen aus `Experimental` . . . 83
2.7.11 Downgrade im Notfall . . . 83
2.7.12 Das equivs-Paket . . . 84
2.7.13 Ein Paket auf das Stable-System portieren . . . 84
2.7.14 Proxy-Server für APT . . . 85
2.7.15 Weitere Lektüre zum Paketmanagement . . . 85

3 Die Systeminitialisierung 87
3.1 Ein Überblick über den Bootstrap-Prozess . . . 87
3.1.1 Stufe 1: das UEFI . . . 88
3.1.2 Stufe 2: der Bootloader . . . 88
3.1.3 Stufe 3: das Mini-Debian-System . . . 90
3.1.4 Stufe 4: das normale Debian-System . . . 90
3.2 Das Rettungssystem . . . 91
3.2.1 GRUB UEFI-Rettungssystem auf USB . . . 92
3.2.2 Linux Live-Rettungssystem auf USB . . . 92
3.2.3 Linux Live-Rettungssystem von GRUB . . . 93
3.3 Systemd . . . 93
3.3.1 Systemd-Init . . . 93
3.3.2 Systemd-Login . . . 95
3.4 Die Kernel-Meldungen . . . 95
3.5 Die Systemmeldungen . . . 95
3.6 Systemmanagement . . . 96
3.7 Weitere Systemüberwachungs-Werkzeuge . . . 96
3.8 Systemkonfiguration . . . 96
3.8.1 Der Rechnername . . . 96
3.8.2 Das Dateisystem . . . 98
3.8.3 Initialisierung der Netzwerkschnittstellen . . . 98
3.8.4 Initialisierung eines Cloud-Systems . . . 98
3.8.5 Anpassungsbeispiel zur Optimierung des sshd-Dienstes . . . 98
3.9 Das udev-System . . . 99
3.10 Die Kernel-Modul-Initialisierung . . . 100

4 Authentifizierung und Zugriffskontrolle 101
4.1 Normale Unix-Authentifizierung . . . 101
4.2 Verwalten von Konten- und Passwortinformationen . . . 103
4.3 Ein gutes Passwort . . . 103
4.4 Verschlüsselte Passwörter erstellen . . . 104
4.5 PAM und NSS . . . 104
4.5.1 Konfigurationsdateien, auf die PAM und NSS zugreifen . . . 105
4.5.2 Modernes zentralisiertes Systemmanagement . . . 105
4.5.3 ”Warum unterstützt GNU su nicht die wheel-Gruppe” . . . 106
4.5.4 Schärfere Passwortregeln . . . 106
4.6 Sicherheit der Authentifizierung . . . 107
4.6.1 Sicheres Passwort im Internet . . . 107
4.6.2 Secure Shell (sichere Shell) . . . 108

4.6.3 Zusätzliche Sicherheitsmaßnahmen für das Internet . . . 108
4.6.4 Sichern des root-Passworts . . . 108
4.7 Andere Möglichkeiten zur Zugriffskontrolle . . . 109
4.7.1 Zugriffssteuerungslisten (Access control lists/ACLs) . . . 109
4.7.2 sudo . . . 110
4.7.3 PolicyKit . . . 110
4.7.4 Den Zugriff auf einige Server-Dienste einschränken . . . 111
4.7.5 Linux Sicherheits-Funktionalitäten . . . 111

5 Netzwerkkonfiguration 113
5.1 Die elementare Netzwerkinfrastruktur . . . 113
5.1.1 Die Auflösung des Rechnernamens . . . 113
5.1.2 Der Netzwerkschnittstellenname . . . 115
5.1.3 Der Netzwerkadressbereich für das LAN . . . 116
5.1.4 Unterstützung für Netzwerkgeräte . . . 116
5.2 Moderne Netzwerkkonfiguration für Arbeitsplatzsysteme . . . 116
5.2.1 Grafische Netzwerkkonfigurations-Werkzeuge . . . 117
5.3 Moderne Netzwerkkonfiguration ohne grafische Oberfläche . . . 117
5.4 Moderne Netzwerkkonfiguration für Cloud-Systeme . . . 118
5.4.1 Moderne Netzwerkkonfiguration für Cloud-Systeme mit DHCP . . . 118
5.4.2 Moderne Netzwerkkonfiguration für Cloud-Systeme mit statischer IP-Adresse . . . 118
5.4.3 Moderne Netzwerkkonfiguration für Cloud-Systeme mit NetworkManager . . . 118
5.5 Netzwerkkonfiguration auf unterster Ebene . . . 119
5.5.1 iproute2-Befehle . . . 119
5.5.2 Sichere Basis-Netzwerkoperationen . . . 119
5.6 Netzwerkoptimierung . . . 119
5.6.1 Die optimale MTU finden . . . 119
5.6.2 WAN-TCP-Optimierung . . . 122
5.7 Die Netfilter-Infrastruktur . . . 122

6 Netzwerkapplikationen 124
6.1 Webbrowser . . . 124
6.1.1 Fälschen der User-Agent-Angabe . . . 125
6.1.2 Browser-Erweiterung . . . 125
6.2 Das Mail-System . . . 125
6.2.1 Grundlagen des E-Mail-Systems . . . 125
6.2.2 Einschränkungen moderner E-Mail-Dienste . . . 126
6.2.3 Historische Erwartungen an E-Mail-Dienste . . . 127
6.2.4 Mail Transfer Agent (MTA) . . . 127

6.2.4.1 Die Konfiguration von exim4 . . . 127
6.2.4.2 Die Konfiguration von Postfix mit SASL . . . 129
6.2.4.3 Die Mail-Adress-Konfiguration . . . 130
6.2.4.4 Grundlegende MTA-Operationen . . . 131
6.3 Der Server für Fernzugriff (SSH) und Hilfsprogramme . . . 131
6.3.1 Grundlagen von SSH . . . 133
6.3.2 Benutzername auf dem fernen Rechner . . . 134
6.3.3 Verbindungen ohne Passwörter für die ferne Seite . . . 134
6.3.4 Der Umgang mit fremden SSH-Clients . . . 134
6.3.5 Einrichten von ssh-agent . . . 134
6.3.6 Eine Mail versenden auf einem fernen Rechner . . . 135
6.3.7 Portweiterleitung für SMTP-/POP3-Tunnelung . . . 135
6.3.8 Wie Sie das ferne System über SSH herunterfahren . . . 135
6.3.9 Fehlersuche bei SSH . . . 135
6.4 Der Print-Server und Hilfsprogramme . . . 136
6.5 Weitere Netzwerkanwendungs-Server . . . 136
6.6 Weitere Netzwerkanwendungs-Clients . . . 137
6.7 Diagnose von System-Daemons . . . 137

7 GUI-System 139
7.1 GUI-Arbeitsplatzumgebung . . . 139
7.2 GUI-Kommunikationsprotokoll . . . 140
7.3 GUI-Infrastruktur . . . 141
7.4 GUI-Anwendungen . . . 142
7.5 Benutzerverzeichnisse . . . 142
7.6 Schriften . . . 142
7.6.1 Basis-Schriftarten . . . 142
7.6.2 Schriftenrasterung . . . 145
7.7 Sandbox . . . 146
7.8 Arbeitsplatz-Fernzugriff (Remote Desktop) . . . 147
7.9 X-Server-Verbindungen . . . 147
7.9.1 Lokale Verbindung zum X-Server . . . 147
7.9.2 Fernverbindung auf den X-Server . . . 148
7.9.3 Chroot-Verbindung zum X-Server . . . 148
7.10 Zwischenablage . . . 148

8 I18N und L10N 150

8.1 Das Gebietsschema (Locale) . . . 150
8.1.1 Argumentation für UTF-8-Gebietsschemata . . . 150
8.1.2 Die Neukonfiguration des Gebietsschemas (Locale) . . . 151
8.1.3 Dateinamenkodierung . . . 152
8.1.4 Lokalisierte Meldungen und übersetzte Dokumentation . . . 152
8.1.5 Auswirkungen des Gebietsschemas . . . 153
8.2 Die Tastatureingabe . . . 153
8.2.1 Tastatureingabe für die Linux-Konsole und X-Window . . . 154
8.2.2 Tastatureingabe für Wayland . . . 154
8.2.3 Unterstützung für Eingabemethoden mit IBus . . . 155
8.2.4 Unterstützung für Eingabemethoden mit Fcitx . . . 155
8.2.5 Ein Beispiel für Japanisch . . . 157
8.3 Die Bildschirmausgabe . . . 157
8.3.1 Die Terminal-Konfiguration . . . 158
8.3.2 Unbekannte Zeichenbreite bei ostasiatischen Zeichen . . . 158

9 Systemtipps 159

9.1 Tipps für die Konsole . . . 159
9.1.1 Shell-Aktivitäten sauber aufzeichnen . . . 159
9.1.2 Das Programm screen . . . 160
9.1.3 Durch Verzeichnisse navigieren . . . 161
9.1.4 Readline-Wrapper . . . 161
9.1.5 Durchsuchen eines Quellcode-Baums . . . 162
9.2 Anpassen von vim . . . 162
9.2.1 Anpassen von vim mit internen Funktionalitäten . . . 162
9.2.2 Anpassen von vim mit externen Paketen . . . 164
9.3 Datenaufzeichnung und -darstellung . . . 165
9.3.1 Der log-Daemon . . . 165
9.3.2 Analyseprogramme für Logdateien . . . 166
9.3.3 Angepasste Anzeige von Textdaten . . . 166
9.3.4 Angepasste Anzeige von Datum und Uhrzeit . . . 166
9.3.5 Farbige Shell-Ausgabe . . . 167
9.3.6 Farbige Befehle . . . 167
9.3.7 Aufzeichnen von Editor-Aktivitäten für komplexe Wiederholungen . . . 168
9.3.8 Die Bildschirmanzeige einer X-Anwendung aufzeichnen . . . 168
9.3.9 Aufzeichnen von Änderungen in Konfigurationsdateien . . . 169
9.4 Überwachen, Steuern und Starten von Programmaktivitäten . . . 169
9.4.1 Zeitmessung für einen Prozess . . . 170

9.4.2 Die Ablaufpriorität . . . 170
9.4.3 Der Befehl ps . . . 170
9.4.4 Der Befehl top . . . 170
9.4.5 Dateien auflisten, die von einem Prozess geöffnet wurden . . . 171
9.4.6 Programmaktivitäten verfolgen . . . 171
9.4.7 Identifikation von Prozessen, die Dateien oder Sockets verwenden . . . 171
9.4.8 Einen Befehl mit festem Intervall wiederholt ausführen . . . 171
9.4.9 Einen Befehl wiederholt mit einer Schleife über verschiedene Dateien ausführen . . . 172
9.4.10 Ein Programm aus der grafischen Oberfläche heraus starten . . . 172
9.4.11 Anpassen des zu startenden Programms . . . 173
9.4.12 Einen Prozess beenden (kill) . . . 174
9.4.13 Einmalige Aufgaben planen . . . 174
9.4.14 Regelmäßige Aufgaben planen . . . 176
9.4.15 Aufgaben ereignisgesteuert planen . . . 176
9.4.16 Die Alt-SysRq-Tastenkombination . . . 176
9.5 Tipps zur Systempflege . . . 177
9.5.1 Wer ist/war im System aktiv? . . . 177
9.5.2 Allen eine Warnung schicken . . . 178
9.5.3 Hardware-Identifikation . . . 178
9.5.4 Hardware-Konfiguration . . . 178
9.5.5 System- und Hardware-Zeit . . . 178
9.5.6 Die Audio-Infrastruktur . . . 180
9.5.7 Deaktivieren des Bildschirmschoners . . . 180
9.5.8 Ausschalten von Pieptönen . . . 181
9.5.9 Arbeitsspeichernutzung . . . 181
9.5.10 Systemsicherheits- und Integritätsüberprüfung . . . 181
9.6 Tipps zur Speicherung von Daten . . . 182
9.6.1 Verwendung des Plattenplatzes . . . 183
9.6.2 Konfiguration der Festplattenpartitionen . . . 183
9.6.3 Zugriff auf Partitionen über die UUID-Kennung . . . 184
9.6.4 LVM2 . . . 184
9.6.5 Konfiguration von Dateisystemen . . . 185
9.6.6 Dateisystemerzeugung und Integritätsüberprüfung . . . 186
9.6.7 Optimierung von Dateisystemen über mount-Optionen . . . 186
9.6.8 Optimierung von Dateisystemen über den Superblock . . . 186
9.6.9 Optimierung der Festplatte . . . 187
9.6.10 Optimierung von Solid State Disks . . . 187
9.6.11 SMART verwenden, um Festplattenausfälle vorherzusehen . . . 187
9.6.12 Angeben eines Verzeichnisses für temporäre Dateien über $TMPDIR . . . 188

9.6.13 Vergrößerung des nutzbaren Speicherplatzes mittels LVM . . . 188
9.6.14 Vergrößerung des nutzbaren Speicherplatzes über das Einbinden anderer Partitionen . . . 188
9.6.15 Vergrößerung des nutzbaren Speicherplatzes, indem ein anderes Verzeichnis mit ”mount --bind” eingebunden wird . . . 188
9.6.16 Vergrößerung des nutzbaren Speicherplatzes, indem ein anderes Verzeichnis mit ”Overlay-mounting” eingebunden wird . . . 189
9.6.17 Vergrößerung des nutzbaren Speicherplatzes über einen symbolischen Link . . . 189
9.7 Das Festplatten-Abbild . . . 189
9.7.1 Erzeugung der Festplatten-Abbild-Datei . . . 189
9.7.2 Direkt auf eine Festplatte schreiben . . . 190
9.7.3 Einbinden der Festplatten-Abbild-Datei . . . 190
9.7.4 Eine Festplatten-Abbild-Datei bereinigen . . . 192
9.7.5 Eine leere Abbild-Datei erstellen . . . 192
9.7.6 Erstellen einer ISO9660-Abbild-Datei . . . 193
9.7.7 Direkt auf die CD/DVD-R/RW schreiben . . . 194
9.7.8 Einbinden einer ISO9660-Abbild-Datei . . . 194
9.8 Binärdaten . . . 194
9.8.1 Betrachten und Bearbeiten von Binärdaten . . . 194
9.8.2 Dateien bearbeiten, ohne das Dateisystem einzuhängen . . . 195
9.8.3 Datenredundanz . . . 195
9.8.4 Datenwiederherstellung und forensische Analyse . . . 195
9.8.5 Aufsplitten einer großen in mehrere kleine Dateien . . . 196
9.8.6 Leeren von Dateiinhalten . . . 196
9.8.7 Dummy-Dateien . . . 196
9.8.8 Eine vollständige Festplatte löschen . . . 197
9.8.9 Einen ungenutzten Bereich einer Festplatte löschen . . . 197
9.8.10 Wiederherstellen von gelöschten, aber noch geöffneten Dateien . . . 198
9.8.11 Alle harten Links suchen . . . 198
9.8.12 Unsichtbarer Verbrauch von Festplattenplatz . . . 199
9.9 Tipps zur Datenverschlüsselung . . . 199
9.9.1 Verschlüsselung von Wechseldatenträgern mit dm-crypt/LUKS . . . 199
9.9.2 Einbinden verschlüsselter Laufwerke mit dm-crypt/LUKS . . . 200
9.10 Der Kernel . . . 200
9.10.1 Kernel-Parameter . . . 200
9.10.2 Kernel-Header . . . 201
9.10.3 Kompilieren des Kernels und dazugehöriger Module . . . 201
9.10.4 Kompilieren des Kernel-Quellcodes: Empfehlung des Debian-Kernel-Teams . . . 202
9.10.5 Hardware-Treiber und Firmware . . . 202
9.11 Virtualisierte Systeme . . . 203

9.11.1 Virtualisierungs- und Emulationswerkzeuge . . . 203
9.11.2 Arbeitsablauf bei Virtualisierung . . . 204
9.11.3 Einbinden des virtuellen Festplatten-Images . . . 206
9.11.4 Chroot-System . . . 206
9.11.5 System mit mehreren Arbeitsplatzumgebungen . . . 207

10 Datenmanagement 208

10.1 Austauschen, kopieren und archivieren von Dateien . . . 208
10.1.1 Archivierungs- und Kompressionswerkzeuge . . . 209
10.1.2 Kopier- und Synchronisationswerkzeuge . . . 209
10.1.3 Aufrufe für Archivierungsoperationen . . . 211
10.1.4 Aufrufe für Kopieroperationen . . . 211
10.1.5 Aufrufe für die Auswahl von Dateien . . . 212
10.1.6 Archivierungsmedien . . . 213
10.1.7 Wechseldatenträger . . . 214
10.1.8 Dateisystemauswahl für den Datenaustausch . . . 215
10.1.9 Datenaustausch über das Netzwerk . . . 217
10.2 Datensicherung und -wiederherstellung . . . 217
10.2.1 Richtlinien für Datensicherung und -wiederherstellung . . . 218
10.2.2 Programmsammlungen für Datensicherungsaufgaben . . . 219
10.2.3 Backup-Tipps . . . 219
10.2.3.1 Grafische Backup-Software . . . 221
10.2.3.2 Automatisches Backup, durch ein mount-Ereignis ausgelöst . . . 221
10.2.3.3 Automatisches Backup, durch ein timer-Ereignis ausgelöst . . . 222
10.3 Datensicherheits-Infrastruktur . . . 223
10.3.1 Schlüsselverwaltung für GnuPG . . . 223
10.3.2 Verwendung von GnuPG mit Dateien . . . 225
10.3.3 Verwendung von GnuPG mit Mutt . . . 225
10.3.4 Verwendung von GnuPG mit Vim . . . 225
10.3.5 Die MD5-Prüfsumme . . . 227
10.3.6 Passwort-Schlüsselbund . . . 227
10.4 Werkzeuge zur Quellcode-Zusammenführung (merge) . . . 227
10.4.1 Unterschiede für Quelldateien extrahieren . . . 227
10.4.2 Aktualisierungen für Quelldateien zusammenführen . . . 229
10.4.3 Interaktives Zusammenführen (merge) . . . 229
10.5 Git . . . 229
10.5.1 Konfiguration eines Git-Clients . . . 229
10.5.2 Grundlegende Git-Befehle . . . 230
10.5.3 Git-Tipps . . . 231
10.5.4 Weitere Referenzen zu Git . . . 233
10.5.5 Andere Versionskontrollsysteme . . . 233

11 Datenkonvertierung 234
11.1 Werkzeuge für Textkonvertierung . . . 234
11.1.1 Konvertieren einer Textdatei mit iconv . . . 235
11.1.2 Prüfen mit iconv, ob eine Datei UTF-8-kodiert ist . . . 236
11.1.3 Dateinamen konvertieren mit iconv . . . 236
11.1.4 EOL-Konvertierung . . . 237
11.1.5 TAB-Konvertierung . . . 237
11.1.6 Editoren mit automatischer Konvertierung . . . 237
11.1.7 Extrahieren von reinem Text . . . 238
11.1.8 Hervorheben und Formatieren von reinen Textdaten . . . 238
11.2 XML-Daten . . . 238
11.2.1 Grundlegende Hinweise für XML . . . 240
11.2.2 XML-Verarbeitung . . . 241
11.2.3 Extrahierung von XML-Daten . . . 241
11.2.4 XML Lint . . . 242
11.3 Textsatz . . . 242
11.3.1 roff-Textsatz . . . 242
11.3.2 TeX/LaTeX . . . 243
11.3.3 Schöner Ausdruck einer Handbuchseite . . . 244
11.3.4 Erstellen einer Handbuchseite . . . 244
11.4 Druckfähige Daten . . . 244
11.4.1 Ghostscript . . . 245
11.4.2 Zwei PS- oder PDF-Dateien zusammenführen . . . 245
11.4.3 Werkzeuge für druckfähige Daten . . . 245
11.4.4 Drucken mit CUPS . . . 245
11.5 Konvertierung von Mail-Daten . . . 247
11.5.1 Grundlagen zu Mail-Daten . . . 247
11.6 Werkzeuge für Grafikdaten . . . 248
11.6.1 Werkzeuge für Grafikdaten (Metapaket) . . . 248
11.6.2 Werkzeuge für Grafikdaten (GUI) . . . 249
11.6.3 Werkzeuge für Grafikdaten (Konsolen-Befehle) . . . 249
11.7 Verschiedene Datenkonvertierungen . . . 249

12 Programmierung 252
12.1 Das Shell-Skript . . . 252
12.1.1 POSIX-Shell-Kompatibilität . . . 253
12.1.2 Shell-Parameter . . . 253
12.1.3 Bedingte Ausdrücke in der Shell . . . 255
12.1.4 Shell-Schleifen . . . 255

12.1.5 Shell-Umgebungsvariablen . . . 256
12.1.6 Befehlsabfolge auf der Shell . . . 256
12.1.7 Hilfsprogramme für Shell-Skripte . . . 257
12.2 Skriptverarbeitung in Interpreter-Sprachen . . . 259
12.2.1 Fehlersuche im Code für Interpreter-Sprachen . . . 259
12.2.2 Grafisches GUI-Programm und Shell-Skripte . . . 259
12.2.3 Eigene Aktionen im Dateimanager . . . 260
12.2.4 Verrücktes bei kurzen Perl-Skripten . . . 260
12.3 Programmieren in kompilierten Sprachen . . . 261
12.3.1 C . . . 261
12.3.2 Ein einfaches C-Programm (gcc) . . . 262
12.3.3 Flex - ein besseres Lex . . . 262
12.3.4 Bison - ein besseres Yacc . . . 263
12.4 Werkzeuge zur statischen Code-Analyse . . . 264
12.5 Fehlersuche (Debugging) . . . 266
12.5.1 Grundlegende Ausführung von gdb . . . 266
12.5.2 Fehlersuche (Debugging) in einem Debian-Paket . . . 266
12.5.3 Gewinnen von Backtrace-Informationen . . . 267
12.5.4 Erweiterte gdb-Befehle . . . 268
12.5.5 Überprüfen der Abhängigkeiten von Bibliotheken . . . 268
12.5.6 Werkzeuge zur dynamischen Aufrufverfolgung . . . 268
12.5.7 Fehleranalyse bei X-Fehlern . . . 269
12.5.8 Werkzeuge zur Erkennung von Speicherlecks . . . 269
12.5.9 Disassemblieren von Binärdateien . . . 269
12.6 Bauwerkzeuge . . . 269
12.6.1 Make . . . 270
12.6.2 Autotools . . . 270
12.6.2.1 Kompilieren und Installieren eines Programms . . . 271
12.6.2.2 Deinstallation eines Programms . . . 271
12.6.3 Meson . . . 271
12.7 Web . . . 272
12.8 Die Quellcode-Übersetzung . . . 272
12.9 Erstellen von Debian-Paketen . . . 272

A Anhang **274**

A.1 Das Debian-Labyrinth . . . 274
A.2 Copyright-Vergangenheit . . . 274
A.3 Dokumentenformat . . . 275

Tabellenverzeichnis

1.1 Liste interessanter Textmodus-Programm-Pakete . . . 5
1.2 Liste informativer Dokumentationspakete . . . 5
1.3 Auflistung der Verwendung wichtiger Verzeichnisse . . . 8
1.4 Bedeutungen des ersten Zeichens der "`ls -l`"-Ausgabe . . . 9
1.5 Der numerische Modus für Dateiberechtigungen in chmod(1)-Befehlen . . . 11
1.6 Beispiele für **umask**-Werte . . . 11
1.7 Liste erwähnenswerter systemweiter Gruppen . . . 13
1.8 Liste erwähnenswerter, vom System angebotenen Gruppen zur Ausführung besonderer Befehle . . . 13
1.9 Liste der Arten von Zeitstempeln . . . 13
1.10 Liste spezieller Gerätedateien . . . 17
1.11 Die Tastaturbefehle von MC . . . 19
1.12 Die Reaktion auf die Enter-Taste in MC . . . 21
1.13 Liste von Shell-Programmen . . . 22
1.14 Liste der Tastaturbefehle für bash . . . 23
1.15 Liste von Mausoperationen und zugehörigen Tastenaktionen auf Debian-Systemen . . . 24
1.16 Liste grundlegender Vim-Tastenkürzel . . . 26
1.17 Liste grundlegender Unix-Befehle . . . 28
1.18 Die drei Teile des locale-Wertes . . . 30
1.19 Liste mit Empfehlungen zum Gebietsschema . . . 30
1.20 Liste der Werte von "`$HOME`" . . . 31
1.21 Shell-Glob-Suchmuster . . . 32
1.22 Befehls-Beendigungs-Codes . . . 33
1.23 Abfolgen von Shell-Befehlen . . . 34
1.24 Vordefinierte Datei-Deskriptoren . . . 34
1.25 Metazeichen für BRE und ERE . . . 37
1.26 Der Ersetzungsausdruck . . . 38
1.27 Liste von Skript-Schnipseln für die Befehlsweiterleitung . . . 41

2.1 Liste von Debians Paketmanagement-Werkzeugen . . . 43
2.2 Liste von Debian-Archiv-Seiten . . . 47

2.3 Liste der Debian-Archiv-Bereiche . . . 47
2.4 Zusammenhanag zwischen Suite und Codename . . . 48
2.5 Liste wichtiger Websites zur Lösung von Paketproblemen . . . 53
2.6 Grundlegende Paketmanagement-Operationen mit apt(8), aptitude(8) und apt-get(8) / apt-cache(8) . . . 56
2.7 Erwähnenswerte Befehlsoptionen für aptitude(8) . . . 57
2.8 Liste der Tastaturkürzel für aptitude . . . 58
2.9 Liste der Ansichten für aptitude . . . 59
2.10 Die Kategorisierung von Standard-Paketansichten . . . 59
2.11 Liste von aptitudes Regex-Formeln . . . 61
2.12 Protokolldateien für Paketaktivitäten . . . 62
2.13 Liste erweiterter Paketmanagement-Operationen . . . 66
2.14 Inhalt der Metadaten des Debian-Archivs . . . 68
2.15 Namensstruktur von Debian-Paketen . . . 71
2.16 In den einzelnen Komponenten von Debian-Paketnamen zu verwendende Zeichen . . . 71
2.17 Erwähnenswerte Dateien, die durch `dpkg` erzeugt werden . . . 72
2.18 Liste erwähnenswerter Pin-Prioritäts-Werte für **apt-pinning** . . . 80
2.19 Liste von Proxy-Hilfsprogrammen speziell für das Debian-Archiv . . . 85

3.1 Liste der Bootloader . . . 88
3.2 Bedeutung der Bestandteile eines Menüeintrags in `/boot/grub/grub.cfg` . . . 89
3.3 Liste von Boot-Hilfsprogrammen für das Debian-System . . . 91
3.4 Liste der Schwellwerte zur Filterung von Kernel-Fehler-Meldungen . . . 95
3.5 Liste typischer `journalctl`-Befehle . . . 96
3.6 Liste typischer `systemctl`-Befehle . . . 97
3.7 Liste weiterer Überwachungsbefehle unter `systemd` . . . 98

4.1 Wichtige Konfigurationsdateien für pam_unix(8) . . . 101
4.2 Inhalt des zweiten Eintrags in "`/etc/passwd`" . . . 102
4.3 Liste von Befehlen zur Verwaltung von Konteninformationen . . . 103
4.4 Liste der Werkzeuge zur Passworterzeugung . . . 104
4.5 Liste von PAM- und NSS-Systemen . . . 105
4.6 Liste von Konfigurationsdateien, auf die PAM und NSS zugreifen . . . 106
4.7 Liste von unsicheren und sicheren Diensten und Ports . . . 107
4.8 Liste von Werkzeugen, die zusätzliche Sicherheitsmaßnahmen ermöglichen . . . 108

5.1 Liste von Werkzeugen zur Netzwerkkonfiguration . . . 114
5.2 Liste der Netzwerkadressbereiche . . . 116
5.3 Gegenüberstellung von `net-tools`- und `iproute2`-Befehlen . . . 119
5.4 Liste von Basis-Netzwerkbefehlen . . . 120
5.5 Liste von Werkzeugen zur Netzwerkoptimierung . . . 120

5.6 Wesentliche Grundregeln für den optimalen MTU-Wert . . . 121
5.7 Liste von Firewall-Werkzeugen . . . 122

6.1 Liste der Webbrowser . . . 124
6.2 Liste der Mail User Agents (MUA) . . . 126
6.3 Liste von Paketen für grundlegende Mail Transfer Agents . . . 128
6.4 Liste von wichtigen Postfix-Handbuchseiten . . . 129
6.5 Liste von Konfigurationsdateien für Mail-Adressen . . . 130
6.6 Liste grundlegender MTA-Operationen . . . 132
6.7 Liste von Servern für Fernzugriff und Hilfsprogrammen . . . 132
6.8 Liste von SSH-Konfigurationsdateien . . . 133
6.9 Liste von Beispielen zum Start einer SSH-Verbindung auf einem Client . . . 133
6.10 Liste freier SSH-Clients für andere Plattformen . . . 134
6.11 Liste von Print-Servern und Hilfsprogrammen . . . 136
6.12 Liste von weiteren Netzwerkanwendungs-Servern . . . 137
6.13 Liste von Netzwerkanwendungs-Clients . . . 138
6.14 Liste populärer RFCs . . . 138

7.1 Liste der Arbeitsplatzumgebungen . . . 139
7.2 Liste erwähnenswerter GUI-Infrastruktur-Pakete . . . 141
7.3 Liste erwähnenswerter GUI-Anwendungen . . . 143
7.4 Liste erwähnenswerter TrueType- und OpenType-Schriftarten . . . 144
7.5 Liste erwähnenswerter Pakete für Schriftumgebungen . . . 145
7.6 Liste erwähnenswerter Pakete für Sandbox-Umgebungen . . . 146
7.7 Liste erwähnenswerter Server für Fernzugriff . . . 147
7.8 Liste der Verbindungsmethoden zum X-Server . . . 147
7.9 Liste von Programmen zum Beeinflussen der Zwischenablage . . . 149

8.1 Liste von IBus- und dessen Plugin-Paketen . . . 155
8.2 Liste von Fcitx5- und dessen Plugin-Paketen . . . 156

9.1 Liste von Programmen für Konsolenaktivitäten . . . 159
9.2 Liste von Tastaturkürzeln für screen . . . 161
9.3 Informationen zur Initialisierung von `vim` . . . 165
9.4 Liste von System-Log-Analyzern . . . 166
9.5 Anzeigebeispiele von Datum und Uhrzeit für den Befehl "`ls -l`" mit **time style value**-Einstellung . . . 167
9.6 Liste von Werkzeugen zur Bildbearbeitung . . . 168
9.7 Liste von Paketen zur Aufzeichnung der Konfigurations-Historie . . . 169
9.8 Liste von Werkzeugen zur Überwachung und Steuerung von Programmaktivitäten . . . 169
9.9 Liste der nice-Werte für die Ablaufpriorität . . . 170

9.10 Liste von ps-Befehlen . . . 170
9.11 Liste von häufig verwendeten Signalen für den kill-Befehl . . . 175
9.12 Liste erwähnenswerter SAK-Befehlstasten . . . 177
9.13 Liste von Werkzeugen zur Hardware-Identifikation . . . 178
9.14 Liste von Werkzeugen zur Hardware-Konfiguration . . . 179
9.15 Liste von Audio-Paketen . . . 180
9.16 Liste von Befehlen zur Deaktivierung des Bildschirmschoners . . . 181
9.17 Liste der gemeldeten Arbeitsspeichergrößen . . . 182
9.18 Liste von Werkzeugen für eine Systemsicherheits- und Integritätsüberprüfung . . . 182
9.19 Liste von Paketen für die Partitionierung . . . 183
9.20 Liste von Paketen für das Dateisystem-Management . . . 185
9.21 Liste von Paketen zum Betrachten und Bearbeiten von Binärdaten . . . 195
9.22 Liste von Paketen zur Bearbeitung von Dateien, ohne die Dateisysteme einzuhängen . . . 195
9.23 Liste von Werkzeugen, um Redundanz für Dateien hinzuzufügen . . . 195
9.24 Liste von Paketen für Datenwiederherstellung und forensische Analysen . . . 196
9.25 Liste von Werkzeugen zur Datenverschlüsselung . . . 199
9.26 Liste von Schlüsselpaketen für die Neukompilierung des Kernels auf einem Debian-System . . . 201
9.27 Liste von Virtualisierungswerkzeugen . . . 205

10.1 Liste von Archivierungs- und Kompressionswerkzeugen . . . 210
10.2 Liste von Kopier- und Synchronisationswerkzeugen . . . 211
10.3 Liste von Dateisystemen für Wechseldatenträger mit typischen Anwendungsszenarien . . . 216
10.4 Liste von Netzwerkdiensten mit typischen Anwendungsszenarien . . . 217
10.5 Liste von Datensicherungsprogrammen . . . 220
10.6 Liste von Werkzeugen für die Datensicherheits-Infrastruktur . . . 223
10.7 Liste von GNU-Privacy-Guard-Befehlen für die Schlüsselverwaltung . . . 224
10.8 Liste der Bedeutungen des Vertrauenscodes . . . 224
10.9 Liste von GNU-Privacy-Guard-Befehlen mit Dateien . . . 226
10.10 Liste von Werkzeugen zur Quellcode-Zusammenführung . . . 228
10.11 Liste von zu Git gehörigen Paketen und Befehlen . . . 230
10.12 Wichtige Git-Befehle . . . 231
10.13 Git-Tipps . . . 232
10.14 Liste anderer Versionskontrollsystem-Werkzeuge . . . 233

11.1 Liste von Textkonvertierungs-Werkzeugen . . . 234
11.2 Liste von Werten für die Zeichenkodierung und deren Verwendung . . . 235
11.3 Liste der EOL-Codes für verschiedene Plattformen . . . 237
11.4 Liste der Befehle zur TAB-Konvertierung aus den Paketen `bsdmainutils` und `coreutils` . . . 237
11.5 Liste von Werkzeugen zum Extrahieren von reinen Textdaten . . . 239

11.6 Liste von Werkzeugen für Hervorhebung/Formatierung von Textdaten . . . 239
11.7 Liste von vordefinierten Entitäten für XML . . . 240
11.8 Liste von XML-Werkzeugen . . . 241
11.9 Liste von DSSSL-Werkzeugen . . . 241
11.10 Liste von Werkzeugen zur Extrahierung von XML-Daten . . . 242
11.11 Liste von XML-Druck-Werkzeugen . . . 242
11.12 Liste von Textsatz-Werkzeugen . . . 243
11.13 Liste von Paketen, die bei der Erstellung einer Handbuchseite helfen . . . 244
11.14 Liste von Ghostscript-PostScript-Interpretern . . . 245
11.15 Liste von Werkzeugen für druckfähige Daten . . . 246
11.16 Liste von Paketen zur Konvertierung von Mail-Daten . . . 247
11.17 Liste von Werkzeugen für Grafikdaten (Metapaket) . . . 248
11.18 Liste von Werkzeugen für Grafikdaten (GUI) . . . 249
11.19 Liste von Werkzeugen für Grafikdaten (Konsolen-Befehle) . . . 250
11.20 Liste verschiedener Werkzeuge zur Datenkonvertierung . . . 251

12.1 Liste typischer Bashisms . . . 253
12.2 Liste von Shell-Parametern . . . 254
12.3 Liste von Parameterauswertungen . . . 254
12.4 Liste von Shell-Parameterersetzungen . . . 254
12.5 Liste von Dateivergleichsoperatoren in bedingten Ausdrücken . . . 255
12.6 Liste von String-Vergleichsoperatoren im bedingten Ausdruck . . . 256
12.7 Liste der Pakete, die kleine Hilfsprogramme für Shell-Skripte enthalten . . . 258
12.8 Liste von Interpreter-betreffenden Paketen . . . 258
12.9 Liste von Dialog-Programmen . . . 260
12.10 Liste von Compiler-betreffenden Paketen . . . 261
12.11 Liste Yacc-kompatibler LALR-Parser-Generatoren . . . 263
12.12 Liste von Werkzeugen für die statische Code-Analyse . . . 265
12.13 Liste von Debugging-Paketen . . . 266
12.14 Liste erweiterter gdb-Befehle . . . 268
12.15 Liste von Werkzeugen zur Erkennung von Speicherlecks . . . 269
12.16 Liste von Paketen mit Bauwerkzeugen . . . 269
12.17 Liste von automatischen make-Variablen . . . 270
12.18 Liste von make-Variablenexpandierungen . . . 270
12.19 Liste von Programmen zur Übersetzung von Quellcode . . . 272

Zusammenfassung

Dieses Buch ist frei; Sie dürfen es unter den Bedingungen der GNU General Public License in jeder Version, die verträglich mit den Debian Richtlinien für Freie Software (DFSG) ist, weiterverteilen und/oder verändern.

Vorwort

Diese Debian-Referenz (Version 2.153) (2026-11-29 05:38:00 UTC) soll für die Zeit nach der Installation einen groben Überblick über das Debian-System in Form eines Benutzerhandbuchs bieten.

Es spricht diejenigen Leser an, die bereit sind, Shell-Skripte zu lernen, aber nicht bereit sind, alle C-Quellen zu lesen, um herauszufinden, wie das GNU/Linux-System genau funktioniert.

Anweisungen zur Installation eines Debian-Systems finden Sie unter:

- Debian GNU/Linux Installationsanleitung für das aktuelle Stable-System
- Debian GNU/Linux Installationsanleitung für das Testing-System

Haftungsausschluss

Jegliche Gewährleistung wird ausgeschlossen. Alle Handelsmarken sind Eigentum ihrer jeweiligen Markeninhaber.

Das Debian-System selbst ist ein bewegliches Ziel. Das macht es schwer, diese Dokumentation aktuell und korrekt zu halten. Obgleich die aktuelle `Testing`-Version des Debian-Systems beim Schreiben dieser Referenz als Basis genutzt wurde, mag einiges vom Inhalt bereits veraltet sein, wenn Sie dies lesen.

Bitte behandeln Sie dieses Dokument als zweitrangige Referenz. Dieses Dokument ersetzt nicht die maßgeblichen Handbücher. Autor und Helfer/Übersetzer übernehmen keine Verantwortung für die Folgen von Fehlern, Auslassungen oder Zweideutigkeiten in diesem Dokument.

Was ist Debian

Das Debian-Projekt ist eine Vereinigung von Einzelpersonen, die es sich zur gemeinsamen Aufgabe gemacht haben, ein freies Betriebssystem zu erstellen. Seine Distribution zeichnet sich durch folgende Aspekte aus:

- Verpflichtung zur Freiheit von Software: Debian-Gesellschaftsvertrag und Debian-Richtlinien für Freie Software (DFSG);
- internet-basierte, verteilte, unbezahlte und freiwillige Leistung: https://www.debian.org;
- eine große Anzahl vorkompilierter, hochqualitativer Software-Pakete;
- Fokus auf Stabilität und Sicherheit mit einfachem Zugang zu Sicherheitsaktualisierungen;
- Fokus auf leichte Aktualisierung auf die neuesten Software-Pakete über das `testing`-Archiv;
- eine große Zahl unterstützter Hardware-Architekturen.

Freie Software in Debian stammt von GNU, Linux, BSD, X, ISC, Apache, Ghostscript, Common Unix Printing System , Samba, GNOME, KDE, Mozilla, LibreOffice, Vim, TeX, LaTeX, DocBook, Perl, Python, Tcl, Java, Ruby, PHP, Berkeley DB, MariaDB, PostgreSQL, SQLite, Exim, Postfix, Mutt, FreeBSD, OpenBSD, Plan 9 und vielen weiteren unabhängigen Freie-Software-Projekten. Debian integriert diese Vielzahl freier Software in ein Ökosystem.

Über dieses Dokument

Leitregeln

Folgende Leitregeln wurden beim Erstellen dieses Dokuments verfolgt:

- liefere einen Überblick und überspringe Sonderfälle (**das Gesamtbild**);
- halte es kurz und einfach (Keep It Short and Simple, **KISS**);
- erfinde das Rad nicht neu (benutze Verweise auf **bestehende Referenzen**);
- Fokus auf nicht-grafische Werkzeuge und Konsolen (benutze **Shell-Beispiele**);
- Sei objektiv (benutze popcon usw.).

Tipp
Ich habe versucht, hierarchische Aspekte und die unteren Ebenen des Systems zu erläutern.

Voraussetzungen



Warnung
Sie können davon ausgehen, auch außerhalb dieser Dokumentation leicht selbst Antworten zu finden. Dieses Dokument liefert Ihnen lediglich effiziente Ausgangssituationen.

Verwenden Sie folgende primäre Informationsquellen, um selbst nach Lösungen suchen:

- die Debian-Website unter https://www.debian.org für grundsätzliche Informationen;
- die Dokumentation im Verzeichnis "`/usr/share/doc/`*`paketname`*";
- die **Handbuchseiten (manpages)** im Unix-Stil: "`dpkg -L` *`paketname`* `|grep '/man/man.*/'`";
- die **Infoseiten** im GNU-Stil: "`dpkg -L` *`paketname`* `|grep '/info/'`";
- die Fehlerdatenbank: https://bugs.debian.org/*paketname*;
- das Debian Wiki unter https://wiki.debian.org/ für sich schnell ändernde und spezielle Themen;
- die Single UNIX Specification von Open Group's Homepage The UNIX System Home Page
- die freie Enzyklopädie von Wikipedia unter https://www.wikipedia.org/.
- Das Debian Administrationshandbuch
- Die HOWTOs von The Linux Documentation Project (TLDP)

Anmerkung
Für detaillierte Dokumentation zu einem speziellen Programm/Paket müssen Sie möglicherweise das entsprechende Dokumentationspaket mit "`-doc`" als Anhang hinter dem Namen installieren.

Konventionen

Dieses Dokument bietet Informationen im folgenden vereinfachten Präsentationsstil mit bash(1) Shell-Befehl-Beispielen:

```
# command-in-root-account
$ command-in-user-account
```

Diese Shell-Prompts unterscheiden sich abhängig vom verwendeten Benutzer und sind deckungsgleich mit dem Setzen von Umgebungsvariablen wie "`PS1='\$'`" und "`PS2=' '`". Diese Werte wurden zwecks Lesbarkeit in diesem Dokument ausgewählt und sind nicht zwingend typisch für ein aktuell installiertes System.

Alle Befehlsbeispiele in dieser Referenz wurden mit der englischen Gebietsschema-Einstellung "`LANG=en_US.UTF8`" ausgeführt. Bitte erwarten Sie nicht, dass Platzhalter wie *`command-in-root-account`* und *`command-in-user-account`* in den Befehlsbeispielen übersetzt sind. Sie sind bewusst unübersetzt, um alle Beispiele aktuell halten zu können.

Anmerkung
Lesen Sie dazu die Bedeutung der Umgebungsvariablen "$PS1" und "$PS2" in bash(1).

Eine vom Systemadministrator geforderte **Aktion** ist in der Imperativ-Form (Befehlsform) geschrieben, z.B. "Enter-Taste drücken nach jedem Eingeben eines Befehls in der Shell".

Spalten wie **Beschreibung** und ähnliche in einer Tabelle können eine **Nominalphrase** im Sinne der Richtlinien für die Paket-Kurzbeschreibungen enthalten, wobei führende Artikel wie "ein" oder "der" weggelassen werden. Sie können alternativ auch eine Infinitivphrase als **Nominalphrase** ohne das "zu" am Anfang enthalten, folgend den Richtlinien für Befehls-Kurzbeschreibungen in Handbuchseiten. Dies mag einigen Leuten lustig erscheinen, ist aber meine beabsichtigte Stilwahl, um diese Dokumentation so einfach wie möglich zu halten. Diese **Nominalphrasen** beginnen nicht mit einem Großbuchstaben und enden nicht mit einem Punkt, gemäß den Richtlinien für Kurzbeschreibungen.

Anmerkung
Korrekte Nomen inklusive Befehlsnamen bleiben von der Groß-/Kleinschreibung her immer gleich, unabhängig von Ihrer Position im Satz.

Ein **Befehlsschnipsel**, der in einem Textabschnitt zitiert wird, ist in Schreibmaschinen-Schriftart zwischen doppelten Anführungszeichen dargestellt, z.B. "`aptitude safe-upgrade`".

Ein **Ausschnitt von Daten** aus einer Konfigurationsdatei, der in einem Textabschnitt zitiert wird, ist in Schreibmaschinen-Schriftart zwischen doppelten Anführungszeichen dargestellt, z.B. "`deb-src`".

Ein **Befehl** wird mit seinem Namen in Schreibmaschinen-Schriftart und optional gefolgt von der Nummer seines Handbuchseiten-Abschnitts in Klammern dargestellt, z.B. bash(1). Geben Sie Folgendes ein, um weitere Informationen zu erhalten:

```
$ man 1 bash
```

Eine **Handbuchseite (manpage)** wird mit ihrem Namen in Schreibmaschinen-Schriftart gefolgt von der Nummer ihres Handbuchseiten-Abschnitts in Klammern dargestellt, z.B. sources.list(5). Geben Sie Folgendes ein, um weitere Informationen zu erhalten:

```
$ man 5 sources.list
```

Eine **Infoseite** wird mit dem Befehlsschnipsel in Schreibmaschinen-Schriftart zwischen doppelten Anführungszeichen dargestellt, z.B. "`info make`". Geben Sie Folgendes ein, um weitere Informationen zu erhalten:

```
$ info make
```

Ein **Dateiname** wird in Schreibmaschinen-Schriftart zwischen doppelten Anführungszeichen dargestellt, z.B. "`/etc/passwd`". Geben Sie Folgendes ein, um z.B. den Inhalt von Konfigurationsdateien anzuzeigen:

```
$ sensible-pager "/etc/passwd"
```

Ein **Verzeichnisname** wird in Schreibmaschinen-Schriftart zwischen doppelten Anführungszeichen dargestellt, z.B. "`/etc/apt/`". Geben Sie Folgendes ein, um den Verzeichnisinhalt anzuzeigen:

```
$ mc "/etc/apt/"
```

Ein **Paketname** wird mit seinem Namen in Schreibmaschinen-Schriftart dargestellt, z.B. `vim`. Geben Sie Folgendes ein, um weitere Informationen zu dem Paket zu erhalten:

```
$ dpkg -L vim
$ apt-cache show vim
$ aptitude show vim
```

Bei **Dokumentation** wird der Ablageort entweder über den Dateinamen in Schreibmaschinen-Schriftart zwischen doppelten Anführungszeichen angezeigt, z.B. "`/usr/share/doc/base-passwd/users-and-groups.txt.gz`" und "`/usr/share/doc/base-passwd/users-and-groups.html`"; oder über seine URL, z.B. https://www.debian.org. Geben Sie Folgendes ein, um die Dokumentation zu lesen:

```
$ zcat "/usr/share/doc/base-passwd/users-and-groups.txt.gz" | sensible-pager
$ sensible-browser "/usr/share/doc/base-passwd/users-and-groups.html"
$ sensible-browser "https://www.debian.org"
```

Eine **Umgebungsvariable** wird mit ihrem Namen und einem führenden "$" in Schreibmaschinen-Schriftart zwischen doppelten Anführungszeichen dargestellt, z.B. "$VAR". Geben Sie Folgendes ein, um ihren derzeitigen Wert zu erhalten:

```
$ echo "$TERM"
```

Der Popularitäts-Wettbewerb (popcon)

Die Daten des Debian Popularitäts-Wettbewerbs (Debian Popularity Contest / popcon) stellen ein objektives Messinstrument für die Beliebtheit eines jeden Pakets dar. Sie wurden am 2026-11-28 09:48:51 UTC heruntergeladen und enthalten die gesamte Menge von 299780 Meldungen über 229395 Binärpakete und 26 Architekturen.

Anmerkung
Bitte beachten Sie, dass das `amd64`-Archiv in `unstable` derzeit nur 78709 Pakete enthält. Die popcon-Daten enthalten auch Berichte von vielen alten Systeminstallationen.

Die popcon-Zahl mit einem vorangestellten "V:" für "votes" wird berechnet über "1000 * (Anzahl der popcon-Berichte, dass das Paket auf dem PC kürzlich ausgeführt wurde) / (Gesamtanzahl der popcon-Berichte)".

Die popcon-Zahl mit einem vorangestellten "I:" für "installs" wird berechnet über "1000 * (Anzahl der popcon-Berichte, dass das Paket auf dem PC installiert ist) / (Gesamtanzahl der popcon-Berichte)".

Anmerkung
Die popcon-Daten sollten nicht als absolutes Maß für die Wichtigkeit des Pakets angesehen werden. Es gibt viele Faktoren, die die Statistik beeinflußen können. Zum Beispiel könnten einige Systeme, die an dem Popularitäts-Wettbewerb teilnehmen, eingebundene Verzeichnisse wie "`/usr/bin`" mit der Option "`noatime`" haben, um die System-Performance zu verbessern, und so wäre "votes" für solche Systeme letztlich deaktiviert.

Die Paketgröße

Die Paketgrößen-Daten werden auch als objektives Maß für jedes Paket angeboten. Sie basieren auf der Angabe "`Installed-Size:`", die vom Befehl "`apt-cache show`" oder "`aptitude show`" angezeigt werden (derzeit von der `amd64`-Architektur für die `Unstable`-Veröffentlichung). Die Größe wird in KiB (Kibibyte = Einheit für 1024 Byte) angegeben.

Anmerkung
Ein Paket mit einer sehr kleinen numerischen Paketgröße kann darauf hindeuten, dass das Paket in der `Unstable`-Veröffentlichung ein Dummy-Paket ist, das andere Pakete mit signifikantem Inhalt über die Abhängigkeiten installiert. Dummy-Pakete ermöglichen sanfte Versionsübergänge oder das Aufsplitten von Paketen in mehrere kleinere Einzelpakete.

Anmerkung
Eine Paketgröße gefolgt von "(*)" zeigt an, dass das Paket in der `Unstable`-Veröffentlichung fehlt und stattdessen die Paketgröße aus der `Experimental`-Veröffentlichung angezeigt wird.

Fehlerbericht gegen dieses Dokument

Falls Sie irgendwelche Fehler in diesem Dokument finden, melden Sie diese gegen das Paket `debian-reference`; Sie können dazu reportbug(1) verwenden. Fügen Sie bitte Korrekturvorschläge für den Fehler bei, als "`diff -u`" gegen den reinen Text oder die Quellen.

Gedächtnisstützen für neue Benutzer

Hier einige Gedächtnisstützen für neue Benutzer:

- Sichern Sie Ihre Daten
  - Lesen Sie dazu Abschnitt 10.2.
- Halten Sie Passwörter und Schlüssel verschlossen
- KISS - halte es kurz und einfach (keep it simple stupid)
  - Machen Sie Ihr System nicht unnötig kompliziert
- Lesen Sie die Protokolldateien Ihres System
  - Der **ERSTE** Fehler ist der, der zählt
- RTFM - lesen Sie das tolle Handbuch (read the fine manual)
- Suchen Sie im Internet, bevor Sie Fragen stellen
- Arbeiten Sie nicht als root, wenn es nicht erforderlich ist
- Spielen Sie nicht mit dem Paketmanagementsystem herum
- Geben Sie keine Befehle ein, die Sie nicht verstehen
- Ändern Sie keine Dateizugriffsrechte
- Verlassen Sie die root-Shell nicht, bevor Sie Ihre Änderungen **GETESTET** haben
- Halten Sie immer ein alternatives Boot-Medium parat (USB-Stick, CD-ROM, …)

Einige Zitate für neue Benutzer

Hier einige interessante Zitate von den Debian-Mailinglisten, die helfen können, neue Benutzer aufzuklären.

- ”Dies ist Unix. Es gibt Ihnen genug Seil, damit Sie sich selbst erhängen können.” --- Miquel van Smoorenburg `<miquels at cistron.nl>`
- ”Unix IST benutzerfreundlich… Es ist nur sehr wählerisch mit der Entscheidung, wer seine Freunde sind.” --- Tollef Fog Heen `<tollef at add.no>`

Auf Wikipedia gibt es den Artikel ”Unix-Philosophie”, der interessante Zitate enthält.

Kapitel 1

GNU/Linux-Lehrstunde

Ich denke, ein Computersystem zu erlernen ist wie das Erlernen einer Fremdsprache. Obwohl Anleitungen und Dokumentation hilfreich sind, müssen Sie es selbst einüben. Um Ihnen einen sanften Start zu verschaffen, werde ich hier einige grundsätzliche Dinge ausführen.

Das kraftvolle Design von Debian GNU/Linux stammt von dem Unix-Betriebssystem, was einem Multiuser- und Multitasking-Betriebssystem ist. Sie müssen lernen, die Vorteile aus der Kraft dieser Funktionalitäten und den Ähnlichkeiten zwischen Unix und GNU/Linux zu ziehen.

Scheuen Sie sich nicht, Unix-orientierte Texte zu lesen und stützen Sie sich nicht ausschließlich auf GNU/Linux-orientierte Texte, da Sie dann viele nützliche Informationen verpassen würden.

Anmerkung
Falls Sie bereits für eine Weile irgendein Unix-ähnliches System mit Befehlszeilenwerkzeugen genutzt haben, wissen Sie möglicherweise bereits alles, was ich hier beschreibe. Bitte nutzen Sie dies zum Realitäts-Check und zur Auffrischung.

1.1 Grundlagen für die Konsole

1.1.1 Die Shell-Eingabeaufforderung (Shell-Prompt)

Nach dem Starten des Systems sehen Sie einen textbasierten Anmeldebildschirm, wenn Sie keine grafische Oberfläche (GUI) wie GNOME oder KDE installiert haben. Wenn wir davon ausgehen, dass Ihr Rechnername `foo` lautet, dann sieht Ihr Anmeldebildschirm wie folgt aus:

Haben Sie eine GUI-Oberfläche installiert, können Sie trotzdem einen solchen textbasierten Anmeldebildschirm bekommen, indem Sie Strg-Alt-F3 drücken; mit Strg-Alt-F2 kehren Sie wieder zur GUI-Umgebung zurück (mehr dazu unter Abschnitt 1.1.6).

```
foo login:
```

In dem Anmeldebildschirm geben Sie Ihren Benutzernamen ein, z.B. `penguin` und drücken die Enter-Taste, dann Ihr Passwort und nochmals Enter.

Anmerkung
Gemäß der Unix-Tradition muss bei Benutzername und Passwort auf Groß- und Kleinschreibung geachtet werden. Im Benutzernamen werden für gewöhnlich nur Kleinbuchstaben verwendet. Das erste Benutzerkonto wird normalerweise während der Installation angelegt. Weitere Benutzerkonten können durch root mit dem Befehl adduser(8) erstellt werden.

Das System startet mit einer Grußnachricht, die in "`/etc/motd`" (Message Of The Day, Meldung des Tages) gespeichert ist, und zeigt einen Befehls-Prompt an.

```
Debian GNU/Linux 12 foo tty3

foo login: penguin
Password:

Linux foo 6.5.0-0.deb12.4-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.10-1~bpo12+1 (2023-11-23) ↩
     x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Last login: Wed Dec 20 09:39:00 JST 2023 on tty3
foo:~$
```

Jetzt sind Sie in der Shell. Die Shell interpretiert die Befehle, die Sie eingeben.

1.1.2 Die Shell-Eingabeaufforderung auf einem GUI-System

Falls Sie während der Installation eine grafische GUI-Oberfläche installiert haben, bekommen Sie nach dem Systemstart einen grafischen Anmeldebildschirm. Geben Sie zur Anmeldung Benutzername und Passwort des unprivilegierten Benutzerkontos ein. Verwenden Sie dabei die TAB-Taste, um zwischen Benutzername und Passwort zu wechseln, oder benutzen Sie dazu die primäre Maustaste.

In der GUI-Oberfläche bekommen Sie eine Shell-Eingabeaufforderung, indem Sie einen `X-Terminal-Emulator` wie gnome-terminal(1), konsole(1) oder xterm(1) starten. In der GNOME-Arbeitsplatzumgebung erreichen Sie dies, indem Sie die SUPER-Taste (Windows-Taste) drücken und "terminal" in das Suchfeld eingeben.

Bei einigen anderen Arbeitsplatzumgebungen (wie `fluxbox`) gibt es möglicherweise keine offensichtliche Möglichkeit, ein Menü zu öffnen. Versuchen Sie in diesem Fall einfach einen Rechtsklick auf den Desktop-Hintergrund und hoffen, dass ein Menü erscheint.

1.1.3 Das root-Benutzerkonto

Das root-Benutzerkonto wird auch Superuser oder privilegierter Benutzer genannt. Mit diesem Benutzerkonto können Sie folgende Administrationsaufgaben erledigen:

- lesen, schreiben und löschen aller Dateien auf dem System unabhängig von deren Zugriffsrechten;
- setzen von Eigentümer und Zugriffsrechten jeglicher Dateien auf dem System;
- setzen des Passworts von jedem unprivilegierten Benutzer auf dem System;
- anmelden an jedem Benutzerkonto ohne dessen Passwort.

Diese uneingeschränkten Rechte des root-Benutzerkontos erfordern von Ihnen, dass Sie sich besonnen und verantwortungsvoll verhalten, wenn Sie es benutzen.



Warnung
Teilen Sie das root-Passwort niemals anderen mit.

Anmerkung
Bestimmte Dateizugriffsrechte einer Datei (inklusive Hardware-Geräte wie CD-ROM usw., die auf einem Debian-System nichts anderes als eine Datei sind) können dazu führen, dass Sie für nicht-root-Benutzer unbenutzbar sind. Obwohl die Verwendung des root-Benutzerkontos ein schneller Weg ist, solche Situationen zu testen, sollte die Lösung das korrekte Einstellen der Datei-Zugriffsrechte und/oder das Hinzufügen des Benutzers in entsprechende Gruppen sein (siehe dazu Abschnitt 1.2.3).

1.1.4 Die root-shell-Eingabeaufforderung

Hier ein paar grundlegende Methoden, um mittels root-Passwort eine root-shell-Eingabeaufforderung zu bekommen:

- Geben Sie `root` im textbasierten Anmeldebildschirm ein.
- Tippen Sie in jeglicher Shell "`su -l`" ein.
  - Hierdurch werden NICHT die Umgebungseinstellungen des angemeldeten Benutzers beibehalten.
- Tippen Sie in jeglicher Benutzer-Shell-Eingabeaufforderung "`su`" ein.
  - Hierdurch werden teilweise die Umgebungseinstellungen des angemeldeten Benutzers beibehalten.

1.1.5 Systemadministrations-Werkzeuge mit grafischer Oberfläche (GUI)

Wenn sich über Ihr Arbeitsplatz-Menü keine GUI-Systemadministrations-Werkzeuge mit den benötigten Rechten starten lassen, können Sie diese von einer root-shell-Eingabeaufforderung aus starten, wie z.B. von gnome-terminal(1), konsole(1) oder xterm(1). Lesen Sie dazu Abschnitt 1.1.4 und Abschnitt 7.9.



Warnung
Starten Sie niemals das GUI-Display/den Sitzungsmanager als root, indem Sie `root` im Anmeldebildschirm des Displaymanagers (z.B. gdm3(1)) eingeben.
Lassen Sie niemals vertrauensunwürdige GUI-Programme mit Fernanbindung in einem X-Window laufen, wenn dort kritische Daten angezeigt werden, da diese Programme Ihren grafischen Bildschirminhalt abgreifen können.

1.1.6 Virtuelle Konsolen

In einem Standard-Debian-System sind sechs VT100-ähnliche textbasierte Konsolen verfügbar, zwischen denen Sie hin und her schalten können, und in denen die Befehls-Shell direkt auf dem Linux-Host ausgeführt werden kann. Solange Sie sich nicht in einer GUI-Umgebung befinden, können Sie mit der linken `Alt-Taste` und gleichzeitig einer der Funktionstasten `F1` — `F6` zwischen den virtuellen Konsolen umschalten. Jede textbasierte Konsole erlaubt eine eigenständige Anmeldung mit einem Benutzerkonto und bietet eine Mehrbenutzer-Umgebung. Diese Mehrbenutzer-Umgebung ist eine tolle Unix-Funktionalität und macht sehr leicht süchtig.

Wenn Sie sich in der grafischen GUI-Umgebung befinden, bekommen Sie mit `Strg-Alt-F3` (d.h. die linke `Strg-Taste`, die linke `Alt-Taste` und `F3` gleichzeitig) Zugang zur textbasierten Konsole 3. Sie können zur GUI-Oberfläche zurückkehren, die normalerweise auf der virtuellen Konsole 2 läuft, indem Sie `Alt-F2` drücken.

Alternativ können Sie auch von der Befehlszeile aus zu einer anderen virtuellen Konsole wechseln, z.B. zur Konsole 3:

```
# chvt 3
```

1.1.7 Wie Sie die Eingabeaufforderung wieder verlassen

Drücken Sie in der Eingabeaufforderung `Strg-D` (also die linke `Strg-Taste` und die `d-Taste` gleichzeitig), um die Shell-Aktivitäten zu beenden. Wenn Sie sich auf der textbasierten Konsole befinden, kehren Sie hiermit zum Anmeldebildschirm zurück. Obwohl bei diesen Steuerzeichen von ”control D” mit großem D gesprochen wird, müssen Sie hier nicht die Umschalt-Taste drücken. Für `Strg-D` wird auch die Kurzform `^D` benutzt. Alternativ können Sie auch ”exit” eingeben.

Wenn Sie sich in einem X-Terminal-Emulator(1) befinden, können Sie hiermit das `X-Terminal-Emulator`-Fenster schließen.

1.1.8 Wie Sie das System herunterfahren

Wie jedes andere moderne Betriebssystem, bei dem Dateioperationen das Zwischenspeichern (Caching) beinhalten, um die Performance zu erhöhen, erfordert auch das Debian-System eine Prozedur zum sauberen Herunterfahren, bevor gefahrlos abgeschaltet werden kann. Dies dient dazu, die Integrität der Daten aufrecht zu erhalten, indem alle Änderungen auf die Platte geschrieben werden. Falls Energiekontrolle per Software verfügbar ist, wird im Rahmen des Herunterfahrens die Spannung automatisch abgeschaltet. (Andernfalls müssen Sie den Ein-/Ausschaltknopf mehrere Sekunden gedrückt halten, nachdem das Herunterfahren abgeschlossen ist).

Sie können das System im normalen Mehrbenutzermodus auf der Befehlszeile herunterfahren mittels:

```
# shutdown -h now
```

Sie können das System im Einzelbenutzermodus auf der Befehlszeile herunterfahren mittels:

```
# poweroff -i -f
```

Lesen Sie auch Abschnitt 6.3.8.

1.1.9 Eine Konsole wiederherstellen

Wenn der Bildschirm verrückt spielt, nachdem Sie verrückte Dinge wie ”`cat` *`irgendeine-binärdatei`*” gemacht haben, geben Sie ”`reset`” an der Eingabeaufforderung ein. Sie können den Befehl möglicherweise nicht sehen, während Sie ihn eingeben. Mit ”`clear`” können Sie bei Bedarf auch den Bildschirm leeren.

1.1.10 Zusätzliche Paketempfehlungen für Neulinge

Obwohl selbst die Minimalinstallation eines Debian-Systems ohne jegliche Arbeitsplatzumgebung die grundlegenden Unix-Funktionalitäten bietet, ist es eine gute Idee, mittels apt-get(8) einige Befehlszeilen- und curses-basierte Zeichen-Terminal-Pakete zusätzlich zu installieren, wie z.B. `mc` und `vim`; Anfänger können dazu für einen ersten Versuch folgendes verwenden:

```
# apt-get update
 ...
# apt-get install mc vim sudo aptitude
 ...
```

Falls Sie diese Pakete bereits installiert haben, werden keine neuen Pakete installiert.

Es könnte eine gute Idee sein, einiges an informativer Dokumentation zu lesen.

Sie sollten vielleicht einige dieser Pakete installieren, indem Sie folgendes eingeben:

```
# apt-get install package_name
```

Paket	Popcon	Größe	Beschreibung
mc	V:42, I:172	1590	Ein Textmodus-Dateimanager mit Vollbildschirm-Ansicht
sudo	V:739, I:870	6817	Ein Programm, das Benutzern eingeschränkte root-Privilegien einräumt
vim	V:73, I:330	4415	Unix Texteditor Vi IMproved, ein Texteditor für Programmierer (Standardversion)
vim-tiny	V:55, I:979	1931	Unix Texteditor Vi IMproved, ein Texteditor für Programmierer (Kompaktversion)
emacs-nox	V:3, I:12	46564	Emacs vom GNU-Projekt, der Lisp-basierte erweiterbare Texteditor
w3m	V:11, I:125	2853	Textmodus-WWW-Browser
gpm	V:8.0, I:8.7	524	Unix-ähnliches Kopieren-und-Einfügen auf der Textkonsole (Daemon)

Tabelle 1.1: Liste interessanter Textmodus-Programm-Pakete

Paket	Popcon	Größe	Beschreibung
doc-debian	I:887	185	Dokumentation des Debian-Projekts und andere Dokumente
debian-policy	I:6.4	5147	Debian Policy-Handbuch und zugehörige Dokumente
developers-reference	V:0.2, I:2.0	2656	Richtlinien und Informationen für Debian-Entwickler
debmake-doc	I:0.32	11807	Leitfaden für Debian-Betreuer
debian-history	I:0.49	6513	Vergangenheit des Debian-Projekts
debian-faq	I:886	791	Debian FAQ

Tabelle 1.2: Liste informativer Dokumentationspakete

1.1.11 Ein zusätzliches Benutzerkonto

Falls Sie Ihr Haupt-Benutzerkonto nicht für die folgenden Trainingsaktivitäten nutzen möchten, können Sie mit folgendem Befehl ein Trainings-Benutzerkonto erstellen, hier z.B. `fish`:

```
# adduser fish
```

Beantworten Sie alle Fragen.

Dadurch wird ein neues Konto namens `fish` erstellt. Nach Ihren Übungen können Sie dieses Benutzerkonto und das dazugehörige Heimatverzeichnis entfernen, indem Sie folgendes verwenden:

```
# deluser --remove-home fish
```

Auf nicht-Debian-Systemen sowie speziell angepassten Debian-Systemen müssen für die oben aufgeführten Aktivitäten die niederschwelligen Werkzeuge useradd(8) und userdel(8) verwendet werden.

1.1.12 sudo-Konfiguration

Für ein typisches Einzelbenutzer-Arbeitsplatzsystem, wie z.B. einen Debian-Desktopauf einem Laptop, ist es gängig, wie folgt eine einfache Konfiguration für sudo(8) einzurichten, so dass der nicht-privilegierten Benutzer, z.B. `penguin`, lediglich mit seinem Benutzer-Passwort (aber ohne das root-Passwort) administrative Rechte bekommen kann:

```
# echo "penguin  ALL=(ALL) ALL" >> /etc/sudoers
```

Alternativ dazu ist es auch verbreitet, dem nicht-privilegierten Benutzer, hier `penguin`, wie folgt administrative Rechte zu gewähren, ohne dass dieser irgendein Passwort eingeben muss:

```
# echo "penguin  ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers
```

Dieser Trick sollte nur für einen Einzelbenutzer-Arbeitsplatz angewandt werden, den Sie selbst administrieren und auf dem Sie der einzige Benutzer sind.



Warnung
Richten Sie keine Benutzerkonten regulärer Benutzer auf Mehrbenutzersystemen derartig ein; dies wäre sehr schlecht für die Systemsicherheit.



Achtung
Das Passwort und Benutzerkonto von `penguin` im obigen Beispiel erfordert ebenso viel Schutz wie das root-Passwort und das root-Benutzerkonto.
Administrative Rechte in diesem Zusammenhang gehören zu jemandem, der berechtigt ist, Systemadministrationsaufgaben auf dem Arbeitsplatzsystem durchzuführen. Gewähren Sie niemals einem Manager in der Verwaltungsabteilung Ihrer Firma oder Ihrem Chef derartige Privilegien, außer diese sind dazu autorisiert und befähigt.

Anmerkung
Um Leuten Zugriffsrechte auf limitierte Geräte und Dateien zu verschaffen, sollten Sie die Verwendung von **group** in Betracht ziehen, um eingeschränkte Rechte einzurichten, statt die root-Privilegien via sudo(8) dafür zu nutzen. Mit einer durchdachten und vorsichtigen Konfiguration kann sudo(8) auf einem System, das Sie sich mit mehreren Leuten teilen, limitierte administrative Rechte einräumen, ohne dass Sie anderen das root-Passwort mitteilen müssen. Dies kann Ihnen auf Systemen helfen, wo Sie sich die Verantwortung mit mehreren Administratoren teilen, so dass Sie sagen können, wer was gemacht hat. Auf der anderen Seite möchten Sie vielleicht nicht, dass irgendjemand sonst solche Privilegien hat.

1.1.13 Zeit zum Spielen

Jetzt sind Sie soweit, dass Sie mit dem Debian-System ohne Risiko herumspielen können, solange Sie das nichtprivilegierte Benutzerkonto verwenden.

Wir lernen das Debian-System mit folgenden Aspekten eines Unix-ähnlichen Systems kennen:

- Abschnitt 1.2
- Abschnitt 1.3
- Abschnitt 1.4
- Abschnitt 1.5
- Abschnitt 1.6

1.2 Unix-ähnliches Dateisystem

In GNU/Linux und anderen Unix-ähnlichen Betriebssystemen sind Dateien in Verzeichnis-Strukturen organisiert. Alle Dateien und Verzeichnisse sind in einem großen Verzeichnisbaum unterhalb von "/" eingeordnet. Dies wird Baum genannt, weil das Dateisystem, wenn Sie es aufmalen, wie ein Baum aussieht, nur dass er auf dem Kopf steht.

Diese Dateien und Verzeichnisse können auf mehrere Geräte verteilt sein. mount(8) dient dazu, Dateisysteme, die auf anderen Geräten erkannt werden, in den großen Verzeichnisbaum einzubinden. Umgekehrt dient umount(8) dazu, es wieder daraus zu entfernen. Auf aktuellen Linux-Kernel kann mount(8) mit bestimmten Optionen Teile

des Verzeichnisbaums zusätzlich an anderer Stelle anknüpfen oder die zusätzlichen Eigenschaften shared, private, slave oder unbindable verwenden. Unterstützte mount-Optionen für verschiedene Dateisystemtypen finden Sie in "`/usr/share/doc/linux-doc-*/Documentation/filesystems/`".

Verzeichnisse auf Unix-Systemen werden bei einigen anderen Betriebssystemen auch **Ordner** genannt. Bitte beachten Sie ebenfalls, dass es auf einem Unix-System kein Konzept zur Bezeichnung von **Laufwerken** gibt, das vergleichbar mit z.B. "`A:`" wäre. Es gibt ein Dateisystem, und dort ist alles enthalten. Dies ist ein gewaltiger Unterschied verglichen mit Windows.

1.2.1 Unix-Dateigrundlagen

Hier einige Grundlagen zu Dateien unter Unix:

- Bei Dateinamen ist **Groß-/Kleinschreibung immer relevant**. Das heißt, "`MEINEDATEI`" und "`MeineDatei`" sind zwei unterschiedliche Dateien.
- Das **Wurzel-Verzeichnis (root)** bedeutet Wurzel des Dateisystems und wird einfach als "`/`" dargestellt. Verwechseln Sie dies nicht mit dem Heimatverzeichnis des root-Benutzers "`/root`".
- Jedes Verzeichnis hat einen Namen, der jegliche Buchstaben oder Symbole **außer "`/`"** enthalten kann. Das Wurzelverzeichnis root ist eine Ausnahme; dessen Name ist "`/`" (ausgesprochen "slash" oder "das root-Verzeichnis") und es kann nicht umbenannt werden.
- Jede Datei oder jedes Verzeichnis ist über einen **voll qualifizierten Dateinamen**, **absoluten Dateinamen** oder **Pfad** gekennzeichnet, der die Abfolge der Verzeichnisse enthält, die zu dessen Erreichen passiert werden müssen. Die drei oben angegebenen Ausdrücke sind alle gleichbedeutend.
- Alle **voll qualifizierten Dateinamen** beginnen mit dem "`/`"-Verzeichnis, und zwischen zwei Verzeichnissen oder zwischen Verzeichnis und Datei steht ein weiteres "`/`". Das erste "`/`" ist das Wurzelverzeichnis, und die anderen "`/`" sind jeweils Trenner zwischen den einzelnen Unterverzeichnissen, bis der letzte Eintrag erreicht ist, was der Name der eigentlichen Datei ist. Die Wörter, die hier verwendet werden, können verwirrend sein. Nehmen Sie den folgenden **voll qualifizierten Dateinamen** als Beispiel: "`/usr/share/keytables/de.map.gz`". Allerdings spricht man von "`de.map.gz`" alleine als Dateiname.
- Das Wurzelverzeichnis enthält eine Anzahl von Unterverzeichnisse, wie "`/etc/`" und "`/usr/`". Diese haben wiederum nochmals weitere Unterverzeichnisse, z. B. "`/etc/systemd/`" und "`/usr/local/`". Das Ganze komplett betrachtet wird als **Verzeichnisbaum** bezeichnet. Sie können den absoluten Dateinamen als Route vom Fuss des Baums ("`/`") zu dessen Ende einer Verzweigung (einer Datei) betrachten. Manchmal hören Sie auch, wie Leute vom Verzeichnisbaum reden, als wäre es ein **Familien**-Stammbaum, der alle direkten Nachkommen einer einzigen Figur (dem Wurzelverzeichnis "`/`") einschließt: demzufolge haben die Unterverzeichnisse **Eltern**, und ein Pfad zeigt die vollständige Herkunft einer Datei. Es gibt auch relative Pfade, die irgendwo beginnen (nicht beim Wurzelverzeichnis). Sie sollten dabei in Erinnerung behalten, dass "`../`" immer das jeweils übergeordnete Verzeichnis (Eltern-Verzeichnis) bezeichnet. Diese Bezeichnungsweise gilt auch für andere Strukturen, die dem Verzeichnisbaum ähnlich sind, wie hierarchische Datenstrukturen.
- Es gibt keine spezielle Verzeichnispfad-Namenskomponente, die einem physikalischen Gerät, wie einer Festplatte, entspricht. Dies unterscheidet sich von RT-11, CP/M, OpenVMS, MS-DOS, AmigaOS und Microsoft Windows, wo der Pfad einen Gerätenamen wie "`C:\`" enthält. Allerdings existieren Verzeichniseinträge, die sich als Teil des normalen Dateisystems auf physikalische Geräte beziehen. Siehe dazu Abschnitt 1.2.2.

Anmerkung
Obwohl Sie nahezu alle Buchstaben oder Symbole in einem Dateinamen verwenden **können**, ist es in der Praxis eine schlechte Idee, dies zu tun. Es ist besser, alle Zeichen zu vermeiden, die auf der Befehlszeile oft eine spezielle Bedeutung haben, darunter Leerzeichen, Tabulatoren, Newlines, und andere spezielle Zeichen wie `{ } ( ) [ ] ' ` " \ / > < | ; ! # & ^ * % @ $`. Wenn Sie innerhalb des Namens einzelne Wörter trennen möchten, sind Punkte, Bindestriche und Unterstriche hierfür eine gute Wahl. Sie können auch die Wörter jeweils am Anfang großschreiben, "`SoWieHier`". Erfahrene Benutzer tendieren dazu, Leerzeichen in Dateinamen zu vermeiden.

Anmerkung
Das Wort ”root” kann entweder ”root-Benutzer” oder ”root-Verzeichnis” (Wurzelverzeichnis) bedeuten. Am Zusammenhang bei der Verwendung sollten Sie erkennen können, was gemeint ist.

Anmerkung
Das Wort **Pfad** wird nicht nur wie oben für **voll qualifizierte Dateinamen** verwendet, sondern auch für den **Befehls-Suchpfad**. Die jeweilige Bedeutung erschließt sich für gewöhnlich aus dem Zusammenhang.

Detaillierte bewährte Methoden für die Dateihierarchie sind im Filesystem Hierarchy Standard beschrieben (in ”`/usr/share/`” und unter hier(7)). Sie sollten folgendes als Ausgangspunkt nutzen:

Verzeichnis	Verwendung des Verzeichnisses
`/`	das Wurzelverzeichnis (root-Verzeichnis)
`/etc/`	systemweite Konfigurationsdateien
`/var/log/`	System-Protokolldateien
`/home/`	die Heimatverzeichnisse aller nicht-privilegierten Benutzer

Tabelle 1.3: Auflistung der Verwendung wichtiger Verzeichnisse

1.2.2 Dateisystem-Internas

Gemäß der **Unix-Tradition** stellt das Debian GNU/Linux-System ein Dateisystem bereit, unterhalb dessen alle physikalischen Daten auf Festplatten und anderen Speichermedien liegen; alles für das Zusammenspiel mit Hardware-Geräten wie Konsolenbildschirmen und von fern zugreifenden seriellen Konsolen benötigte wird in vereinheitlichter Art und Weise unterhalb von ”`/dev/`” abgebildet.

Alle Dateien, Verzeichnisse, benannten Pipes (named pipes; ein Verfahren, mit dem zwei Progamme Daten austauschen können) oder physikalischen Geräte auf einem Debian GNU/Linux-System haben eine Datenstruktur, Inode genannt, die die mit ihr verbundenen Attribute beschreibt, wie der Benutzer, der deren Eigentümer (owner) ist, die Gruppe, der sie angehören, der Zeitpunkt des letzten Zugriffs usw. Die Idee, nahezu alles im Dateisystem darzustellen, war eine Unix-Innovation, und moderne Linux-Kernel haben diese Idee sogar noch weiterentwickelt. Jetzt kann man sogar Informationen über die Prozesse, die auf dem Computer laufen, im Dateisystem finden.

Die abstrakte und vereinheitlichte Abbildung von physikalischen Datensätzen und internen Prozessen ist sehr leistungsfähig, da sie es uns erlaubt, für die gleiche Funktion auf vielen total unterschiedlichen Geräten denselben Befehl zu verwenden. Es ist sogar möglich, die Art und Weise, wie der Kernel arbeitet, zu verändern, indem man Daten in spezielle Dateien schreibt, die mit laufenden Prozessen verbunden sind.

Tipp
Wenn Sie wissen möchten, wie der Verzeichisbaum und die Gerätedateien verknüpft sind, führen Sie mount(8) ohne Argumente aus.

1.2.3 Dateisystem-Berechtigungen

Dateisystem-Berechtigungen auf Unix-ähnlichen Systemen werden für drei Kategorien beteiligter Benutzer festgelegt:

- der **Benutzer** (user), der Eigentümer der Datei ist (**u**);
- andere Benutzer der **Gruppe** (group), zu der die Datei gehört (**g**);

- alle **anderen** (other) Benutzer (**o**), auch ”world” (Welt) oder ”everyone” (jeder) genannt.

Bei Dateien gibt es Berechtigung für die folgenden Aktionen:

- Die **Lese**-Berechtigung (read, **r**) erlaubt dem Rechteinhaber, den Inhalt der Datei zu betrachten.
- Die **Schreib**-Berechtigung (write, **w**) erlaubt dem Rechteinhaber, die Datei zu verändern.
- Die **Ausführungs**-Berechtigung (execute, **x**) erlaubt dem Rechteinhaber, die Datei als Befehl auszuführen.

Bei Verzeichnissen gibt es Berechtigung für die folgenden Aktionen:

- Die **Lese**-Berechtigung (read, **r**) erlaubt dem Rechteinhaber, den Inhalt des Verzeichnisses aufzulisten.
- Die **Schreib**-Berechtigung (write, **w**) erlaubt dem Rechteinhaber, Dateien zu dem Verzeichnis hinzuzufügen oder Dateien zu löschen.
- Die **Ausführungs**-Berechtigung (execute, **x**) erlaubt dem Rechteinhaber, auf Dateien in dem Verzeichnis zuzugreifen.

Hierbei bedeutet die **Ausführungs**-Berechtigung für ein Verzeichnis nicht nur, dass der Inhalt der Dateien in diesem Verzeichnis betrachtet werden kann, sondern auch deren Attribute, wie die Dateigröße und der Zeitpunkt der letzten Änderung.

ls(1) wird verwendet, um Informationen über die Zugriffsrechte (und mehr) von Dateien und Verzeichnissen anzuzeigen. Wenn es mit der Option ”`-l`” aufgerufen wird, werden die folgenden Informationen in der angegebenen Reihenfolge angezeigt:

- **Typ der Datei** (erstes Zeichen);
- **Berechtigungen** der Datei (neun Zeichen, bestehend aus drei Zeichen jeweils für Benutzer, Gruppe und andere, in dieser Reihenfolge);
- **Anzahl der harten Links** auf die Datei;
- Name des **Benutzers**, der Eigentümer der Datei ist;
- Name der **Gruppe**, zu der die Datei gehört;
- **Größe** der Datei in Zeichen (Byte);
- **Datum und Uhrzeit** der Datei (mtime);
- **Name** der Datei.

Zeichen	Bedeutung
-	normale Datei
d	Verzeichnis
l	symbolischer Link
c	Geräte-Node für zeichenorientierte Geräte
b	Geräte-Node für blockorientierte Geräte
p	benannte Pipe (named pipe)
s	Socket

Tabelle 1.4: Bedeutungen des ersten Zeichens der ”`ls -l`”-Ausgabe

chown(1) wird von dem root-Benutzer verwendet, um den Eigentümer einer Datei zu ändern. chgrp(1) wird vom Eigentümer einer Datei oder von root benutzt, um die Gruppe zu ändern, zu der die Datei gehört. chmod(1) wird vom Eigentümer oder dem root-Benutzer verwendet, um die Zugriffsberechtigungen für Datei und Verzeichnis zu ändern. Die grundsätzliche Syntax, um die Datei `foo` zu bearbeiten, ist wie folgt:

```
# chown newowner foo
# chgrp newgroup foo
# chmod  [ugoa][+-=][rwxXst][,...] foo
```

Sie können zum Beispiel mit folgenden Befehlen einen Verzeichnisbaum so manipulieren, dass `foo` sein Eigentümer wird und `bar` die Gruppe:

```
# cd /some/location/
# chown -R foo:bar .
# chmod -R ug+rwX,o=rX .
```

Es gibt noch drei weitere, spezielle Zugriffs-Bits:

- das **Setze Benutzer-ID**-Bit (**s**, oder **S** statt dem **x** des Benutzers);
- das **Setze Gruppen-ID**-Bit (**s**, oder **S** statt dem **x** der Gruppe);
- das **sticky (klebrig)**-Bit (**t**, oder **T** statt dem **x** der "anderen").

Hierbei enthält die Ausgabe von "`ls -l`" für diese Bits die jeweiligen **Großbuchstaben**, wenn die Ausführungs-Bits (x), die bei dieser Ausgabe versteckt sind, **nicht gesetzt** sind.

Das Setzen des **Setze Benutzer-ID**-Bits einer ausführbaren Datei erlaubt dem Benutzer, die ausführbare Datei unter der Benutzer-ID der Datei (zum Beispiel **root**) auszuführen. Ähnlich dazu erlaubt das Setzen des Bits **Setze Gruppen-ID** einer ausführbaren Datei dem Benutzer, die ausführbare Datei unter der Gruppen-ID der Datei (zum Beispiel **root**) auszuführen. Da diese Einstellungen Sicherheitsrisiken verursachen können, erfordert deren Aktivierung besondere Vorsicht.

Das Setzen des **Setze Gruppen-ID**-Bits eines Verzeichnisses aktiviert das BSD-ähnliche Dateierstellungs-Schema, bei dem alle in dem Verzeichnis erzeugten Dateien der gleichen **Gruppe** angehören wie das Verzeichnis selbst.

Das Setzen des **sticky (klebrig)**-Bits eines Verzeichnisses verhindert, dass eine Datei in dem Verzeichnis von einem Benutzer gelöscht wird, der nicht Eigentümer der Datei ist. Um den Inhalt einer Datei in einem für alle schreibbaren Verzeichnis wie "`/tmp`" oder in durch die Gruppe schreibbaren Verzeichnissen sicherzustellen, muss nicht nur die **Schreib**-Berechtigung für die Datei zurückgenommen werden, sondern auch das **sticky (klebrig)**-Bit für das Verzeichnis. Ansonsten könnte jeder Benutzer, der Schreibberechtigung in dem Verzeichnis hat, die Datei löschen und eine neue mit dem gleichen Namen (aber eventuell anderem Inhalt) erstellen.

Hier einige interessante Beispiele von Dateiberechtigungen:

```
$ ls -l /etc/passwd /etc/shadow /dev/ppp /usr/sbin/exim4
crw------T 1 root root   108, 0 Oct 16 20:57 /dev/ppp
-rw-r--r-- 1 root root     2761 Aug 30 10:38 /etc/passwd
-rw-r----- 1 root shadow   1695 Aug 30 10:38 /etc/shadow
-rwsr-xr-x 1 root root   973824 Sep 23 20:04 /usr/sbin/exim4
$ ls -ld /tmp /var/tmp /usr/local /var/mail /usr/src
drwxrwxrwt 14 root root  20480 Oct 16 21:25 /tmp
drwxrwsr-x 10 root staff  4096 Sep 29 22:50 /usr/local
drwxr-xr-x 10 root root   4096 Oct 11 00:28 /usr/src
drwxrwsr-x  2 root mail   4096 Oct 15 21:40 /var/mail
drwxrwxrwt  3 root root   4096 Oct 16 21:20 /var/tmp
```

Es gibt eine alternative numerische Möglichkeit, um Dateiberechtigungen mit chmod(1) darzustellen. Dieser numerische Modus verwendet 3 oder 4 einstellige Oktalzahlen (Basis=8).

Dies klingt kompliziert, aber es ist letztendig ganz einfach. Wenn Sie die ersten Spalten (2-10) der "`ls -l`"-Ausgabe anschauen und sie in binärer Darstellung (Basis = 2) lesen ("-" ist "0" und "rwx" sind "1"), werden Sie die letzten 3 Ziffern des numerischen Modus' im Oktalformat (Basis = 8) erkennen.

Probieren Sie zum Beispiel folgendes:

Ziffer	Bedeutung
1. Ziffer (optional)	Summe der Bits **Setze Benutzer-ID** (=4), **Setze Gruppen-ID** (=2) und **sticky (klebrig)** (=1)
2. Ziffer	Summer der Rechte **Lesen** (= 4), **Schreiben** (= 2) und **Ausführen** (= 1) für den **Benutzer**
3. Ziffer	identisch für die **Gruppe**
4. Ziffer	identisch für **alle anderen**

Tabelle 1.5: Der numerische Modus für Dateiberechtigungen in chmod(1)-Befehlen

```
$ touch foo bar
$ chmod u=rw,go=r foo
$ chmod 644 bar
$ ls -l foo bar
-rw-r--r-- 1 penguin penguin 0 Oct 16 21:39 bar
-rw-r--r-- 1 penguin penguin 0 Oct 16 21:35 foo
```

Tipp
Wenn Sie mit einem Shell-Skript auf Informationen von "`ls -l`" zugreifen müssen, sollten Sie geeignete Befehle wie test(1), stat(1) und readlink(1) verwenden. Auch in die Shell integrierte Dinge wie "`[`" oder "`test`" können genutzt werden.

1.2.4 Steuerung der Berechtigungen für neu erzeugte Dateien: umask

Welche Berechtigungen einer neu erzeugten Datei oder einem neu erzeugten Verzeichnis zugewiesen werden, wird durch den in die Shell integrierten Befehl `umask` festgelegt/eingeschränkt. Lesen Sie dazu dash(1), bash(1) und bash-builtins(7).

```
(file permissions) = (requested file permissions) & ~(umask value)
```

umask	erzeugte Dateirechte	erzeugte Verzeichnisrechte	Verwendung
`0022`	`-rw-r--r--`	`-rwxr-xr-x`	schreibbar nur durch den Benutzer
`0002`	`-rw-rw-r--`	`-rwxrwxr-x`	schreibbar durch die Gruppe

Tabelle 1.6: Beispiele für **umask**-Werte

Das Debian-System verwendet standardmäßig das UPG-Schema (user private group, eineprivate Gruppe für jeden Benutzer). Eine UPG wird erstellt, immer wenn ein Benutzer dem System hinzugefügt wird. Eine UPG hat den gleichen Namen wie der Benutzer, für den sie erstellt wurde, und dieser Benutzer ist das alleinige Mitglied in dieser Gruppe. Durch das UPG-Schema ist es sicher, die umask auf `0002` zu setzen, da jeder Benutzer seine eigene private Gruppe hat. (Bei einigen Unix-Varianten ist es gängig, alle normalen Benutzer so einzurichten, dass sie einer einzigen **`users`**-Gruppe angehören und es ist in solchen Fällen eine gute Idee, umask aus Sicherheitsgründen auf `0022` zu setzen.)

Tipp
Sie aktivieren das UPG-Schema, indem Sie "`umask 002`" zur Datei `~/.bashrc` hinzufügen.

1.2.5 Berechtigungen für Gruppen von Benutzern (group)



Warnung
Stellen sie sicher, dass noch nicht gespeicherte Änderungen gespeichert werden, bevor Sie einen Reboot oder ähnliches ausführen.

Sie können mit zwei Schritten den Benutzer `penguin` zu der Gruppe `bird` hinzufügen:

- Ändern der Gruppenkonfiguration mit einem der folgenden Befehle:
  - Führen Sie "`sudo usermod -aG bird penguin`" aus.
  - Führen Sie "`sudo adduser penguin bird`" aus (nur auf typischen Debian-Systemen).
  - Führen Sie "`sudo vigr`" für `/etc/group` aus,sowie "`sudo vigr -s`" für `/etc/gshadow`, um `penguin` der Zeile für die Gruppe `bird` hinzuzufügen.
- Übernehmen Sie die Konfiguration mit einer der folgenden Möglichkeiten:
  - Kalt-Neustart und neu anmelden (die beste Option).
  - Abmelden und neu anmelden über die grafische Oberfläche (für moderne Desktop-Umgebungen funktioniert dies möglicherweise nicht).

Sie können mit zwei Schritten den Benutzer `penguin` aus der Gruppe `bird` entfernen:

- Ändern der Gruppenkonfiguration mit einem der folgenden Befehle:
  - Führen Sie "`sudo usermod -rG bird penguin`" aus.
  - Führen Sie "`sudo deluser penguin bird`" aus (nur auf typischen Debian-Systemen).
  - Führen Sie "`sudo vigr`" für `/etc/group`, sowie "`sudo vigr -s`" für `/etc/gshadow` aus, um `penguin` aus der Zeile für die Gruppe `bird` zu entfernen.
- Übernehmen Sie die Konfiguration mit einer der folgenden Möglichkeiten:
  - Kalt-Neustart und neu anmelden (die beste Option).
  - Führen Sie "`kill -TERM -1`" aus sowie einige korrigierende Aktionen wie "`systemctl restart NetworkManager`
  - Abmelden über die grafische Oberfläche ist für den GNOME-Desktop keine Option.

Jegliche Versuche eines Warm-Neustarts sind bei modernen Arbeitsplatzsystemen ein fragiler Ersatz für einen Kalt-Neustart.

Anmerkung
Alternativ können Sie Benutzer während des Authentifizierungsprozesses dynamisch zu Gruppen hinzufügen, indem Sie die Zeile "`auth optional pam_group.so`" zu "`/etc/pam.d/common-auth`" hinzufügen und "`/etc/security/group.conf`" einrichten. Lesen Sie dazu Kapitel 4.)

Hardware-Geräte sind nur eine andere Art von Datei auf dem Debian-System. Wenn Sie Probleme haben, von einem Benutzerkonto aus auf Geräte wie USB-Speichersticks oder CD-ROM-Laufwerke zuzugreifen, sollten Sie den Benutzer zu einem Mitglied der entsprechenden Gruppe machen.

Einige erwähnenswerte systemweite Gruppen erlauben ihren Mitglieder, ohne `root`-Privilegien auf die jeweiligen Dateien und Geräte zuzugreifen:

Gruppe	Beschreibung der Dateien und Geräte, auf die zugegriffen werden kann
`dialout`	voller und direkter Zugriff auf serielle Ports (”`/dev/ttyS[0-3]`”)
`dip`	eingeschränkter Zugriff auf serielle Ports für **Dialup IP**-(Einwahl-)Verbindungen zu vertrauenswürden Gegenstellen
`cdrom`	CD-ROM- und DVD+/-RW-Laufwerke
`audio`	Audiogeräte (z.B. Soundkarten)
`video`	Videogeräte (z.B. Grafikkarten)
`scanner`	Scanner
`adm`	Systemüberwachungs-Protokolle
`staff`	einige Verzeichnisse für untergeordnete administrative Tätigkeiten: ”`/usr/local`”, ”`/home`”

Tabelle 1.7: Liste erwähnenswerter systemweiter Gruppen

Tipp
Sie müssen der Gruppe `dialout` angehören, um ein Modem umzukonfigurieren, sich irgendwo einzuwählen usw. Wenn jedoch `root` vordefinierte Konfigurationsdateien für vertrauenswürdige Gegenstellen in ”`/etc/ppp/peers/`” anlegt, müssen Sie lediglich Mitglied der Gruppe `dip` sein, um mit den Befehlen pppd(8), pon(1) und poff(1) **Dialup IP**-(Einwahl-)Verbindungen herzustellen.

Einige erwähnenswerte, vom System angebotene Gruppen erlauben Ihren Mitgliedern, besondere Befehle ohne `root`-Rechte auszuführen:

Gruppe	Befehle, die ausgeführt werden können
`sudo`	Einen Befehl mit Superuser-Privilegien ausführen
`lpadmin`	Befehle ausführen, um Drucker zur Druckerdatenbank hinzuzufügen, zu modifizieren und zu entfernen

Tabelle 1.8: Liste erwähnenswerter, vom System angebotenen Gruppen zur Ausführung besonderer Befehle

Eine vollständige Liste der vom System angebotenen Benutzer und Gruppen finden Sie in der letzten Version des Dokuments ”Users and Groups” in ”`/usr/share/doc/base-passwd/users-and-groups.html`” aus dem Paket `base-passwd`.

Lesen Sie passwd(5), group(5), shadow(5), newgrp(1), vipw(8), vigr(8) und pam_group(8) für Informationen über Befehle zur Verwaltung des Benutzer- und Gruppensystems.

1.2.6 Zeitstempel

Es gibt drei Arten von Zeitstempeln für eine GNU/Linux-Datei:

Art	Bedeutung (historische Unix-Definition)
mtime	Zeitpunkt der letzten Änderung an der Datei (`ls -l`)
ctime	Zeitpunkt der letzten Statusänderung der Datei (`ls -lc`)
atime	Zeitpunkt des letzten Zugriffs auf die Datei (`ls -lu`)

Tabelle 1.9: Liste der Arten von Zeitstempeln

Anmerkung
ctime ist nicht der Zeitpunkt der Dateierzeugung.

Anmerkung
Der wirkliche Wert von **atime** auf einem GNU/Linux-System könnte sich von dem mit der historischen Unix-Definition konformen Variante unterscheiden.

- Das Überschreiben einer Datei ändert die **mtime**-, **ctime**- und **atime**-Attribute der Datei.
- Das Ändern von Eigentümer oder Zugriffsrechten ändert die **ctime**- und **atime**-Attribute der Datei.
- Das Lesen einer Datei ändert das **atime**-Attribut der Datei auf einem historischen Unix-System.
- Das Lesen einer Datei ändert das **atime**-Attribut der Datei auf einem GNU/Linux-System, wenn das Dateisystem mit dem Argument "`strictatime`" eingebunden wurde.
- Das erstmalige Lesen einer Datei oder das Lesen nach einem Tag ändert das **atime**-Attribut der Datei auf einem GNU/Linux-System, wenn das Dateisystem mit dem Argument "`relatime`" eingebunden wurde (dies ist das Standardverhalten seit Linux 2.6.30).
- Das Lesen einer Datei ändert nicht das **atime**-Attribut der Datei auf einem GNU/Linux-System, wenn das Dateisystem mit dem Argument "`noatime`" eingebunden wurde.

Anmerkung
Die mount-Argument "`noatime`" und "`relatime`" wurden eingeführt, um in normalen Anwendungssituationen die Leistungsfähigkeit der Systeme beim Lesen vom Dateisystem zu verbessern. Der einfache Vorgang zum Lesen einer Datei beinhaltet bei aktivierter "`strictatime`"-Option den zeitaufwendigen Schreibvorgang zum Aktualisieren des **atime**-Attributs. Jedoch wird das **atime**-Attribut selten benutzt, außer bei der mbox(5)-Datei. Lesen Sie dazu mount(8).

Verwenden Sie den Befehl touch(1), um den Zeitstempel vorhandener Dateien zu ändern.

Zeitstempel werden in der Ausgabe des `ls`-Befehls lokalisiert angezeigt, wenn ein nicht-englisches Gebietsschema (Locale) eingestellt ist (bei "`de_DE.UTF-8`" also im deutschen Datumsformat.

```
$ LANG=C  ls -l foo
-rw-rw-r-- 1 penguin penguin 0 Oct 16 21:35 foo
$ LANG=en_US.UTF-8  ls -l foo
-rw-rw-r-- 1 penguin penguin 0 Oct 16 21:35 foo
$ LANG=fr_FR.UTF-8  ls -l foo
-rw-rw-r-- 1 penguin penguin 0 oct. 16 21:35 foo
```

Tipp
Lesen Sie Abschnitt 9.3.4, um die Ausgabe von "`ls -l`" anzupassen.

1.2.7 Links

Es gibt zwei Methoden, um die Datei "`foo`" mit einem anderen Dateinamen, z.B. "`bar`" zu verbinden:

- Hard link
  - doppelter Name für eine existierende Datei
  - "`ln foo bar`"
- symbolischer Link (auch Symlink genannt)
  - eine spezielle Datei, die über deren Namen auf eine andere Datei verweist

- "ln -s foo bar"

Schauen Sie sich folgendes Beispiel bezüglich Änderungen bei der Linkanzahl und den feinen Unterschieden im Resultat des rm-Befehls an:

```
$ umask 002
$ echo "Original Content" > foo
$ ls -li foo
1449840 -rw-rw-r-- 1 penguin penguin 17 Oct 16 21:42 foo
$ ln foo bar     # hard link
$ ln -s foo baz  # symlink
$ ls -li foo bar baz
1449840 -rw-rw-r-- 2 penguin penguin 17 Oct 16 21:42 bar
1450180 lrwxrwxrwx 1 penguin penguin  3 Oct 16 21:47 baz -> foo
1449840 -rw-rw-r-- 2 penguin penguin 17 Oct 16 21:42 foo
$ rm foo
$ echo "New Content" > foo
$ ls -li foo bar baz
1449840 -rw-rw-r-- 1 penguin penguin 17 Oct 16 21:42 bar
1450180 lrwxrwxrwx 1 penguin penguin  3 Oct 16 21:47 baz -> foo
1450183 -rw-rw-r-- 1 penguin penguin 12 Oct 16 21:48 foo
$ cat bar
Original Content
$ cat baz
New Content
```

Der harte Link kann innerhalb des gleichen Dateisystems erstellt werden und teilt sich die gleiche Inode-Nummer mit der Originaldatei, welche Sie mittels der Option "-i" des ls(1)-Befehls herausfinden können.

Der Symlink hat immer nominal die Dateiberechtigungen "rwxrwxrwx", wie in dem obigen Beispiel zu sehen, wobei die letztlich wirksamen Zugriffsrechte durch die Berechtigungen der Datei, auf die verwiesen wird, bestimmt werden.



Achtung
Es ist grundsätzlich eine gute Idee, keine komplizierten symbolischen oder harten Link zu erstellen, außer Sie haben einen sehr guten Grund dafür. Es könnte Ihnen sonst Albträume bescheren, wenn die logische Kombination von symbolischen Links zu Schleifen im Dateisystem führt.

Anmerkung
Im allgemeinen sollten Sie symbolische Links gegenüber harten Links bevorzugen, außer Sie haben einen guten Grund, einen harten Link zu verwenden.

Das Verzeichnis "." verweist auf das Verzeichnis, in dem es erscheint, daher steht die Anzahl der Links in jedem neuen Verzeichnis auf 2. ".." verweist auf das jeweilige Elternverzeichnis, daher erhöht sich die Anzahl der Links für dieses Verzeichnis mit jedem Hinzufügen eines neuen Unterverzeichnisses.

Wenn Sie gerade von Windows zu Linux wechseln, wird Ihnen bald klar werden, wie gut das Verfahren zum Verknüpfen von Dateinamen bei Unix aufgebaut ist, verglichen mit dem Windows-Equivalent der "Shortcuts". Weil es im Dateisystem implementiert ist, bemerken Anwendungen keinen Unterschied zwischen der verknüpften Datei und dem Original. Im Fall eines harten Links gibt es auch tatsächlich keinen Unterschied.

1.2.8 Benannte Pipes (FIFOs)

Eine benannte Pipe (named pipe) ist eine Datei, die sich wie ein Rohr verhält: wenn Sie etwas in die Datei hineingeben, kommt es am anderen Ende wieder heraus. Daher wird sie auch FIFO genannt, vom englischen First-In-First-Out: das erste, was Sie in das Rohr hineingeben, kommt als erstes am anderen Ende wieder heraus.

Wenn Sie in eine benannte Pipe schreiben, wird der Prozess, der in die Pipe schreibt, nicht eher beendet, als bis die zu schreibenden Informationen von der Pipe gelesen wurden. Wenn Sie aus einer benannten Pipe lesen, wartet der lesende Prozess, bis es nichts mehr zu lesen gibt, bevor er beendet wird. Die Größe der Pipe ist immer Null --- sie speichert keine Informationen, sie verbindet lediglich zwei Prozesse, wie dies die Shell mittels "|" tut. Da diese Pipe allerdings einen Namen hat, müssen die beiden Prozesse nicht auf der gleichen Befehlszeile und nicht einmal unter dem gleichen Benutzerkonto laufen. Pipes waren eine sehr einflußreiche Innovation von Unix.

Probieren Sie zum Beispiel folgendes:

```
$ cd; mkfifo mypipe
$ echo "hello" >mypipe & # put into background
[1] 8022
$ ls -l mypipe
prw-rw-r-- 1 penguin penguin 0 Oct 16 21:49 mypipe
$ cat mypipe
hello
[1]+  Done                    echo "hello" >mypipe
$ ls mypipe
mypipe
$ rm mypipe
```

1.2.9 Sockets

Sockets werden von allen Internet-Übertragungen, Datenbanken und dem Betriebssystem selbst umfassend genutzt. Sie sind den benannten Pipes (FIFO) ähnlich und erlauben Prozessen, Informationen sogar zwischen verschiedenen Computern auszutauschen. Für die Nutzung von Sockets müssen diese Prozesse nicht zur gleichen Zeit laufen und auch nicht Kindprozesse desselben Urprozesses sein. Ein Socket ist der Endpunkt für die Interprozesskommunikation (IPC). Der Austausch von Informationen kann über das Netzwerk zwischen verschiedenen Rechnern stattfinden. Die beiden meist verwendeten sind der Internet-Socket und der Unix-Domain-Socket.

Tipp
"`ss -t -u -a`" (aus dem `iproute2`-Paket) bietet eine sehr nützliche Übersicht aller auf einem entsprechenden System geöffneten Sockets.

1.2.10 Gerätedateien

Gerätedateien verweisen auf physikalische oder virtuelle Geräte auf Ihrem System, wie Ihre Festplatte, Grafikkarte, Bildschirm oder Tastatur. Ein Beispiel für ein virtuelles Gerät ist die Konsole, repräsentiert durch "`/dev/console`".

Es gibt zwei Arten von Gerätedateien:

- **Zeichenorientierte Geräte**
  - Zugriff immer nur auf ein Zeichen gleichzeitig
  - 1 Zeichen = 1 Byte
  - z.B. Tastatur, serieller Port, …
- **Blockorientierte Geräte**
  - Zugriff in größeren Einheiten, Blocks genannt
  - 1 Block > 1 Byte
  - z.B. Festplatten, …

Sie können Gerätedateien lesen und schreiben, allerdings enthalten diese möglicherweise Binärdaten, die für Menschen nur unverständliches Kauderwelsch darstellen. Daten direkt in diese Dateien zu schreiben, ist manchmal zur Fehlersuche bei Verbindungsproblemen nützlich. Zum Beispiel können Sie eine Textdatei zur Drucker-Gerätedatei "`/dev/lp0`" schicken oder Modembefehle an den entsprechenden seriellen Port "`/dev/ttyS0`". Aber wenn hierbei nicht mit Vorsicht gearbeitet wird, kann dies eine große Katastrophe verursachen. Seien Sie also auf der Hut.

Anmerkung
Für den normalen Zugriff auf den Drucker verwenden Sie lp(1).

Die Geräte-Node-Nummer wird angezeigt, indem Sie ls(1) wie folgt ausführen:

```
$ ls -l /dev/sda /dev/sr0 /dev/ttyS0 /dev/zero
brw-rw---T  1 root disk     8,  0 Oct 16 20:57 /dev/sda
brw-rw---T+ 1 root cdrom   11,  0 Oct 16 21:53 /dev/sr0
crw-rw---T  1 root dialout  4, 64 Oct 16 20:57 /dev/ttyS0
crw-rw-rw-  1 root root     1,  5 Oct 16 20:57 /dev/zero
```

- "`/dev/sda`" hat die major-Gerätenummer 8 und die minor-Gerätenummer 0. Es ist lesbar/schreibbar für Benutzer, die der Gruppe `disk` angehören.
- "`/dev/sr0`" hat die major-Gerätenummer 11 und die minor-Gerätenummer 0. Es ist lesbar/schreibbar für Benutzer, die der Gruppe `cdrom` angehören.
- "`/dev/ttyS0`" hat die major-Gerätenummer 4 und die minor-Gerätenummer 64. Es ist lesbar/schreibbar für Benutzer, die der Gruppe `dialout` angehören.
- "`/dev/zero`" hat die major-Gerätenummer 1 und die minor-Gerätenummer 5. Es ist lesbar/schreibbar für jeden.

Auf modernen Linux-Systemen wird das Dateisystem unterhalb von "`/dev/`" automatisch durch den udev(7)-Mechanismus bestückt.

1.2.11 Spezielle Gerätedateien

Es gibt einige spezielle Gerätedateien:

Gerätedatei	Aktion	Beschreibung der Antwort
`/dev/null`	lesen	gibt das "end-of-file"-Zeichen (EOF) zurück
`/dev/null`	schreiben	gibt nichts zurück (ein bodenloses Loch für Datenmüll)
`/dev/zero`	lesen	gibt das "`\0`"-Zeichen (NUL) zurück (das ist nicht dasselbe wie die Zahl Null in ASCII)
`/dev/random`	lesen	gibt zufällige Zeichen von einem realen Zufallszahlengenerator zurück, der echte Entropie liefert (langsam)
`/dev/urandom`	lesen	gibt zufällige Zeichen von einem kryptografisch sicheren Pseudo-Zufallszahlengenerator zurück
`/dev/full`	schreiben	gibt den "disk-full"-Fehler (ENOSPC, Platte voll) zurück

Tabelle 1.10: Liste spezieller Gerätedateien

Diese werden oft in Verbindung mit der Shell-Umleitung genutzt (lesen Sie Abschnitt 1.5.8).

1.2.12 procfs und sysfs

procfs und sysfs, eingebunden unter "/proc" und "/sys", sind Pseudo-Dateisysteme und bringen interne Datenstrukturen des Kernel in den Userspace. Mit anderen Worten: diese Einträge sind virtuell, sie fungieren als komfortables Fenster zu den Operationen des Betriebssystems.

Das Verzeichnis "/proc" enthält (neben anderen Dingen) ein Unterverzeichnis für jeden Prozess, der auf dem System läuft, welches nach der Prozess-ID (PID) benannt ist. Systemwerkzeuge, die auf Prozessinformationen zugreifen, wie ps(1), bekommen ihre Informationen aus dieser Verzeichnisstruktur.

Die Verzeichnisse unterhalb von "/proc/sys/" enthalten Schnittstellen, um bestimmte Kernel-Parameter zur Laufzeit zu ändern. (Sie können dies möglicherweise auch mit dem spezialisierten Befehl sysctl(8) oder seiner Preload-/Konfigurationsdatei "/etc/sysctl.d/*.conf" erreichen.)

Die Leute geraten regelmäßig in Panik, wenn Sie eine bestimmte Datei bemerken - "/proc/kcore" - die wirklich riesig ist. Dies ist (mehr oder weniger) eine Kopie des Inhalts vom Arbeitsspeicher Ihres Rechners. Sie wird für Fehlersuche im Kernel verwendet. Es ist eine virtuelle Datei, die auf den Arbeitsspeicher verweist, sorgen Sie sich daher nicht über ihre Größe.

Das Verzeichnis unterhalb von "/sys" enthält vom Kernel exportierte Datenstrukturen, deren Attribute sowie die Anbindungen zwischen ihnen. Es enthält auch Schnittstellen, um bestimmte Kernel-Parameter zur Laufzeit zu ändern.

Lesen Sie "proc.txt(.gz)", "sysfs.txt(.gz)" sowie weitere dazugehörige Dokumente aus der Linux-Kernel-Dokumentation ("/usr/share/doc/linux-doc-*/Documentation/filesystems/*") aus dem Paket linux-doc-*

1.2.13 tmpfs

tmpfs ist ein temporäres Dateisystem, das alle Dateien im virtuellen Speicher hält. Die Daten des tmpfs im page cache des Speichers können - falls nötig - in den Swap-Bereich auf der Festplatte ausgelagert werden.

Das Verzeichnis "/run" wird im frühen Stadium des Boot-Prozesses als tmpfs eingebunden. Dadurch ist es möglich, in dieses Verzeichnis zu schreiben, auch wenn "/" als read-only (nur lesen) eingebunden ist. Dies ist der neue Ablageort für die Speicherung kurzlebiger Statusdateien und ersetzt verschiedene Orte, die im Filesystem Hierarchy Standard Version 2.3 beschrieben sind:

- "/var/run" → "/run"
- "/var/lock" → "/run/lock"
- "/dev/shm" → "/run/shm"

Lesen Sie "tmpfs.txt(.gz)" in der Linux-Kernel-Dokumentation ("/usr/share/doc/linux-doc-2.6.*/Documenta
aus dem Paket linux-doc-2.6.*.

1.3 Midnight Commander (MC)

Midnight Commander (MC) ist ein GNU "Schweizer Messer" für die Linux-Konsole und andere Terminal-Umgebungen. Er bringt Neulingen eine menügeführte Konsole, die viel einfacher zu erlernen ist wie die Standard-Unix-Befehle.

Sie müssen möglicherweise wie folgt das Midnight Commander-Paket mit Namen "mc" installieren:

```
$ sudo apt-get install mc
```

Verwenden Sie den mc(1)-Befehl, um das Debian-System zu erforschen. Dies ist der beste Weg, um zu lernen. Entdecken Sie einige interessante Orte lediglich durch Verwendung der Pfeiltasten und der Enter-Taste.

- "/etc" und dessen Unterverzeichnisse;
- "/var/log" und dessen Unterverzeichnisse;
- "/usr/share/doc" und dessen Unterverzeichnisse;
- "/usr/sbin" und "/usr/bin".

1.3.1 Anpassen des MC

Um beim Schließen des MC das Arbeitsverzeichnis zu speichern und beim nächsten Start mit `cd` dorthin zu wechseln, empfehle ich, "`~/.bashrc`" zu modifizieren, um ein Skript dort zu integrieren, das vom `mc`-Paket bereitgestellt wird:

```
. /usr/lib/mc/mc.sh
```

Den Grund finden Sie unter mc(1) (unter der Option "`-P`"). (Falls Sie nicht verstehen, worum genau es hier geht, können Sie dies später erledigen.)

1.3.2 Starten von MC

MC kann wie folgt gestartet werden:

```
$ mc
```

MC kümmert sich über sein Menü um alle Dateioperationen und erfordert dabei nur minimale Anstrengungen des Benutzers. Drücken Sie einfach F1, um die Hilfe-Seiten zu bekommen. Sie können mit dem MC herumspielen, indem Sie einfach die Pfeiltasten und Funktionstasten drücken.

Anmerkung
In einigen Konsolen, wie gnome-terminal(1), könnten Tastendrücke der Funktionstasten vom Konsolenprogramm "gestohlen" werden. Bei `gnome-terminal` können Sie diese Funktionalität über "Bearbeiten" → "Einstellungen" → "Tastenkürzel" deaktivieren.

Wenn Sie Probleme mit der Zeichenkodierung feststellen und Zeichenmüll angezeigt wird, kann das Hinzufügen von "`-a`" zur MC-Befehlszeile helfen, Probleme zu vermeiden.

Falls dies Ihre Anzeigeprobleme mit MC nicht löst, lesen Sie Abschnitt 8.3.1.

1.3.3 Dateimanager in MC

Die Standardanzeige sind zwei Verzeichnisfenster, welche Dateilisten enthalten. Ein anderer nützlicher Modus ist, das rechte Fenster auf "Info" einzustellen, um dort Informationen zu Dateizugriffsrechten und anderem zu erhalten. Wenn der gpm(8)-Daemon läuft, kann man auch auf textbasierten Linux-Konsolen eine Maus verwenden. (Stellen Sie sicher, dass Sie die Umschalt-Taste drücken, um im MC das normale Ausschneiden-und-Einfügen-Verhalten zu bekommen.)

Taste	**Tastaturbefehl**
`F1`	Hilfe-Menü
`F3`	interner Dateibetrachter
`F4`	interner Texteditor
`F9`	Aufklapp-Menü aktivieren
`F10`	Midnight Commander beenden
`Tab`	zwischen den beiden Fenstern wechseln
`Einfg` oder `Strg-T`	Datei für eine mehrere Dateien betreffende Operation wie Kopieren markieren
`Del`	Datei löschen (seien Sie vorsichtig --- stellen Sie MC auf "sicheres Löschen")
Pfeiltasten	selbsterklärend

Tabelle 1.11: Die Tastaturbefehle von MC

1.3.4 Befehlszeilentricks in MC

- `cd`-Befehl wechselt das Verzeichnis, das im gewählten Fenster angezeigt wird.
- `Strg-Enter` oder `Alt-Enter` kopiert einen Dateinamen auf die Befehlszeile. Nutzen Sie dies mit cp(1) und mv(1) zusammen mit dem Editieren auf der Befehlszeile.
- `Alt-Tab` zeigt Auswahlmöglichkeiten für die Auto-Vervollständigung von Dateinamen.
- Die Start-Verzeichnisse für beide Fenster können MC als Argumente angegeben werden, zum Beispiel "`mc /etc /root`".
- `Esc` + `n-Taste` → `Fn` (d.h. `Esc` + `1` → `F1`, usw.; `Esc` + `0` → `F10`).
- Das Drücken von `Esc` vor einer anderen Buchstabentaste hat den gleichen Effekt wie das Drücken von `Alt` und der anderen Taste zusammen, d. h. drücken Sie `Esc` + `c` für `Alt-C`. `Esc` wird Meta-Taste genannt und manchmal als "`M-`" dargestellt.

1.3.5 Der interne Texteditor in MC

Der interne Editor hat ein interessantes Ausschneiden-und-Einfügen-Schema: das Drücken von `F3` setzt den Startpunkt einer zu markierenden Auswahl, ein weiteres `F3` markiert das Ende der Auswahl und hebt die Auswahl hervor. Dann können Sie Ihren Cursor verschieben. Wenn Sie `F6` drücken, wird der ausgewählte Bereich an die Cursor-Position verschoben. Wenn Sie `F5` drücken, wird der ausgewählte Bereich kopiert und an der Cursor-Position eingefügt. `F2` sichert die Datei. `F10` beendet das Ganze. Die meisten Pfeiltasten funktionieren intuitiv.

Dieser Editor kann direkt mit einer Datei gestartet werden, indem einer der folgenden Befehle benutzt wird:

```
$ mc -e filename_to_edit
```

```
$ mcedit filename_to_edit
```

Dies ist kein Multi-Fenster-Editor, aber man kann mehrere Linux-Konsolen verwenden, um den gleichen Effekt zu erreichen. Um Inhalte zwischen verschiedenen Fenstern hin und her zu kopieren, verwenden Sie die Alt-F*n*-Tasten, um zwischen den virtuellen Konsolen zu wechseln, und "Datei→Datei einfügen" oder "Datei→Kopie in Datei", um Teile einer Datei in eine andere Datei zu kopieren.

Dieser interne Editor kann auf Wunsch durch jeden anderen externen Editor ersetzt werden.

Außerdem verwenden viele Programme die Umgebungsvariablen "`$EDITOR`" oder "`$VISUAL`", um festzulegen, welcher Editor genutzt wird. Wenn Ihnen vim(1) oder nano(1) nicht behagen, möchten Sie diese vielleicht auf "`mcedit`" setzen, indem Sie folgende Zeilen zu "`~/.bashrc`" hinzufügen:

```
export EDITOR=mcedit
export VISUAL=mcedit
```

Ich empfehle, diese wenn möglich auf "`vim`" zu setzen.

Wenn Sie vim(1) nicht mögen, können Sie für die meisten Systemwartungsaufgaben auch mcedit(1) benutzen.

1.3.6 Der interne Dateibetrachter in MC

MC ist ein sehr toller Dateibetrachter. Er ist ein großartiges Werkzeug, um nach Wörtern in Dokumenten zu suchen. Ich verwende ihn immer für Dateien im `/usr/share/doc`-Verzeichnis. Dies ist der schnellste Weg, um Massen von Linux-Informationen zu durchsuchen. Dieser Betrachter kann direkt gestartet werden, indem einer der folgenden Befehle benutzt wird:

```
$ mc -v path/to/filename_to_view
```

```
$ mcview path/to/filename_to_view
```

1.3.7 Autostart-Funktionalitäten von MC

Drücken Sie Enter auf einer Datei und das zugehörige Programm wird den Inhalt der Datei verarbeiten (lesen Sie Abschnitt 9.4.11). Dies ist eine sehr praktische MC-Funktionalität.

Dateityp	Reaktion auf die Enter-Taste
ausführbare Datei	Befehl ausführen
Handbuchseiten-Datei	Inhalt an Betrachter-Software weiterleiten
html-Datei	Inhalt an Webbrowser weiterleiten
"`*.tar.gz`"- und "`*.deb`"-Datei	den Inhalt anzeigen, als wäre es ein Unterverzeichnis

Tabelle 1.12: Die Reaktion auf die Enter-Taste in MC

Damit diese Betrachter- und virtuellen Dateifunktionalitäten korrekt funktionieren können, sollte bei Dateien, deren Inhalt betrachtet werden kann, nicht die Ausführungs-Berechtigung gesetzt sein. Ändern Sie deren Status ansonsten mit chmod(1) oder über das MC-Dateimenü.

1.3.8 Virtuelles Dateisystem von MC

MC kann verwendet werden, um über das Internet auf Dateien zuzugreifen. Rufen Sie das Menü auf über F9, "`Enter`" und "h" aktiviert das virtuelle Dateisystem. Geben Sie eine URL in der Form "`sh://[benutzer@]rechner[:optionen]/`" ein, wodurch (mittels `ssh`) ein auf einem fernen Rechner liegendes Verzeichnis abgerufen wird, das dann wie ein lokales erscheint.

1.4 Die grundlegende Unix-ähnliche Arbeitsumgebung

Obwohl MC es Ihnen erlaubt, fast alles zu tun, ist es sehr wichtig für Sie, zu lernen, wie die Befehlszeilenwerkzeuge am Shell-Prompt aufgerufen werden, und mit der Unix-ähnlichen Arbeitsumgebung vertraut zu werden.

1.4.1 Die Login-Shell

Da die Login-Shell von einigen Programmen zur Systeminitialisierung verwendet werden könnte, ist es klug, diese auf bash(1) eingestellt zu lassen (vermeiden Sie es also auch, mittels chsh(1) die Login-Shell zu ändern).

Wenn Sie eine andere interaktive Shell nutzen möchten, stellen Sie dies in der GUI-Konfiguration des Terminal-Emulator-Programms ein oder legen Sie es in `~/.bashrc` fest, z.B. indem Sie dort "`exec /usr/bin/zsh -i -l`" oder "`exec /usr/bin/fish -i -l`" angeben.

Tipp
Obwohl sich POSIX-konforme Shells die grundsätzliche Syntax teilen, können sie sich beim Verhalten für Dinge wie Shell-Variablen oder Glob-Ersetzungen unterscheiden. Bitte konsultieren Sie die jeweilige Dokumentation bezüglich weiterer Details.

In dieser Lehrstunde ist mit interaktiver Shell immer `bash` gemeint.

1.4.2 Anpassen der bash

Sie können das Verhalten von bash(1) mittels "`~/.bashrc`" anpassen.

Probieren Sie zum Beispiel folgendes:

Paket	Popcon	Größe	POSIX-Shell	Beschreibung
bash	V:900, I:1000	7309	Ja	Bash: die GNU Bourne Again SHell (De-Facto-Standard)
bash-completion	V:41, I:964	1952	Nicht verfügbar	konfigurierbare Vervollständigung für die bash-Shell
dash	V:933, I:999	211	Ja	Debian Almquist Shell, gut für Shell-Scripting
zsh	V:38, I:67	2571	Ja	Z Shell: die Standard-Shell mit vielen Erweiterungen
tcsh	V:4, I:14	1366	Nein	TENEX C Shell: eine erweiterte Version der Berkeley csh
mksh	V:5.0, I:7.3	7713	Ja	Eine Version der Korn-Shell
csh	V:1.0, I:5.0	348	Nein	OpenBSD C-Shell, eine Version der Berkeley csh
sash	V:0.3, I:4.6	1249	Ja	Stand-alone Shell mit integrierten Befehlen (nicht gedacht für Standard "/usr/bin/sh")
ksh	V:0.2, I:7.5	65	Ja	die echte AT&T-Version der Korn Shell
rc	V:0.08, I:0.69	182	Nein	Implementierung der AT&T Plan 9 rc-Shell
posh	V:0.01, I:0.21	195	Ja	Policy-compliant (grundsatz-konforme) Ordinary SHell (pdksh-Abkömmling)

Tabelle 1.13: Liste von Shell-Programmen

```
# enable bash-completion
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

# CD upon exiting MC
. /usr/lib/mc/mc.sh

# set CDPATH to a good one
CDPATH=.:/usr/share/doc:~:~/Desktop:~
export CDPATH

PATH="${PATH+$PATH:}/usr/sbin:/sbin"
# set PATH so it includes user's private bin if it exists
if [ -d ~/bin ] ; then
  PATH="~/bin${PATH+:$PATH}"
fi
export PATH

EDITOR=vim
export EDITOR
```

Tipp
Es gibt weitere Tipps zur Anpassung der bash, zum Beispiel Abschnitt 9.3.6 in Kapitel 9.

Tipp
Das `bash-completion`-Paket enthält eine konfigurierbare Auto-Vervollständigung für `bash`.

1.4.3 Spezielle Tastendrücke

In einer Unix-ähnlichen Umgebung gibt es ein paar Tastendrücke, die spezielle Bedeutungen haben. Bitte beachten Sie, dass auf einer normalen textbasierten Linux-Konsole nur die linke `Strg`- und `Alt`-Taste wie erwartet funktionieren. Hier ein paar erwähnenswerte Tastendrücke, die Sie sich merken sollten:

Taste	Beschreibung des Tastaturbefehls
`Ctrl-U`	alle Zeichen vor dem Cursor löschen
`Ctrl-H`	ein Zeichen vor dem Cursor löschen
`Ctrl-D`	Eingabe beenden (Shell beenden, wenn Sie eine benutzen)
`Ctrl-C`	einen laufenden Prozess beenden
`Ctrl-Z`	Programm vorübergehend stoppen, indem es in den Hintergrund geschoben wird
`Ctrl-S`	Ausgabe auf den Bildschirm anhalten
`Ctrl-Q`	Ausgabe auf den Bildschirm wieder starten
`Ctrl-Alt-Del`	Neu starten/anhalten des Systems, siehe dazu inittab(5)
`Linke-Alt-Taste` (optional `Windows-Taste`)	Meta-Taste für Emacs und ähnliche Benutzerschnittstellen
`Up-arrow`	Die zuletzt ausgeführten Befehlen der `bash` anzeigen
`Ctrl-R`	Inkrementelle Suche in den zuletzt ausgeführten Befehlen der `bash`
`Tab`	Auto-Vervollständigung des Dateinamens auf der Befehlszeile in der `bash`
`Ctrl-V Tab`	Eingabe von `Tab` ohne Auto-Vervollständigung auf der Befehlszeile in der `bash`

Tabelle 1.14: Liste der Tastaturbefehle für bash

Tipp
Die Terminal-Funktionalität von `Strg-S` kann mittels stty(1) deaktiviert werden.

1.4.4 Mausoperationen

Bei Mausoperationen für text-basierte Anwendungen werden auf Debian-Systemen über einige Kniffe zwei 2 Stile miteinander vermischt:

- Traditionelle Mausoperationen im Unix-Stil:
  - Verwendung von 3 Tasten (Klicken)
  - Verwendung von PRIMARY
  - wird verwendet von X-Anwendungen wie `xterm` sowie von textbasierten Anwendungen in der Linux-Konsole
- Moderne Mausoperationen im GUI-Stil:
  - Verwendung von 2 Knöpfen (Ziehen + Klicken)
  - Verwendung von PRIMARY und ZWISCHENABLAGE
  - wird verwendet in modernen GUI-Anwendungen wie `gnome-terminal`

Aktion	Reaktion
Links-Klick und Ziehen der Maus	Bereich als PRIMARY-Auswahl festlegen
Links-Klick	Beginn des Bereich für die PRIMARY-Auswahl festlegen
Rechts-Klick (traditionell)	Ende des Bereich für die PRIMARY-Auswahl festlegen
Rechts-Klick (modern)	Kontextmenü (ausschneiden/kopieren/einfügen)
Mittel-Klick oder Umschalt-`Einfüg`	PRIMARY-Auswahl an Cursor-Position einfügen
`Ctrl-X`	PRIMARY-Auswahl ausschneiden und in ZWISCHENABLAGE einfügen
`Strg-C` (`Umschalt-Strg-C` im Terminal)	PRIMARY-Auswahl in ZWISCHENABLAGE kopieren
`Ctrl-V`	Inhalt der ZWISCHENABLAGE an Cursor-Position einfügen

Tabelle 1.15: Liste von Mausoperationen und zugehörigen Tastenaktionen auf Debian-Systemen

Hierbei ist die PRIMARY-Auswahl der hervorgehobene Textbereich. Innerhalb des Terminal-Programms wird `Umschalt-Str` anstelle von `Strg-C` zum Kopieren benutzt, um zu vermeiden, dass laufende Programme beendet werden.

Das Mittelrad auf einer modernen Radmaus wird als mittlerer Mausknopf betrachtet und kann für den Mittel-Klick verwendet werden. Das gleichzeitige Drücken des rechten und linken Maus-Knopfes wirkt in einem 2-Knopf-Maus-System wie ein Mittel-Klick.

Um eine Maus in textbasierten Linux-Konsolen verwenden zu können, müssen Sie gpm(8) als Daemon laufen haben.

1.4.5 Der Pager

Der less(1)-Befehl ist ein erweiterter Pager (Dateiinhalt-Browser). Er liest die Datei, die per Befehlsargument oder Standardeingabe angegeben wird. Drücken Sie "h", wenn Sie beim Anzeigen von Dateiinhalten mit `less` Hilfe benötigen. Er hat viel mehr Fähigkeiten als more(1) und kann sogar noch weiter aufgebohrt werden, indem "`eval $(lesspipe)`" oder "`eval $(lessfile)`" im Shell-Startskript ausgeführt wird. Weiteres dazu finden Sie in "`/usr/share/`. Die Option "`-R`" erlaubt die Ausgabe von Rohformat-Zeichen und aktiviert ANSI-Color-Escape-Sequenzen. Lesen Sie dazu less(1).

Tipp
In `less` können Sie "h" eingeben, um den Hilfe-Bildschirm anzuzeigen, mit "`/`" oder "?" können Sie nach einer Zeichenfolge suchen, und mit "`-i`" schalten Sie zwischen Groß- und Kleinschreibung um.

1.4.6 Der Texteditor

Sie sollten lernen, eine der Varianten von Vim oder Emacs zu nutzen, die auf Unix-ähnlichen Systemen beliebt sind.

Ich denke, sich an Vim-Befehle zu gewöhnen, ist das richtige, da der Vi-Editor in der Linux-/Unix-Welt immer vorhanden ist. (Eigentlich sind der originale `vi` oder auch der neue `nvi` Programme, die Sie überall finden. Ich habe stattdessen für Neulinge Vim ausgewählt, da es über `F1` eine Hilfe anbietet, den anderen sehr ähnlich und gleichzeitig noch leistungsfähiger ist.)

Wenn Sie stattdessen entweder Emacs oder XEmacs als Ihren Editor auswählen, ist dies in der Tat eine andere gute Wahl, speziell zum Programmieren. Emacs enthält ebenso eine Fülle an weiteren Funktionen, dazu gehört die Arbeit als Newsreader, Verzeichnis-Editor, Mail-Programm usw. Wenn Sie ihn zum Programmieren oder Editieren von Shell-Skripten verwenden, erkennt er intelligent das Format, an dem Sie arbeiten, und versucht Sie dabei zu unterstützen. Einige Leute behaupten, dass das einzige Programm, dass Sie unter Linux benötigen, Emacs sei. Jetzt zehn Minuten Emacs lernen kann später Stunden an Zeit sparen. Es wird dringend empfohlen, beim Erlernen von Emacs das GNU Emacs-Handbuch als Referenz zur Hand zu haben.

All diese Programme enthalten gewöhnlich Unterrichtseinheiten für Sie, damit Sie sie erlernen können. Starten Sie Vim, indem Sie "`vim`" eingeben, und drücken Sie die F1-Taste. Sie sollten zumindest die ersten 35 Zeilen lesen. Absolvieren Sie dann den Online-Trainingskurs, indem Sie mit dem Cursor auf "`|tutor|`" gehen, und drücken Sie `Strg-]`.

Anmerkung
Gute Editoren wie Vim und Emacs können Texte mit UTF-8 und anderen exotischen Kodierungen korrekt darstellen. Es ist eine gute Idee, für die GUI-Umgebung eine UTF-8-Locale zu verwenden und dafür benötigte Programme und Schriften zu installieren. Editoren haben Optionen, um die Dateikodierung unabhängig von der GUI-Umgebung zu setzen. Bitte ziehen Sie deren Dokumentation bezüglich Multibyte-Text zu Rate.

1.4.7 Einen Standard-Texteditor einstellen

Debian enthält eine Reihe verschiedener Editoren. Wir empfehlen wie oben bereits erwähnt, das `vim`-Paket zu installieren.

Debian bietet über den Befehl "`/usr/bin/editor`" einen einheitlichen Zugriff auf den Standard-Editor des Systems, so dass andere Programme (z. B. reportbug(1)) ihn aufrufen können. Sie können ihn wie folgt ändern:

```
$ sudo update-alternatives --config editor
```

Ich empfehle Neulingen, die Auswahl "`/usr/bin/vim.basic`" gegenüber `/usr/bin/vim.tiny` zu bevorzugen, da er Syntaxhervorhebung beherrscht.

Tipp
Viele Programme nutzen die Umgebungsvariablen "`$EDITOR`" oder "`$VISUAL`", um zu entscheiden, welcher Editor verwendet wird (lesen Sie dazu Abschnitt 1.3.5 und Abschnitt 9.4.11). Zwecks Konsistenz auf Debian-Systemen sollten Sie dies auf "`/usr/bin/editor`" setzen. (Früher war "`$EDITOR`" auf "`ed`" und "`$VISUAL`" auf "`vi`" gesetzt.)

1.4.8 Verwenden von vim

Der aktuelle vim(1) startet selbsttätig mit der schlauen "`nocompatible`"-Option und wechselt in den `NORMAL`-Modus.[1]

Bitte nutzen Sie das "`vimtutor`"-Programm, um die Verwendung von `vim` über einen interaktiven Einführungskurs zu erlernen.

Das `vim`-Programm ändert seine Reaktion auf Tastatureingaben abhängig vom aktiven **Modus**: Tastatureingaben in den Puffer erfolgen überwiegend im `EINFÜGEN`- oder `ERSETZEN`-Modus. Das Bewegen des Cursors findet meist im `NORMAL`-Modus statt. Interaktive Auswahlen erledigen Sie im `VISUELL`-Modus. Die Eingabe von "`:`" im `NORMAL`-Modus schaltet den **Modus** in den `Ex`-Modus um. Im `Ex`-Modus werden Befehle akzeptiert.

Tipp
Vim bringt das **Netrw**-Paket mit. Netrw unterstützt das Lesen und Schreiben von Dateien sowie das Durchsuchen von Verzeichnissen über das Netzwerk und lokal! Probieren Sie Netrw, indem Sie "`vim .`" ausführen (also ein Punkt als Argument zu vim) und lesen Sie das Handbuch mittels "`:help netrw`".

Informationen zur fortgeschrittenen Konfiguration von `vim` finden Sie in Abschnitt 9.2.

1.4.9 Aufzeichnen der Shell-Aktivitäten

Die Ausgabe eines Shell-Befehls könnte aus Ihrem Bildschirm herauslaufen und für immer verloren sein. Es ist nützlich, Shell-Aktivitäten in einer Datei zu protokollieren, um sie später kontrollieren zu können. Diese Art der Aufzeichnung ist besonders wichtig für jegliche Systemadministrationsaufgaben.

[1] Auch ältere `vim`-Versionen können im schlauen "`nocompatible`"-Modus genutzt werden, wenn Sie mit der Option "`-N`" gestartet werden.

Modus	Tastatureingabe	Aktion
NORMAL	`:help|only`	die Hilfedatei anzeigen
NORMAL	`:e dateiname.ext`	neuen Puffer öffnen, um `dateiname.ext` zu bearbeiten
NORMAL	`:w`	Originaldatei mit aktuellem Puffer überschreiben
NORMAL	`:w filename.ext`	aktuellen Puffer in `filename.ext` schreiben
NORMAL	`:q`	`vim` beenden
NORMAL	`:q!`	Beenden von `vim` erzwingen
NORMAL	`:only`	alle anderen vorhandenen Fenster schließen
NORMAL	`:set nocompatible?`	prüfen, ob `vim` im schlauen `nocompatible`-Modus läuft
NORMAL	`:set nocompatible`	`vim` in den schlauen `nocompatible`-Modus versetzen
NORMAL	`i`	in den `EINGABE`-Modus wechseln
NORMAL	`R`	in den `ERSETZEN`-Modus wechseln
NORMAL	`v`	in den `VISUELL`-Modus wechseln
NORMAL	`V`	in den zeilenweise `VISUELLEN` Modus wechseln
NORMAL	Strg-`V`	in den blockweise `VISUELLEN` Modus wechseln
außer `TERMINAL-JOB`	ESC-Taste	in den `NORMAL`-Modus wechseln
NORMAL	`:term`	in den `TERMINAL-JOB`-Modus wechseln
`TERMINAL-NORMAL`	`i`	in den `TERMINAL-JOB`-Modus wechseln
`TERMINAL-JOB`	Strg-`W N` (oder Strg-`\` Strg-`N`)	in den `TERMINAL-NORMAL`-Modus wechseln
`TERMINAL-JOB`	Strg-`W :`	in den `Ex`-Modus in `TERMINAL-NORMAL` wechseln

Tabelle 1.16: Liste grundlegender Vim-Tastenkürzel

Tipp
Der neue Vim (Version>=8.2) kann über seinen `TERMINAL-JOB`-Modus verwendet werden, um Shell-Aktivitäten sauber aufzuzeichnen. Näheres in Abschnitt 1.4.8.

Die grundsätzliche Methode, um die Aktivitäten auf der Shell aufzuzeichnen ist, sie via script(1) laufen zu lassen.

Probieren Sie zum Beispiel folgendes:

```
$ script
Script started, file is typescript
```

Führen Sie alle gewünschten Shell-Befehle via `script` aus.

Drücken Sie `Strg-D`, um `script` zu beenden.

```
$ vim typescript
```

Lesen Sie Abschnitt 9.1.1 .

1.4.10 Grundlegende Unix-Befehle

Wir wollen einige grundlegende Unix-Befehle lernen. Ich verwende hier "Unix" in seiner allgemeinen Bedeutung. Alle Unix-Clone-Betriebssysteme bieten normalerweise entsprechende gleichbedeutende Befehle. Das Debian-System ist hier keine Ausnahme. Sorgen Sie sich nicht, wenn einige Befehle jetzt bei Ihnen nicht wie gewünscht funktionieren. Falls `alias` in der Shell verwendet wird, sind die zugehörigen Befehlsausgaben unterschiedlich. Diese Beispiele sind nicht dazu gedacht, in dieser Reihenfolge ausgeführt zu werden.

Probieren Sie die folgenden Befehle von einem nicht-privilegierten Benutzerkonto:

Anmerkung
Unix hat die Tradition, Dateinamen zu verstecken, die mit einem "`.`" beginnen. Dies sind traditionell Dateien, die Konfigurationsinformationen und Benutzereinstellungen enthalten.
Bezüglich des `cd`-Befehls schauen Sie unter bash-builtins(7).
Der Standard-Pager auf einem reinen Debian-System ist more(1), der nicht rückwärts scrollen kann. Indem Sie das `less`-Paket mittels "`apt-get install less`" installieren, wird less(1) zum Standard-Pager und Sie können mit den Pfeiltasten rückwärts scrollen.
Die Zeichen "`[`" und "`]`" in dem regulären Ausdruck des obigen Befehls "`ps aux | grep -e "[e]xim4*"`" vermeiden, dass `grep` sich selbst findet. Das "`4*`" in dem regulären Ausdruck steht für 0 oder mehr Wiederholungen des Zeichens "4", wodurch `grep` sowohl "`exim`" wie auch "`exim4`" findet. Obwohl "`*`" in dem Dateinamen und dem regulären Ausdruck verwendet wird, ist ihre Bedeutung unterschiedlich. Lernen Sie die regulären Ausdrücke von grep(1).

Bitte durchlaufen Sie zur Übung verschiedene Verzeichnisse und erforschen Sie das System. Wenn Sie Fragen zu einem der Konsolenbefehle haben, seien Sie sicher, dass Sie die Handbuchseite gelesen haben.

Probieren Sie zum Beispiel folgendes:

```
$ man man
$ man bash
$ man builtins
$ man grep
$ man ls
```

An den Stil der Handbuchseiten kann man sich eventuell etwas schwer gewöhnen, da sie eher knapp gehalten sind, speziell die älteren, traditionellen. Aber wenn Sie sich einmal daran gewöhnt haben, werden Sie die knappe Form zu schätzen wissen.

Bitte beachten Sie, dass viele der Unix-ähnlichen Befehle von GNU und BSD eine kurze Hilfe anzeigen, wenn Sie auf eine der folgenden Arten aufgerufen werden (oder in einigen Fällen ohne jegliche Argumente):

Befehl	Beschreibung
`pwd`	Name des derzeitigen/Arbeits-Verzeichnisses
`whoami`	derzeitigen Benutzernamen anzeigen
`id`	derzeitige Benutzeridentität anzeigen (Name, uid (Benutzer-ID), gid (Gruppen-ID) und zugehörige Gruppen)
`file foo`	den Dateityp der Datei "*foo*" anzeigen
`type -p befehlsname`	den Speicherort des Befehls *befehlsname* anzeigen
`which befehlsname`	, ,
`type befehlsname`	Informationen zum Befehl *befehlsname* anzeigen
`apropos schlüsselwort`	Befehle mit Bezug zu *schlüsselwort* finden
`man -k schlüsselwort`	, ,
`whatis befehlsname`	eine einzeilige Kurzbeschreibung des Befehls *befehlsname* anzeigen
`man -a befehlsname`	Beschreibung zum Befehl *befehlsname* anzeigen (Unix-Stil)
`info befehlsname`	eher lange Beschreibung zum Befehl "*befehlsname*" anzeigen (GNU-Stil)
`ls`	Inhalte eines Verzeichnisses auflisten (keine versteckten Dateien und Verzeichnisse)
`ls -a`	Inhalte eines Verzeichnisses auflisten (alle Dateien und Verzeichnisse)
`ls -A`	Inhalte eines Verzeichnisses auflisten (nahezu alle Dateien und Verzeichnisse, "`..`" und "`.`" werden übersprungen)
`ls -la`	alle Inhalte eines Verzeichnisses mit detaillierten Informationen auflisten
`ls -lai`	alle Inhalte eines Verzeichnisses mit Inode-Nummer und detaillierten Informationen auflisten
`ls -d`	alle Verzeichnisse im derzeitigen Verzeichnis auflisten
`tree`	Verzeichnisbaum-Inhalte anzeigen
`lsof foo`	"geöffnet"-Status der Datei *foo* anzeigen
`lsof -p pid`	Dateien auflisten, die von dem Prozess mit der ID *pid* geöffnet wurden
`mkdir foo`	ein neues Verzeichnis *foo* im derzeitigen Verzeichnis erstellen
`rmdir foo`	das Verzeichnis *foo* im derzeitigen Verzeichnis löschen
`cd foo`	in das Verzeichnis *foo* im derzeitigen Verzeichnis oder im in der Variable "`$CDPATH`" enthaltenen Verzeichnis wechseln
`cd /`	in das root-Verzeichnis wechseln
`cd`	in das Heimatverzeichnis des derzeitigen Benutzers wechseln
`cd /foo`	in das Verzeichnis mit dem absoluten Pfad "*/foo*" wechseln
`cd ..`	in das übergeordnete Verzeichnis (Eltern-Verzeichnis) wechseln
`cd ~foo`	in das Heimatverzeichnis des Benutzers *foo* wechseln
`cd -`	in das vorherige Verzeichnis wechseln
`</etc/motd pager`	Inhalt von "`/etc/motd`" mit dem Standard-Pager anzeigen
`touch leeredatei`	eine leere Datei mit Namen *leeredatei* erzeugen
`cp foo bar`	die vorhandene Datei *foo* in eine neue Datei *bar* kopieren
`rm unnützedatei`	die Datei *unnützedatei* entfernen
`mv foo bar`	die vorhandene Datei *foo* umbenennen zum neuen Namen *bar* (*bar* darf nicht existieren)
`mv foo bar`	die vorhandene Datei *foo* an einen neuen Ort *bar/foo* verschieben (das Verzeichnis *bar* muss existieren)
`mv foo bar/baz`	die vorhandene Datei *foo* an einen neuen Ort verschieben und gleichzeitig umbenennen nach *bar/baz* (das Verzeichnis *bar* muss existieren, aber der Dateiname *bar/baz* darf nicht existieren)
`chmod 600 foo`	die vorhandene Datei *foo* als nicht lesbar und nicht schreibbar für andere Benutzer einstellen (und nicht ausführbar für alle)
`chmod 644 foo`	die vorhandene Datei *foo* als lesbar, aber nicht schreibbar für andere Benutzer einstellen (und nicht ausführbar für alle)
`chmod 755 foo`	die vorhandene Datei *foo* als lesbar, aber nicht schreibbar für andere Benutzer einstellen (und als ausführbar für alle)
`find . -name Suchmuster`	passende Dateinamen finden durch Verwendung von Shell-*Suchmuster* (langsamer)
`locate -d . Suchmuster`	passende Dateinamen finden durch Verwendung von Shell-*Suchmuster* (schneller, nutzt eine regelmäßig erstellte Datenbank)

```
$ commandname --help
$ commandname -h
```

1.5 Der einfache Shell-Befehl

Jetzt haben Sie ein Gefühl, wie das Debian-System genutzt wird. Lassen Sie uns tiefer in den Mechanismus der Befehlsausführung einsteigen. Eine genaue Beschreibung finden Sie unter bash(1).

Ein einfacher Befehl ist eine Abfolge mehrerer Komponenten.

1. Zuweisung von Variablen (optional)
2. Befehlsname
3. Argumente (optional)
4. Weiterleitungen (optional: > , >> , < , << usw.)
5. Steuer-Operator (optional: && , | | , *newline* , ; , & , (,))

1.5.1 Befehlsausführung und Umgebungsvariablen

Die Werte einiger Umgebungsvariablen verändern teilweise das Verhalten von Unix-Befehlen.

Die Standardwerte von Umgebungsvariablen werden zunächst vom PAM-System gesetzt und dann eventuell von manchen Anwendungsprogrammen wieder zurückgesetzt.

- Das PAM-System (wie `pam_env`) könnte Umgebungsvariablen über "`/etc/pam.conf`", "`/etc/environment`" und "`/etc/default/locale`" setzen.
- Eine moderne grafische Desktop-Umgebung, von einer User-Mode systemd-Unit gestartet (siehe systemd.unit(5)) setzt die Umgebungsvariablen über eine Datei im Verzeichnis "`~/.config/environment.d/`".
- Benutzer-spezifische Programminitialisierungen könnten Umgebungsvariablen über "`~/.profile`", "`~/.bash_profile`" und "`~/.bashrc`" zurücksetzen.

1.5.2 Die "$LANG"-Variable

Das Standard-Gebietsschema wird über die Umgebungsvariable "$LANG" festgelegt; sie hat die Form "`LANG=xx_YY.UTF-8`" und wird vom Installer oder einer späteren GUI-Konfiguration (in GNOME z.B. über "Einstellungen" → "Region & Sprache" → "Sprache" / "Formate") gesetzt.

Anmerkung
Ich empfehle Ihnen, die Systemumgebung für den Anfang direkt über die "$LANG"-Variable zu konfigurieren und die Finger von den "$LC_*"-Variablen zu lassen, außer es ist absolut nötig.

Der vollständige Wert für das Gebietsschema (Locale), der an die "$LANG"-Variable übergeben wird, besteht aus drei Teilen: "`xx_YY.ZZZZ`".

Eine typische Befehlsausführung verwendet eine Shell-Sequenz wie die folgende:

locale-Wert	Bedeutung
xx	Sprach-Code laut ISO 639 (in Kleinbuchstaben) wie z.B. ”de”
YY	Länder-Code laut ISO 3166 (in Großbuchstaben) wie z.B. ”DE”
ZZZZ	Zeichenkodierung (am besten immer auf ”UTF-8” setzen)

Tabelle 1.18: Die drei Teile des locale-Wertes

empfohlenes Gebietsschema	Sprache (Gebiet)
en_US.UTF-8	Englisch (USA)
en_GB.UTF-8	Englisch (Großbritannien)
fr_FR.UTF-8	Französisch (Frankreich)
de_DE.UTF-8	Deutsch (Deutschland)
it_IT.UTF-8	Italienisch (Italien)
es_ES.UTF-8	Spanisch (Spanien)
ca_ES.UTF-8	Katalanisch (Spanien)
sv_SE.UTF-8	Schwedisch (Schweden)
pt_BR.UTF-8	Portugiesisch (Brasilien)
ru_RU.UTF-8	Russisch (Russland)
zh_CN.UTF-8	Chinesisch (Volksrepublik China)
zh_TW.UTF-8	Chinesisch (Taiwan)
ja_JP.UTF-8	Japanisch (Japan)
ko_KR.UTF-8	Koreanisch (Republik Korea)
vi_VN.UTF-8	Vietnamesisch (Vietnam)

Tabelle 1.19: Liste mit Empfehlungen zum Gebietsschema

```
$ echo $LANG
en_US.UTF-8
$ date -u
Wed 19 May 2021 03:18:43 PM UTC
$ LANG=fr_FR.UTF-8 date -u
mer. 19 mai 2021 15:19:02 UTC
```

Hier wird das Programm date(1) mit verschiedenen Gebietsschema-Einstellungen ausgeführt.

- Beim ersten Befehl ist ”$LANG” auf den System-locale-Standardwert ”en_US.UTF-8” gesetzt.
- Beim zweiten Befehl ist ”$LANG” auf den französischen UTF-8-locale-Wert ”fr_FR.UTF-8” gesetzt.

Die meisten Befehlsausführungen haben keine voreingestellten Definitionen für Umgebungsvariablen. Bei dem obigen Beispiel können Sie alternativ auch folgendes ausführen:

```
$ LANG=fr_FR.UTF-8
$ date -u
mer. 19 mai 2021 15:19:24 UTC
```

Tipp
Wenn Sie mit einem nicht auf Englisch eingerichteten System einen Fehlerbericht einreichen, ist es eine gute Idee, die nötigen Befehle mit ”en_US.UTF-8” als Gebietsschema laufen zu lassen und zu überprüfen.

Tolle Details zur Konfiguration des Gebietsschemas finden Sie in Abschnitt 8.1.

1.5.3 Die ”$PATH”-Variable

Wenn Sie einen Befehl in die Shell eingeben, sucht die Shell in den durch die Umgebungsvariable ”$PATH” definierten Verzeichnissen nach diesem Befehl. Der Wert der Umgebungsvariablen ”$PATH” wird auch als Shell-Suchpfad bezeichnet.

In der Standard-Debian-Installation enthält die Umgebungsvariable ”$PATH” bei normalen Benutzerkonten nicht die Verzeichnisse ”/usr/sbin” und ”/usr/sbin”. Zum Beispiel muss der `ifconfig`-Befehl mit dem vollständigen Pfad aufgerufen werden, also ”/usr/sbin/ifconfig”. Ähnlich dazu ist der `ip`-Befehl in ”/usr/bin” abgelegt.

Sie können die Umgebungsvariable ”$PATH” der Bash-Shell über die Dateien ”~/.bash_profile” oder ”~/.bashrc” verändern.

1.5.4 Die ”$HOME”-Variable

Viele Befehle speichern Konfigurationsdateien im Heimatverzeichnis des jeweiligen Benutzers. Über diese läßt sich das Verhalten der Befehle anpassen. Der Ort des Heimatverzeichnisses ist in der Umgebungsvariable ”$HOME” gespeichert.

Wert von ”$HOME”	Situation bei Programmausführung
/	Programm ausgeführt vom init-Prozess (Daemon)
/root	Programm ausgeführt von der root-Shell
/home/*normaler_benutzer*	Programm ausgeführt von der Shell des normalen Benutzers
/home/*normaler_benutzer*	Programm ausgeführt vom GUI-Desktop-Menü des normalen Benutzers
/home/*normaler_benutzer*	Programm ausgeführt von root mittels ”sudo programm”
/root	Programm ausgeführt von root mittels ”sudo -H programm”

Tabelle 1.20: Liste der Werte von ”$HOME”

Tipp
Die Shell expandiert ”~/” zum Heimatverzeichnis des aktuellen Benutzers, also ”$HOME/”, und ”~foo/” zum Heimatverzeichnis des Benutzers `foo`, also ”/home/foo/”.

Schauen Sie in Abschnitt 12.1.5, wenn $HOME für Ihr Programm nicht verfügbar ist.

1.5.5 Befehlszeilen-Optionen

Einige Befehle akzeptieren Argumente. Argumente, die mit ”-” oder ”--” beginnen, werden Optionen genannt und steuern das Verhalten des Befehls.

```
$ date
Thu 20 May 2021 01:08:08 AM JST
$ date -R
Thu, 20 May 2021 01:08:12 +0900
```

Hier ändert das Befehlszeilen-Argument ”-R” das Verhalten von date(1) zwecks Ausgabe einer RFC2822-konformen Datumsangabe.

Shell-Glob-Suchmuster	Beschreibung der Regel
*	Dateiname, der nicht mit "." beginnt
.*	Dateiname, der mit "." beginnt (in Bash 5.2 und später sind "." und ".." hier nicht enthalten)
?	genau ein Zeichen
[…]	genau eins der in Klammern eingeschlossenen Zeichen
[a-z]	genau ein Zeichen von denen zwischen "a" und "z"
[!…]	genau ein Zeichen von denen, die nicht in Klammern eingeschlossen sind (außer "!")

Tabelle 1.21: Shell-Glob-Suchmuster

1.5.6 Shell-Glob

Oft möchten Sie einen Befehl auf eine Gruppe von Dateien anwenden, ohne all die Dateinamen einzutippen. Die Dateinamen-Suchmuster mittels Shell-**Glob** (manchmal auch **Wildcards** genannt) erfüllen diese Bedürfnisse.

Anmerkung
Ein "/" in einem Pfadnamen wird über die Shell-Glob-Suchmuster in obiger Tabelle nicht gefunden.

Probieren Sie zum Beispiel folgendes:

```
$ mkdir foo; cd foo; touch b1.txt b2.txt b3.c b4.h .b5.txt ..b6.txt
$ mkdir bar; cd bar; touch b1.txt b2.txt b3.c b4.h .b5.txt ..b6.txt ; cd -
$ echo b*.txt
b1.txt b2.txt
$ echo b*
b1.txt b2.txt b3.c b4.h bar
$ echo */*.txt
bar/b1.txt bar/b2.txt
$ echo *.[hc]
b3.c b4.h
$ echo .*
.b5.txt ..b6.txt
$ echo b[!1-3]*
b4.h bar
$ echo b*/b[!1-3]*
bar/b4.h
$ cd ..; rm -rf junk
```

Weitere Details finden Sie in glob(7).

Anmerkung
Anders als die normale Dateinamenerweiterung der Shell findet das Shell-Suchmuster "*", das in find(1) mit -name abgefragt wird, den führenden "." des Dateinamens (neue POSIX-Funktionalität).

Anmerkung
bash kann dahingehend angepasst werden, sein glob-Verhalten mittels der integrierten shopt-Optionen wie "dotglob", "noglob", "nocaseglob", "nullglob", "extglob" usw. zu verändern. Lesen Sie dazu bash(1).

1.5.7 Rückgabewert eines Befehls

Jeder Befehl gibt seinen Beendigungsstatus (Exit-Status, Variable: "$?") als Rückgabewert zurück.

Befehls-Beendigungsstatus	numerischer Rückgabewert	logischer Rückgabewert
Erfolg	Null, 0	**WAHR**
Fehler	nicht-Null, -1	**FALSCH**

Tabelle 1.22: Befehls-Beendigungs-Codes

Probieren Sie zum Beispiel folgendes:

```
$ [ 1 = 1 ] ; echo $?
0
$ [ 1 = 2 ] ; echo $?
1
```

Anmerkung
Bitte beachten Sie, dass im logischen Zusammenhang für die Shell **Erfolg** als logisches **WAHR** betrachtet wird, was 0 (Null) als Wert bedeutet. Dies ist nicht ganz intuitiv und daher soll hier daran erinnert werden.

1.5.8 Typische Befehlssequenzen und Shell-Weiterleitungen

Versuchen Sie, sich folgende Shell-Befehlsfolgen zu merken, die in einer Zeile eingegeben werden:

Das Debian-System ist ein Multi-Tasking-System. Hintergrundprozesse erlauben es Benutzern, mehrere Programme in einer einzigen Shell laufen zu lassen. Zur Verwaltung von Hintergrundprozessen werden die Shell-Builtins `jobs`, `fg`, `bg` und `kill` benutzt. Bitte lesen Sie die entsprechenden Abschnitte von bash(1) bezüglich ”SIGNALS”, ”JOB CONTROL” und bash-builtins(1).

Probieren Sie zum Beispiel folgendes:

```
$ </etc/motd pager
```

```
$ pager </etc/motd
```

```
$ pager /etc/motd
```

```
$ cat /etc/motd | pager
```

Obwohl alle 4 Beispiele von Shell-Weiterleitungen dasselbe anzeigen, führt das letzte Beispiel einen separaten `cat`-Befehl aus und verschwendet ohne Grund zusätzliche Ressourcen.

Die Shell erlaubt Ihnen, Dateien mittels dem `exec`-Builtin mit einem frei wählbaren Datei-Deskriptor zu öffnen.

```
$ echo Hello >foo
$ exec 3<foo 4>bar  # open files
$ cat <&3 >&4       # redirect stdin to 3, stdout to 4
$ exec 3<&- 4>&-    # close files
$ cat bar
Hello
```

Die Datei-Deskriptoren 0-2 sind fest vordefiniert.

Befehlsabfolge	Beschreibung
`befehl &`	Ausführung von `befehl` im **Hintergrund** in einer Unter-Shell
`befehl1 | befehl2`	Weiterleitung (**pipe**) der Standardausgabe von `befehl1` an die Standardeingabe von `befehl2` (**gleichzeitige** Ausführung)
`befehl1 2>&1 | befehl2`	Weiterleitung (**pipe**) der Standardausgabe und Standardfehler von `befehl1` an die Standardeingabe von `befehl2` (**gleichzeitige** Ausführung)
`befehl1 ; befehl2`	**Aufeinander folgende** Ausführung von `befehl1` und `befehl2`
`befehl1 && befehl2`	Ausführung von `befehl1`; falls erfolgreich, **anschließende** Ausführung von `befehl2` (Erfolg zurückgeben, wenn beide (`befehl1` **und** `befehl2`) erfolgreich sind)
`befehl1 || befehl2`	Ausführung von `befehl1`; falls nicht erfolgreich, **anschließende** Ausführung von `befehl2` (Erfolg zurückgeben, wenn `befehl1` **oder** `befehl2` erfolgreich ist)
`befehl > foo`	Weiterleitung der Standardausgabe von `befehl` in die Datei `foo` (Datei überschreiben)
`befehl 2> foo`	Weiterleitung des Standardfehlers von `befehl` in die Datei `foo` (Datei überschreiben)
`befehl >> foo`	Weiterleitung der Standardausgabe von `befehl` in die Datei `foo` (an Datei anhängen)
`befehl 2>> foo`	Weiterleitung des Standardfehlers von `befehl` in die Datei `foo` (an Datei anhängen)
`befehl > foo 2>&1`	Weiterleitung der Standardausgabe und Standardfehler von `befehl` in die Datei `foo`
`befehl < foo`	Weiterleitung der Standardeingabe von `befehl` in die Datei `foo`
`befehl << begrenzung`	Weiterleitung der Standardeingabe von `befehl` in die folgenden Zeilen, bis ”`begrenzung`” erreicht ist
`befehl <<- delimiter`	Weiterleitung der Standardeingabe von `befehl` in die folgenden Zeilen, bis ”`begrenzung`” erreicht ist (die führenden Tab-Zeichen werden von den Eingabezeilen entfernt)

Tabelle 1.23: Abfolgen von Shell-Befehlen

Gerät	Beschreibung	Datei-Deskriptor
`stdin`	Standardeingabe	0
`stdout`	Standardausgabe	1
`stderr`	Standardfehler	2

Tabelle 1.24: Vordefinierte Datei-Deskriptoren

1.5.9 Befehls-Alias

Sie können einen Alias für häufig verwendete Befehle definieren.

Probieren Sie zum Beispiel folgendes:

```
$ alias la='ls -la'
```

Jetzt funktioniert "`la`" als Kurzform für "`ls -la`", was alle Dateien in einem langen Listenformat auflistet.

Sie können alle vorhandenen Alias-Definitionen mit a`lias` anzeigen (lesen Sie dazu in bash(1) den Abschnitt "SHELL BUILTIN COMMANDS").

```
$ alias
...
alias la='ls -la'
```

Sie können den exakten Pfad oder die Identität eines Befehls mit `type` herausfinden (lesen Sie dazu in bash(1) den Abschnitt "SHELL BUILTIN COMMANDS").

Probieren Sie zum Beispiel folgendes:

```
$ type ls
ls is hashed (/bin/ls)
$ type la
la is aliased to ls -la
$ type echo
echo is a shell builtin
$ type file
file is /usr/bin/file
```

Hier wurde kürzlich nach "`ls`" gesucht, nach "`file`" hingegen nicht, daher ist "`ls`" "hashed", d.h. die Shell hat einen internen Eintrag für einen schnellen Zugriff auf den Speicherort des "`ls`"-Befehls.

Tipp
Lesen Sie Abschnitt 9.3.6.

1.6 Unix-ähnliche Textverarbeitung

In Unix-ähnlichen Arbeitsumgebungen können Textverarbeitungsvorgänge durchgeführt werden, indem der Text durch Abfolgen mehrerer Standardwerkzeuge geleitet wird. Dies war eine weitere bedeutende Unix-Innovation.

1.6.1 Unix-Textverarbeitungswerkzeuge

Es gibt ein paar Standard-Textverarbeitungswerkzeuge, die auf Unix-ähnlichen Systemen sehr oft verwendet werden.

- Ohne Verwendung regulärer Ausdrücke:
  - cat(1) verkettet Dateien und gibt den vollständigen Inhalt aus.
  - tac(1) verkettet Dateien und gibt den Inhalt in umgekehrter Reihenfolge aus.
  - cut(1) wählt Teile von Zeilen und ausgegebenen Inhalten aus.
  - head(1) gibt die ersten Zeilen von Dateien aus.
  - tail(1) gibt die letzten Zeilen von Dateien aus.
  - sort(1) sortiert Zeilen von Textdateien.

  - uniq(1) entfernt doppelte Zeilen aus einer sortierten Datei.
  - tr(1) übersetzt oder löscht Zeichen.
  - diff(1) vergleicht Dateien Zeile für Zeile.
- Mit Verwendung grundlegender regulärer Ausdrücke (basic regular expressions / **BRE**) als Standardeinstellung:
  - ed(1) ist ein primitiver Zeilen-Editor.
  - sed(1) ist ein Stream-Editor.
  - grep(1) findet Text über Suchmuster.
  - vim(1) ist ein Bildschirm-gestützter Editor.
  - emacs(1) ist ein Bildschirm-gestützter Editor (leicht erweiterte **BRE**)
- Mit Verwendung erweiterter regulärer Ausdrücke (extended regular expressions / **ERE**):
  - awk(1) führt einfache Textverarbeitung durch.
  - egrep(1) findet Text über Suchmuster.
  - `tcl`(3tcl) kann jede erdenkliche Art von Textverarbeitung durchführen. Lesen Sie dazu re_syntax(3). Oft zusammen mit `tk`(3tk) verwendet.
  - perl(1) kann jede erdenkliche Art von Textverarbeitung durchführen. Lesen Sie dazu perlre(1).
  - pcre2grep(1) aus dem `pcre2-utils`-Paket findet Text über Perl-kompatible reguläre Ausdrücke (PCRE).
  - python3(1) mit dem `re`-Modul kann jede erdenkliche Art von Textverarbeitung durchführen. Lesen Sie dazu "`/usr/share/doc/python3/html/index.html`".

Wenn Sie sich nicht sicher sind, was genau diese Befehle tun, verwenden Sie bitte "`man befehl`", um es selbst herauszufinden.

Anmerkung
Sortierreihenfolge und Bereichsausdruck sind abhängig vom Gebietsschema (Locale). Wenn Sie das traditionelle Verhalten eines Befehls erreichen möchten, verwenden Sie die Locale **C** oder **C.UTF-8** statt der normalen **UTF-8**-Locales (siehe Abschnitt 8.1).

Anmerkung
Die regulären Perl-Erweiterungen (perlre(1)), Perl-kompatiblen regulären Ausdrücke (PCRE) sowie regulären Python-Erweiterungen (bereitgestellt durch das `re`-Modul) enthalten viele gängige Erweiterungen zu den normalen **erweiterten regulären Ausdrücken (ERE)**.

1.6.2 Reguläre Ausdrücke

Reguläre Ausdrücke werden in vielen Textverarbeitungswerkzeugen verwendet. Sie sind den Shell Globs ähnlich, aber viel komplexer und leistungsfähiger.

Ein regulärer Ausdruck beschreibt ein Muster und besteht aus Textzeichen und **Metazeichen**.

Ein **Metazeichen** ist einfach ein Zeichen mit einer speziellen Bedeutung. Es gibt zwei Hauptgruppen, **BRE** und **ERE**, abhängig von den Textwerkzeugen, wie unten beschrieben.

Die regulären Ausdrücke von **`emacs`** gehören grundsätzlich zu den **BRE**, sind aber erweitert worden, damit "+" und "?" wie bei den **ERE** als **Metazeichen** behandelt werden. Es ist daher nicht nötig, sie bei den regulären Ausdrücken von `emacs` mit "\" zu schützen.

grep(1) kann verwendet werden, um eine Textsuche mittels regulärer Ausdrücke durchzuführen.

Probieren Sie zum Beispiel folgendes:

BRE	**ERE**	**Beschreibung des regulären Ausdrucks**
`\ . [ ] ^ $ *`	`\ . [ ] ^ $ *`	allgemeine **Metazeichen**
`\+ \? \( \) \{ \} |`		nur bei BRE: durch "\" geschützte **Metazeichen**
	`+ ? ( ) { } |`	nur bei ERE: nicht durch "\" geschützte **Metazeichen**
c	c	findet **Nicht-Metazeichen** "c"
`\c`	`\c`	findet den Buchstaben "c", auch wenn "c" selbst ein **Metazeichen** ist
`.`	`.`	findet jegliches Zeichen inklusive Newline
^	^	Position am Anfang einer Zeichenkette
$	$	Position am Ende einer Zeichenkette
`\<`	`\<`	Position am Anfang eines Wortes
`\>`	`\>`	Position am Ende eines Wortes
`[abc…]`	`[abc…]`	findet irgendein Zeichen von "abc…"
`[^abc…]`	`[^abc…]`	findet irgendein Zeichen außer "abc…"
`r*`	`r*`	findet keinen oder mehrere reguläre Ausdrücke festgelegt durch "r"
`r\+`	`r+`	findet einen oder mehrere reguläre Ausdrücke festgelegt durch "r"
`r\?`	`r?`	findet keinen oder einen regulären Ausdruck festgelegt durch "r"
`r1|r2`	`r1|r2`	findet einen der regulären Ausdrücke festgelegt durch "r1" oder "r2"
`\(r1|r2\)`	`(r1|r2)`	findet einen der regulären Ausdrücke festgelegt durch "r1" oder "r2" und behandelt es als einen **in Klammern** gesetzten regulären Ausdruck

Tabelle 1.25: Metazeichen für BRE und ERE

```
$ egrep 'GNU.*LICENSE|Yoyodyne' /usr/share/common-licenses/GPL
GNU GENERAL PUBLIC LICENSE
GNU GENERAL PUBLIC LICENSE
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
```

Tipp
Lesen Sie Abschnitt 9.3.6.

1.6.3 Ersetzungsausdrücke

Bei den Ersetzungsausdrücken haben einige Zeichen spezielle Bedeutungen.

Ersetzungsausdruck	Beschreibung des Textes, der den Ersetzungsausdruck ersetzen soll
&	was der reguläre Ausdruck gefunden hat (benutzen Sie \& in emacs)
\n	was der n-te **in Klammern** gesetzte reguläre Ausdruck gefunden hat (”n” ist ein Zahl)

Tabelle 1.26: Der Ersetzungsausdruck

Bei Ersetzungsausdrücken für Perl wird ”$&” statt ”&” verwendet und ”$n” statt ”\n”.

Probieren Sie zum Beispiel folgendes:

```
$ echo zzz1abc2efg3hij4 | \
sed -e 's/\(1[a-z]*\)[0-9]*\(.*\)$/=&=/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
sed -E -e 's/(1[a-z]*)[0-9]*(.*)$/=&=/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
perl -pe 's/(1[a-z]*)[0-9]*(.*)$/=$&=/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
sed -e 's/\(1[a-z]*\)[0-9]*\(.*\)$/\2===\1/'
zzzefg3hij4===1abc
$ echo zzz1abc2efg3hij4 | \
sed -E -e 's/(1[a-z]*)[0-9]*(.*)$/\2===\1/'
zzzefg3hij4===1abc
$ echo zzz1abc2efg3hij4 | \
perl -pe 's/(1[a-z]*)[0-9]*(.*)$/$2===$1/'
zzzefg3hij4===1abc
```

Achten Sie hier bitte besonders auf die Art der **in Klammern** gesetzten regulären Ausdrücke und wie die gefundenen Zeichenketten bei der Textersetzung mit verschiedenen Werkzeugen genutzt werden.

Diese regulären Ausdrücke können in einigen Editoren auch zur Cursor-Verschiebung und Textersetzung verwendet werden.

Der Back-Slash ”\” am Ende der Zeile in der Shell-Befehlszeile schützt das Newline als Leerraum-Zeichen und erlaubt die Fortsetzung der Shell-Befehlszeileneingabe in die nächste Zeile.

Bitte lesen Sie alle zugehörigen Handbuchseiten, um diese Befehle zu erlernen.

1.6.4 Globale Ersetzungen mit regulären Ausdrücken

Der Befehl ed(1) ersetzt hier alle Vorkommen von ”VON_REGEX” in ”datei” durch ”ZU_TEXT”.

```
$ ed file <<EOF
,s/FROM_REGEX/TO_TEXT/g
w
q
EOF
```

Der Befehl sed(1) ersetzt hier alle Vorkommen von ”VON_REGEX” in ”datei” durch ”ZU_TEXT”.

```
$ sed -i -e 's/FROM_REGEX/TO_TEXT/g' file
```

Der Befehl vim(1) ersetzt hier durch Verwendung der ex(1)-Befehle alle Vorkommen von ”VON_REGEX” in ”datei” durch ”ZU_TEXT”.

```
$ vim '+%s/FROM_REGEX/TO_TEXT/gc' '+update' '+q' file
```

Tipp
Die ”c”-Markierung in dem obigen Beispiel gewährleistet die interaktive Ersetzung jedes Vorkommmnisses.

Mehrere Dateien (”datei1”, ”datei2” und ”datei3”) können auf ähnlichem Wege mittels regulärer Ausdrücke mit vim(1) oder perl(1) verarbeitet werden.

```
$ vim '+argdo %s/FROM_REGEX/TO_TEXT/gce|update' '+q' file1 file2 file3
```

Tipp
Die Option ”e” in dem obigen Beispiel verhindert, dass der ”No match”-Fehler die Zuweisung unterbricht.

```
$ perl -i -p -e 's/FROM_REGEX/TO_TEXT/g;' file1 file2 file3
```

In dem perl(1)-Beispiel steht ”-i” für direkte Bearbeitung der jeweiligen Zieldatei und ”-p” für implizites Abarbeiten aller angegebenen Dateien mittels Schleifen.

Tipp
Durch die Verwendung von ”-i.bak” statt ”-i” wird jede Originaldatei gesichert, dabei wird ”.bak” zum Dateinamen hinzugefügt. Dies erlaubt eine einfachere Wiederherstellung bei Fehlern in komplexen Situationen.

Anmerkung
ed(1) und vim(1) gehören zu den **BRE**, perl(1) zu den **ERE**.

1.6.5 Extrahieren von Daten aus einer Textdatei-Tabelle

Wir nehmen an, eine Textdatei namens ”DPL” enthält die Namen einiger Debian-Projektleiter der Jahre vor 2004 sowie deren Ernennungsdatum in einem durch Leerzeichen getrennten Format:

```
Ian     Murdock   August  1993
Bruce   Perens    April   1996
Ian     Jackson   January 1998
Wichert Akkerman  January 1999
Ben     Collins   April   2001
Bdale   Garbee    April   2002
Martin  Michlmayr March   2003
```

Tipp
Lesen Sie ”Eine kurze Geschichte von Debian ” bezüglich der neuesten Vergangenheit zum Debian-Projektleiter.

Awk wird oft verwendet, um Daten aus dieser Art von Dateien zu extrahieren.

Probieren Sie zum Beispiel folgendes:

```
$ awk '{ print $3 }' <DPL                    # month started
August
April
January
January
April
April
March
$ awk '($1=="Ian") { print }' <DPL           # DPL called Ian
Ian     Murdock   August  1993
Ian     Jackson   January 1998
$ awk '($2=="Perens") { print $3,$4 }' <DPL # When Perens started
April 1996
```

Auch Shells wie Bash können verwendet werden, um diese Art von Dateien zu verarbeiten.

Probieren Sie zum Beispiel folgendes:

```
$ while read first last month year; do
    echo $month
  done <DPL
... same output as the first Awk example
```

Hier verwendet der Builtin-Befehl `read` Zeichen aus ”`$IFS`” (interne Feldseparatoren), um Zeilen in einzelne Wörter aufzusplitten.

Wenn Sie ”`$IFS`” in ”`:`” ändern, können Sie auf nette Weise ”`/etc/passwd`” mit der Shell verarbeiten.

```
$ oldIFS="$IFS"   # save old value
$ IFS=':'
$ while read user password uid gid rest_of_line; do
    if [ "$user" = "bozo" ]; then
      echo "$user's ID is $uid"
    fi
  done < /etc/passwd
bozo's ID is 1000
$ IFS="$oldIFS"   # restore old value
```

(Wenn Awk dies erledigen soll, verwenden Sie ”FS=`':'`”, um den Feldseparator festzulegen.)

IFS wird auch von der Shell verwendet, um die Ergebnisse von Parameterexpansion, Befehlsersetzung und arithmetischer Expansion aufzusplitten. Diese werden nicht innerhalb von doppelten oder einfachen Anführungszeichen dargestellt. Der Standardwert von IFS ist die Kombination von *Space*, *Tab* und *Newline*.

Seien Sie vorsichtig bei der Verwendung dieser Shell-IFS-Tricks. Verrückte Dinge passieren, wenn die Shell einige Teile des Skripts als eigene **Eingabe** interpretiert.

```
$ IFS=":,"                          # use ":" and "," as IFS
$ echo IFS=$IFS,   IFS="$IFS"       # echo is a Bash builtin
IFS=  , IFS=:,
$ date -R                           # just a command output
Sat, 23 Aug 2003 08:30:15 +0200
$ echo $(date -R)                   # sub shell --> input to main shell
Sat  23 Aug 2003 08 30 36 +0200
$ unset IFS                         # reset IFS to the default
$ echo $(date -R)
Sat, 23 Aug 2003 08:30:50 +0200
```

1.6.6 Skript-Schnipsel für die Befehlsweiterleitung

Die folgenden Befehlsketten tun nette Dinge als Teil einer Weiterleitung:

Skript-Schnipsel (in einer Zeile eingeben)	**Auswirkung des Befehls**
`find /usr -print`	findet alle Dateien in "`/usr`"
`seq 1 100`	gibt die Zahlen 1 bis 100 aus
`| xargs -n 1 befehl`	führt <befehl> wiederholt aus, jeweils mit jedem Element aus der Weiterleitung als Argument
`| xargs -n 1 echo`	splittet durch Leerraum getrennte Elemente aus der Weiterleitung in einzelne Zeilen auf
`| xargs echo`	führt alle Zeilen aus der Weiterleitung in eine Zeile zusammen
`| grep -e regex_muster`	extrahiert Zeilen aus der Weiterleitung, die das Regex-Suchmuster enthalten
`| grep -v -e regex_muster`	extrahiert Zeilen aus der Weiterleitung, die nicht das Regex-Suchmuster enthalten
`| cut -d: -f3 -`	extrahiert das dritte Feld aus der Weiterleitung, separiert durch "`:`" (z.B. aus der passwd-Datei)
`| awk '{ print $3 }'`	extrahiert das dritte Feld aus der Weiterleitung, separiert durch Leerraum-Zeichen
`| awk -F'\t' '{ print $3 }'`	extrahiert das dritte Feld aus der Weiterleitung, separiert durch Tab
`| col -bx`	entfernt Backspace und expandiert Tabs zu Leerzeichen
`| expand -`	expandiert Tabs
`| sort| uniq`	sortiert Inhalte und entfernt Duplikate
`| tr 'A-Z' 'a-z'`	konvertiert Großschreibung in Kleinschreibung
`| tr -d '\n'`	führt mehrere Zeilen in eine zusammen
`| tr -d '\r'`	entfernt CR (carriage return/Wagenrücklauf)
`| sed 's/^/# /'`	fügt "#" zum Anfang jeder Zeile hinzu
`| sed 's/\.ext//g'`	entfernt "`.ext`"
`| sed -n -e 2p`	gibt die zweite Zeile aus
`| head -n 2 -`	gibt die ersten zwei Zeilen aus
`| tail -n 2 -`	gibt die letzten zwei Zeilen aus

Tabelle 1.27: Liste von Skript-Schnipseln für die Befehlsweiterleitung

Ein einzeiliges Shell-Skript kann durch Verwendung von find(1) und xargs(1) über Schleifen viele Dateien in Folge abarbeiten und so ziemlich komplexe Aufgaben erledigen. Lesen Sie dazu Abschnitt 10.1.5 und Abschnitt 9.4.9.

Wenn die Verwendung des interaktiven Shell-Modus' zu kompliziert wird, ziehen Sie bitte in Erwägung, ein Shell-Skript zu schreiben (weiteres dazu in Abschnitt 12.1).

Kapitel 2

Debian-Paketmanagement

Anmerkung
Dieses Kapitel geht davon aus, dass `Trixie` die aktuelle stabile Veröffentlichung ist.
Die Paketquellen für das APT-System werden in diesem Dokument mit dem Oberbegriff **sources.list** bezeichnet . Sie können die entsprechenden Eintragungen sowohl in der Datei "`/etc/apt/sources.list`" durchführen, sowie in irgendeiner "`/etc/apt/sources.list.d/*.list`"- oder "`/etc/apt/sources.list.d/*.sources`"-Datei.

2.1 Grundvoraussetzungen für das Debian-Paketmanagement

2.1.1 Debian-Paketmanagement

Debian ist eine Organisation Freiwilliger, die **konsistente** Distributionen vorkompilierter Binärpakete von freier Software erstellt und sie über ihr Archiv verteilt.

Das Debian-Archiv wird von vielen Spiegel-Servern via HTTP und FTP bereitgestellt. Es ist auch als CD-ROM/DVD erhältlich.

Das derzeit empfohlene Debian-Werkzeug für Paketmanagement, das all diese Ressourcen nutzen kann, ist das Advanced Packaging Tool (APT).

Debians Paketmanagement-System, ermöglicht dem Nutzer (**wenn es korrekt genutzt wird**), **konsistente Sätze von Binärpaketen** auf dem System zu installieren. Derzeit sind für die amd64-Architektur 78709 Pakete verfügbar.

Das Debian-Paketmanagement-System hat eine reichhaltige Geschichte und bietet eine Auswahl an Frontend-Benutzerprog die genutzt werden können sowie an verschiedenen Varianten des Archivzugriffs. Derzeit empfehlen wir folgendes:

- apt(8) für alle interaktiven Befehlszeilen-Operationen, inklusive Paketinstallationen und -entfernungen sowie Distributions-Upgrades;
- apt-get(8) für Paketverwaltungsoperationen mittels Skripten; auch eine Ausweichlösung für den Fall, dass `apt` nicht verfügbar ist (z.B. auf älteren Debian-Systemen);
- aptitude(8) für ein interaktives Text-Interface zur Verwaltung der installierten Pakete und für die Suche nach verfügbaren Paketen.

Paket	Popcon	Größe	Beschreibung
dpkg	V:909, I:1000	5086	niederschwelliges Paketmanagement-System für Debian (datei-basiert)
apt	V:882, I:1000	5025	APT-Frontend zur Paketverwaltung mit text-basierter Bedienung: apt/apt-get/apt-cache
aptitude	V:32, I:154	4837	APT-Frontend für interaktives Paketmanagement mit Vollbildschirm-Konsole: aptitude(8)
tasksel	V:42, I:985	352	APT-Frontend zur Installation von vorausgewählten Programmgruppen (Tasks): tasksel(8)
unattended-upgrades	V:142, I:190	374	Erweiterungspaket für APT, um die automatische Installation von Sicherheits-Aktualisierungen zu ermöglichen
gnome-software	V:159, I:260	4732	Software Center für GNOME (grafisches Frontend für APT)
synaptic	V:31, I:262	7264	grafischer Paketmanager (grafisches GTK-Frontend für APT)
apt-utils	V:383, I:996	1187	APT-Hilfsprogramme: apt-extracttemplates(1), apt-ftparchive(1) und apt-sortpkgs(1)
apt-listchanges	V:451, I:895	563	Benachrichtigungswerkzeug für die Änderungshistorie von Paketen
apt-listbugs	V:4.7, I:6.4	512	Auflistung kritischer Fehler vor jeder APT-Installation
apt-file	V:14, I:52	76	APT-Paketsuch-Werkzeug - Befehlszeilen-Interface
apt-rdepends	V:0.4, I:4.2	39	rekursive Auflistung von Paketabhängigkeiten

Tabelle 2.1: Liste von Debians Paketmanagement-Werkzeugen

2.1.2 Paketkonfiguration

Hier einige wichtige Punkte für die Paketkonfiguration auf einem Debian-System:

- Bei einem modernen Desktop-System ist es eine gute Idee, nach Änderungen an der Paketkonfiguration oder nach Paket-Upgrades einen Reboot durchzuführen, um sicherzustellen, dass das System weiter korrekt funktioniert.
- Die händische Konfiguration durch den Systemadministrator wird respektiert. In anderen Worten: das Paketkonfigurationssystem führt keine aufdringliche Konfiguration durch, nur weil es einfach wäre.
- Jedes Paket enthält sein eigenes Konfigurationsskript mit einer standardisierten Benutzerschnittstelle namens debconf(7), um Sie bei der Erstinstallation des Pakets zu unterstützen.
- Debian-Entwickler versuchen ihr Bestes, um mittels der Paketkonfigurationsskripte für einen fehlerfreien Ablauf der Paketaktualisierungen zu sorgen.
- Dem Systemadministrator steht die volle Funktionalität der paketierten Software zur Verfügung, aber Funktionen mit Sicherheitsrisiken sind standardmäßig deaktiviert.
- Wenn Sie von Hand einen Dienst aktivieren und so Sicherheitsrisiken hervorrufen, sind Sie selbst für die Beherrschung der Risiken verantwortlich.
- Esoterische Konfigurationen könnten manuell durch den Systemadministrator aktiviert worden sein. Dies kann Behinderungen mit beliebten, grundlegenden Systemkonfigurations-Programmen verursachen.

2.1.3 Grundsätzliche Vorsichtsmaßnahmen



Warnung
Installieren Sie nicht Pakete aus einer wahllosen Mischung von Debian-Suiten (Stable, Testing, Unstable, Experimental). Sie können dadurch die Paketkonsistenz beschädigen und deren Korrektur erfordert ein tiefes Verständnis der Systemverwaltung wie Compiler-ABI, Bibliotheks-Versionen, Interpreter-Funktionalitäten usw.

Als Neuling unter den Debian-Systemadministratoren sollten Sie bei der **`Stable`**-Veröffentlichung bleiben und nur Sicherheitsaktualisierungen einspielen. Bis Sie das Debian-System gut verstehen, sollten Sie folgende Vorsichtsmaßnahmen einhalten:

- Fügen Sie nicht **`testing`** oder **`unstable`** in Ihre "`sources.list`" ein.
- Vermischen Sie in Ihrer "`sources.list`" nicht das Standard-Debian mit anderen Nicht-Debian-Archiven wie Ubuntu.
- Erstellen Sie keine "`/etc/apt/preferences`"-Datei.
- Ändern Sie nicht über Konfigurationsdateien das Standardverhalten der Paketmanagement-Werkzeuge, wenn Sie die vollständigen Auswirkungen nicht kennen.
- Installieren Sie nicht irgendwelche Pakete mit "`dpkg -i` *`irgendein_paket`*".
- Installieren Sie niemals irgendwelche Pakete mit "`dpkg --force-all -i` *`irgendein_paket`*".
- Löschen oder verändern Sie keine Dateien in "`/var/lib/dpkg/`".
- Überschreiben Sie keine Systemdateien, indem Sie Software-Programme installieren, die direkt aus den Quellen übersetzt wurden.
  - Installieren Sie solche Pakete, falls nötig, in "`/usr/local`" oder "`/opt`".

Die nicht-kompatiblen Effekte, die das Verletzen von obigen Maßnahmen im Debian-Paketmanagement verursachen können, führen im schlimmsten Falle zu einem nicht mehr verwendbaren System.

Ein ernsthafter Debian-Systemadministrator, der missionskritische Server betreibt, sollte besondere Vorsicht walten lassen.

- Installieren Sie nicht irgendwelche Pakete von Debian, die Sicherheitsaktualisierungen beinhalten, ohne sie unter sicheren Bedingungen sorgfältig mit Ihrer speziellen Konfiguration getestet zu haben.
  - Sie als Systemadministrator sind am Ende für Ihr System verantwortlich.
  - Die lange Stabilitäts-Historie des Debian-Systems ist für sich alleine keine Garantie.

2.1.4 Leben mit den ewigen Aktualisierungen



Achtung
Für Ihre **Produktions-Server** wird die `Stable`-Suite inklusive der Sicherheitsaktualisierungen empfohlen. Dasselbe gilt für Desktop-Rechner, für die nur eingeschränkt administrative Betreuung gewährleistet ist.

Trotz der obigen Warnungen weiß ich, dass viele Leser dieses Dokuments trotzdem die neuere `Testing`- oder `Unstable`-Suite nutzen möchten.

Die Erleuchtung, die wir Ihnen mit den folgenden Informationen bieten, bewahrt Sie vor der ewigen karmischen Qual der Upgrade-Hölle und ermöglicht Ihnen, das Debian-Nirvana zu erreichen.

Diese Liste zielt auf ein **selbst-administriertes** Arbeitsplatzsystem ab:

- Nutzen Sie die `Testing`-Suite, da dies letztlich ein Rolling-Release ist, das von Debians Qualitässicherungs-Infrastruktur profitiert, wie Debians Continuous-Integration-Unterstützung, der Regel, dass nur Quellcode hochgeladen werden kann sowie die Nachverfolgung von Versionsübergängen bei Bibliotheken. Die Pakete in der `Testing`-Suite werden häufig genug aktualisiert, um stets die neuesten Funktionalitäten bieten zu können.
- Setzen Sie den zur `Testing`-Suite passenden Codenamen (während `trixie` der aktuellen `stable`-Suite entspricht, ist dies "`forky`") in Ihrer "`sources.list`".

- Eine gewisse Zeit nach Freigabe einer neuer Debian-Veröffentlichung (ca. einen Monat, oder nach Ihrer eigenen Beurteilung auch früher oder später), ändern Sie diesen Namen in Ihrer "`sources.list`" händisch in den neuen Codenamen für `Testing`. Die Debian-User- oder -Developer-Mailinglisten sind auch gute Quellen für diese Informationen.

Die Verwendung von `Unstable` wird nicht empfohlen. Die `Unstable`-Suite ist für Entwickler **gut geeignet zur Fehlersuche in Paketen**, bringt aber für normale Arbeitsplatznutzer auch gewisse unnötige Risiken mit sich. Selbst wenn die `Unstable`-Suite von Debian die meiste Zeit sehr stabil erscheint, hat es in der Vergangenheit bereits Probleme mit bestimmten Paketen gegeben, von denen einige nicht einfach zu beheben waren.

Hier einige Tipps für grundlegende Vorsichtsmaßnahmen, wie Sie eine schnelle und einfache Systemwiederherstellung bei Fehlern in Debian-Paketen gewährleisten können:

- Stellen Sie eine System-Rettungsumgebung bereit, siehe Abschnitt 3.2.
- Erstellen Sie ein **Dual-boot**-fähiges System, das Debians `Stable`-Suite auf einer anderen Partition installiert hat.
- Ziehen Sie in Betracht, `apt-listbugs` zu installieren, um vor einer Paketaktualisierung relevante Informationen aus der Debian-Fehlerdatenbank (BTS) einsehen zu können.
- Lernen Sie genug über die Infrastruktur des Paketsystems, um das Problem umgehen zu können.



Achtung
Falls Sie keine dieser Vorsichsmaßnahmen durchführen können, sind Sie möglicherweise nicht bereit für die `Testing`- oder `Unstable`-Suite.

2.1.5 Grundlagen über das Debian-Archiv

Tipp
Die offiziellen Richtlinien für das Debian-Archiv sind im Debian Policy-Handbuch, Kapitel 2 - The Debian Archive festgeschrieben.

Lassen Sie uns das Debian-Archiv aus der Sicht eines Benutzers betrachten.

Für einen System-User erfolgt der Zugriff auf das Debian-Archiv über das APT-System.

Die Paketquellen für das APT-System werden hier unter dem Oberbegriff **source list** behandelt und sind in sources.list(5) beschrieben.

Für ein `trixie`-System mit typischem HTTP-Zugriff sieht die **source list** im modernen deb822-Stil in "`/etc/apt/sources.list.d/debian.sources`" wie folgt aus:

```
Types: deb deb-src
URIs: http://deb.debian.org/debian/
Suites: trixie trixie-updates
Components: main non-free-firmware contrib non-free

Types: deb deb-src
URIs: http://security.debian.org/debian-security/
Suites: trixie-security
Components: main non-free-firmware contrib non-free
```

Tipp
Falls die **source list** noch in der alten einzeiligen Darstellung in "`/etc/apt/sources.list`" oder "`/etc/apt/sources.list.d/*.list`" vorliegt, aktualisieren Sie sie mit:

```
$ sudo apt modernize-sources
```

Die Kernpunkte der **source list** im deb822-Stil sind folgende:

- Die Konfiguration wird hier in "`/etc/apt/sources.list.d/*.sources`"-Dateien festgelegt.
- Jeder Block von Zeilen (getrennt durch eine Leerzeile) definiert eine Paketquelle für das APT-System.
- Der "`Types:`"-Eintrag definiert den Typ der Quelle wie "`deb`" (für Binärpakete) und "`deb-src`" (für Quellpakete).
- Der "`URIs:`"-Eintrag definiert die Liste von Wurzel-URI-Adressen für das Debian-Archiv.
- Der "`Suites:`"-Eintrag definiert die Liste der Distributionsnamen, entweder über den Namen der Suite oder den Codenamen.
- Der "`Components:`"-Eintrag definiert die Liste der verwendeten Archivbereiche aus dem Debian-Archiv.

Die Definition für "`deb-src`" kann problemlose weggelassen werden, wenn die Konfiguration lediglich für `aptitude` ist, da `aptitude` entsprechende Metadaten nicht auswertet. Dies beschleunigt die Aktualisierung der Archiv-Metadaten.

Die URL kann "`https://`", "`http://`", "`ftp://`", "`file://`", ... sein.

Zeilen, die mit "#" beginnen, sind Kommentare und werden ignoriert.

Hier neige ich dazu, den Codenamen "`trixie`" oder "`forky`" zu verwenden statt dem Namen der Suite (`stable` oder "`testing`"), um Überraschungen zu vermeiden, wenn das nächste `Stable`-Release veröffentlicht wird.

Tipp
Falls in dem obigen Beispiel "`sid`" statt "`trixie`" benutzt wird, ist die Zeile "`deb: http://security.debian.org/ ...`" für Sicherheitsaktualisierungen in "`/etc/apt/sources.list`" (oder der entsprechende deb822-Eintrag) nicht nötig, weil es kein Archiv für Sicherheitsaktualisierungen für "`Sid`" (`Unstable`) gibt.

Hier eine Liste der URL von Debian-Archiv-Seiten und Suite- oder Codenamen, die in der Konfigurationsdatei nach einer `trixie`-Veröffentlichung verwendet werden.



Achtung
Nur die reine **`Stable`**-Veröffentlichung mit Sicherheitsaktualisierungen bietet beste Stabilität. Zum größten Teil die **`Stable`**-Veröffentlichung zu verwenden, aber vermischt mit einigen Paketen aus **`Testing`** oder **`Unstable`**, birgt ein höheres Risiko, als reines **`Unstable`** zu nutzen (aufgrund von unpassenden Bibliotheksversionen usw). Wenn Sie wirklich die neueste Version einiger Programme unter **`Stable`** benötigen, verwenden Sie bitte Pakete von stable-updates und backports (lesen Sie dazu Abschnitt 2.7.4). Diese Dienste müssen mit besonderer Sorgfalt genutzt werden.



Achtung
Sie sollten grundsätzlich nur eine der Suiten `stable`, `testing` oder `unstable` pro "`deb`"-Zeile angeben. Wenn Sie irgendeine Kombination von `stable`, `testing` und `unstable` in einer "`deb`"-Zeile verwenden, bremst dies die APT-Programme aus, wobei am Ende aber nur das neueste Archiv verwendet wird. Eine Auflistung mehrerer Einträge macht Sinn, wenn die "`/etc/apt/preferences`"-Datei genutzt wird und dort klare Richtlinien festgelegt sind (lesen Sie dazu Abschnitt 2.7.7).

Archiv-URL	Suite-Name	Codename	Zweck der Paketquelle
http://deb.debian.org/debian/	`stable`	`trixie`	Quasi statische `stable`-Veröffentlichung, umfangreich getestet
http://deb.debian.org/debian/	`stable-updates`	`trixie-updates`	Dringende Aktualisierungen für die `stable`-Suite (wichtig)
http://security.debian.org/debian-security/	`stable-security`	`trixie-security`	Sicherheitsaktualisierungen für die `stable`-Suite (wichtig)
http://deb.debian.org/debian/	`testing`	`forky`	Dynamische `testing`-Veröffentlichung, annehmbar getestet, Änderungen nach kurzer Wartezeit
http://deb.debian.org/debian/	`unstable`	`sid`	Dynamische `unstable`-Veröffentlichung, nur minimale Tests, Änderungen ohne Wartezeit
http://deb.debian.org/debian/	`experimental`	Nicht verfügbar	Experimentelle Vorveröffentlichung von Entwicklern (optional, nur für Entwickler)
http://deb.debian.org/debian/	`stable-proposed-updates`	`trixie-proposed-updates`	Aktualisierungen für die nächste `stable`-Zwischenveröffentlichung (optional)
http://deb.debian.org/debian/	`stable-backports`	`trixie-backports`	Grobe Sammlung von vorkompilierten Paketen überwiegend aus der `testing`-Suite (optional)
http://security.debian.org/debian-security/	`testing-security`	`forky-security`	Weder aktiv unterstützt noch genutzt vom Security-Team

Tabelle 2.2: Liste von Debian-Archiv-Seiten

Tipp
Für ein Debian-System mit der `Stable`-Veröffentlichung ist es eine gute Idee, wie in obigem Beispiel Zeilen/Einträge mit "`http://security.debian.org/`" in der "`sources.list`" zu integrieren, um Sicherheitsaktualisierungen zu erhalten.

Anmerkung
Sicherheitsprobleme im `Stable`-Archiv werden vom Debian-Sicherheits-Team behoben. Dies war immer ziemlich gründlich und zuverlässig. Sicherheitsprobleme im `Testing`-Archiv werden unter Umständen durch das Debian-Testing-Sicherheits-Team behoben. Aus verschiedenen Gründen funktioniert dies aber nicht so gründlich wie für `Stable` und Sie müssen möglicherweise warten, bis die Pakete mit den Fehlerkorrekturen von `Unstable` nach `Testing` migriert sind. Sicherheitsprobleme in `Unstable` werden durch die jeweiligen Paketbetreuer behoben. Aktiv betreute `Unstable`-Pakete sind für gewöhnlich in einem ziemlich guten Zustand, da Sicherheitskorrekturen von Upstream mit einfließen. In der Debian Sicherheits-FAQ finden Sie Informationen, wie Debian mit Sicherheitslücken umgeht.

Bereich	Anzahl der Pakete	Kriterien der Paketkomponenten
`main`	77218	DFSG-konform und keine Abhängigkeiten zu `non-free`
`non-free-firmware`	85	nicht DFSG-konforme Firmware; oft erforderlich, um ein angemessenes Ergebnis bei einer Systeminstallation zu erhalten
`contrib`	365	DFSG-konform, aber mit Abhängigkeiten zu `non-free`
`non-free`	1041	nicht DFSG-konform und nicht in `non-free-firmware`

Tabelle 2.3: Liste der Debian-Archiv-Bereiche

Die Paketanzahl in obiger Liste gilt für die amd64-Architektur. Der `main`-Bereich stellt das Debian-System dar (lesen Sie dazu Abschnitt 2.1.6).

Die Struktur des Debian-Archivs können am besten erforschen, indem Sie mit Ihrem Browser die jeweilige Archiv-URL mit einem angehängten `dists` oder `pool` besuchen.

Die Distribution wird auf zwei Arten referenziert, über die Suite oder über den Codenamen. Das Wort ”Distribution” wird in vielen Dokumentationen synonym zum Namen der Suite verwendet. Die Beziehung zwischen Suite und Codename kann wie folgt zusammengefasst werden:

Zeitraum	**Suite = `Stable`**	**Suite = `Testing`**	**Suite = `Unstable`**
nach der Veröffentlichung von `Trixie`	Codename = `Trixie`	Codename = `Forky`	Codename = `Sid`
nach der Veröffentlichung von `Forky`	Codename = `Forky`	Codename = `Duke`	Codename = `Sid`

Tabelle 2.4: Zusammenhanag zwischen Suite und Codename

Die Historie der Codenamen ist beschrieben in der Debian FAQ: Kapitel 6.2.1 Welche Codenamen wurden in der Vergangenheit verwendet?

In der strengeren Debian-Archiv-Terminologie wird das Wort ”Sektion” (section) speziell für die Kategorisierung der Pakete durch Anwendungen genutzt (obwohl der Begriff ”main section” auch verwendet werden könnte, um den Bereich ”main” im Debian-Archiv zu beschreiben).

Immer wenn ein Debian-Entwickler (DD, von Debian developer) ein Paket in das `Unstable`-Archiv hochlädt (via incoming), muss der DD dafür sorgen, dass das hochgeladene Paket mit dem aktuellen Satz der Pakete im `Unstable`-Archiv kompatibel ist.

Falls der DD diese Kompatibilität absichtlich verletzt (für wichtige Bibliotheksaktualisierungen usw.), gibt es gewöhnlich eine Ankündigung dazu auf der Mailingliste debian-devel.

Bevor eine Gruppe von Paketen durch das Wartungsskript des Debian-Archivs von `Unstable` nach `Testing` verschoben wird, prüft das Wartungsskript nicht nur das Alter (rund 2-10 Tage alt) und den Status der veröffentlichungskritischen Fehler (RC bugs) für die Pakete, sondern versucht auch die Kompatibilität der Pakete mit dem aktuellen Satz der Pakete im `Testing`-Archiv sicherzustellen. Dieser Prozess macht `Testing` sehr aktuell und nutzbar.

Durch den abgestuften, vom Release-Team gesteuerten Archiv-Freeze-Prozess ist das `Testing`-Archiv bereits gereift, um es auf diesem Wege - mit einigen manuellen Eingriffen - vollständig konsistent und fehlerfrei zu machen. Dann wird die neue `Stable`-Veröffentlichung erzeugt, indem der Codename für das alte `Testing`-Archiv dem neuen `Stable`-Archiv zugewiesen wird; anschließend wird ein neuer Codename für das neue `Testing`-Archiv erstellt. Die anfänglichen Inhalte des neuen `Testing` entsprechen exakt dem des neu veröffentlichten `Stable`.

Sowohl das `Unstable`- wie auch das `Testing`-Archiv können aus unterschiedlichen Gründen unter vorübergehenden Störungen leiden:

- Upload eines beschädigten Pakets in das Archiv (meistens bei `Unstable`);
- Verzögerung, bis die neuen Pakete im Archiv akzeptiert werden (meistens bei `Unstable`);
- Timing-Probleme bei der Archiv-Synchronisation (sowohl bei `Testing` wie auch bei `Unstable`);
- Manuelle Eingriffe in das Archiv wie das Entfernen eines Pakets (überwiegend bei `Testing`).

Falls Sie sich also jemals entscheiden, diese Archive zu verwenden, sollten Sie in der Lage sein, diese Arten von Störungen zu beheben oder zu umgehen.



Achtung
Für die ersten Monate nach einer neuen `Stable`-Veröffentlichung sollten die meisten Desktop-Benutzer das `Stable`-Archiv inklusive der Sicherheitsaktualisierungen verwenden, selbst wenn sie normalerweise `Unstable` oder `Testing` benutzen. In dieser Übergangszeit sind sowohl das `Unstable`- wie auch das `Testing`-Archiv für die meisten Leute nicht gut. Das System ist in dieser Zeit nur schwer in lauffähigem Zustand zu halten, da `Unstable` Fluten von gravierenden Hochrüstungen der Kernpakete ertragen muss, und auch `Testing` ist nicht von Nutzen, weil sein Inhalt überwiegend dem von `Stable` entspricht, aber ohne die Unterstützung für Sicherheitsaktualisierungen (Debian Testing-Security-Ankündigung 12/2008). Nach einem Monat (evtl. mehr) könnte das `Unstable`-oder `Testing`Archiv wieder nützlich sein, wenn Sie vorsichtig sind.

Tipp
Wenn Sie das `Testing`-Archiv beobachten, stellen Sie fest, dass ein durch die Entfernung eines Pakets verursachtes Problem normalerweise umgangen wird, indem man das entsprechende Paket aus `Unstable` installiert, welches zwecks Behebung des Fehlers hochgeladen wurde.

Näheres über Begriffsdefinitionen zum Debian-Archiv finden Sie im Debian Policy-Handbuch:

- ”Sections” (Sektionen);
- ”Priorities” (Prioritäten);
- ”Base system” (Grundsystem);
- ”Essential packages” (grundlegende Pakete).

2.1.6 Debian ist zu 100% freie Software

Debian besteht aus folgenden Gründen zu 100% aus freier Software:

- Debian installiert standardmäßig nur freie Software und respektiert so diedie Freiheit des Benutzers.
- Debian stellt in `main` nur freie Software bereit.
- Debian empfiehlt, nur freie Software aus `main` zu verwenden.
- Kein Paket in `main` hängt von Paketen in `non-free`, `non-free-firmware` oder `contrib` ab bzw. empfiehlt diese.

Einige Leute fragen sich, ob sich folgende Fakten widersprechen oder nicht:

- ”Debian wird zu 100% frei bleiben” (der erste Punkt des Debian-Gesellschaftsvertrags).
- Debian-Server beherbergen etliche `non-free-firmware`-, `non-free`- und `contrib`-Pakete.

Sie widersprechen sich nicht, und zwar aus folgenden Gründen:

- Das Debian-System ist zu 100% frei und seine Pakete werden von Debian-Servern im `main`-Bereich vorgehalten.
- Pakete außerhalb des Debian-Systems befinden sich auf den Debian-Servern in den Bereichen `non-free`, `non-free-fi` und `contrib`.

Dies wird auch präzise unter Punkt 4 und 5 des Debian-Gesellschaftsvertrags erklärt:

- Unsere Prioritäten sind unsere Anwender und Freie Software

  - Wir orientieren uns an den Bedürfnissen unserer Anwender und der Gemeinschaft für Freie Software. Deren Interessen stehen an erster Stelle. Wir werden unsere Nutzer bei ihrer Arbeit mit den verschiedensten Rechnerumgebungen unterstützen. Wir haben nichts gegen unfreie Arbeiten, die darauf abzielen, auf Debian-Systemen verwendet zu werden oder versuchen, eine Gebühr von Personen, die solche Arbeiten erstellen oder verwenden, einzufordern. Wir erlauben anderen, Distributionen zu erstellen, die das Debian-System und andere Arbeiten enthalten, ohne dafür irgendwelche Gebühren zu erheben. Um diese Ziele zu fördern, bieten wir ein integriertes System von hoher Qualität an, das die gerade beschriebene Nutzung nicht durch rechtliche Einschränkungen verhindert.

- Arbeiten, die nicht unseren Standards für Freie Software genügen

  - Wir wissen, dass einige unserer Anwender unbedingt Arbeiten einsetzen müssen, die nicht den Debian-Richtlinien für Freie Software entsprechen. Für solche Arbeiten haben wir die Bereiche "`non-free`", "`non-free-firmware`" und "`contrib`" in unserem Archiv eingerichtet. Die Pakete in diesen Bereichen sind nicht Bestandteil des Debian-Systems, wurden aber trotzdem für den Einsatz mit Debian vorbereitet. Wir empfehlen den CD-Herstellern, die jeweiligen Lizenzbestimmungen der Pakete in diesen Bereichen zu studieren und selbst zu entscheiden, ob sie die Pakete mit ihren CDs verteilen dürfen. Obwohl unfreie Arbeiten nicht Bestandteil von Debian sind, unterstützen wir ihren Einsatz und bieten Infrastruktur für nicht-freie Pakete an (z.B. unsere Fehlerdatenbank und die Mailinglisten). Offizielle Installationsmedien von Debian könnten Firmware enthalten, die sonst nicht Teil des Debian-Systems sind; dies soll es ermöglichen, Debian auf Hardware einzusetzen, die solche Firmware erfordert.

Anmerkung
Der aktuelle Text des 5. Absatzes im derzeitigen Debian-Gesellschaftsvertrag unterscheidet sich geringfügig von obigem Text. Diese redaktionelle Abweichung ist beabsichtigt, um dieses Dokument konsistent zu machen, ohne den realen Inhalt des Gesellschaftsvertrags ändern zu müssen.

Benutzer sollten sich der Risiken bewußt sein, die durch die Verwendung von Paketen aus `non-free`, `non-free-firmware` und `contrib` entstehen:

- fehlende Freiheit bei diesen Software-Paketen;
- keine Unterstützung von Debian für solche Software-Pakete (Debian kann keine vernünftige Unterstützung für Software bieten, bei der nicht auf den Quellcode zugegriffen werden kann);
- Verunreinigung Ihres zu 100% freien Debian-Systems.

Die Debian-Richtlinien für Freie Software (DFSG) sind der Freie-Software-Standard für Debian. Debian interpretiert "Software" im weitesten Sinne inklusive Dokumentation, Firmware, Logos und künstlerische Arbeiten in einem Paket. Aufgrund dessen sind Debians Freie-Software-Standards sehr streng.

Typische `non-free`-, `non-free-firmware`- und `contrib`-Pakete enthalten frei verteilbare Pakete der folgenden Typen:

- Dokumentationspakete unter der GNU-Lizenz für freie Dokumentation mit unveränderlichen Abschnitten wie diejenigen für GCC und Make (meistens zu finden in der Sektion `non-free/doc`).
- Firmware-Pakete, die Binärdaten ohne Quelltext enthalten, wie diejenigen, die in Abschnitt 9.10.5 unter `non-free-firmw` aufgeführt sind (meistens zu finden in der Sektion `non-free-firmware/kernel`).
- Spiele- und Schriftarten-Pakete mit Einschränkungen für die kommerzielle Verwendung und/oder inhaltliche Veränderung.

Bitte beachten Sie, dass die Anzahl der Pakete in `non-free`, `non-free-firmware` und `contrib` weniger als 2% der Pakete in `main` beträgt. Den Zugriff auf die Bereiche `non-free`, `non-free-firmware` und `contrib` des Archivs zu ermöglichen, verschleiert nicht die Quellen der Pakete. Die interaktive Verwendung von aptitude(8) im Vollbildmodus erlaubt Ihnen die vollständige Ansicht und Kontrolle darüber, welche Pakete aus welchen Bereichen installiert sind, um Ihr System so frei zu halten, wie Sie möchten.

2.1.7 Paketabhängigkeiten

Das Debian-System stellt einen konsistenten Satz von Binärpaketen bereit und verwendet dafür einen Deklarationsmechanismus für versionierte Binärabhängigkeiten über Felder in der control-Datei. Hier eine leicht vereinfachte Definition dieser Felder:

- ”Depends” / Hängt ab von
  - Depends deklariert eine absolute Abhängigkeit und alle Pakete, die in diesem Feld aufgelistet sind, müssen zur selben Zeit oder im Voraus installiert sein.
- ”Pre-Depends” / Hängt ab (vorher) von
  - Pre-Depends ist ähnlich wie Depends, nur dass die vollständige Installation der aufgelisteten Pakete im Voraus erforderlich ist.
- ”Recommends” / Empfiehlt
  - Recommends deklariert eine starke, aber nicht absolute Abhängigkeit. Die meisten Benutzer würden das Paket nicht installiert haben wollen, solange nicht auch alle in diesem Feld aufgelisteten Pakete installiert sind.
- ”Suggests” / Schlägt vor
  - Suggests deklariert eine schwache Abhängigkeit. Viele Benutzer dieses Pakets würden davon profitieren, auch die in diesem Feld aufgelisteten Pakete zu installieren, aber selbst ohne diese bietet das Paket vernünftige Funktionalität.
- ”Enhances” / Wertet auf
  - Enhances deklariert eine schwache Abhängigkeit wie Suggests, nur funktioniert sie in der entgegengesetzten Richtung.
- ”Breaks” / Beschädigt
  - Breaks deklariert eine Paket-Inkompatibilität, gewöhnlich mit einer Versionsangabe. Grundsätzlich lösen Sie etwaige Probleme, indem Sie alle in diesem Feld aufgelisteten Pakete auf den neuesten Stand bringen.
- ”Conflicts” / Kollidiert
  - Conflicts deklariert eine absolute Inkompatibilität. Alle in diesem Feld aufgelisteten Pakete müssen entfernt werden, um dieses Paket zu installieren.
- ”Replaces” / Ersetzt
  - Replaces wird deklariert, wenn Dateien, die von diesem Paket installiert werden, andere Dateien in den aufgelisteten Paketen ersetzen.
- ”Provides” / Stellt bereit
  - Provides wird deklariert, wenn dieses Paket alle Dateien und Funktionalitäten der hier aufgelisteten Pakete bereitstellt.

Anmerkung
Bitte beachten Sie, dass die gleichzeitige Deklaration von ”Provides”, ”Conflicts” und ”Replaces” für ein virtuelles Paket eine vernünftige Konfiguration ist. So wird sichergestellt, dass nur ein reelles Paket, das dieses virtuelle Paket bereit stellt, zur jeweiligen Zeit installiert sein kann.

Die offizielle Definition inklusive der Quellpaket-Abhängigkeiten finden Sie im Debian Policy-Handbuch: Kapitel 7 - Declaring relationships between packages.

2.1.8 Die Ereignisabfolge für das Paketmanagement

Hier folgt eine vereinfachte Ereignisabfolge für das APT-Paketmanagement.

- **Aktualisierung der Paketlisten / Update** (”`apt update`”, ”`aptitude update`” oder ”`apt-get update`”):
  1. Archiv-Metadaten vom Archiv-Server beziehen;
  2. Neuaufbau und Aktualisierung der lokalen Metadaten für die Verwendung durch APT.
- **Paketaktualisierung / Upgrade** (”`apt upgrade`” und ”`apt full-upgrade`” oder ”`aptitude safe-upgrade`” und ”`aptitude full-upgrade`” oder ”`apt-get upgrade`” und ”`apt-get dist-upgrade`”):
  1. Auswahl der Installationskandidat-Version für alle installierten Pakete, was üblicherweise die jeweils letzte verfügbare Version ist (in Abschnitt 2.7.7 finden Sie Infos, wie Sie Ausnahmen festlegen);
  2. Auflösung der Paketabhängigkeiten;
  3. Ausgewählte Binärpakete vom Archiv-Server beziehen, falls die jeweilige Installationskandidat-Version sich von der installierten Version unterscheidet;
  4. Entpacken der heruntergeladenen Binärpakete;
  5. Ausführen des **preinst**-Skripts;
  6. Installieren der Binärdateien;
  7. Ausführen des **postinst**-Skripts.
- **Installieren / Install** (”`apt install …`”, ”`aptitude install …`” oder ”`apt-get install …`”):
  1. Auswählen der Pakete über die Angabe auf der Befehlszeile
  2. Auflösung der Paketabhängigkeiten;
  3. Beziehen der ausgewählten Binärpakete vom Archiv-Server;
  4. Entpacken der heruntergeladenen Binärpakete;
  5. Ausführen des **preinst**-Skripts;
  6. Installieren der Binärdateien;
  7. Ausführen des **postinst**-Skripts.
- **Entfernen / Remove** (”`apt remove …`”, ”`aptitude remove …`” oder ”`apt-get remove …`”):
  1. Auswählen der Pakete über die Angabe auf der Befehlszeile
  2. Auflösung der Paketabhängigkeiten;
  3. Ausführen des **prerm**-Skripts;
  4. Löschen der installierten Dateien **außer** den Konfigurationsdateien;
  5. Ausführen des **postrm**-Skripts.
- **Entfernen inkl. der Konfigurationsdateien / Purge** (”`apt purge`”, ”`aptitude purge …`” oder ”`apt-get purge …`”):
  1. Auswählen der Pakete über die Angabe auf der Befehlszeile
  2. Auflösung der Paketabhängigkeiten;
  3. Ausführen des **prerm**-Skripts;
  4. Löschen der installierten Dateien **inklusive** der Konfigurationsdateien;
  5. Ausführen des **postrm**-Skripts.

Um des Gesamtbildes willen habe ich hier absichtlich technische Details übersprungen.

2.1.9 Erste Hilfe bei Paketmanagement-Problemen

Sie sollten die offizielle Dokumentation lesen. Das erste Dokument ist hierbei die Debian-spezifische Datei "`/usr/share/do`
Weitere Dokumentation in "`/usr/share/doc/`*`paketname`*`/`" sollten Sie ebenfalls konsultieren. Wenn Sie die Shell wie in Abschnitt 1.4.2 beschrieben eingerichtet haben, geben Sie folgendes ein:

```
$ cd package_name
$ pager README.Debian
$ mc
```

Sie müssen unter Umständen das entsprechende Dokumentationspaket installieren (hängen Sie dazu "`-doc`" an den Namen an), um die Dokumentation verfügbar zu haben.

Wenn Sie Probleme mit einem bestimmten Paket haben, sollten Sie unbedingt zuerst die Seiten der Debian-Fehlerdatenbank (BTS) besuchen.

Website	Befehl
Homepage der Debian-Fehlerdatenbank (BTS)	`sensible-browser "https://bugs.debian.org/"`
Fehlerberichte gegen ein bestimmtes Paket	`sensible-browser "https://bugs.debian.org/`*`paketname`*`"`
Fehlerbericht mit einer bekannten Fehlernummer	`sensible-browser "https://bugs.debian.org/`*`fehlernummer`*`"`

Tabelle 2.5: Liste wichtiger Websites zur Lösung von Paketproblemen

Suchen Sie bei Google mit Suchbegriffen, die "`site:debian.org`", "`site:wiki.debian.org`", "`site:lists.debian.`
usw. enthalten.

Wenn Sie einen Fehlerbericht einreichen, nutzen Sie bitte den Befehl reportbug(1).

2.1.10 Wie Sie Debian-Pakete auswählen

Wenn Sie auf mehr als zwei ähnliche Pakete stoßen und sich fragen, welches Sie installieren sollen, ohne dabei nach dem "trial and error"-Verfahren (ausprobieren und schauen, welches das richtige ist) vorgehen zu müssen, sollten Sie ein wenig Ihren **gesunden Menschenverstand** benutzen. Die folgenden Punkte können als gute Hinweise auf passende Pakete angesehen werden:

- Essentiell: ja > nein;
- Bereich: main > contrib > non-free;
- Priorität: erforderlich > wichtig > standard > optional > extra;
- Tasks: Pakete, die in Tasks wie "Desktop-Umgebungen" aufgelistet sind;
- Pakete, die über eine Paketabhängigkeit ausgewählt werden (z.B. `gcc-16` über `gcc`) ;
- Popcon: höhere Einstufung bei den vote- und install-Werten;
- Changelog (Änderunsprotokoll): regelmäßige Aktualisierungen durch den Betreuer;
- BTS (Fehlerdatenbank): keine veröffentlichungskritischen Fehler (RC-Bugs, d.h. keine Fehler mit Schweregrad critical, grave oder serious);
- BTS (Fehlerdatenbank): Betreuer reagiert auf Fehlerberichte;
- BTS (Fehlerdatenbank): höhere Anzahl von kürzlich behobenen Fehlern;

- BTS (Fehlerdatenbank): niedrigere Anzahl von offenen Fehlern mit Schweregrad (Severity) verschieden von wishlist.

Debian ist ein Freiwilligenprojekt mit verteiltem Entwicklungsmodell, dessen Archiv viele Pakete unterschiedlicher Zielsetzung und Qualität enthält. Sie müssen Ihre eigene Entscheidung treffen, was Sie damit anfangen.

2.1.11 Wie Sie mit Konflikten umgehen

Welche Debian-Suite Sie auch immer entschieden haben zu nutzen, Sie werden immer den Wunsch haben, Versionen von Programmen laufen zu lassen, die nicht in dieser Suite enthalten sind. Selbst wenn Sie Binärpakete von solchen Programmen in anderen Debian-Suites oder in nicht-Debian-Quellen finden, könnten deren Abhängigkeiten mit Ihrem aktuellen Debian-Systemen kollidieren.

Obwohl Sie das Paketmanagement anpassen können, z.B. mit der **apt-pinning**-Technik, um solche nicht-synchronen Binärpakete zu installieren (wie in Abschnitt 2.7.7 beschrieben), haben diese Ansätze nur eine begrenzte Reichweite, da sie auch die entsprechenden Programme oder Ihr System beschädigen können.

Bevor Sie all zu brutal versuchen, solchen nicht-synchronen Pakete zu installieren, sollten Sie vorher alle verfügbaren alternativen und sicheren Lösungen in Erwägung ziehen, die mit Ihrem aktuellen Debian-System kompatibel sind:

- Installieren solcher Programme unter Verwendung eines passenden Binärpakets von Upstream in einer Sandbox (siehe Abschnitt 7.7).
  - Viele grafische Programme wie LibreOffice und GNOME-Anwendungen sind als Flatpak-, Snap- oder AppImage-Pakete verfügbar.
- Erzeugen Sie eine chroot- oder ähnliche Umgebung und lassen Sie dort solche Programme laufen (lesen Sie dazu Abschnitt 9.11).
  - CLI-Befehle (Konsolen-Programme) können auf einfache Art in einer kompatiblen Chroot-Umgebung ausgeführt werden (siehe Abschnitt 9.11.4).
  - Systeme mit mehreren grafischen Arbeitsplatzumgebungen können einfach ohne einen Reboot getestet werden (siehe Abschnitt 9.11.5).
- Bauen Sie die gewünschten Versionen der Binärpakete selbst, so dass sie mit Ihrem laufenden Debian-System kompatibel sind.
  - Dies ist allerdings keine triviale Aufgabe (siehe Abschnitt 2.7.13).

2.2 Grundlegende Paketmanagement-Operationen

Paketmanagement-Operationen können im Debian-System mit vielen, auf APT aufbauenden Werkzeugen durchgeführt werden. Wir beschreiben hier drei grundlegende Paketmanagement-Werkzeuge: `apt`, `apt-get` / `apt-cache` und `aptitude`.

Für Paketmanagement-Operationen inklusive Installation oder Aktualisierung der Paket-Metadaten benötigen Sie root-Privilegien.

2.2.1 `apt` contra `apt-get` / `apt-cache` contra `aptitude`

Obwohl `aptitude` ein tolles interaktives Werkzeug ist, das auch der Autor überwiegend verwendet, sollten Sie einige warnende Fakten kennen:

- `aptitude` wird nicht für das Upgrade eines Debian-`Stable`-Systems (Hochrüstung von einer Stable-Version zur nächsten) empfohlen.

  - Dafür empfehlen wir "`apt full-upgrade`" oder "`apt-get dist-upgrade`". Lesen Sie dazu den Fehlerbericht #411280.

- Manchmal empfiehlt `aptitude` beim System-Upgrade von `Testing`- oder `Unstable`-Systemen die massenweise Entfernung von Paketen.
  - Diese Situation hat schon manchen Systemadministrator erschreckt. Aber: keine Panik.
  - Dies wurde scheinbar überwiegend dadurch verursacht, dass eine Schieflage bei den Versionsnummern von Paketen bestand, die von Metapaketen wie `gnome-core` abhängen oder diese empfehlen.
  - Sie können dies beheben, indem Sie "Noch ausstehende Aktionen abbrechen" im Menü von `aptitude` wählen, `aptitude` beenden und "`apt full-upgrade`" verwenden.

Die Befehle `apt-get` und `apt-cache` sind sehr **grundlegende** APT-basierte Paketmanagement-Werkzeuge.

- `apt-get` und `apt-cache` bieten lediglich eine Befehlszeilen-Oberfläche.
- `apt-get` eignet sich sehr gut für die **großen Systemhochrüstungen** von einer Debian-Veröffentlichung auf die nächste, usw.
- `apt-get` bietet einen **robusten** Mechanismus zur Auflösung von Paketabhängigkeiten.
- `apt-get` ist weniger anspruchsvoll bezüglich der Hardware-Ressourcen. Es verbraucht weniger Speicher und läuft schneller.
- `apt-cache` bietet eine **grundlegende**, auf regulären Ausdrücken basierende Suche über Paketname und -beschreibung.
- `apt-get` und `apt-cache` können über `/etc/apt/preferences` mehrere Versionen des gleichen Pakets verwalten, aber dies ist ziemlich mühselig.

Das `apt`-Programm ist eine Befehlszeilenschnittstelle für das Paketmanagement. Es ist vom Grundsatz her ein Wrapper-Skript für `apt-get`, `apt-cache` und ähnliche Programme und als Bedienoberfläche für den Benutzer gedacht; einige für die interaktive Nutzung zweckmäßige Optionen sind standardmäßig aktiviert.

- `apt` bietet eine nette Fortschrittsanzeige bei der Paketinstallation mittels `apt install`.
- Standardmäßig **entfernt** `apt` die zwischengespeicherten `.deb`-Dateien nach der erfolgreichen Installation.

Tipp
Benutzern wird empfohlen, den neuen apt(8)-Befehl zu verwenden, der für die **interaktive** Nutzung durch den Benutzer gedacht ist. In Shell-Skripten sollten Sie apt-get(8) und apt-cache(8) einsetzen.

Der Befehl `aptitude` ist das **vielseitigste** APT-basierte Paketmanagement-Werkzeug.

- `aptitude` bietet eine interaktive textbasierte Benutzeroberfläche mit Vollbildschirm-Ansicht.
- `aptitude` enthält auch eine Befehlszeilen-Benutzerschnittstelle.
- `aptitude` eignet sich sehr gut für das **tägliche interaktive Paketmanagement**, wie die Überprüfung installierter Pakete und die Suche nach verfügbaren Paketen.
- `aptitude` ist anspruchsvoller bezüglich der Hardware-Ressourcen. Es verbraucht mehr Speicher und läuft geringfügig langsamer.
- `aptitude` bietet eine **erweiterte**, auf regulären Ausdrücken basierende Suche über alle Paket-Metadaten.
- `aptitude` kann mehrere Versionen des gleichen Pakets verwalten, ohne dabei `/etc/apt/preferences` zu verwenden, und es ist sehr intuitiv.

apt-Syntax	**aptitude-Syntax**	**apt-get/apt-cache-Syntax**	**Beschreibung**
`apt update`	`aptitude update`	`apt-get update`	Aktualisieren der Paket-Metadaten
`apt install foo`	`aptitude install foo`	`apt-get install foo`	Installieren der Installationskandidat-Version von ”`foo`” inklusive seiner Abhängigkeiten
`apt upgrade`	`aptitude safe-upgrade`	`apt-get upgrade`	Installieren der Installationskandidat-Version aller installierten Pakete, ohne irgendwelche anderen Pakete zu entfernen
`apt full-upgrade`	`aptitude full-upgrade`	`apt-get dist-upgrade`	Installieren der Installationskandidat-Version aller installierten Pakete und Entfernen anderer Pakete, falls nötig
`apt remove foo`	`aptitude remove foo`	`apt-get remove foo`	Entfernen des Pakets ”`foo`” bei gleichzeitiger Erhaltung seiner Konfigurationsdateien
`apt autoremove`	Nicht verfügbar	`apt-get autoremove`	Entfernen aller automatisch installierten, nicht mehr benötigten Pakete
`apt purge foo`	`aptitude purge foo`	`apt-get purge foo`	Vollständiges Entfernen des Pakets ”`foo`” und Löschung seiner Konfigurationsdateien
`apt clean`	`aptitude clean`	`apt-get clean`	Vollständiges Leeren des lokalen Depots heruntergeladener Paketdateien
`apt autoclean`	`aptitude autoclean`	`apt-get autoclean`	Leeren des lokalen Depots heruntergeladener Paketdateien für veraltete Pakete
`apt show foo`	`aptitude show foo`	`apt-cache show foo`	Anzeigen von detaillierten Informationen über das Paket ”`foo`”
`apt search` *`regex`*	`aptitude search` *`regex`*	`apt-cache search` *`regex`*	Suchen nach Paketen, auf die der reguläre Ausdruck *regex* zutrifft
Nicht verfügbar	`aptitude why` *`regex`*	Nicht verfügbar	Erklären, warum die Pakete, auf die der reguläre Ausdruck *regex* zutrifft, installiert werden sollen
Nicht verfügbar	`aptitude why-not` *`regex`*	Nicht verfügbar	Erklären, warum die Pakete, auf die der reguläre Ausdruck *regex* zutrifft, nicht installiert werden sollen
`apt list --manual-installed`	`aptitude search '~i!~M'`	`apt-mark showmanual`	Händisch installierte Paket auflisten

Tabelle 2.6: Grundlegende Paketmanagement-Operationen mit apt(8), aptitude(8) und apt-get(8) / apt-cache(8)

2.2.2 Grundlegende Paketmanagement-Operationen auf der Befehlszeile

Hier einige grundlegende Paketmanagement-Operationen auf der Befehlszeile, die apt(8), aptitude(8) und apt-get(8) / apt-cache(8) verwenden.

`apt`/`apt-get` und `aptitude` können ohne größere Probleme gemischt verwendet werden.

Der Befehl "`aptitude why` *`regex`*" kann mehr Informationen ausgeben, wenn "-v" verwendet wird ("`aptitude -v why` *`regex`*"). Ähnliche Informationen erhalten Sie mit `apt rdepends` *`paket`*" oder "`apt-cache rdepends` *`paket`*.

Wenn `aptitude` im Befehlszeilenmodus gestartet wird und Probleme wie z.B. Paketkonflikte anzeigt, können Sie im Nachhinein in die interaktive Vollbildschirm-Ansicht wechseln, indem Sie am Prompt die Taste "e" drücken.

Anmerkung
Obwohl der `aptitude`-Befehl reichhaltige Funktionen mit sich bringt, wie z.B. seinen erweiterten Paketabhängigkeitsauflöser, hat diese Komplexität einige Rückentwicklungen verursacht (oder könnte sie auch in Zukunft verursachen), wie z.B. Fehler #411123, Fehler #514930 und Fehler #570377. Im Zweifel bevorzugen Sie bitte die `apt-`, `apt-get-` und `apt-cache`-Befehle.

Befehlsoptionen können direkt im Anschluß an den "`aptitude`"-Befehl angegeben werden.

Befehlsoptionen	Beschreibung
`-s`	das Resultat des Befehls simulieren
`-d`	nur Herunterladen, aber nicht installieren/aktualisieren
`-D`	kurze Erläuterungen vor automatischen Installationen und Entfernungen anzeigen

Tabelle 2.7: Erwähnenswerte Befehlsoptionen für aptitude(8)

Weitere Informationen finden Sie unter aptitude(8) und im "aptitude-Benutzerhandbuch" in "`/usr/share/doc/aptitude/R`

2.2.3 Interaktive Nutzung von aptitude

Für interaktives Paketmanagement starten Sie `aptitude` von der Konsolen-Shell aus im interaktiven Modus mit:

```
$ sudo aptitude -u
Password:
```

Die lokale Kopie der Archivinformationen wird aktualisiert und die Paketliste in der Vollbildschirm-Ansicht angezeigt. Aptitude legt seine Konfiguration in "`~/.aptitude/config`" ab.

Tipp
Wenn Sie root's Konfiguration verwenden wollen statt der des unprivilegierten Benutzers, verwenden Sie in obigem Ausdruck "`sudo -H aptitude` …" statt "`sudo aptitude` …".

Tipp
`Aptitude` merkt automatisch **bestimmte Aktionen** vor, wenn es interaktiv gestartet wird. Wenn Sie dies nicht möchten, können Sie sie über das Menü zurücksetzen: "Aktionen" → "Nicht abgeschlossene Aktionen abbrechen".

Taste	Tastaturbefehl
F10 oder Strg-t	Menü
?	**Hilfe** für Tastaturkürzel anzeigen (ausführlichere Auflistung)
F10 → Hilfe → Handbuch	Benutzerhandbuch anzeigen
u	Paketinformationen (Metadaten) aktualisieren
+	Paket zur **Aktualisierung (upgrade)** oder **Installation (install)** vormerken
-	Paket zum **Entfernen (remove)** vormerken (Konfigurationsdateien erhalten)
_	Paket zum **vollständigen Entfernen (purge)** vormerken (Konfigurationsdateien löschen)
=	Paket auf **Zurückhalten (hold)** setzen
U	Alle aktualisierbaren Pakete markieren (die gleiche Funktion wie **full-upgrade**)
g	Das **Herunterladen** und **Installieren** aller markierten Pakete starten
q	Aktuelle Ansicht schließen und Änderungen speichern
x	Aktuelle Ansicht schließen und Änderungen verwerfen
Enter	Informationen über ein Paket anzeigen
C	Änderungsprotokoll (Changelog) eines Pakets anzeigen
l	Ändern des Filters für die Paketanzeige
/	Suche nach dem ersten Treffer
\	Wiederholen der letzten Suche

Tabelle 2.8: Liste der Tastaturkürzel für aptitude

2.2.4 Tastaturkürzel von aptitude

Es folgen einige erwähnenswerte Tastenkürzel, um in der Vollbildschirm-Ansicht den Status von Paketen abzufragen und ”geplante Aktionen” vorzumerken.

Die Angabe des Dateinamens auf der Befehlszeile sowie das Eingabefeld nach dem Drücken von ”l” und ”//” nutzen aptitudes Regex (regulären Ausdruck) wie unten beschrieben. Damit aptitudes regulärer Ausdruck exakt auf einen Paketnamen zutrifft, können Sie eine Zeichenkette verwenden, die mit ”~n” beginnt, gefolgt von dem Paketnamen.

Tipp
Im interaktiven Modus müssen Sie ”U” drücken, um alle installierten Pakete auf die **Installationskandidat-Version** zu aktualisieren. Andernfalls werden nur die markierten Pakete und solche mit versionierten Abhängigkeiten zu diesen auf die **Installationskandidat-Version** aktualisiert.

2.2.5 Paketansichten in aptitude

In der interaktiven Vollbildschirm-Ansicht von aptitude(8) werden Pakete in der Paketliste wie im folgenden Beispiel angezeigt:

```
idA   libsmbclient                          -2220kB 3.0.25a-1  3.0.25a-2
```

Diese Zeile hat die folgenden Bedeutungen (von links nach rechts):

- die Markierung für den aktuellen Zustand (der erste Buchstabe);
- die Markierung für die geplante Aktion (der zweite Buchstabe);
- die Markierung für ”automatisch installiert” (der dritte Buchstabe);

- der Paketname;
- die voraussichtliche Änderung beim verwendeten Platz auf der Festplatte, ausgelöst durch die geplante Aktion;
- die aktuelle Version des Pakets;
- die Installationskandidat-Version des Pakets.

Tipp
Eine vollständige Liste der Markierungen finden Sie am unteren Ende der **Hilfe**-Ansicht, die Ihnen durch Drücken von "?" angezeigt wird.

Die **Installationskandidat-Version** wird aufgrund der aktuellen lokalen Voreinstellungen ausgewählt (lesen Sie dazu apt_preferences(5) und Abschnitt 2.7.7).

Im Menü unter `Ansichten` sind verschiedene Arten von Paketansichten verfügbar:

Ansicht	Beschreibung der Ansicht
`Package View`	laut Tabelle 2.10 (Standardeinstellung)
`Audit Recommendations`	Pakete auflisten, die von installierten Paketen empfohlen werden, aber noch nicht installiert sind
`Flat Package List`	Pakete ohne Kategorisierung anzeigen (für Verwendung mit Regex)
`Source Package View`	Pakete gruppiert nach Quellpaketen anzeigen

Tabelle 2.9: Liste der Ansichten für aptitude

Die Standard-`Paketansicht` kategorisiert Pakete ähnlich wie `dselect` plus einiger zusätzlicher Funktionalitäten.

Kategorie	Beschreibung der Ansicht
`Upgradable Packages`	Pakete organisiert nach `Sektion` → `Bereich` → `Paket`
`New Packages`	, ,
`Installed Packages`	, ,
`Not Installed Packages`	, ,
`Obsolete and Locally Created Packages`	, ,
`Virtual Packages`	Pakete mit der gleichen Funktionalität auflisten
`Tasks`	Pakete mit verschiedenen Funktionalitäten auflisten, die grundsätzlich für eine bestimmte Aufgabe benötigt werden

Tabelle 2.10: Die Kategorisierung von Standard-Paketansichten

Tipp
Die `Tasks`-Ansicht kann verwendet werden, um aus verschiedenen Paketen für eine Aufgabe auszuwählen.

2.2.6 Optionen für Suchmethoden mit aptitude

Aptitude bietet verschiedene Optionen, damit Sie über seine Regex-Formel (regulären Ausdruck) nach Paketen suchen können.

- Shell-Befehlszeile:

  - "`aptitude search` *`aptitude_regex`*" zur Anzeige von Installationsstatus, Paketname und Kurzbeschreibung passender Pakete
  - "`aptitude show` *`paketname`*" zur Anzeige detaillierter Informationen für ein Paket
- Interaktive Vollbildschirm-Ansicht:
  - "`l`" zur Einschränkung der Paketansicht auf zutreffende Pakete
  - "`/`" für die Suche nach einem passenden Paket
  - "`\`" für die Rückwärts-Suche nach einem passenden Paket
  - "`n`" zum Finden des nächsten Treffers
  - "`N`" zum Finden des nächsten Treffers in umgekehrter Richtung

---

Tipp
Die obige Zeichenkette *paketname* muss exakt auf den Paketnamen zutreffen, außer sie beginnt mit "~", um anzuzeigen, dass es ein regulärer Ausdruck ist.

2.2.7 Aptitudes Regex-Formel

Die Regex-Formel (regulärer Ausdruck) ist ähnlich wie bei mutt ein **erweiterter regulärer Ausdruck (ERE)** (lesen Sie dazu Abschnitt 1.6.2); die Bedeutung der `aptitude`-spezifischen speziellen Regelerweiterungen ist im folgenden erklärt:

- Der Regex-Teil ist der gleiche erweiterte reguläre Ausdruck (**ERE**) wie in Unix-typischen Textverarbeitungswerkzeugen und nutzt "^", "`.*`", "$" usw., wie es auch egrep(1), awk(1) und perl(1) tun.
- Der Typ einer Abhängigkeit zur Festlegung von Wechselbeziehungen zwischen Paketen ist einer aus der folgenden Liste: depends (hängt ab von), predepends (hängt ab (vorher) von), recommends (empfiehlt), suggests (schlägt vor), conflicts (kollidiert mit), replaces (ersetzt), provides (stellt bereit).
- Der Typ der Standardabhängigkeit ist "depends".

Tipp
Wenn *regex_muster* ein Null-String ist, geben Sie "~T" direkt hinter dem Befehl an.

Hier einige Abkürzungen:

- "`~P`*`name`*" == "`~Dprovides:`*`name`*"
- "`~C`*`name`*" == "`~Dconflicts:`*`name`*"
- "…`~W name`" == "`(…|name)`"

Benutzer, denen `mutt` geläufig ist, werden sich auch hier schnell zurechtfinden, da mutt die Inspiration für die Syntax des regulären Ausdrucks war. Lesen Sie "SEARCHING, LIMITING, AND EXPRESSIONS" im "Benutzerhandbuch" unter "`/usr/share/doc/aptitude/README`".

Anmerkung
Seit der `Lenny`-Version von aptitude(8) kann für die Suche über reguläre Ausdrücke statt der alten **kurzen Form** (wie z.B. "~b") auch die neue **lange Form** (entsprechend "`?broken`") verwendet werden. Leerzeichen " " werden jetzt als Regex-Abschluß-Zeichen gewertet, zusätzlich zu dem Tilde-Zeichen "~". Weitere Details zur Syntax der neuen **langen Form** finden Sie im Benutzerhandbuch.

Beschreibung der erweiterten Regel	**Regex-Formel**
Treffer auf Paketname	`~n`*`regex_name`*
Treffer auf Paketbeschreibung	`~d`*`regex_beschreibung`*
Treffer auf Task-Name	`~t`*`regex_task`*
Treffer auf Debtag	`~G`*`regex_debtag`*
Treffer auf Paketbetreuer	`~m`*`regex_betreuer`*
Treffer auf Paketsektion	`~s`*`regex_sektion`*
Treffer auf Paketversion	`~V`*`regex_version`*
Treffer auf Archiv	`~A{trixie,forky,sid}`
Treffer auf Paket-Herkunft	`~O{debian,…}`
Treffer auf Priorität	`~p{extra,important,optional,required,standard}`
Treffer auf essentielle Pakete	`~E`
Treffer auf virtuelle Pakete	`~v`
Treffer auf neue Pakete	`~N`
Treffer auf noch nicht abgeschlossene Aktion	`~a{install,upgrade,downgrade,remove,purge,hold,keep}`
Treffer auf installierte Pakete	`~i`
Treffer auf Pakete mit der **A**-Markierung (automatisch installierte Pakete)	`~M`
Treffer auf Pakete ohne **A**-Markierung (vom Administrator manuell zur Installation ausgewählte Pakete)	`~i!~M`
Treffer auf installierte und aktualisierbare Pakete	`~U`
Treffer auf entfernte Pakete, deren Konfigurationsdateien aber noch vorhanden sind	`~c`
Treffer auf Pakete, die entfernt oder vollständig inklusive der Konfigurationsdateien entfernt wurden oder entfernt werden können	`~g`
Treffer auf Pakete, die eine beschädigte Abhängigkeit melden	`~b`
Treffer auf Pakete, die eine beschädigte Abhängigkeit vom Typ *typ* melden	`~B`*`typ`*
Treffer auf Pakete, die eine Abhängigkeit vom Typ *typ* definiert haben	`~D[`*`typ`*`:]`*`muster`*
Treffer auf Pakete, die eine beschädigte Abhängigkeit vom Typ *typ* definiert haben	`~DB[`*`typ`*`:]`*`muster`*
Treffer auf Pakete, zu denen auf *muster* passende Pakete die Abhängigkeit *typ* definiert haben	`~R[`*`typ`*`:]`*`muster`*
Treffer auf Pakete, zu denen auf *muster* passende Pakete die beschädigte Abhängigkeit *typ* definiert haben	`~RB[`*`typ`*`:]`*`muster`*
Treffer auf Pakete, von denen andere installierte Pakete abhängig sind	`~R~i`
Treffer auf Pakete, von denen keine anderen installierten Pakete abhängig sind	`!~R~i`
Treffer auf Pakete, von denen andere installierte Pakete abhängig sind oder empfohlen werden	`~R~i|~Rrecommends:~i`
Treffer auf Pakete, die auf *muster* passen, mit gefilterter Version	`~S filter `*`muster`*
Treffer auf alle Pakete (true/wahr)	`~T`
Treffer auf kein Paket (false/unwahr)	`~F`

Tabelle 2.11: Liste von aptitudes Regex-Formeln

2.2.8 Abhängigkeitsauflösung bei aptitude

Die Auswahl eines Pakets in `aptitude` bringt nicht nur Pakete mit, die in seiner "`Depends:`"-Liste stehen, sondern auch solche in "`Recommends:`", falls die Einstellung im Menü unter `F10` → "Optionen" → "Einstellungen" → "Abhängigkeitsbehandlung" entsprechend gesetzt ist. Diese automatisch installierten Pakete werden automatisch entfernt, wenn sie laut `aptitude` nicht mehr benötigt werden.

Der Schalter, der das "auto install"-Verhalten von `aptitude` steuert, kann auch über den Befehl apt-mark(8) aus dem `apt`-Paket verändert werden.

2.2.9 Protokollierung der Paketaktivitäten

Sie können vergangene Paketaktivitäten in den Protokolldateien nachkontrollieren.

Datei	Inhalt
`/var/log/dpkg.log`	Protokollierung der Operationen auf `dpkg`-Ebene für alle Paketaktivitäten
`/var/log/apt/term.log`	Protokollierung der grundlegenden APT-Aktivitäten
`/var/log/aptitude`	Protokollierung der Aktivitäten des `aptitude`-Befehls

Tabelle 2.12: Protokolldateien für Paketaktivitäten

In der Realität ist es nicht so einfach, aus diesen Protokollen auf die Schnelle aussagekräftige Informationen herauszuziehen. Abschnitt 9.3.9 gibt Ihnen hilfreiche Infos, wie Sie dies einfacher bewältigen können.

2.3 Beispiele für aptitude-Operationen

Hier ein paar Beispiele für aptitude(8)-Operationen.

2.3.1 Suchen nach interessanten Paketen

Mit `aptitude` können Sie nach Paketen suchen, die Ihren Anforderungen entsprechen, entweder über die Paketbeschreibung oder über die Liste in "Tasks".

2.3.2 Auflisten von Paketen mit Regex-Suche auf den Paketnamen

Der folgende Befehl listet Pakete auf, bei denen die Regex-Suche nach dem Paketnamen zutrifft:

```
$ aptitude search '~n(pam|nss).*ldap'
v   libnss-ldap                                                -
p   libnss-ldapd                                               - NSS module for using LDAP  ↩
    as a naming service
v   libpam-ldap                                                -
p   libpam-ldapd                                               - PAM module for using LDAP  ↩
    as an authentication service
```

Dies ist für Sie ziemlich praktisch, um den exakten Namen eines Pakets zu finden.

2.3.3 Durchsuchen mit der Regex-Suche

Wenn Sie den regulären Ausdruck "`~dipv6`" in der "Neuen einfachen Paketansicht" im "`l`"-Dialog eingeben, werden die angezeigten Pakete auf diejenigen eingeschränkt, bei denen der Ausdruck auf die Paketbeschreibung zutrifft, und Sie können deren Informationen interaktiv durchsuchen.

2.3.4 Entfernte Pakete endgültig löschen

Sie können alle verbliebenen Konfigurationsdateien entfernter Pakete endgültig löschen.

Überprüfen Sie die Resultate des folgenden Befehls:

```
# aptitude search '~c'
```

Wenn Sie glauben, dass die aufgelisteten Pakete vollständig entfernt werden können, führen Sie folgenden Befehl aus:

```
# aptitude purge '~c'
```

Sie möchten das gleiche vielleicht im interaktiven Modus durchführen, um eine detailgenaue Kontrolle über den Vorgang zu haben:

Sie geben den regulären Ausdruck ”~c” unter der ”Neuen Paketansicht” in dem ”`l`”-Dialog ein. Dadurch werden die angezeigten Pakete auf solche eingeschränkt, auf die der reguläre Ausdruck zutrifft, d.h. ”gelöscht, aber nicht vollständig inklusive der Konfigurationsdateien entfernt”. All diese, auf den regulären Ausdruck zutreffenden Pakete können angezeigt werden, indem Sie auf der höchsten Ebene der Anzeigehierachie (z.B. ”Nicht installierte Pakete”) ”`[`” drücken.

Dann drücken Sie auf einem Eintrag in der höchsten Anzeigehierachie (z.B. auf ”Nicht installierte Pakete”) ”_”. Nur Pakete, auf die der reguläre Ausdruck passt, werden auf diese Art vollständig inklusive der Konfigurationsdateien entfernt. Sie können jedes dieser Pakete interaktiv von der vollständigen Entfernung ausschließen, indem Sie auf dem entsprechenden Paket ”=” drücken.

Diese Methode ist ziemlich praktisch und funktioniert für viele andere Befehlskürzel.

2.3.5 Automatisch/manuell-Installationsstatus bereinigen

Hier beschreibe ich, wie Sie den automatisch/manuell-Installationsstatus bereinigen können (nach der Verwendung von anderen Paketinstallationsprogrammen o.ä.):

1. Starten Sie `aptitude` im interaktiven Modus als root.
2. Drücken Sie ”u”, ”U”, ”`f`” und ”g”, um die Paketliste sowie die Pakete zu aktualisieren.
3. Drücken Sie ”`l`” und setzen Sie den Anzeigefilter auf ”`~i(~R~i|~Rrecommends:~i)`”; drücken Sie dann ”`M`” mit der Markierung auf ”`Installierte Pakete`”, um deren Status auf ”Automatisch installiert” zu ändern.
4. Drücken Sie ”`l`”, setzen Sie den Anzeigefilter auf ”`~required|~pimportant|~pstandard|~E`” und drücken Sie ”`m`” mit der Markierung auf ”`Installierte Pakete`”, um deren Status auf ”Manuell installiert” zu ändern.
5. Drücken Sie ”`l`”, setzen Sie den Anzeigefilter auf ”`~i!~M`” und entfernen Sie nicht genutzte Pakete, indem Sie ”`-`” auf jedem davon drücken (wenn die Markierung auf ”`Installierte Pakete`” steht, können Sie mit ”`[`” die Liste aufklappen, so dass alle Pakete sichtbar werden).
6. Drücken Sie ”`l`” um den Anzeigefilter auf ”`~i`” zu setzen; drücken Sie dann ”`m`”, während die Markierung auf ”`Tasks`” steht, um diese Pakete auf ”Manuell installiert” zu setzen.
7. Beenden Sie `aptitude`.
8. Führen Sie ”`apt-get -s autoremove|less`” als root aus, um zu überprüfen, welche Pakete derzeit von apt-get als ungenutzt gemeldet werden.
9. Starten Sie `aptitude` erneut im interaktiven Modus und markieren Sie Pakete, bei denen dies nötig ist, mit ”`m`” als ”Manuell installiert”.
10. Führen Sie erneut ”`apt-get -s autoremove|less`” als root aus, um erneut zu überprüfen, ob bei ENTFERNT nur die gewünschten Pakete enthalten sind.

11. Führen Sie "`apt-get autoremove | less`" als root aus, um nicht genutzte Pakete automatisch zu entfernen.

Die Aktion "m" mit der Markierung auf "`Tasks`" ist eine optionale Aktion, um die Situation massenhaft zu entfernender Pakete in der Zukunft zu vermeiden.

2.3.6 Systemweite Hochrüstung

Anmerkung
Beim Wechsel auf einen neue Veröffentlichung sollten Sie eine Neuinstallation in Erwägung ziehen, auch wenn Debian wie unten beschrieben auch auf die neue Version hochgerüstet werden kann. So haben Sie eine Chance, Müll, der sich über die Dauer der Zeit angesammelt hat, loszuwerden und bekommen die beste Kombination der aktuellsten Pakete. Natürlich sollten Sie ein vollständiges System-Backup machen und an einem sicheren Platz ablegen (lesen Sie dazu Abschnitt 10.2), bevor Sie neu installieren. Ich empfehle für einen sanften Übergang eine Dual-Boot-Installation unter Verwendung separater Partitionen.

Sie können eine systemweite Hochrüstung auf eine neue Veröffentlichung durchführen, indem Sie die Inhalte der "`sources.list`" ändern, so dass sie auf die neue Veröffentlichung zeigt; führen Sie danach "`apt update; apt dist-upgrade`" aus.

Um von `stable` nach `testing` bzw. `unstable` zu aktualisieren, während `trixie`-das aktualle `stable` ist , ersetzen Sie "`trixie`" in dem "`sources.list`"-Beispiel aus Abschnitt 2.1.5 durch "`forky`" bzw. "`sid`".

In der Realität könnte es vielleicht einige Komplikationen aufgrund von Problemen mit Paketübergängen geben, meistens wegen Paketabhängigkeiten. Je größer die Unterschiede bei der Hochrüstung sind, um so wahrscheinlicher werden Sie gravierendere Probleme bekommen. Für die Hochrüstung des alten `Stable` auf das neue `Stable` nach dessen Veröffentlichung können Sie die neuen Veröffentlichungshinweise (Release Notes) lesen. Folgen Sie exakt der dort beschriebenen Prozedur, um die möglichen Probleme zu minimieren.

Wenn Sie sich entscheiden, vor der offiziellen Freigabe der nächsten Veröffentlichung von `Stable` zu `Testing` zu wechseln, gibt es keine Veröffentlichunshinweise, die Ihnen helfen könnten. Die Unterschiede zwischen `Stable` und `Testing` könnten seit der letzten `Stable`-Veröffentlichung ziemlich groß geworden sein und die Situation für eine Hochrüstung recht komplex machen.

Sie sollten vorbeugende Schritte für das vollständige Upgrade durchführen, indem Sie aktuellste Informationen von den Mailinglisten sammeln, und Ihren gesunden Menschenverstand einsetzen.

1. Lesen Sie die vorherigen "Veröffentlichungshinweise (Release Notes)".
2. Machen Sie eine vollständige Sicherung (Backup) von Ihrem System (speziell Daten und Konfigurationsdateien).
3. Halten Sie ein boot-fähiges Medium bereit für den Fall eines beschädigten Bootloaders.
4. Informieren Sie die Benutzer des Systems rechtzeitig vorher.
5. Zeichnen Sie die Aktivitäten während des Upgrades mit script(1) auf.
6. Wenden Sie "unmarkauto" für erforderliche Pakete an, also z.B. "`aptitude unmarkauto vim`", um solche Pakete als "Manuell installiert" zu markieren und zu verhindern, dass sie entfernt werden.
7. Minimieren Sie die Anzahl installierter Pakete, um die Wahrscheinlichkeit von Paketkonflikten zu reduzieren, entfernen Sie z.B. Desktop-Task-Pakete.
8. Entfernen Sie die Datei "`/etc/apt/preferences`" (um **apt-pinning** zu deaktivieren).
9. Versuchen Sie, das Upgrade in mehreren Schritten durchzuführen: `oldstable` → `stable` → `testing` → `unstable`.
10. Aktualisieren Sie Ihre "`sources.list`", so dass sie nur auf das neue Archiv verweist, und führen Sie "`aptitude update`" aus.

11. Installieren Sie neue **Kern-Pakete** separat im Voraus, z.B. ”`aptitude install perl`” (optional).
12. Führen Sie ”`apt-get -s dist-upgrade`” aus, um die Auswirkungen im Voraus abschätzen zu können.
13. Führen Sie als letztes den Befehl ”`apt-get dist-upgrade`” aus.



Achtung
Es ist nicht klug, bei Hochrüstung einer `Stable`-Veröffentlichung auf ein anderes Stable ein oder mehrere Debian-Hauptveröffentlichungen zu überspringen (also z.B. von Debian 5.0 direkt auf 7.0 hochzurüsten und dabei 6.0 zu überspringen).



Achtung
In früheren ”Veröffentlichungshinweisen” wurden für GCC, Linux-Kernel, initrd-Werkzeuge, Glibc, Perl, APT-Werkzeuge usw. spezielle Vorsichtsmaßnahmen bei systemweiten Upgrades erwähnt.



Achtung
Die ”Veröffentlichungshinweise” decken unter Umständen nicht alle möglichen Fälle ab. Wenn Sie Konfigurationen tief im Inneren des Systems ändern, könnte Ihre nächste Systemhochrüstung mit Meldungen wie ”... segfault after upgrade ...” fehlschlagen.

Informationen zu täglichen Upgrades in `Unstable` finden Sie in Abschnitt 2.4.3.

2.4 Erweiterte Paketmanagement-Operationen

2.4.1 Erweiterte Paketmanagement-Operationen auf der Befehlszeile

Hier eine Liste weiterer Paketmanagement-Operationen, die `aptitude` nicht unterstützt, weil es auf einer anderen Ebene arbeitet oder weil ihm die nötigen Funktionen fehlen.

Anmerkung
Für ein Paket mit der multi-arch-Funktionalität müssen Sie unter Umständen bei einigen Befehlen den Architekturnamen mit angeben. Verwenden Sie zum Beispiel ”`dpkg -L libglib2.0-0:amd64`”, um die Inhalte des Pakets `libglib2.0-0` für die `amd64`-Architektur anzuzeigen.



Achtung
Grundlegende Paketwerkzeuge wie ”`dpkg -i ...`” und ”`debi ...`” sollten vom Systemadministrator mit Vorsicht eingesetzt werden. Erforderliche Paketabhängigkeiten werden dabei nicht automatisch aufgelöst. Dpkg’s Befehlszeilenoption ”`--force-all`” und ähnliche (schauen Sie in dpkg(1)) dürfen nur von Experten eingesetzt werden. Sie zu verwenden ohne deren Auswirkungen vollständig zu verstehen könnte das komplette System beschädigen.

Bitte beachten Sie folgendes:

- Alle System-Konfigurations- und Installationsbefehle müssen von root ausgeführt werden.

Befehl	Aktion
`COLUMNS=120 dpkg -l` *`paketnamen_muster`*	Status eines installierten Pakets für einen Fehlerbericht auflisten
`dpkg -L` *`paketname`*	Inhalte eines installierten Pakets auflisten
`dpkg -L` *`paketname`* `| egrep '/usr/share/man/man.*/.+'`	Handbuchseiten (manpages) für ein installiertes Paket auflisten
`dpkg -S` *`dateinamen_muster`*	Installierte Pakete auflisten, die auf das Muster passende Dateien enthalten
`apt-file search` *`dateinamen_muster`*	Pakete im Archiv auflisten, die auf das Muster passende Dateien enthalten
`apt-file list` *`paketnamen_muster`*	Inhalte von auf das Muster passenden Paketen im Archiv auflisten
`dpkg-reconfigure` *`paketname`*	Dieses Paket neu konfigurieren
`dpkg-reconfigure -plow` *`paketname`*	Dieses Paket mit detaillierten Fragen neu konfigurieren
`configure-debian`	Pakete über das Vollbildschirm-Menü neu konfigurieren
`dpkg --audit`	System nach teilweise installierten Paketen durchsuchen
`dpkg --configure -a`	Alle teilweise installierten Pakete konfigurieren
`apt-cache policy` *`binärpaketname`*	Verfügbare Version, Priorität und Archivinformationen eines Binärpakets anzeigen
`apt-cache madison` *`paketname`*	Verfügbare Version und Archivinformationen eines Pakets anzeigen
`apt-cache showsrc` *`binärpaketname`*	Quellpaket-Informationen eines Binärpakets anzeigen
`apt-get build-dep` *`paketname`*	Zum Bau eines Pakets benötigte Pakete installieren
`aptitude build-dep` *`paketname`*	Zum Bau eines Pakets benötigte Pakete installieren
`apt-get source` *`paketname`*	Ein Quellpaket herunterladen (aus dem Standardarchiv)
`dget` *`URL für dsc-Datei`*	Ein Quellpaket herunterladen (aus einem anderen Archiv)
`dpkg-source -x` *`paketname_version-debian.revision.dsc`*	Einen Quellcodebaum aus einem Satz von Quellpaketen erstellen (”`*.orig.tar.gz`” und ”`*.debian.tar.gz`” / ”`*.diff.gz`”)
`debuild binary`	Paket(e) aus einem lokalen Quellcodebaum bauen
`make-kpkg kernel_image`	Ein Kernelpaket aus einem Kernel-Quellcodebaum bauen
`make-kpkg --initrd kernel_image`	Ein Kernelpaket mit aktivierter initramfs aus einem Kernel-Quellcodebaum bauen
`dpkg -i` *`paketname_version-debian.revision_arch`*`.deb`	Ein lokales Paket in das System installieren
`apt install /pfad/zu/`*`paketname`*`.deb`	Ein lokales Paket in das System installieren und dabei versuchen, die Abhängigkeiten automatisch aufzulösen
`debi` *`paketname_version-debian.revision_arch`*`.dsc`	Lokale(s) Paket(e) in das System installieren
`dpkg --get-selections '*' >selection.txt`	`dpkg`-Paketauswahlinformationen sichern
`dpkg --set-selections <selection.txt`	`dpkg`-Paketauswahlinformationen setzen
`echo` *`paketname`* `hold | dpkg --set-selections`	`dpkg`-Paketauswahlinformationen eines Pakets auf **hold** (halten) setzen (gleichbedeutend mit ”`aptitude hold` *`paketname`*”)

Tabelle 2.13: Liste erweiterter Paketmanagement-Operationen

- Anders als `aptitude`, das reguläre Ausdrücke verwendet (mehr dazu in Abschnitt 1.6.2), nutzen andere Paketmanagemer Befehle Suchmuster wie Shell-Globs (Details in Abschnitt 1.5.6).
- Für apt-file(1) aus dem `apt-file`-Paket muss zuvor "`apt-file update`" ausgeführt werden.
- configure-debian(8) aus dem `configure-debian`-Paket führt dpkg-reconfigure(8) als Backend im Hintergrund aus.
- dpkg-reconfigure(8) führt Paketskripte aus, die debconf(1) als Backend im Hintergrund verwenden.
- Die Befehle "`apt-get build-dep`", "`apt-get source`" und "`apt-cache showsrc`" erfordern einen "`deb-src`"-Eintrag in Ihrer "`sources.list`".
- dget(1), debuild(1) und debi(1) erfordern die Installation des `devscripts`-Pakets.
- Infos zur Prozedur des (Neu-)Paketierens mittels "`apt-get source`" finden Sie in Abschnitt 2.7.13.
- Der `make-kpkg`-Befehl erfordert die Installation des `kernel-package`-Pakets (lesen Sie Abschnitt 9.10).
- Grundsätzliche Informationen über das Paketieren finden Sie in Abschnitt 12.9.

2.4.2 Verifizierung von installierten Paketdateien

Die Installation von `debsums` ermöglicht die Verifizierung installierter Pakete über MD5sum-Werte aus der Datei "`/var/lib/dpkg/info/*.md5sums`" mittels debsums(1). Details darüber, wie MD5sum arbeitet, finden Sie in Abschnitt 10.3.5.

Anmerkung
Da die MD5sum-Datenbank durch Eindringlinge gefälscht werden könnte, ist debsums(1) als Werkzeug für die Systemsicherheit nur von begrenztem Nutzen. Es eignet sich aber gut dafür, lokale Veränderungen durch den Administrator oder durch Beschädigungen aufgrund von Fehlern des Speichermediums zu erkennen.

2.4.3 Absicherungen für den Fall von Paketproblemen

Viele Benutzer bevorzugen die **Testing**- oder **Unstable**-Veröffentlichung des Debian-Systems wegen seiner neuen Funktionen und Pakete. Dies macht deren System jedoch anfällig für kritische Paketfehler.

Die Installation des `apt-listbugs`-Pakets schützt Ihr System vor kritischen Fehlern, indem bei Upgrades durch das APT-System automatisch die Debian-Fehlerdatenbank (BTS) bezüglich kritischer Fehler abgefragt wird.

Die Installation des `apt-listchanges`-Pakets stellt Ihnen wichtige Neuigkeiten aus "`NEWS.Debian`" zur Verfügung, wenn Upgrades durch das APT-System durchgeführt werden.

2.4.4 Durchsuchen der Paket-Metadaten

Obwohl der Besuch der Debian-Site https://packages.debian.org/ heutzutage einfache Möglichkeiten bietet, die Paket-Metadaten zu durchsuchen, wollen wir uns auch die traditionellen Wege anschauen.

Die Befehle grep-dctrl(1), grep-status(1) und grep-available(1) können verwendet werden, um jegliche Datei zu durchsuchen, die das grundsätzliche Format einer control-Datei für ein Debian-Paket hat.

"`dpkg -S` *`dateinamen_muster`*" kann verwendet werden, um durch `dpkg` installierte Pakete zu finden, die auf das Suchmuster passende Dateinamen enthalten. Hiermit werden jedoch keine Dateien gefunden, die durch Skripte der Paketbetreuer erzeugt wurden.

Wenn Sie eine tiefergehende Suche der dpkg-Metadaten benötigen, müssen Sie den Befehl "`grep -e regex_muster *`" im Verzeichnis "`/var/lib/dpkg/info/`" ausführen. So können Sie Wörter finden, die in Paketskripten und Texten für Abfragen bei der Installation auftauchen.

Wenn Sie Paketabhängigkeiten rekursiv abfragen möchten, sollten Sie apt-rdepends(8) verwenden.

2.5 Internas des Debian-Paketmanagements

Lassen Sie uns betrachten, wie das Debian-Paketmanagement-System intern funktioniert. Dies sollte Ihnen dabei helfen, eigene Lösungen für gewisse Paketprobleme zu finden.

2.5.1 Archiv-Metadaten

Metadaten-Dateien für die jeweilige Distribution sind auf jedem Debian-Spiegel unter "`dist/`*`codename`*" abgelegt, z.B. auf "`http://deb.debian.org/debian/`". Sie können die Archivstruktur mit einem Webbrowser durchsuchen. Es gibt sechs Arten von wichtigen Metadaten:

Datei	Speicherort	Inhalt
`Release`	Wurzelverzeichnis der Distribution	Archivbeschreibung und Integritätsinformationen
`Release.gpg`	Wurzelverzeichnis der Distribution	Signaturdatei für die "`Release`"-Datei, mit dem Archiv-Schlüssel signiert
`Contents-`*`architektur`*	Wurzelverzeichnis der Distribution	Liste aller Dateien für alle Pakete in dem entsprechenden Archiv
`Release`	Wurzelverzeichnis aller Distributions-/Bereichs-/Architektur-Kombinationen	Archivbeschreibung, die zur Festlegung von apt_preferences(5) verwendet wird
`Packages`	Wurzelverzeichnis aller Distributions-/Bereichs-/Binärarchitektur-Kombinationen	vereinigte `debian/control` für Binärpakete
`Sources`	Wurzelverzeichnis aller Distributions-/Bereichs-/Quellen-Kombinationen	vereinigte `debian/control` für Quellpakete

Tabelle 2.14: Inhalt der Metadaten des Debian-Archivs

Im aktuellen Archiv sind diese Metadaten als komprimierte und differenzielle Dateien abgelegt, um Netzwerkverkehr zu reduzieren.

2.5.2 "Release"-Datei im Wurzelverzeichnis und Authentizität

Tipp
Die "Release"-Datei im Wurzelverzeichnis wird verwendet, um die Archive im **Secure-APT**-System zu signieren.

Jede Suite im Debian-Archiv hat im Wurzelverzeichnis eine "`Release`"-Datei, z.B. "`http://deb.debian.org/debian/d`
wie hier:

```
Origin: Debian
Label: Debian
Suite: unstable
Codename: sid
Date: Sat, 14 May 2011 08:20:50 UTC
Valid-Until: Sat, 21 May 2011 08:20:50 UTC
Architectures: alpha amd64 armel hppa hurd-i386 i386 ia64 kfreebsd-amd64 kfreebsd-i386 mips ↩
    mipsel powerpc s390 sparc
Components: main contrib non-free
Description: Debian x.y Unstable - Not Released
```

```
MD5Sum:
 bdc8fa4b3f5e4a715dd0d56d176fc789 18876880 Contents-alpha.gz
 9469a03c94b85e010d116aeeab9614c0 19441880 Contents-amd64.gz
 3d68e206d7faa3aded660dc0996054fe 19203165 Contents-armel.gz
...
```

Anmerkung
Hier finden Sie den Grund dafür, warum ich in Abschnitt 2.1.5 ”Suite” und ”Codename” verwende. Die ”Distribution” wird benutzt, um sowohl auf ”Suite” wie auch auf ”Codename” zu verweisen. Alle Namen der Archiv-Bereiche, die von dem Archiv angeboten werden, sind unter ”Components” aufgelistet.

Die Integrität der ”`Release`”-Datei im Wurzelverzeichnis wird über die kryptografische Infrastruktur namens Secure-APT verifiziert, wie in apt-secure(8) beschrieben.

- Die kryptografische Signaturdatei ”`Release.gpg`” wird aus der authentischen ”`Release`”-Datei im Wurzelverzeichnis und dem geheimen Debian-Archiv-Schlüssel erzeugt.
- Die öffentlichen Debian-Archiv-Schlüssel werden lokal mittels des aktuellsten `debian-archive-keyring`-Pakets installiert.
- Das **Secure-APT**-System verifiziert automatisch kryptografisch die Integrität der heruntergeladenen ”`Release`”-Datei im Wurzelverzeichnis über diese ”`Release.gpg`”-Datei und die lokal installierten Debian-Archiv-Schlüssel.
- Die Integrität all der ”`Packages`”- und ”`Sources`”-Dateien wird über MD5sum-Werte in der ”`Release`”-Datei im Wurzelverzeichnis verifiziert. Die Integrität der Paketdateien wird über MD5sum-Werte in den ”`Packages`”- und ”`Sources`”-Dateien verifiziert. Lesen Sie dazu debsums(1) und Abschnitt 2.4.2.
- Da die kryptografische Signaturverifizierung viel CPU-intensiver ist als die Berechnung von MD5sum-Werten, bietet die Verwendung von MD5sum-Werten für die Pakete bei gleichzeitiger Nutzung einer kryptografischen Signatur für die ”`Release`”-Datei im Wurzelverzeichnis einen guten Kompromiss zwischen Sicherheit und Performance (weiteres in Abschnitt 10.3).

Wenn der **sources.list**-Eintrag die ”`signed-by`”-Option enthält, wird die Integrität der heruntergeladenen ”`Release`”-Datei des obersten Pfad-Levels mittels des angegebenen öffentlichen Schlüssels verifiziert. Dies ist nützlich, wenn die **sources.list** Einträge für nicht-Debian-Archive enthält.

Sie können die Integrität der ”`Release`”-Datei auch händisch mittels `gpg` mit der ”`Release.gpg`”-Datei und dem öffentlichen Schlüssel des Debian-Archivs verifizieren, der auf ftp-master.debian.org zu finden ist.

2.5.3 ”Release”-Dateien im Archivverzeichnis

Tipp
Die ”Release”-Dateien im Archivverzeichnis werden zur Festlegung von apt_preferences(5) genutzt.

Es gibt ”`Release`”-Dateien in allen Archivverzeichnissen, die über **sources.list**-Einträge wie ”`http://deb.debian.org/d` oder ”`http://deb.debian.org/debian/dists/sid/main/binary-amd64/Release`” angegeben werden, wie hier:

```
Archive: unstable
Origin: Debian
Label: Debian
Component: main
Architecture: amd64
```



Achtung
Für "`Archive:`" werden im Debian-Archiv Suite-Namen ("`stable`", "`testing`", "`unstable`", ...) verwendet, im Ubuntu-Archiv jedoch Codenamen ("`trusty`", "`xenial`", "`artful`", ...).

Für einige Archive (wie `experimental` und `trixie-backports`), die Pakete enthalten, welche nicht automatisch installiert werden sollten, z.B. "`http://deb.debian.org/debian/dists/experimental/main/binary-amd64/Re` existiert eine zusätzliche Zeile, wie hier:

```
Archive: experimental
Origin: Debian
Label: Debian
NotAutomatic: yes
Component: main
Architecture: amd64
```

Bitte beachten Sie, dass für normale Archive ohne "`NotAutomatic: yes`" der Wert für die Pin-Priorität 500 ist, während für besondere Archive mit "`NotAutomatic: yes`" dieser Wert 1 ist (lesen Sie dazu apt_preferences(5) und Abschnitt 2.7.7).

2.5.4 Empfangen der Metadaten für ein Paket

Um APT-Werkzeuge wie `aptitude`, `apt-get`, `synaptic`, `apt-file`, `auto-apt` ... zu verwenden, müssen wir die lokalen Kopien der Metadaten, die die Debian-Archivinformationen enthalten, aktualisieren. Diese lokalen Kopien haben die folgenden Dateinamen, passend zu den in der "`sources.list`" festgelegten Namen für `Distribution`, `Bereich` und `Architektur` (weiteres in Abschnitt 2.1.5):

- "`/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribution`*`_Release`"
- "`/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribution`*`_Release.gpg`"
- "`/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribution_area`*`_binary-`*`architektur`*`_Pack`
- "`/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribution_bereich`*`_source_Sources`"
- "`/var/cache/apt/apt-file/deb.debian.org_debian_dists_`*`distribution`*`_Contents-`*`architektur`*`.gz`
  (für `apt-file`)

Die ersten vier Typen obiger Dateien werden von allen entsprechenden APT-Befehlen gemeinsam genutzt und mittels "`apt-get update`" oder "`aptitude update`" aktualisiert. Die "`Packages`"-Metadaten werden aktualisiert, wenn "`deb`" in der "`sources.list`" angegeben ist. "`Sources`"-Metadaten werden aktualisiert, wenn "`deb-src`" in der "`sources.list`" existiert.

Die "`Packages`"- und "`Sources`"-Metadaten enthalten einen "`Filename:`"-Eintrag, der auf den Speicherort der Binär- und Quellpakete verweist. Derzeit sind diese Pakete in dem Verzeichnisbaum unterhalb von "`pool/`" abgelegt zwecks einfacher Übergänge zwischen den Veröffentlichungen.

Lokale Kopien der "`Packages`"-Metadaten können mit Hilfe von `aptitude` interaktiv durchsucht werden. Der spezialisierte Suchbefehl grep-dctrl(1) kann lokale Kopien der "`Packages`"- und "`Sources`"-Metadaten durchsuchen.

Lokale Kopien von "`Contents-`*`architektur`*"-Metadaten können mittels "`apt-file update`" aktualisiert werden und ihr Speicherort unterscheidet sich von den vier anderen. Lesen Sie dazu apt-file(1). (`auto-apt` nutzt standardmäßig andere Speicherorte für die lokale Kopie von "`Contents-`*`architektur`*`.gz`".)

2.5.5 Der Paketstatus für APT

Zusätzlich zu den von außerhalb empfangenen Metadaten speichert das APT-Werkzeug seit `Lenny` seine lokal erzeugten Installationsstatus-Informationen in der Datei "`/var/lib/apt/extended_states`", die von allen APT-Programmen genutzt wird, um automatisch installierte Pakete zu verfolgen.

2.5.6 Der Paketstatus von aptitude

Zusätzlich zu den von außerhalb empfangenen Metadaten speichert der `aptitude`-Befehl seine lokal erzeugten Installationsstatus-Informationen in der Datei "`/var/lib/aptitude/pkgstates`", die nur von aptitude selbst genutzt wird.

2.5.7 Lokale Kopien der empfangenen Pakete

Alle über den APT-Mechanismus von außerhalb empfangenen Pakete werden im Verzeichnis "`/var/cache/apt/archive` abgelegt, bis sie gelöscht werden.

Sie können diese Richtlinie zur Entfernung von zwischengespeicherten Dateien bei `aptitude` unter "`Optionen`" → "`Einstellungen`" festlegen; über das Menü unter "`Aktionen`" → "`Paketcache komplett leeren`" oder "`Nur veraltete Paketdateien löschen`" können Sie dies auch händisch erzwingen.

2.5.8 Debian-Paketdateinamen

Die Dateien von Debian-Paketen haben eine bestimmte Namensstruktur:

Pakettyp	Namensstruktur
Binärpaket (auch deb)	*paketname_upstream-version-debian.revision_architektur*
Binärpaket für den debian-installer (auch udeb)	*paketname_upstream-version-debian.revision_architetur.u*
Quellpaket (Quellcode der Originalprogrammierer)	*paketname_upstream-version-debian.revision*.orig.tar.gz
`1.0`-Quellpaket (Debian-Änderungen)	*paketname_upstream-version-debian.revision*.diff.gz
`3.0 (quilt)`-Quellpaket (Debian-Änderungen)	*paketname_upstream-version-debian.revision*.debian.tar.g
Quellpaket (Beschreibung)	*paketname_upstream-version-debian.revision*.dsc

Tabelle 2.15: Namensstruktur von Debian-Paketen

Tipp
Hier sind nur die grundlegenden Quellpaketformate beschrieben. Lesen Sie weitere Details in dpkg-source(1).

Namenskomponente	zu verwendende Zeichen (erweiterter regulärer Ausdruck)	Existenz
paketname	`[a-z0-9][-a-z0-9.+]+`	erforderlich
epoche`:`	`[0-9]+:`	optional
upstream-version	`[-a-zA-Z0-9.+:]+`	erforderlich
debian.revision	`[a-zA-Z0-9.+~]+`	optional

Tabelle 2.16: In den einzelnen Komponenten von Debian-Paketnamen zu verwendende Zeichen

Anmerkung
Sie können die Reihenfolge von Paketversionen (welche Versionsnummer höher ist) mit dpkg(1) überprüfen, z.B. mittels "`dpkg --compare-versions 7.0 gt 7.~pre1 ; echo $?`".

Anmerkung
Der debian-installer (d-i) nutzt udeb als Dateinamenerweiterung für seine Binärpakete statt des normalen deb. Ein udeb-Paket ist ein reduziertes deb-Paket, in dem einige nicht-essentielle Bestandteile wie Dokumentation entfernt wurden, um Speicherplatz zu sparen; dazu wurden die Anforderungen an die Paketrichtlinien etwas gelockert. Sowohl deb- wie auch udeb-Pakete verwenden die gleiche Paketstruktur. Das ”u” steht für micro (sehr klein).

2.5.9 Der dpkg-Befehl

dpkg(1) ist das Werkzeug, das beim Debian-Paketmanagement auf der untersten Ebene arbeitet. Es ist sehr leistungsfähig und muss mit Vorsicht verwendet werden.

Beim Installieren eines Pakets ”*paketname*” arbeitet dpkg folgende Schritte ab:

1. die deb-Datei auspacken (gleichbedeutend mit ”ar -x”);
2. ausführen von ”*paketname*.preinst” mittels debconf(1);
3. installieren des Paketinhalts in das System (gleichbedeutend zu ”tar -x”);
4. ausführen von ”*paketname*.postinst” mittels debconf(1).

Das debconf-System bietet eine standardisierte Benutzer-Schnittstelle mit Unterstützung für I18N und L10N (Näheres in Kapitel 8).

Datei	Beschreibung des Inhalts
/var/lib/dpkg/info/*paketname*.conffiles	Liste der Konfigurationsdateien (durch den Benutzer änderbar)
/var/lib/dpkg/info/*paketname*.list	Liste von Dateien und Verzeichnissen, die durch das Paket installiert werden
/var/lib/dpkg/info/*paketname*.md5sums	Liste der MD5-Hash-Werte für Dateien, die durch das Paket installiert werden
/var/lib/dpkg/info/*paketname*.preinst	Paket-Skript, das vor der Paketinstallation ausgeführt wird
/var/lib/dpkg/info/*paketname*.postinst	Paket-Skript, das nach der Paketinstallation ausgeführt wird
/var/lib/dpkg/info/*paketname*.prerm	Paket-Skript, das vor der Paketentfernung ausgeführt wird
/var/lib/dpkg/info/*paketname*.postrm	Paket-Skript, das nach der Paketentfernung ausgeführt wird
/var/lib/dpkg/info/*paketname*.config	Paket-Skript für das debconf-System
/var/lib/dpkg/alternatives/*paketname*	alternative-Informationen, die durch den update-alternatives-Befehl genutzt werden
/var/lib/dpkg/available	availability-Informationen (Verfügbarkeit) für alle Pakete
/var/lib/dpkg/diversions	die diversions-Informationen, die durch dpkg(1) genutzt und durch dpkg-divert(1) gesetzt werden
/var/lib/dpkg/statoverride	die stat-override-Informationen, die durch dpkg(1) genutzt und durch dpkg-statoverride(1) gesetzt werden
/var/lib/dpkg/status	Status-Informationen für alle Pakete
/var/lib/dpkg/status-old	Backup der ersten Generation von ”var/lib/dpkg/status”
/var/backups/dpkg.status*	Backup der zweiten Generation und ältere von ”var/lib/dpkg/status”

Tabelle 2.17: Erwähnenswerte Dateien, die durch dpkg erzeugt werden

Die ”status”-Datei wird auch von Werkzeugen wie dpkg(1), ”dselect update” und ”apt-get -u dselect-upgrade” verwendet.

Der spezialisierte Suchbefehl grep-dctrl(1) kann lokale Kopien der ”status”- und ”available”-Metadaten durchsuchen.

Tipp
In der debian-installer-Umgebung wird der Befehl `udpkg` benutzt, um `udeb`-Pakete zu öffnen. `udpkg` ist eine reduzierte Variante des `dpkg`-Befehls.

2.5.10 Der update-alternatives-Befehl

Das Debian-System hat Mechanismen, um Programme, die sich bei der Funktionalität ein wenig überlappen, friedlich nebeneinander zu installieren, und zwar mittels update-alternatives(1). Zum Beispiel können Sie den `vi`-Befehl so einrichten, dass er `vim` ausführt, wenn Sie sowohl das `vim` wie auch das `nvi`-Paket installiert haben.

```
$ ls -l $(type -p vi)
lrwxrwxrwx 1 root root 20 2007-03-24 19:05 /usr/bin/vi -> /etc/alternatives/vi
$ sudo update-alternatives --display vi
...
$ sudo update-alternatives --config vi
  Selection    Command
 -----------------------------------------------
      1        /usr/bin/vim
*+    2        /usr/bin/nvi

Enter to keep the default[*], or type selection number: 1
```

Das Debian-alternatives-System setzt seine Auswahl als symbolischer Link in "`/etc/alternatives/`". Der Auswahlprozess nutzt die entsprechende Datei in "`/var/lib/dpkg/alternatives/`".

2.5.11 Der dpkg-statoverride-Befehl

Stat overrides, bereitgestellt durch den dpkg-statoverride(1)-Befehl, sind ein Weg, dpkg(1) mitzuteilen, dass bei Installation eines Pakets für eine **Datei** ein anderer Eigentümer oder andere Berechtigungen verwendet werden sollen. Falls "`--update`" angegeben wurde und die Datei bereits existiert, werden direkt der neue Eigentümer oder neue Berechtigungen gesetzt.



Achtung
Die direkte Abänderung von Benutzer oder Berechtigungen durch den Systemadministrator (über die Befehle `chmod` oder `chown`) bei einer **Datei**, die zu einem Paket gehört, wird bei der nächsten Aktualisierung des Pakets wieder aufgehoben.

Anmerkung
Ich verwende hier den Begriff **Datei**, jedoch kann dies jegliches Objekt im Dateisystem sein, dass durch `dpkg` behandelt wird, inklusive Verzeichnissen, Gerätedateien usw.

2.5.12 Der dpkg-divert-Befehl

Diversions, bereitgestellt durch den dpkg-divert(1)-Befehl, bieten die Möglichkeit, dpkg(1) zu zwingen, eine Datei nicht an Ihren Standardort zu installieren, sondern an einen umgelenkten Speicherort (**diverted**). `dpkg-divert` ist zur Verwendung durch Paketverwaltungs-Skripte gedacht. Die beiläufige Verwendung durch den Systemadministrator ist nicht mehr zeitgemäß.

2.6 Wiederherstellung eines beschädigten Systems

Bei einem `testing`- oder `unstable`-System sollte der Administrator darauf vorbereitet sein, das System im Falle einer Beschädigung des Paketmanagements wiederherstellen zu können.



Achtung
Einige Methoden, die hier beschrieben werden, beinhalten ein hohes Risiko. Sie wurden gewarnt!

2.6.1 Inkompatibilität mit alter Benutzerkonfiguration

Wenn ein Desktop-GUI-Programm nach signifikanten Upgrades von Upstream Instabilitäten erkennen lässt, sollten Sie Behinderungen durch alte lokal erstellte Konfigurationsdateien als Ursache in Betracht ziehen. Falls das Programm unter einem neu erstellten Benutzerkonto stabil läuft, ist diese Hypothese bestätigt. (Dies ist ein Fehler beim Paketieren und wird für gewöhnlich durch den Paketersteller vermieden.)

Um das Programm wieder stabil zu bekommen, sollten Sie die entsprechenden lokalen Konfigurationsdateien beiseite schaffen und das GUI-Programm neu starten. Sie müssen möglicherweise die Inhalte alter Konfigurationsdateien lesen, um die Konfigurationsinformationen später wiederherstellen zu können. (Löschen Sie sie nicht zu früh.)

2.6.2 Fehler beim Zwischenspeichern der Paketdaten

Fehler beim Zwischenspeichern der Paketdaten führen zu verblüffenden Meldungen, wie "GPG error: ... invalid: BADSIG ..." bei APT.

Sie sollten alle zwischengespeicherten Daten mittels "`sudo rm -rf /var/lib/apt/* `" entfernen und es erneut versuchen. (Falls Sie `apt-cacher-ng` verwenden, führen Sie auch "`sudo rm -rf /var/cache/apt-cacher-ng/* `" aus.)

2.6.3 Systemrettung mit dem dpkg-Befehl

Das `dpkg`-Werkzeug arbeitet tief im Inneren und kann auch ohne Netzwerkverbindung funktionieren.

Gehen wir davon aus, dass das Paket "`foo`" beschädigt ist und wieder funktionsfähig gemacht werden soll.

Sie könnten im cache-Verzeichnis "`/var/cache/apt/archives/`" zwischengespeicherte Kopien von älteren, fehlerfreien Versionen des Pakets `foo` finden. (Falls nicht, können Sie diese auch aus dem Archiv unter https://snapshot.debian.o herunterladen oder von einem funktionierenden System herüber kopieren.)

Ist das System boot-fähig, könnte es unter Umständen möglich sein, das Paket mit folgendem Befehl zu installieren:

```
# dpkg -i /path/to/foo_old_version_arch.deb
```

Falls der Versuch, ein Paket auf diese Art zu installieren, aufgrund von verletzten Abhängigkeiten fehlschlägt, können Sie als letzten Ausweg Abhängigkeiten mit den `dpkg`-Optionen "`--ignore-depends`", "`--force-depends`" und weiteren überschreiben. Dabei müssen Sie besondere Sorgfalt darauf verwenden, die Abhängigkeiten später korrekt wiederherzustellen. Details finden Sie in dpkg(8).

Anmerkung
Falls Ihr System ernsthaft beschädigt ist, sollten ein vollständiges Backup aller Daten an einem sicheren Ort ablegen (siehe Abschnitt 10.2) und eine saubere Neuinstallation durchführen. Dies ist weniger zeitaufwändig und führt am Ende zu einem besseren Ergebnis.

Tipp
Wenn die Beschädigung des Systems nur gering ist, könnten Sie das ganze System möglicherweise mit dem APT-System auf eine ältere Version zurückrüsten, wie in Abschnitt 2.7.11 beschrieben.

2.6.4 Fehlgeschlagene Installation aufgrund von fehlenden Abhängigkeiten

Wenn Sie die Installation eines Pakets über "`sudo dpkg -i ...`" auf einem System erzwingen, auf dem nicht alle erforderlichen Pakete vorhanden sind, wird die Paketinstallation mit dem Status "`nicht vollständig installiert`" fehlschlagen.

Sie sollten alle benötigten Pakete über wiederholte Aufrufe von "`sudo dpkg -i ...`" installieren, oder mittels:

```
# apt --fix-broken install
```

Konfigurieren Sie dann alle unvollständig installierten Pakete mit folgendem Befehl:

```
# dpkg --configure -a
```

2.6.5 Mehrere Pakete mit überlappenden Dateien

Paketmanagementsysteme für das Debian-Archiv wie aptitude(8) oder apt-get(8) versuchen erst gar nicht, Pakete mit überlappenden Dateien über Paketabhängigkeiten zu installieren (weiteres in Abschnitt 2.1.7).

Fehler durch einen Paketbetreuer oder die Einrichtung von inkonsistent gemischten Archivquellen durch den Systemadministrator (lesen Sie Abschnitt 2.7.6) könnten Situationen mit falsch definierten Paketabhängigkeiten hervorrufen. Wenn Sie in solch einer Situation ein Paket mit überlappenden Dateien mittels aptitude(8) oder apt-get(8) installieren, stellt dpkg(1), das das Paket entpackt, sicher, dass ein Fehler an das aufrufende Programm zurückgegeben wird, ohne dass vorhandene Dateien überschrieben werden.



Achtung
Die Verwendung von Paketen aus Drittquellen bringt durch Betreuerskripte ein erhebliches Risiko für das System mit sich, da diese mit root-Privilegien ausgeführt werden und jegliche Aktion auf dem System ausführen können. Der dpkg(1)-Befehl schützt lediglich vor dem Überschreiben beim Entpacken.

Sie können solche beschädigten Installationen umgehen, indem Sie vorher das alte betroffene Paket *`altes-paket`* entfernen:

```
$ sudo dpkg -P old-package
```

2.6.6 Behebung von Problemen aufgrund von beschädigtem Paketskript

Wenn ein Befehl in einem Paketskript aus irgendeinem Grund einen Fehler zurückgibt und das Skript aufgrund des Fehlers abgebrochen wird, beendet das Paketmanagement die Aktion und es bleiben halb-installierte Pakete zurück. Enthält ein Paket Programmierfehler in seinen Skripten zum Löschen des Pakets, könnte es sein, dass das Paket nicht mehr entfernt werden kann und ziemlich unangenehm wird.

Wenn das Paket "*`paketname`*" ein Problem mit den Skripten hat, sollten Sie in folgenden Paketskripten nachschauen:

- "`/var/lib/dpkg/info/`*`paketname`*`.preinst`"
- "`/var/lib/dpkg/info/`*`paketname`*`.postinst`"

- "`/var/lib/dpkg/info/paketname.prerm`"
- "`/var/lib/dpkg/info/paketname.postrm`"

Editieren Sie die angebotenen Paketskripte von Grund auf mittels folgender Techniken:

- deaktivieren Sie die problematische Zeile, indem Sie ein "#" voranstellen;
- erzwingen Sie, dass das Skript einen Rückgabewert für Erfolg zurück gibt, indem Sie der fraglichen Zeile "`|| true`" anhängen.

Konfigurieren Sie dann alle unvollständig installierten Pakete mit folgendem Befehl:

```
# dpkg --configure -a
```

2.6.7 Paketauswahldaten wiederherstellen

Falls "`/var/lib/dpkg/status`" aus irgendeinem Grund beschädigt wird, verliert das Debian-System die Daten über ausgewählte Pakete und kommt dadurch erheblich zu Schaden. Sie finden alte "`/var/lib/dpkg/status`"-Dateien in "`/var/lib/dpkg/status-old`" oder "`/var/backups/dpkg.status.*`".

Es könnte eine gute Idee sein, "`/var/backups/`" auf einer separaten Partition abzulegen, da dieses Verzeichnis viele wichtige Systemdaten enthält.

Bei ernsten Beschädigungen empfehle ich eine frische Neuinstallation, nachdem eine Sicherung der Daten durchgeführt wurde. Sogar wenn alles in "`/var/`" verloren ist, können Sie trotzdem noch etliche Informationen aus Verzeichnissen in "`/usr/share/doc/`" wiederherstellen, um Ihre neue Installation zu lenken:

Installieren Sie ein minimales (Desktop-)System.

```
# mkdir -p /path/to/old/system
```

Binden Sie die alten Systempartitionen in "`/pfad/zum/alten/system/`" ein.

```
# cd /path/to/old/system/usr/share/doc
# ls -1 >~/ls1.txt
# cd /usr/share/doc
# ls -1 >>~/ls1.txt
# cd
# sort ls1.txt | uniq | less
```

Jetzt haben Sie eine Liste mit Paketnamen, die Sie installieren können. (Es könnten auch einige Zeichenfolgen darunter sein, die keine Paketnamen sind, wie z.B. "`texmf`".)

2.7 Tipps für das Paketmanagement

Der Einfachheit halber sind die **sources.list**-Beispiele in diesem Abschnitt im einzeiligen Stil angegeben, nach Freigabe von `trixie`.

2.7.1 Wer hat das Paket hochgeladen?

Obwohl der Betreuername, der in "`/var/lib/dpkg/available`" und "`/usr/share/doc/package_name/changelog`" aufgelistet ist, Informationen darüber gibt, "wer hinter den Paketaktivitäten steht", sind Informationen über den Uploader (die Person, die das Paket wirklich hochgeladen hat) ein wenig verschleiert. who-uploads(1) aus dem `devscripts`-Paket identifiziert die realen Uploader von Debian-Quellpaketen.

2.7.2 Die Download-Bandbreite für APT einschränken

Falls Sie die Download-Bandbreite für APT einschränken möchten, sagen wir z.B. auf 800Kib/sec (=100kiB/sec), sollten Sie APT mit folgenden Konfigurationsparametern einrichten:

```
APT::Acquire::http::Dl-Limit "800";
```

2.7.3 Automatisches Herunterladen und Aktualisieren von Paketen

Das `apt`-Paket enthält ein eigenes cron-Skript (”`/etc/cron.daily/apt`”), um den automatischen Download von Paketen zu unterstützen. Durch die Installation des `unattended-upgrades`-Pakets kann dieses Skript erweitert werden, so dass auch die automatische Aktualisierung von Paketen durchgeführt wird. Sie können dies über Parameter in ”`/etc/apt/apt.conf.d/02backup`” und ”`/etc/apt/apt.conf.d/50unattended-upgrades`” noch weiter anpassen, wie in ”`/usr/share/doc/unattended-upgrades/README`” beschrieben.

Das `unattended-upgrades`-Paket ist hauptsächlich für Sicherheits-Upgrades auf `Stable`-Systemen gedacht. Wenn das Risiko, ein vorhandenes `Stable`-System über einen automatischen Upgrade-Prozess zu beschädigen, geringer ist als das, welches von einem Angreifer ausgeht, der über Ausnutzung einer Sicherheitslücke das System schädigt, sollten Sie ein automatisches Upgrade in Erwägung ziehen, das diese Lücke über ein Sicherheits-Upgrade schließt. Nutzen Sie dazu folgende Konfigurationsparameter:

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::Unattended-Upgrade "1";
```

Wenn Sie ein `Testing`- oder `Unstable`-System laufen haben, sollten Sie kein automatisches Upgrade verwenden, da sonst mit Sicherheit eines Tages das System beschädigt würde. Aber sogar für solch ein `Testing`- oder `Unstable`-System möchten Sie vielleicht in Vorbereitung für ein interaktives Upgrade die Pakete schon mal automatisch herunterladen, um Zeit zu sparen. Verwenden Sie dazu folgende Konfigurationsparameter:

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::Unattended-Upgrade "0";
```

2.7.4 Aktualisierungen und Backports

Es gibt stable-updates- (”`trixie`-updates”, während `trixie` als `stable` gilt) und backports.debian.org-Archive, die aktualisierte Pakete für `stable` bereitstellen.

Um diese Archive zu nutzen, listen Sie alle erforderlichen Archive in der ”`/etc/apt/sources.list`”-Datei auf, wie hier:

```
deb http://deb.debian.org/debian/ trixie main non-free-firmware contrib non-free
deb http://security.debian.org/debian-security trixie-security main non-free-firmware ↩
    contrib non-free
deb http://deb.debian.org/debian/ trixie-updates main non-free-firmware contrib non-free
deb http://deb.debian.org/debian/ trixie-backports main non-free-firmware contrib non-free
```

Es ist nicht nötig, einen Wert für die Pin-Priorität in ”`/etc/apt/preferences`” zu setzen. Wenn neuere Pakete verfügbar werden, werden mit der Standardkonfiguration passende Upgrades bereitgestellt (siehe Abschnitt 2.5.3).

- Alle installierten veralteten Pakete werden auf die neuen von `trixie-updates` aktualisiert.
- Nur manuell installierte veraltete Pakete von `trixie-backports` werden auf die neuen Versionen von `trixie-backpor` aktualisiert.

Wann immer Sie ein Paket namens ”*paketname*” inklusive seiner Abhängigkeiten von `trixie-backports` manuell installieren möchten, verwenden Sie den folgenden Befehl, wobei Sie die Zielveröffentlichung mit der Option ”`-t`” festlegen:

```
$ sudo apt-get install -t trixie-backports package-name
```



Warnung
Installieren Sie nicht zu viele Pakete von backports.debian.org. Dies könnte zu Abhängigkeitsproblemen führen. In Abschnitt 2.1.11 finden Sie alternative Lösungen.

2.7.5 Externe Paketarchive



Warnung
Sie sollten sich bewusst sein, dass das externe Paket root-Privilegien für Ihr System bekommt. Sie sollten nur externe Paketarchive nutzen, denen Sie vertrauen. In Abschnitt 2.1.11finden Sie alternative Lösungen.

Sie können secure-APT mit Debian-kompatiblen externen Paketarchiven verwenden, indem Sie diese zur **sources.list** hinzufügen und den zugehörigen Archivschlüssel zum ”`/etc/apt/trusted.gpg.d/`”-Verzeichnis. Lesen Sie dazu sources.list(5) und apt-secure(8).

2.7.6 Pakete aus verschiedenen Paketarchiven ohne apt-pinning



Achtung
Die Installation von Paketen aus gemischten Paketquellen wird von der offiziellen Debian-Distribution nicht unterstützt, außer für ein paar bestimmte Archivkombinationen, wie z.B. `stable` mit Sicherheits-Updates und stable-updates.

Hier ein Beispiel von Befehlen, über die man spezielle neuere Upstream-Versionen von Paketen aus `Unstable` verwendet, während ansonsten `Testing` genutzt wird:

1. Ändern Sie die Datei ”`/etc/apt/sources.list`” vorübergehend in ”`unstable`”.
2. Führen Sie ”`aptitude update`” aus.
3. Führen Sie ”`aptitude install packetname`” aus.
4. Stellen Sie die Originalversion von ”`/etc/apt/sources.list`” für `testing` wieder her.
5. Führen Sie ”`aptitude update`” aus.

Bei diesem manuellen Vorgehen erstellen Sie keine ”`/etc/apt/preferences`”-Datei und müssen sich auch keine Sorgen über **apt-pinning** machen. Allerdings ist dies sehr mühsam.



Achtung
Wenn Sie gemischte Paketquellen verwenden, müssen Sie die Kompatibilität selbst sicherstellen, da Debian diese in solchem Falle nicht garantieren kann. Falls Paketinkompatibilitäten existieren, könnten Sie Ihr System beschädigen. Sie müssen in der Lage sein, diese technischen Anforderungen zu beurteilen. Die Verwendung von gemischten Quellen zufällig ausgewählter Archive ist eine absolut optionale Operation und nicht zu empfehlen.

Die grundsätzlichen Regeln für die Installation von Paketen aus unterschiedlichen Archiven sind wie folgt:

- Nicht-binäre Pakete (”`Architecture: all`”) sind recht **unproblematisch** zu installieren:
  - Dokumentationspakete: keine speziellen Anforderungen;
  - Pakete von Interpreter-Programmen: ein kompatibler Interpreter muss verfügbar sein.
- Bei Binärpaketen (nicht ”`Architecture: all`”) gibt es gewöhnlich viele Stolpersteine und ihre Installation könnte **problematisch** sein:
  - Kompatibilität zu Bibliotheksversionen (inklusive ”`libc`”);
  - Kompatibilität zu damit in Verbindung stehenden Hilfswerkzeugen;
  - Kernel ABI-Kompatibilität;
  - C++ ABI-Kompatibilität;
  - …

Anmerkung
Um die Installation eines Pakets **unproblematischer** zu machen, existieren möglicherweise einige nicht-freie Binärprogramm-Pakete mit vollständig statisch gelinkten Bibliotheken. Sie sollten diese immer auf ABI-Kompatibilitätsprobleme usw. kontrollieren.

Anmerkung
Die Installation von Binärpaketen aus nicht-Debian-Archiven ist grundsätzlich eine schlechte Idee, außer vielleicht um für eine kurze Zeit die Installation von beschädigten Paketen zu vermeiden. Sie sollten vorher alle anderen, verfügbaren sicheren Alternativen in Erwägung ziehen, die mit Ihrem aktuellen Debian-System kompatibel sind (lesen Sie dazu Abschnitt 2.1.11).

2.7.7 Die Installationskandidat-Version anpassen mit apt-pinning



Warnung
Die Verwendung der **apt-pinning**-Technik durch einen unerfahrenen Benutzer wird sicher große Probleme hervorrufen. Sie müssen die Verwendung dieser Technik vermeiden, außer es ist unbedingt erforderlich. In Abschnitt 2.1.11 finden sie alternative Möglichkeiten.

Ohne eine ”`/etc/apt/preferences`”-Datei wählt das APT-System basierend auf dem Versionseintrag die letzte verfügbare Version als **Installationskandidat-Version** aus. Dies ist der normale Weg und die empfohlene Verwendung des APT-Systems. Alle offiziell unterstützten Archivkombinationen erfordern keine ”`/etc/apt/preferences`”-Datei, da bei Archiven, die nicht als automatische Quelle für Aktualisierungen empfohlen werden, die Einstellung **NotAutomatic** gesetzt ist und diese entsprechend behandelt werden.

Tipp
Die Regel zum Vergleich der Versionsnummern (welche Versionsnummer größer ist) kann z.B. über ”`dpkg --compare-versions ver1.1 gt ver1.1~1; echo $?`” verifiziert werden (näheres unter dpkg(1)).

Wenn Sie regelmäßig Pakete aus gemischten Quellen installieren (siehe Abschnitt 2.7.6), können Sie diese komplizierten Operationen automatisieren, indem Sie eine Datei ”`/etc/apt/preferences`” mit entsprechenden Einträgen erstellen; so beeinflussen Sie die Auswahlregel für die **Installationskandidat-Version**; dies ist in apt_preferences(5) beschrieben. Dies Verfahren wird **apt-pinning** genannt.

Wenn Sie **apt-pinning** verwenden, müssen Sie die Kompatibilität der Pakete selbst sicherstellen, da Debian diese in einem solchen Fall nicht garantieren kann. Apt-pinning ist ein absolut optionales Verfahren und nicht zu empfehlen.

Für die apt_preferences(5)-Regeln werden die Release-Dateien im Archiv-Wurzelverzeichnis verwendet (näheres in Abschnitt 2.5.3). Daher funktioniert **apt-pinning** bei normalen Debian-Archiven und Debian-Archiven für Sicherheitsaktualisierungen nur mit dem "Suite"-Namen. (Dies ist anders als bei Ubuntu-Archiven.) Zum Beispiel können wir "`Pin: release a=unstable`" in "`/etc/apt/preferences`" verwenden, aber nicht "`Pin: release a=sid`".

Wenn Sie Debian-fremde Archive als Teil von **apt-pinning** verwenden, sollten Sie kontrollieren, wofür diese gedacht sind und ob sie glaubwürdig sind. So sind zum Beispiel Ubuntu und Debian nicht dazu gedacht, miteinander vermischt zu werden.

Anmerkung
Selbst wenn Sie keine "`/etc/apt/preferences`"-Datei erstellen, können Sie auch ohne **apt-pinning** ziemlich komplexe Systemoperationen durchführen (lesen Sie Abschnitt 2.6.3 und Abschnitt 2.7.6).

Hier eine vereinfachte Beschreibung der Technik hinter **apt-pinning**:

Das APT-System wählt für ein **Upgrade** das Paket mit der höchsten Pin-Priorität aus den verfügbaren (in "`/etc/apt/sourc` definierten) Paketquellen als **Installationskandidat-Version** aus. Wenn die Pin-Priorität des Pakets größer als 1000 ist, wird die Einschränkung für **Upgrades** nicht beachtet, um ein Downgrade (eine Zurückrüstung auf eine ältere Version) zu ermöglichen (siehe Abschnitt 2.7.11).

Werte für die Pin-Priorität eines jeden Pakets werden über Einträge in der "`/etc/apt/preferences`"-Datei festgelegt oder verwenden deren Standardwert.

Pin-Priorität	apt-pinning-Effekte auf das Paket
1001	das Paket installieren, auch wenn dies ein Downgrade des Pakets bedeutet
990	wird als Standardwert für das Archiv der **Zielveröffentlichung** verwendet
500	wird als Standardwert für das **normale** Archiv verwendet
100	wird als Standardwert für Archive mit **NotAutomatic** und **ButAutomaticUpgrades** verwendet
100	wird für das **installierte** Paket verwendet
1	wird als Standardwert für Archive mit **NotAutomatic** verwendet
-1	das Paket **niemals installieren**, selbst wenn es empfohlen wird

Tabelle 2.18: Liste erwähnenswerter Pin-Prioritäts-Werte für **apt-pinning**

Die gewünschte **Zieldistribution** (target release) kann über die Befehlszeilen-Option angegeben werden, z.B. "`apt-get install -t testing irgendein-paket`".

Die **NotAutomatic**- und **ButAutomaticUpgrades**-Archive werden durch Archiv-Server definiert, die in ihren Release-Dateien im Archiv-Wurzelverzeichnis (siehe Abschnitt 2.5.3) sowohl "`NotAutomatic: yes`" wie auch "`ButAutomaticUpg yes`" gesetzt haben. Das **NotAutomatic**-Archiv wird über Server definiert, die in ihren Release-Dateien lediglich "`NotAutomatic: yes`" gesetzt haben.

Die **apt-pinning-Situation** von *package* aus mehreren Archivquellen wird mittels "`apt-cache policy` *package*" angezeigt.

- Eine Zeile beginnend mit "`Package pin:`" listet die Paketversion von **pin** auf, wenn die Zuordnung direkt über *package* definiert ist, z.B. "`Package pin: 0.190`".
- Es existiert keine Zeile mit "`Package pin:`", wenn keine Zuordnung direkt über *package* definiert ist.
- Der Pin-Prioritäts-Wert, der direkt zu *package* gehört, ist rechts neben den Versionseinträgen aufgelistet, z.B. "`0.181 700`".
- "`0`" wird rechts neben den Versionseinträgen aufgelistet, wenn keine Zuordnung direkt mit *package* definiert ist, z.B "`0.181 0`".

- Die Pin-Prioritäts-Werte von Archiven (definiert als ”`Package: *`” in der ”`/etc/apt/preferences`”-Datei) sind links von den Archivpfaden aufgelistet, z.B. ”`100 http://deb.debian.org/debian/ trixie-backports/main Packages`”.

2.7.8 Über ”Recommends” installierte Pakete blockieren



Warnung
Die Verwendung der **apt-pinning**-Technik durch einen unerfahrenen Benutzer wird sicher große Probleme hervorrufen. Sie müssen die Verwendung dieser Technik vermeiden, außer es ist unbedingt erforderlich. In Abschnitt 2.1.11 finden sie alternative Möglichkeiten.

Wenn Sie verhindern möchten, dass bestimmte Pakete über ”Recommends”-Abhängigkeiten (Empfohlen) installiert werden, müssen Sie die Datei ”`/etc/apt/preferences`” erstellen und explizit all diese Pakete ganz oben in der Datei auflisten, wie in diesem Beispiel:

```
Package: package-1
Pin: version *
Pin-Priority: -1

Package: package-2
Pin: version *
Pin-Priority: -1
```

2.7.9 Nutzen von `Testing` mit einigen Paketen aus `Unstable`



Warnung
Die Verwendung der **apt-pinning**-Technik durch einen unerfahrenen Benutzer wird sicher große Probleme hervorrufen. Sie müssen die Verwendung dieser Technik vermeiden, außer es ist unbedingt erforderlich. In Abschnitt 2.1.11 finden sie alternative Möglichkeiten.

Hier ein Beispiel für eine **apt-pinning**-Technik, um neuere Upstream-Versionen einzelner Pakete aus `Unstable` regelmäßig aktualisieren zu können und ansonsten `Testing` zu nutzen. Sie listen alle benötigten Archive in der ”`/etc/apt/sources.list`”-Datei auf, wie hier:

```
deb http://deb.debian.org/debian/ testing main contrib non-free
deb http://deb.debian.org/debian/ unstable main contrib non-free
deb http://security.debian.org/debian-security testing-security main contrib
```

Richten Sie die ”`/etc/apt/preferences`”-Datei wie folgt ein:

```
Package: *
Pin: release a=unstable
Pin-Priority: 100
```

Wenn Sie mit dieser Konfiguration ein Paket namens ”*`paketname`*” aus dem `Unstable`-Archiv inklusive seiner Abhängigkeiten installieren möchten, führen Sie folgenden Befehl aus, wobei über die Option ”`-t`” die Zielveröffentlichung angepasst wird (die Pin-Priorität von `Unstable` wird 990):

```
$ sudo apt-get install -t unstable package-name
```

Mit dieser Konfiguration wird die normale Ausführung von "`apt-get upgrade`" und "`apt-get dist-upgrade`" (oder "`aptitude safe-upgrade`" und "`aptitude full-upgrade`") Pakete, die aus `Testing` installiert wurden, unter Verwendung des derzeitigen `Testing`-Archivs aktualisieren; Pakete, die aus `Unstable` installiert wurden, werden unter Verwendung des derzeitigen `Unstable`-Archivs aktualisiert.



Achtung
Achten Sie darauf, dass Sie nicht den "`testing`"-Eintrag aus der "`/etc/apt/sources.list`"-Datei entfernen. Ohne diesen "`testing`"-Eintrag aktualisiert das APT-System alle Pakete unter Verwendung des (neueren) `Unstable`-Archivs.

Tipp
Ich bearbeite für gewöhnlich die "`/etc/apt/sources.list`"-Datei direkt nach obigen Befehlen, um das "`unstable`"-Archiv auszukommentieren. Dies verhindert, dass der Prozess zur Aktualisierung der Paketdaten aufgrund von zu vielen Einträgen in der "`/etc/apt/sources.list`"-Datei sehr lange dauert; allerdings können dadurch Pakete, die aus `Unstable` installiert wurden, nicht unter Verwendung des derzeitigen `Unstable`-Archivs aktualisiert werden.

Tipp
Wenn "`Pin-Priority: 1`" statt "`Pin-Priority: 100`" in der "`/etc/apt/preferences`"-Datei eingetragen wird, werden bereits installierte Pakete, die den Pin-Prioritäts-Wert 100 haben, nicht mittels dem `Unstable`-Archiv aktualisiert, selbst wenn der "`testing`"-Eintrag in "`/etc/apt/sources.list`" entfernt wird.

Möchten Sie bestimmte Pakete in `Unstable` automatisch ohne vorangestelltes "`-t unstable`" auf aktuellem Stand halten, müssen Sie die "`/etc/apt/preferences`"-Datei erstellen und ganz oben in der Datei all diese Pakete wie folgt auflisten:

```
Package: package-1
Pin: release a=unstable
Pin-Priority: 700

Package: package-2
Pin: release a=unstable
Pin-Priority: 700
```

Dadurch wird die Pin-Priorität für jedes dieser Pakete spezifisch gesetzt. Um zum Beispiel immer die aktuellste `Unstable`-Version dieser "Debian Reference" in Englisch zu installieren, sollten Sie folgende Einträge in Ihrer "`/etc/apt/p` Datei haben:

```
Package: debian-reference-en
Pin: release a=unstable
Pin-Priority: 700

Package: debian-reference-common
Pin: release a=unstable
Pin-Priority: 700
```

Tipp
Diese **apt-pinning**-Technik funktioniert sogar, wenn Sie normalerweise `Stable` verwenden. Dokumentationspakete konnten nach meiner Erfahrung bis jetzt immer problemlos aus dem `Unstable`-Archiv installiert werden.

2.7.10 Nutzen von `Unstable` mit einigen Paketen aus `Experimental`



Warnung
Die Verwendung der **apt-pinning**-Technik durch einen unerfahrenen Benutzer wird sicher große Probleme hervorrufen. Sie müssen die Verwendung dieser Technik vermeiden, außer es ist unbedingt erforderlich. In Abschnitt 2.1.11 finden sie alternative Möglichkeiten.

Hier ein anderes Beispiel der **apt-pinning**-Technik, um neuere Upstream-Versionen einzelner Pakete aus `Experimental` zu integrieren und ansonsten `Testing` zu nutzen. Sie listen alle benötigten Archive in der "`/etc/apt/sources.list`"-Datei auf, wie hier:

```
deb http://deb.debian.org/debian/ unstable main contrib non-free
deb http://deb.debian.org/debian/ experimental main contrib non-free
deb http://security.debian.org/ testing-security main contrib
```

Die Standard-Pin-Priorität für das `Experimental`-Archiv ist immer 1 (<<100), da es ein Archiv mit gesetztem **NotAutomatic** ist (siehe Abschnitt 2.5.3). Es ist nicht nötig, den Wert für die Pin-Priorität explizit in der "`/etc/apt/preferences`"-Datei zu setzen, um das `Experimental`-Archiv nutzen zu können (außer wenn Sie möchten, dass bestimmte Pakete daraus beim nächsten Upgrade automatisch verwendet werden sollen).

2.7.11 Downgrade im Notfall



Warnung
Die Verwendung der **apt-pinning**-Technik durch einen unerfahrenen Benutzer wird sicher große Probleme hervorrufen. Sie müssen die Verwendung dieser Technik vermeiden, außer es ist unbedingt erforderlich. In Abschnitt 2.1.11 finden sie alternative Möglichkeiten.



Achtung
Ein Downgrade wird von Debian aufgrund seines Designs nicht offiziell unterstützt. Es sollte nur als Teil einer Notfall-Rettungsmaßnahme durchgeführt werden. Davon einmal abgesehen ist bekannt, dass es in den meisten Fällen gut funktioniert. Bei kritischen Systemen sollten Sie nach der Wiederherstellung alle wichtigen Daten auf dem System sichern und dann das System von Grund auf neu installieren.

Sie könnten Glück haben und ein Downgrade auf eine ältere Archiv-Version durch Manipulation der **Installationskandidat-Versionen** hinbekommen (lesen Sie dazu Abschnitt 2.7.7). Dies ist die tippfaule Alternative zu vielen langwierigen Befehlen der Art "`dpkg -i` *`beschädigtes-paket_alte-version`*`.deb`" (näheres dazu in Abschnitt 2.6.3).

Suchen Sie Zeilen in der "`/etc/apt/sources.list`"-Datei wie die folgende, die `unstable` referenzieren:

```
deb http://deb.debian.org/debian/ sid main contrib non-free
```

Ersetzen Sie sie durch die folgende, um stattdessen auf `testing` zu verweisen:

```
deb http://deb.debian.org/debian/ forky main contrib non-free
```

Richten Sie die "`/etc/apt/preferences`"-Datei wie folgt ein:

```
Package: *
Pin: release a=testing
Pin-Priority: 1010
```

Führen Sie "`apt-get update; apt-get dist-upgrade`" aus, um ein Downgrade aller Pakete im System zu erzwingen.

Entfernen Sie diese spezielle "`/etc/apt/preferences`"-Datei nach dem Downgrade.

Tipp
Es ist eine gute Idee, so viele Pakete wie möglich zu löschen (nicht vollständig inklusive der Konfigurationsdateien entfernen!), um die Wahrscheinlichkeit von Abhängigkeitsproblemen zu minimieren. Sie müssen bei einem solchen Downgrade unter Umständen einige Pakete von Hand entfernen oder installieren. Linux-Kernel, Bootloader, udev, PAM, APT und Netzwerk-bezogene Pakete sowie deren Konfigurationsdateien erfordern besondere Aufmerksamkeit.

2.7.12 Das equivs-Paket

Wenn Sie ein Programm aus den Quellen übersetzen möchten, um das entsprechende Debian-Paket zu ersetzen, ist es das beste, ein echtes lokal gebautes Paket nach Debian-Art zu erstellen (`*.deb`) und dafür ein privates Archiv zu nutzen.

Haben Sie sich jedoch entschieden, ein Programm aus den Quellen zu kompilieren und es stattdessen in "`/usr/local`" zu installieren, können Sie möglicherweise `equivs` als letzte Rettung einsetzen, um die fehlenden Paketabhängigkeiten zu erfüllen:

```
Package: equivs
Priority: optional
Section: admin
Description: Circumventing Debian package dependencies
 This package provides a tool to create trivial Debian packages.
 Typically these packages contain only dependency information, but they
 can also include normal installed files like other packages do.
 .
 One use for this is to create a metapackage: a package whose sole
 purpose is to declare dependencies and conflicts on other packages so
 that these will be automatically installed, upgraded, or removed.
 .
 Another use is to circumvent dependency checking: by letting dpkg
 think a particular package name and version is installed when it
 isn't, you can work around bugs in other packages' dependencies.
 (Please do still file such bugs, though.)
```

2.7.13 Ein Paket auf das Stable-System portieren



Achtung
Es gibt keine Garantie, dass die hier beschriebene Prozedur ohne händischen Extraaufwand funktioniert, wenn das System sich unterscheidet.

Für partielle Upgrades eines `Stable`-Systems ist das Neubauen von Paketen innerhalb ihrer Umgebung unter Verwendung der Quellpakete sinnvoll. Sie vermeiden so riesige Upgrades aufgrund von Paketabhängigkeiten.

Fügen Sie folgende Einträge zur "`/etc/apt/sources.list`" des `Stable`-Systems hinzu:

```
deb-src http://deb.debian.org/debian unstable  main contrib non-free
```

Installieren Sie für das Kompilieren erforderliche Pakete und laden Sie das Quellpaket wie folgt herunter:

```
# apt-get update
# apt-get dist-upgrade
# apt-get install fakeroot devscripts build-essential
# apt-get build-dep foo
$ apt-get source foo
$ cd foo*
```

Aktualisieren Sie einige Pakete aus der Werkzeugkette wie z.B. `dpkg` und `debhelper` aus Paketquellen für zurückportierte Pakete, falls diese für die Rückportierung erforderlich sind.

Führen Sie folgendes aus:

```
$ dch -i
```

Erhöhen Sie die Paketversion, z.B. indem Sie ein "`+bp1`" in "`debian/changelog`" angehängen.

Bauen Sie die Pakete und installieren Sie sie wie folgt im System:

```
$ debuild
$ cd ..
# debi foo*.changes
```

2.7.14 Proxy-Server für APT

Obwohl das Spiegeln von vollständigen Sektionen des Debian-Archivs Plattenplatz und Netzwerkbandbreite verschwendet, ist es der Einsatz eines lokalen Proxy-Servers für APT wert in Erwägung gezogen zu werden, wenn Sie viele Systeme im LAN (Netzwerk) administrieren. APT kann konfiguriert werden, einen generischen Web-(http-)Proxy-Server wie `squid` (siehe auch Abschnitt 6.5) zu verwenden, wie in apt.conf(5) und "`/usr/share/doc/apt/examples/co` beschrieben. Sie können die Umgebungsvariable "`$http_proxy`" nutzen, um die Proxy-Server-Einstellungen aus der "`/etc/apt/apt.conf`"-Datei zu überschreiben.

Es gibt Proxy-Hilfsprogramme, die für das Debian-Archiv spezialisiert sind. Sie sollten die Fehlerdatenbank (BTS) überprüfen, bevor Sie sie verwenden.

Paket	Popcon	Größe	Beschreibung
`apt-cacher`	V:0.27, I:0.32	265	zwischenspeichernder Proxy-Server für Debian-Paket- und Quell-Dateien (Perl-Programm)
`apt-cacher-ng`	V:3.8, I:3.9	2063	zwischenspeichernder Proxy für die Verteilung von Software-Paketen (kompiliertes C++-Programm)

Tabelle 2.19: Liste von Proxy-Hilfsprogrammen speziell für das Debian-Archiv



Achtung
Sollte Debian einmal seine Archivstruktur umorganisieren, benötigen diese spezialisierten Proxy-Hilfsprogramme eventuell eine Code-Änderung durch den Paketbetreuer und könnten eine Zeit lang nicht funktionsfähig sein. Generische Web-(http-)Proxy-Server sind andererseits robuster und können einfacher mit solchen Änderungen umgehen.

2.7.15 Weitere Lektüre zum Paketmanagement

Folgende Dokumentation bietet sich an, um mehr über das Paketmanagement zu lernen.

- Primäre Dokumentation zum Paketmanagement:

  - aptitude(8), dpkg(1), tasksel(8), apt(8), apt-get(8), apt-config(8), apt-secure(8), sources.list(5), apt.conf(5) und apt_preferences(5);
  - "`/usr/share/doc/apt-doc/guide.html/index.html`" und "`/usr/share/doc/apt-doc/offline.html/in` aus dem `apt-doc`-Paket;
  - "`/usr/share/doc/aptitude/html/en/index.html`" aus dem `aptitude-doc-en`-Paket.
- Offizielle und detaillierte Dokumentation zum Debian-Archiv:
  - "Debian Policy Manual Chapter 2 - The Debian Archive"
  - "Debian Entwickler-Referenz, Kapitel 4 - Ressourcen für Debian-Entwickler, 4.6 Das Debian-Archiv"
  - "Die Debian GNU/Linux-FAQ, Kapitel 6 - Die Debian FTP-Archive"
- Anleitung zum Bau eines Debian-Pakets für Debian-Benutzer:
  - "Debian-Leitfaden für Paketbetreuer"

Kapitel 3

Die Systeminitialisierung

Als Systemadministrator sollten Sie grob wissen, wie das Debian-System gestartet und konfiguriert wird. Obwohl die genauen Details in den Quelldateien der installierten Pakete und deren Dokumentation zu finden sind, ist dies für die meisten von uns ein bisschen viel.

Hier ein knapper Überblick über die wichtigsten Dinge bei der Initialisierung eines Debian-Systems. Da dies ein bewegliches Ziel ist, sollten Sie die neueste Dokumentation lesen:

- Das Debian Linux Kernel Handbook ist die primäre Informationsquelle zum Debian-Kernel.
- bootup(7) beschreibt den System-Boot-Prozess basierend auf `systemd` (derzeitiges Debian-System).
- boot(7) beschreibt den System-Boot-Prozess basierend auf UNIX System V Release 4 (älteres Debian-System).

3.1 Ein Überblick über den Bootstrap-Prozess

Das Computer-System durchläuft verschiedene Phasen des Bootstrap-Prozesses vom Einschalten bis zur Bereitstellung des funktionalen Betriebssystems an den Benutzer.

Der Einfachheit halber beschränke ich meine Betrachtung auf die weit verbreitete PC-Plattform mit einer Standardinstallation.

Der typische Bootstrap-Prozess ist wie eine 4-stufige Rakete. Jede Stufe übergibt die Systemkontrolle an die jeweils nachfolgende Stufe:

- Abschnitt 3.1.1
- Abschnitt 3.1.2
- Abschnitt 3.1.3
- Abschnitt 3.1.4

Natürlich können diese unterschiedlich konfiguriert werden. Wenn Sie zum Beispiel Ihren eigenen Kernel kompilieren, werden Sie unter Umständen den Schritt mit dem Mini-Debian-System überspringen. Gehen Sie daher nicht davon aus, dass dies alles in Ihrem Fall zutrifft, solange Sie es nicht selbst überprüft haben.

3.1.1 Stufe 1: das UEFI

Das Unified Extensible Firmware Interface (UEFI) definiert als Teil der UEFI-Spezifikation einen Boot-Manager. Wenn ein Rechner eingeschaltet wird, ist dieser Boot-Manager die erste Stufe im Boot-Prozess; er prüft die Boot-Konfiguration und führt den dort festgelegten Betriebssystem-Bootloader bzw. Betriebssystem-Kern aus (normalerweise einen Bootloader). Die Boot-Konfiguration ist über Variablen definiert, die im NVRAM gespeichert sind; dazu gehören Variablen, die den Dateisystempfad zum Betriebssystem-Bootloader oder -Kern enthalten.

Eine EFI System Partition (ESP) ist eine spezielle Partition auf einem Datenspeicher auf UEFI-konformen Computersystemen. Nach dem Einschalten des Rechners greift die UEFI-Firmware auf die ESP zu. Auf ihr sind UEFI-Applikationen abgelegt sowie weitere Dateien, die von diesen Applikationen benötigt werden. Zu diesen Applikationen gehören auch die Betriebssystem-Bootloader. (Auf älteren PC-Systemen können auch das BIOS und der MBR zum Einsatz kommen.)

3.1.2 Stufe 2: der Bootloader

Der Bootloader ist die zweite Stufe des Boot-Prozesses und wird durch das UEFI gestartet. Er lädt das System-Kernel-Image und das initrd-Image in den Speicher und übergibt diesen die Kontrolle. Das initrd-Image ist ein Abbild des Wurzeldateisystems und seine Funktionalitäten hängen von dem verwendeten Bootloader ab.

Das Debian-System nutzt normalerweise den Linux-Kernel als Standard-Betriebssystem-Kern. Das initrd-Image für den aktuellen Linux-Kernel der Version 5.x ist technisch gesehen ein initramfs- (initial RAM Filesystem-) Image.

Es gibt mehrere Bootloader und Konfigurationsoptionen:

Paket	Popcon	Größe	initrd	Bootloader	Beschreibung
grub-efi-amd64	I:472	142	Unterstützt	GRUB UEFI	Intelligenter Bootloader, der Festplattenpartitionen und Dateisysteme wie vfat, ext4 … unterstützt (UEFI).
grub-pc	V:17, I:504	479	Unterstützt	GRUB 2	Intelligenter Bootloader, der Festplattenpartitionen und Dateisysteme wie vfat, ext4 … unterstützt (BIOS).
grub-rescue-pc	V:0.04, I:0.55	7269	Unterstützt	GRUB 2	Dies ist das boot-fähige Rettungs-Image von GRUB 2 (CD und Diskette) (PC/BIOS-Version).
grml-rescueboot	V:0.1, I:1.1	39	Nicht verfügbar	GRUB 2 Plug-in	grml-rescueboot fügt Live-Linux ISO-Images zum grub2-Bootmenü hinzu
syslinux	V:3, I:30	325	Unterstützt	Isolinux	Unterstützt das ISO9660-Dateisystem. Dies wird von Boot-CDs verwendet.
syslinux	V:3, I:30	325	Unterstützt	Syslinux	Unterstützt das MSDOS-Dateisystem FAT. Dies wird von Boot-Disketten verwendet.
loadlin	V:0.02, I:0.36	87	Unterstützt	Loadlin	Das gewünschte Betriebssystem wird aus dem laufenden FreeDOS-/MSDOS-System heraus gestartet.
mbr	V:0.2, I:2.6	47	Nicht unterstützt	MBR von Neil Turton	Dies ist freie Software, die den MSDOS-MBR ersetzt. Unterstützt nur Festplattenpartitionen.

Tabelle 3.1: Liste der Bootloader

Auf UEFI-Systemen liest GRUB2 zuerst die ESP-Partition und verwendet die für `search.fs_uuid` in "`/boot/efi/EFI/de` angegebene UUID, um die Partition von GRUB2's Menükonfigurations-Datei "`/boot/grub/grub.cfg`" zu bestimmen.

Der wichtigste Teil von GRUB2's Menükonfigurations-Datei sieht wie folgt aus:

```
menuentry 'Debian GNU/Linux' ... {
        load_video
        insmod gzio
        insmod part_gpt
        insmod ext2
        search --no-floppy --fs-uuid --set=root fe3e1db5-6454-46d6-a14c-071208ebe4b1
        echo    'Loading Linux 5.10.0-6-amd64 ...'
        linux   /boot/vmlinuz-5.10.0-6-amd64 root=UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1 ↩
            ro quiet
        echo    'Loading initial ramdisk ...'
        initrd  /boot/initrd.img-5.10.0-6-amd64
}
```

Dieser Teil von `/boot/grub/grub.cfg` hat folgende Bedeutung:

Einstellung	**Wert**
Geladene GRUB2-Module	`gzio`, `part_gpt`, `ext2`
verwendete Partition für das root-Dateisystem	Partition, die über `UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1` identifiziert wird
Pfad zum Kernel-Image im root-Dateisystem	`/boot/vmlinuz-5.10.0-6-amd64`
verwendete Kernel-Boot-Parameter	"`root=UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1 ro quiet`"
Pfad zum initrd-Image im root-Dateisystem	`/boot/initrd.img-5.10.0-6-amd64`

Tabelle 3.2: Bedeutung der Bestandteile eines Menüeintrags in `/boot/grub/grub.cfg`

Auf einem Debian-System wird "`/boot/grub/grub.cfg`" von dem installierten GRUB-Paket (z.B. `grub-efi-amd64`) verwaltet und die direkte Modifizierung durch den Benutzer ist veraltet. Stattdessen sollten Sie die Konfigurationsdateien in "`/etc/grub.d/`" und "`/etc/default/grub`" anpassen. Danach konfigurieren Sie GRUBs Konfig-Dateien und aktualisieren die NVRAM-Variablen, um automatisch das Debian-System zu booten, indem Sie folgendes ausführen:

```
# dpkg-reconfigure grub-efi-amd64
```

Tipp
Sie können die Anzeige von Kernel-Boot-Meldungen aktivieren, indem Sie das `quiet` aus der "`GRUB_CMDLINE_LINUX_DEFAULT`"-Zeile in "`/etc/default/grub`" entfernen.

Tipp
Ein GRUB-Splash-Hintergrundbild fügen Sie hinzu, indem Sie die Grafikdatei in "`/boot/grub/`" ablegen.

Lesen Sie auch "`info grub`", grub-install(8) und grub-mkconfig(8).

3.1.3 Stufe 3: das Mini-Debian-System

Das Mini-Debian-System ist die dritte Stufe des Boot-Prozesses und wird durch den Bootloader gestartet. Es lässt den System-Kernel mit seinem eigenen Wurzeldateisystem im Speicher laufen. Dies ist ein optionaler, vorbereitender Schritt des Boot-Prozesses.

Anmerkung
Der Begriff "Mini-Debian-System" wurde von dem Autor erfunden, um diese dritte Stufe des Boot-Prozesses in diesem Dokument zu beschreiben. Dieses System wird normalerweise initrd- oder initramfs-System genannt. Ein ähnliches System wird im Speicher auch durch den Debian Installer verwendet.

"`/init`" wird als erstes Programm aus diesem Wurzeldateisystem im Speicher ausgeführt. Es ist ein Programm, das den Kernel im Userspace initialisiert und die Kontrolle an die nächste Stufe übergibt. Dieses Mini-Debian-System bietet Flexibilität für den Boot-Prozess, um zum Beispiel Kernel-Module vor dem Hauptteil des Boot-Prozesses hinzuzufügen oder um das Wurzeldateisystem als verschlüsseltes Dateisystem einzubinden.

- "`/init`" ist ein Shell-Skript, wenn das initfamfs durch `initramfs-tools` erstellt wurde.
  - Sie können diesen Teil des Boot-Prozesses unterbrechen, um eine root-Shell zu bekommen, indem Sie "`break=init`" usw. zu den Kernel-Boot-Parametern hinzufügen. Informationen zu weiteren Unterbrechungsmöglichkeiten finden Sie im "`/init`"-Skript. Diese Shell-Umgebung ist ausgeklügelt genug, um eine gute Überprüfung der Hardware Ihrer Maschine zu ermöglichen.
  - Die verfügbaren Befehle in diesem Mini-Debian-System gehen auf ein GNU-Werkzeug namens busybox(1) zurück und werden auch hauptsächlich von diesem bereitgestellt.
- "`/init`" ist ein binäres `systemd`-Programm, wenn das initramfs durch `dracut` erstellt wurde.
  - Befehle in diesem Mini-Debian-System sind in ihrer Funktionalität auf die systemd(1)-Umgebung reduziert.



Achtung
Sie müssen die Option "`-n`" für den `mount`-Befehl verwenden, wenn Sie sich im Nur-Lese-Wurzeldateisystem befinden.

3.1.4 Stufe 4: das normale Debian-System

Das normale Debian-System ist die vierte Stufe des Boot-Prozesses und wird von dem Mini-Debian-System gestartet. Der System-Kernel des Mini-Debian-Systems läuft in dieser Umgebung weiter. Das verwendete Wurzeldateisystem wird von dem im Arbeitsspeicher umgeschwenkt zu dem auf dem physischen Speichergerät.

Das Programm init wird als erstes Programm mit PID=1 ausgeführt und erledigt die eigentliche Hauptarbeit beim Booten, das Starten verschiedener Programme. Der Standardpfad zum init-Programm ist "`/usr/sbin/init`", aber er kann über einen Kernel-Boot-Parameter wie "`init=/pfad/zum/init-programm`" auch geändert werden.

"`/usr/sbin/init`" wurde nach Debian 8 Jessie (veröffentlicht in 2015) ein symbolischer Link auf "`/lib/systemd/system`

Tipp
Mittels "`ps --pid 1 -f`" können Sie überprüfen, welcher init-Befehl letztlich auf Ihrem System verwendet wird.

Tipp
Im Debian Wiki unter BootProcessSpeedup finden Sie aktuelle Tipps zur Beschleunigung des Boot-Prozesses.

Paket	Popcon	Größe	Beschreibung
systemd	V:925, I:981	11401	Ereignis-basierter init(8)-Daemon für gleichzeitige Ausführung (Alternative zu sysvinit)
cloud-init	V:4.6, I:6.4	3267	Initialisierungssystem für Infrastruktur-Cloud-Instanzen
systemd-sysv	V:921, I:984	77	die Handbuchseiten und Links, die nötig sind, um sysvinit durch systemd zu ersetzen
init-system-helpers	V:754, I:989	132	Hilfsprogramme, um zwischen sysvinit und systemd umschalten zu können
initscripts	V:14, I:49	197	Skripte zur Initialisierung und zum Herunterfahren des Systems
sysvinit-core	V:3.1, I:3.5	368	System-V-ähnliche init(8)-Werkzeuge
sysv-rc	V:27, I:52	85	System-V-ähnlicher Mechanismus zum Wechsel des Runlevels
sysvinit-utils	V:734, I:1000	100	System-V-ähnliche Werkzeuge (startpar(1), bootlogd(8), …)
lsb-base	V:193, I:281	12	Zur Linux Standard Base 3.2 konforme init-Skript-Funktionalität
insserv	V:32, I:52	132	Werkzeug, um die Boot-Reihenfolge unter Verwendung von LSB-konformen init.d-Skript-Abhängigkeiten zu organisieren
kexec-tools	V:1.6, I:5.3	320	Werkzeug für kexec(8)-Neustarts (Warmstarts)
systemd-bootchart	V:0.17, I:0.49	131	Performance-Analyseprogramm für den Boot-Prozess
mingetty	V:0.1, I:2.3	36	getty(8) nur für die Konsole
mgetty	V:0.13, I:0.44	318	Intelligenter getty(8)-Ersatz für Modems

Tabelle 3.3: Liste von Boot-Hilfsprogrammen für das Debian-System

3.2 Das Rettungssystem



Warnung
Führen Sie keine Systemadministrations-Arbeiten am Bootstrap-Prozess durch, wenn Sie kein Rettungssystem verfügbar haben.

Ein Rettungssystem in Reichweite zu haben, erlaubt es uns, folgende Dinge zu erledigen:

- Ein System von einer beschädigten Bootloader-Konfiguration zu booten;
- Eine beschädigte Bootloader-Konfiguration zu korrigieren;
- Extrahieren von Daten von einem nicht boot-fähigen System;
- Anpassen von Dateisystemen, Festplattenpartitionen und LVM-Volumes, inklusive dem root-Dateisystem.

Typischerweise wird ein Rettungssystem als ISO-Image-Datei bereitgestellt und auf einen Wechseldatenträger geschrieben, z.B.:

- auf einen USB-Flash-Speicher, wie in Abschnitt 9.7.2;
- auf eine CD / DVD, wie unter Abschnitt 9.7.7.

Der Einfachheit halber werden in den Beispielen hier Fälle für USB-Speicher aufgeführt, CDs oder DVDs könnten aber auch genutzt werden.

Tipp
Das Booten einiger Bootloader von einem Wechseldatenträger erfordert eventuell die Anpassung bestimmter UEFI-NVRAM-Variablen.

3.2.1 GRUB UEFI-Rettungssystem auf USB

Ein GRUB UEFI-Rettungssystem mit Bootmenü kann gestartet werden, indem der Rechner eingeschaltet wird, während ein USB-Flash-Speicher mit dem ”GRUB UEFI-Rettungssystem auf USB” eingesteckt ist.

Solch ein ”GRUB UEFI-Rettungssystem auf USB” können Sie erstellen, indem Sie das Super Grub2 Disk ISO-Image auf einen USB-Flash-Speicher schreiben.

In dem Fall, dass die Bootloader-Konfiguration z.B. durch die Installation eines anderen Betriebssystems beschädigt wurde, können Sie dies beheben durch:

- Starten Sie das ”GRUB UEFI-Rettungssystem”, um automatisch boot-fähige, installierte Systeme zu detektieren.
- Starten Sie aus dem GRUB-Menü Ihr Debian-System.
- In der Linux-root-Shell geben Sie folgendes ein:

```
# dpkg-reconfigure grub-efi-amd64
```

Anmerkung
Sie können gemäß ”`info grub-mkrescue`” auch ein boot-fähiges GRUB-Rettungs-Image erstellen. Es stellt eine textbasierte GRUB-Shell bereit, aber es kann nicht automatisch boot-fähige, installierte Betriebssysteme detektieren.

3.2.2 Linux Live-Rettungssystem auf USB

Ein Linux Live-Rettungssystem kann gestartet werden, indem der Rechner eingeschaltet wird, während ein USB-Flash-Speicher mit dem ”Linux Live-Rettungssystem auf USB” eingesteckt ist.

Solch ein ”Linux Live-Rettungssystem auf USB” wird erzeugt, indem ein Linux-Live ISO-Image auf einen USB-Flash-Speicher geschrieben wird. Hier einige Beispiele solcher Linux-Live-Images:

- Debian Live-Images
- Kali Linux Live
- Grml Live Linux

Einige Anwendungsfälle für solch ein Linux Live-Rettungssystem:

- Die defekte Bootloader-Konfiguration korrigieren, wenn diese durch die Installation eines anderen Betriebssystems beschädigt wurde:
  - Starten Sie das Linux Live-Rettungssytem;
  - Binden Sie die Partition von Ihrem nicht mehr startfähigen Debian-System, die das root-Dateisystem enthält, als ”`/mnt`” ein.
  - In der Linux-root-Shell geben Sie folgendes ein:

```
# chroot /mnt; dpkg-reconfigure grub-efi-amd64
```

- Ein beschädigtes `dpkg`-Paket instandsetzen (ersetzen durch eine ältere Version):
  - Starten Sie das Linux Live-Rettungssytem;
  - Binden Sie die Partition Ihres Debian-Systems, die das root-Dateisystem enthält, als ”`/mnt`” ein.
  - In der Linux-root-Shell geben Sie folgendes ein:

```
# dpkg --root /mnt -i /mnt/var/cache/apt/archives/dpkg_old_version_amd64.deb
```

- Durchführen von Änderungen, die normalerweise nicht erlaubt sind, wie Dateisystemoperationen am installierten System (siehe dazu Abschnitt 9.6):



Warnung
Die grafische Oberfläche Ihres Linux Live-Systems könnte gesperrt werden, wenn das System zu lange inaktiv war.

Tipp

- Es ist eine gute Idee, ein Passwort zu vergeben, sobald Sie das Linux Live-System gestartet haben.
- Sie können z.B. ein Passwort vergeben über "`sudo passwd` *`user`*" vom Benutzer-Shell-Prompt auf einer der virtuellen Konsolen (siehe dazu Abschnitt 1.1.6).
- Einige Linux Live-Systeme setzen ein Passwort für den Standardbenutzer ("*Benutzer*/*Passwort*"): "Debian Live" = "user/live", "Kali Linux Live" = "kali/kali".

3.2.3 Linux Live-Rettungssystem von GRUB

Dieses Linux Live-Rettungssystem kann über einen GRUB-Menüeintrag gestartet werden. Dazu müssen Sie die GRUB-Konfiguration wie folgt vorbereiten:

- Installieren Sie das Paket `grml-rescueboot`.
- Finden Sie ein Linux-Live ISO-Image, das "`/boot/grub/loopback.cfg`" enthält.
  - ISO-Images aus Abschnitt 3.2.2 enthalten solche eine "`/boot/grub/loopback.cfg`".
- Kopieren Sie ein oder mehrere dieser Linux-Live ISO-Images in das "`/boot/grml/`"-Verzeichnis.
- Aktualisieren Sie das GRUB-Menü durch:

```
# dpkg-reconfigure grub-efi-amd64
```

Sobald Sie jetzt den Rechner einschalten, zeigt das GRUB-Menü mögliche Kandidaten für Linux Live-Rettungssysteme an.

3.3 Systemd

3.3.1 Systemd-Init

Wenn das Debian-System startet, wird `/usr/sbin/init` (ein symbolischer Link auf `/usr/lib/systemd`) gestartet als Init-System-Prozess mit der `PID=1`, Eigentümer ist root (`UID=0`). Lesen Sie dazu systemd(1).

Der `systemd`-Init-Prozess wird - basierend auf den Unit-Konfigurationsdateien (siehe systemd.unit(5)) - in mehrere parallele Prozesse aufgespalten; diese Konfigurationsdateien sind in deklarativem Stil geschrieben, im Unterschied zu dem prozeduralen Stil von SysV.

Die abgespalteten Prozesse werden in individuellen Linux control groups abgelegt, die nach der Unit benannt werden, zu der sie in der privaten systemd-Hierarchie gehören (siehe cgroups und Abschnitt 4.7.5).

Units für den System-Modus werden aus dem in systemd.unit(5) beschriebenen "System Unit Search Path" geladen. Die wichtigsten sind hier in der Reihenfolge ihrer Priorität aufgelistet:

- "`/etc/systemd/system/*`": System-Units, erstellt vom Administrator
- "`/run/systemd/system/*`": Runtime-Units
- "`/lib/systemd/system/*`": System-Units, erstellt vom Distributions-Paketmanager

Deren Abhängigkeiten untereinander sind durch die Regeln "`Wants=`", "`Requires=`", "`Before=`", "`After=`", ... (siehe "MAPPING OF UNIT PROPERTIES TO THEIR INVERSES" in systemd.unit(5)) definiert. Die Ressourcen-Steuerung ist ebenfalls festgelegt (siehe systemd.resource-control(5)).

Die Endung der Unit-Konfigurationsdateien definiert ihren Typ wie folgt:

- ***.service** beschreibt einen Prozess (Dienst), der von `systemd` gesteuert und überwacht wird. Siehe systemd.service(5).
- ***.device** beschreibt ein Gerät, das im sysfs(5) als udev(7)-Gerätedatei abgebildet ist. Siehe systemd.device(5).
- ***.mount** beschreibt einen Einbindungspunkt im System, der von `systemd` gesteuert und überwacht wird. Siehe systemd.mount(5).
- ***.automount** beschreibt einen automatischen Einbindungspunkt im System, der von `systemd` gesteuert und überwacht wird. Siehe systemd.automount(5).
- ***.swap** beschreibt ein Swap-Device oder eine Swap-Datei (zum Auslagern von Arbeitsspeicher auf eine Festplatte), die von `systemd` gesteuert und überwacht wird. Siehe systemd.swap(5).
- ***.path** beschreibt einen Pfad im Dateisystem, der von `systemd` zum Zwecke der pfad-basierten Aktivierung überwacht wird. Siehe systemd.path(5).
- ***.socket** beschreibt einen Socket, der von `systemd` zum Zwecke der socket-basierten Aktivierung gesteuert und überwacht wird. Siehe systemd.socket(5).
- ***.timer** beschreibt einen Timer (Zeitgeber), der von `systemd` zum Zwecker der timer-basierten Aktivierung gesteuert und überwacht wird. Siehe systemd.timer(5).
- ***.slice** verwaltet Ressourcen über cgroups(7). Siehe systemd.slice(5).
- ***.scope** wird programmgesteuert über die Busschnittstellen von `systemd` erzeugt, um Systemprozesse zu verwalten. Siehe systemd.scope(5).
- ***.target** fasst andere Unit-Konfigurationsdateien zu Gruppen zusammen, um Synchronisierungspunkte für den Startprozess zu erstellen. Siehe systemd.target(5).

Während des Systemstarts versucht der `systemd`-Prozess, das Target "`/lib/systemd/system/default.target`" (normalerweise ein symbolischer Link auf "`graphical.target`") zu starten. Als erstes werden dabei einige spezielle Target-Units (siehe systemd.special(7)) wie "`local-fs.target`", "`swap.target`" und "`cryptsetup.target`" aktiviert, um die Dateisysteme einzubinden. Dann werden über die Abhängigkeiten weitere Target-Units aktiviert. Details finden Sie in bootup(7).

`systemd` enthält Funktionalitäten, um die Rückwärtskompatibilität zu SysV zu gewährleisten. Boot-Skripte im SysV-Stil in "`/etc/init.d/rc[0123456S].d/[KS]`*name*" werden immer noch abgearbeitet, und telinit(8)-Befehle werden in Aktivierungsanforderungen für systemd-Units übersetzt.



Achtung
Die emulierten Runlevel 2 bis 4 verweisen über symbolische Links alle auf des gleiche "`multi-user.target`".

3.3.2 Systemd-Login

Wenn sich ein Benutzer über gdm3(8), sshd(8) usw. am Debian-System anmeldet, wird `/lib/systemd/system --user` als User-Service-Manager-Prozess gestartet, Eigentümer ist der entsprechende Benutzer. Lesen Sie dazu systemd(1).

Der `systemd` User-Service-Manager-Prozess wird basierend auf den deklarativen Unit-Konfigurationsdateien (siehe systemd.unit(5) und `user@.service`(5)) in mehrere parallele Prozesse aufgespalten.

Units für den Benutzer-Modus werden aus dem in systemd.unit(5) beschriebenen "User Unit Search Path" geladen. Die wichtigsten sind hier in der Reihenfolge ihrer Priorität aufgelistet:

- "`~/.config/systemd/user/*`": Benutzer-Konfigurations-Units
- "`/etc/systemd/user/*`": Benutzer-Units, erstellt vom Administrator
- "`/run/systemd/user/*`": Runtime-Units
- "`/lib/systemd/user/*`": Benutzer-Units, installiert vom Distributions-Paketmanager

Diese werden auf die gleiche Art verwaltet wie Abschnitt 3.3.1.

3.4 Die Kernel-Meldungen

Die angezeigten Fehlermeldungen des Kernels auf der Konsole können über einen Schwellwert gefiltert werden:

```
# dmesg -n3
```

Wert	Name	Bedeutung
0	KERN_EMERG	System ist unbenutzbar
1	KERN_ALERT	es ist unverzüglich eine Aktion erforderlich
2	KERN_CRIT	Zustände mit kritischen Fehlern
3	KERN_ERR	Zustände mit Fehlern
4	KERN_WARNING	Zustände mit Warnungen
5	KERN_NOTICE	normale, aber erwähnenswerte Zustände
6	KERN_INFO	rein informativ
7	KERN_DEBUG	Nachrichten zur Fehlersuche/-eingrenzung

Tabelle 3.4: Liste der Schwellwerte zur Filterung von Kernel-Fehler-Meldungen

3.5 Die Systemmeldungen

Unter `systemd` werden sowohl Kernel- wie auch Systemmeldungen durch den Journal-Dienst `systemd-journald.servi` (a.k.a `journald`) protokolliert, entweder in Form von Binärdaten unterhalb von "`/var/log/journal`" oder in flüchtigen Binärdaten in "`/run/log/journal/`". Diese binären Protokolldaten können mit dem Befehl journalctl(1) abgefragt werden. Zum Beispiel erhalten Sie das Protokoll vom letzten Boot-Vorgang mit:

```
$ journalctl -b
```

Unter `systemd` könnte das System-Log-Werkzeug rsyslogd(8) deinstalliert sein. Falls es installiert ist, ändert sich sein Verhalten, so dass die volatilen binären Logdaten gelesen werden (statt "`/dev/log`", was vor systemd der Standard war) und traditionelle permanente ASCII-Logdaten erstellt werden. Dies kann über "`/etc/default/rsyslog`" und "`/etc/rsyslog.conf`" sowohl für die Protokolldateien wie auch für die Bildschirmanzeige angepasst werden. Lesen Sie dazu rsyslogd(8) und rsyslog.conf(5) sowie auch Abschnitt 9.3.2.

Tätigkeit	Befehl
Protokoll für Systemdienste und Kernel vom letzten Boot-Vorgang anzeigen	"`journalctl -b --system`"
Protokoll für Dienste des aktuellen Benutzers vom letzten Boot-Vorgang anzeigen	"`journalctl -b --user`"
Protokollinformationen von "`$unit`" vom letzten Boot-Vorgang anzeigen	"`journalctl -b -u $unit`"
Protokollinformationen von "`$unit`" vom letzten Boot-Vorgang anzeigen (im Stil von "`tail -f`")	"`journalctl -b -u $unit -f`"

Tabelle 3.5: Liste typischer `journalctl`-Befehle

3.6 Systemmanagement

`systemd` enthält nicht nur das eigentliche init-System zum Starten des Systems, sondern auch Funktionalitäten zum Systemmanagement mittels dem systemctl(1)-Befehl.

In obigen Beispielen kann "`$unit`" für einen einzelnen Unit-Namen stehen (ein Anhang wie `.service` oder `.target` ist dabei optional), oder auch für die Angabe mehrerer Units (über Suchmuster im Shell-Stil wie "`*`", "`?`" oder "`[]`", die fnmatch(3) verwenden; diese werden auf die primären Namen aller Units angewandt, die derzeit in den Arbeitsspeicher geladen sind).

Befehlen zum Ändern des Systemstatus wird typischerweise ein "`sudo`" vorangestellt, um die nötigen administrativen Rechte anzufordern.

Die Ausgabe von "`systemctl status $unit|$PID|$device`" nutzt farbige Punkte ("●"), um den Unit-Status kompakt zusammenzufassen.

- Ein weisser "●" steht für "inaktiv" oder "deaktiviert".
- Ein roter "●" steht für "fehlgeschlagen" oder "Fehler".
- Ein grüner "●" steht für "aktiv", "wird neu geladen" oder "wird aktiviert".

3.7 Weitere Systemüberwachungs-Werkzeuge

Hier eine Liste von weiteren Befehlsschnipseln zur Systemüberwachung unter `systemd`. Bitte lesen Sie die zugehörigen Handbuchseiten inklusive cgroups(7).

3.8 Systemkonfiguration

3.8.1 Der Rechnername

Der Kernel verwaltet den Rechnernamen (**hostname**) des Systems. Die durch `systemd-hostnamed.service` gestartete System-Unit setzt beim Systemstart den Rechnernamen auf den in "`/etc/hostname`" festgelegten Wert. Diese Datei sollte **nur** den Rechnernamen des Systems enthalten, nicht einen vollqualifizierten Domänennamen.

Um den derzeitigen Rechnernamen auszugeben, führen Sie hostname(1) ohne ein Argument aus.

Tätigkeit	Befehl
Alle verfügbaren Unit-Typen auflisten	"`systemctl list-units --type=help`"
Alle Target-Units im Arbeitsspeicher auflisten	"`systemctl list-units --type=target`"
Alle Service-Units im Arbeitsspeicher auflisten	"`systemctl list-units --type=service`"
Alle Device-Units im Arbeitsspeicher auflisten	"`systemctl list-units --type=device`"
Alle Mount-Units im Arbeitsspeicher auflisten	"`systemctl list-units --type=mount`"
Alle Socket-Units im Arbeitsspeicher auflisten	"`systemctl list-sockets`"
Alle Timer-Units im Arbeitsspeicher auflisten	"`systemctl list-timers`"
"`$unit`" starten	"`systemctl start $unit`"
"`$unit`" stoppen	"`systemctl stop $unit`"
Dienst-spezifische Konfiguration neu laden	"`systemctl reload $unit`"
"`$unit`" stoppen und neu starten	"`systemctl restart $unit`"
"`$unit`" starten und alle anderen stoppen	"`systemctl isolate $unit`"
Zur grafischen Oberfläche wechseln (GUI-System)	"`systemctl isolate graphical`"
Zur Konsolenoberfläche wechseln (Mehrbenutzer-CLI-System)	"`systemctl isolate multi-user`"
Zur Rettungssystem-Oberfläche wechseln (Einzelbenutzer-CLI-System)	"`systemctl isolate rescue`"
Kill-Signal an "`$unit`" senden	"`systemctl kill $unit`"
Prüfen, ob "`$unit`" aktiv ist	"`systemctl is-active $unit`"
Prüfen, ob "`$unit`" fehlgeschlagen ist	"`systemctl is-failed $unit`"
Status von "`$unit\|$PID|device`" prüfen	"`systemctl status $unit\|$PID|$device`"
Eigenschaften von "`$unit\|$job`" anzeigen	"`systemctl show $unit\|$job`"
Fehlgeschlagene "`$unit`" zurücksetzen (Reset)	"`systemctl reset-failed $unit`"
Abhängigkeiten aller Unit-Dienste auflisten	"`systemctl list-dependencies --all`"
Auf dem System installierte Unit-Dateien auflisten	"`systemctl list-unit-files`"
"`$unit`" aktivieren (symbolischen Link hinzufügen)	"`systemctl enable $unit`"
"`$unit`" deaktivieren (symbolischen Link entfernen)	"`systemctl disable $unit`"
"`$unit`" zum Starten bereit machen (symbolischen Link auf "`/dev/null`" entfernen)	"`systemctl unmask $unit`"
"`$unit`" am Starten hindern (symbolischen Link auf "`/dev/null`" hinzufügen)	"`systemctl mask $unit`"
Aktuelles default-Target abrufen	"`systemctl get-default`"
default-Target auf "`graphical`" setzen (grafische Oberfläche, GUI)	"`systemctl set-default graphical`"
default-Target auf "`multi-user`" setzen (Konsolenoberfläche, CLI)	"`systemctl set-default multi-user`"
Job-Umgebungseinstellungen anzeigen	"`systemctl show-environment`"
Job-Umgebungseinstellung "`variable`" auf "`wert`" setzen	"`systemctl set-environment variable=value`"
Job-Umgebungseinstellung "`variable`" löschen	"`systemctl unset-environment variable`"
Alle Unit-Dateien und Daemons neu laden	"`systemctl daemon-reload`"
System herunterfahren	"`systemctl poweroff`"
System herunterfahren und neu starten (Reboot)	"`systemctl reboot`"
System in Standby setzen (Suspend)	"`systemctl suspend`"
System in Ruhezustand setzen (Hibernate)	"`systemctl hibernate`"

Tabelle 3.6: Liste typischer `systemctl`-Befehle

Tätigkeit	Befehl
Dauer der einzelnen Initialisierungsschritte anzeigen	"`systemd-analyze time`"
Alle Units auflisten, sortiert nach der Dauer ihrer Initialisierung	"`systemd-analyze blame`"
"`$unit`"-Datei laden und auf Fehler prüfen	"`systemd-analyze verify $unit`"
Kurze Statusinformationen zum Benutzer der Sitzung vom Aufrufer	"`loginctl user-status`"
Kurze Statusinformationen zur Sitzung vom Aufrufer	"`loginctl session-status`"
Boot-Prozess nachverfolgen über cgroups	"`systemd-cgls`"
Boot-Prozess nachverfolgen über cgroups	"`ps xawf -eo pid,user,cgroup,args`"
Boot-Prozess nachverfolgen über cgroups	sysfs unter "`/sys/fs/cgroup/`" auslesen

Tabelle 3.7: Liste weiterer Überwachungsbefehle unter `systemd`

3.8.2 Das Dateisystem

Die Optionen zum Einbinden normaler Festplatten- und Netzwerkdateisysteme werden in "`/etc/fstab`" festgelegt. See fstab(5) und Abschnitt 9.6.7.

Die Konfiguration verschlüsselter Dateisysteme ist in "`/etc/crypttab`" abgelegt. Siehe crypttab(5).

Software-RAID mit mdadm(8) wird in "`/etc/mdadm/mdadm.conf`" konfiguriert. Siehe mdadm.conf(5).



Warnung
Bei jedem Systemstart werden nach dem Einbinden aller Dateisysteme temporäre Dateien in "`/tmp`", "`/var/lock`" und "`/var/run`" gelöscht.

3.8.3 Initialisierung der Netzwerkschnittstellen

Auf modernen Debian-Desktop-Systemen mit `systemd` erfolgt die Initialisierung von Netzwerkschnittstellen für die Loopback-Schnittstelle `lo` typischerweise durch "`networking.service`" und für andere Schnittstellen durch "`NetworkMar`

Details zur Konfiguration finden Sie in Kapitel 5.

3.8.4 Initialisierung eines Cloud-Systems

Die Cloud-System-Instanz kann als Klon der "Debian Official Cloud Images" oder ähnlicher Images ausgeführt werden. Für solche Systeminstanzen können persönliche Anpassungen wie Rechnername, Dateisystem, Netzwerk, Locale, SSH-Schlüssel, Benutzer und Gruppen über Funktionalitäten aus den Paketen `cloud-init` und `netplan.io` konfiguriert werden. Dabei werden mehrere Datenquellen genutzt, wie Dateien, die im Original-System-Image liegen, und externe Daten, die beim Start der Instanz bereitgestellt werden. Diese Pakete verwenden eine deklarative Systemkonfiguration mittels YAML-Daten.

Näheres finden Sie unter "Cloud Computing with Debian and its descendants", "Cloud-init-Dokumentation" und Abschnitt 5.4.

3.8.5 Anpassungsbeispiel zur Optimierung des sshd-Dienstes

Bei einer Standardinstallation werden viele Netzwerkdienste (siehe Kapitel 6) von `systemd` durch `network.target` als Daemon-Prozess gestartet. "`sshd`" ist hier keine Ausnahme. Als Beispiel dafür, wie man so etwas anpassen kann, wollen wir zeigen, wie Sie "`sshd`" so ändern, dass er nur auf Anfrage (on-demand) startet.

Als erstes deaktivieren Sie die entsprechende Dienst-Unit:

```
$ sudo systemctl stop sshd.service
$ sudo systemctl mask sshd.service
```

Der klassische Weg zur on-demand-Aktivierung von Sockets führte früher über den `inetd`- (oder `xinetd`)-Superserver. Unter `systemd` kann dies über das Hinzufügen von ***.socket** und ***.service** Unit-Konfigurationsdateien erreicht werden.

Eine `sshd.socket` anlegen zur Spezifizierung eines Sockets, der auf Anfragen überwacht wird:

```
[Unit]
Description=SSH Socket for Per-Connection Servers

[Socket]
ListenStream=22
Accept=yes

[Install]
WantedBy=sockets.target
```

Und eine `sshd@.service` als Dienste-Datei passend zu `sshd.socket`:

```
[Unit]
Description=SSH Per-Connection Server

[Service]
ExecStart=-/usr/sbin/sshd -i
StandardInput=socket
```

Dann muss der Dienst neu geladen werden:

```
$ sudo systemctl daemon-reload
```

3.9 Das udev-System

Das udev-System bietet seit dem Linux-Kernel 2.6 Mechanismen zur automatischen Hardware-Erkennung und -initialisierung (lesen Sie dazu udev(7)). Nach der Erkennung eines Gerätes durch den Kernel startet das udev-System einen User-Prozess. Dieser verwendet Informationen aus dem sysfs-Dateisystem (Näheres in Abschnitt 1.2.12), lädt über den Befehl modprobe(8) benötigte Kernel-Module, die die Hardware unterstützen (Details in Abschnitt 3.10), und erstellt die zugehörigen Geräteknoten (device nodes).

Tipp

Falls "`/lib/modules/`*`kernel-version`*`/modules.dep`" mittels depmod(8) aus irgendeinem Grund nicht korrekt erstellt wurde, könnte es beim Laden der Module durch das udev-System Probleme geben. Führen Sie "`depmod -a`" aus, um dies zu beheben.

Für die Regeln zum Einbinden von Dateisystemen in "`/etc/fstab`" müssen Geräteknoten nicht fest zugeordnet sein. Sie können auch UUIDs verwenden, um Geräte einzubinden, statt der Gerätenamen wie "`/dev/sda`". Lesen Sie dazu Abschnitt 9.6.3.

Da das udev-System immer ein wenig im Wandel ist, überlasse ich die Details anderen Dokumenten und beschränke mich hier auf das Nötigste.



Warnung

Versuchen Sie nicht, lange laufende Programme wie Backup-Skripte mit `RUN` in udev-Regeln zu starten, wie in udev(7) erwähnt. Erstellen Sie stattdessen eine saubere systemd.service(5)-Datei und aktivieren Sie diesen Service. Lesen Sie dazu Abschnitt 10.2.3.2.

3.10 Die Kernel-Modul-Initialisierung

Das modprobe(8)-Programm erlaubt es, einen laufenden Linux-Kernel über einen User-Prozess zu konfigurieren, indem Kernel-Module hinzugefügt und entfernt werden. Das udev-System (Näheres in Abschnitt 3.9) automatisiert dessen Aufruf, um bei der Initialisierung des Kernel-Moduls zu helfen.

Es gibt Module, die keine Hardware betreffen, sowie spezielle Hardware-Treiber-Module, die vorab geladen werden müssen. Hierzu werden sie – jeweils eines pro Zeile – in einer Datei mit der Endung `.conf` im Verzeichnis "`/etc/modules-load.d/`" aufgelistet (siehe modules-load.d(5)).

- TUN/TAP-Module, die ein virtuelles Point-to-Point Netzwerkgerät (TUN) und ein virtuelles Ethernet-Netzwerkgerät (TAP) bereitstellen;
- netfilter-Module, die Netfilter-Firewall-Funktionalitäten bereitstellen (lesen Sie dazu iptables(8) und Abschnitt 5.7);
- watchdog timer-Treibermodule.

Die Konfigurationsdateien für das modprobe(8)-Programm sind unterhalb des "`/etc/modprobes.d/`"-Verzeichnisses abgelegt, wie in modprobe.conf(5) beschrieben. (Falls Sie vermeiden möchten, dass einige Kernel-Module automatisch geladen werden, sollten Sie erwägen, diese in die Datei "`/etc/modprobes.d/blacklist`" einzutragen.)

Die Datei "`/lib/modules/`*`version`*`/modules.dep`" (erzeugt durch das Programm depmod(8)) beschreibt Abhängigkeiten zwischen den Modulen; diese Abhängigkeiten werden von modprobe(8) genutzt.

Anmerkung
Wenn Sie Probleme beim Laden von Modulen feststellen, entweder während des Systemstarts oder beim Nachladen mit modprobe(8), kann "`depmod -a`" diese Probleme möglicherweise durch Neuerstellung der "`modules.dep`"-Datei beheben.

Der Befehl modinfo(8) zeigt Informationen über ein Linux-Kernel-Modul an.

Das lsmod(8)-Programm formatiert den Inhalt von "`/proc/modules`" zu einer hübschen Ausgabe, um anzuzeigen, welche Kernel-Module gerade geladen sind.

Tipp
Sie können die Hardware in Ihrem System exakt identifizieren. Lesen Sie dazu Abschnitt 9.5.3.
Möglicherweise wollen Sie Hardware während des Systemstarts konfigurieren, um bestimmte erwartete Hardware-Funktionalitäten zu aktivieren. Näheres finden Sie in Abschnitt 9.5.4.
Unterstützung für spezielle Geräte können Sie unter Umständen durch Neukompilieren des Kernels hinzufügen. Details finden Sie in Abschnitt 9.10.

Kapitel 4

Authentifizierung und Zugriffskontrolle

Wenn eine Person (oder ein Programm) Zugriff auf das System erlangen möchte, wird über die Authentifizierung kontrolliert, ob der Identität vertraut werden kann.



Warnung
Konfigurationsfehler von PAM können Sie aus Ihrem eigenen System aussperren. Sie müssen eine Rettungs-CD zur Hand haben oder eine alternative Boot-Partition einrichten. Um das System wiederherzustellen, booten Sie das System damit und korrigieren Sie alles Nötige von dort.

4.1 Normale Unix-Authentifizierung

Die normale Unix-Authentifizierung wird über das Modul pam_unix(8) von PAM (Pluggable Authentication Modules) bereitgestellt. Dessen drei wichtige Konfigurationsdateien (mit durch "`:`" getrennten Einträgen) sind:

Datei	Berechtigung	Benutzer	Gruppe	Beschreibung
`/etc/passwd`	`-rw-r--r--`	`root`	`root`	(bereinigte) Informationen zum Benutzerkonto
`/etc/shadow`	`-rw-r-----`	`root`	`shadow`	geschützte Informationen zum Benutzerkonto
`/etc/group`	`-rw-r--r--`	`root`	`root`	Informationen zur Gruppe

Tabelle 4.1: Wichtige Konfigurationsdateien für pam_unix(8)

"`/etc/passwd`" enthält Folgendes:

```
 ...
user1:x:1000:1000:User1 Name,,,:/home/user1:/bin/bash
user2:x:1001:1001:User2 Name,,,:/home/user2:/bin/bash
 ...
```

Wie in passwd(5) beschrieben haben die durch "`:`" separierten Einträge dieser Datei folgende Bedeutungen:

- Benutzername (Login-Name);
- Eintrag zur Passwortspezifikation;
- nummerische Benutzer-ID;

- nummerische Gruppen-ID;
- Name oder Kommentarfeld;
- Home-Verzeichnis des Benutzers;
- optionale Angabe des Befehlsinterpreters für den Benutzer.

Der zweite Eintrag von "`/etc/passwd`" wurde früher für das verschlüsselte Passwort benutzt. Seit der Einführung von "`/etc/shadow`" enthält er die Passwortspezifikation:

Inhalt	Bedeutung
(leer)	Konto ohne Passwort
x	das verschlüsselte Passwort ist in "`/etc/shadow`" abgelegt

Tabelle 4.2: Inhalt des zweiten Eintrags in "`/etc/passwd`"

"`/etc/shadow`" enthält Folgendes:

```
 ...
user1:$1$Xop0FYH9$IfxyQwBe9b8tiyIkt2P4F/:13262:0:99999:7:::
user2:$1$vXGZLVbS$ElyErNf/agUDsm1DehJMS/:13261:0:99999:7:::
 ...
```

Wie in shadow(5) beschrieben haben die durch "`:`" separierten Einträge dieser Datei folgende Bedeutungen:

- Benutzername (Login-Name);
- verschlüsseltes Passwort (das "`$1$`" am Anfang zeigt die Verwendung der MD5-Verschlüsselung an; "*" steht für "kein Login");
- Datum der letzten Passwortänderung, ausgedrückt als Anzahl der Tage seit dem 1. Januar 1970;
- Anzahl der Tage, die der Benutzer vor der erneuten Änderung des Passworts warten muss;
- Anzahl der Tage, nach denen der Benutzer das Passwort ändern muss;
- Anzahl der Tage vor dem Verfall des Passworts, während derer der Benutzer vor dem Passwortverfall gewarnt wird;
- Anzahl der Tage, während derer das Passwort noch akzeptiert wird, obwohl es abgelaufen ist;
- Datum, an dem das Konto abläuft, ausgedrückt als Anzahl der Tage seit dem 1. Januar 1970.
- …

"`/etc/group`" enthält Folgendes:

```
group1:x:20:user1,user2
```

Wie in group(5) beschrieben haben die durch "`:`" separierten Einträge dieser Datei folgende Bedeutungen:

- Gruppenname;
- verschlüsseltes Passwort (nicht wirklich benutzt);
- nummerische Gruppen-ID;
- durch "," getrennte Liste von Benutzernamen.

Anmerkung
"`/etc/gshadow`" bietet für "`/etc/group`" ähnliche Funktionalität wie "`/etc/shadow`", diese wird aber nicht wirklich genutzt.

Anmerkung
Die aktuelle Gruppenmitgliedschaft eines Benutzers kann dynamisch angepasst werden, wenn eine Zeile mit "`auth optional pam_group.so`" zu "`/etc/pam.d/common-auth`" hinzugefügt wird und wenn dies in "`/etc/security/group.conf`" gesetzt ist. Lesen Sie dazu pam_group(8).

Anmerkung
Das `base-passwd`-Paket enthält eine verbindliche Liste von Benutzern und Gruppen: "`/usr/share/doc/base-passwd/users-and-groups.html`".

4.2 Verwalten von Konten- und Passwortinformationen

Hier einige erwähnenswerte Befehle zur Verwaltung von Konteninformationen:

Befehl	Funktion
`getent passwd` *`benutzername`*	Konteninformationen von *`benutzername`* anzeigen
`getent shadow` *`benutzername`*	durch shadow geschützte Konteninformationen von *`benutzername`* anzeigen
`getent group` *`gruppenname`*	Gruppeninformationen von *`gruppenname`* anzeigen
`passwd`	Passwort für das Konto verwalten
`passwd -e`	Einmal-Passwort für die Kontenaktivierung setzen
`chage`	Alterungsinformationen des Passworts verwalten

Tabelle 4.3: Liste von Befehlen zur Verwaltung von Konteninformationen

Sie benötigen für einige Funktionen root-Privilegien. Lesen Sie crypt(3) für Informationen zur Passwort- und Datenverschlüsselung.

Anmerkung
Auf Systemen, die mit PAM und NSS eingerichtet sind, wie der Debian-Salsa-Maschine, wird der Inhalt der lokalen Dateien "`/etc/passwd`", "`/etc/group`" und "`/etc/shadow`" auf dem System unter Umständen nicht aktiv verwendet. Obige Befehle sind aber auch in solchen Umgebungen gültig.

4.3 Ein gutes Passwort

Bei der Erstellung eines Kontos während der Systeminstallation oder mit dem passwd(1)-Befehl sollten Sie ein gutes Passwort auswählen, das aus mindestens sechs bis acht Zeichen besteht und ein oder mehrere Zeichen aus folgenden Gruppen enthält (gemäß passwd(1)):

- Kleinbuchstaben;
- Ziffern zwischen 0 und 9;
- Satzzeichen.



Warnung
Wählen Sie keine Wörter für das Passwort, die erraten werden könnten: Kontoname, Sozialversicherungsnummer, Telefonnummer, Adresse, Geburtstag, Name von Familienmitgliedern oder Haustieren, Wörter, die in Lexika auftauchen, einfache Zeichenfolgen wie ”12345” oder ”qwerty”, … alle diese sind eine schlechte Wahl für ein Passwort.

4.4 Verschlüsselte Passwörter erstellen

Es gibt mehrere unabhängige Werkzeuge, um verschlüsselte Passwörter mit Salz zu erzeugen:

Paket	Popcon	Größe	Befehl	Funktion
`whois`	V:25, I:199	384	`mkpasswd`	best-ausgestattetes Frontend für die crypt(3)-Bibliothek
`openssl`	V:853, I:996	2564	`openssl passwd`	Passwort-Hashes berechnen (OpenSSL). `passwd`(1ssl)

Tabelle 4.4: Liste der Werkzeuge zur Passworterzeugung

4.5 PAM und NSS

Moderne Unix-ähnliche Systeme wie Debian stellen dem lokalen Systemadministrator die Mechanismen PAM (Pluggable Authentication Modules) und NSS (Name Service Switch) zur Systemkonfiguration bereit. Deren Funktionen können wie folgt zusammengefasst werden:

- PAM bietet flexible Authentifizierungsmechanismen, die von Anwendungen genutzt werden, und integriert daher auch den Austausch von Passwortdaten.
- NSS besitzt einen flexiblen Name-Service-Mechanismus, der oft von der C-Standard-Bibliothek genutzt wird, um die Benutzer- und Gruppennamen für Programme wie ls(1) und id(1) einzuholen.

Diese PAM- und NSS-Systeme müssen konsistent konfiguriert sein.

Hier einige erwähnenswerte Pakete von PAM- und NSS-Systemen:

- Der ”Linux-PAM System Administrators’ Guide” in `libpam-doc` ist zum Erlernen der PAM-Konfiguration unerlässlich.
- Zum Erlernen der NSS-Konfiguration ist der Abschnitt ”System Databases and Name Service Switch” in `glibc-doc-refe` unerlässlich.

Anmerkung
Eine ausführlichere und aktuellere Liste bekommen Sie mit dem Befehl ”`aptitude search 'libpam-|libnss-'`”. Die Abkürzung NSS kann auch ”Network Security Service” bedeuten, was etwas anderes ist als ”Name Service Switch”.

Anmerkung
PAM ist der grundlegendste Weg, um Umgebungsvariablen für jegliche Programme mit einem systemweiten Standardwert vorzubelegen.

Unter systemd ist das Paket `libpam-systemd` installiert, um die Login-Informationen der Benutzer zu verwalten; dazu werden die Nutzersitzungen in der `systemd`-Kontrollgruppen-Hierarchie für logind registriert. Details finden Sie in systemd-logind(8), logind.conf(5) und pam_systemd(8).

Paket	Popcon	Größe	Beschreibung
libpam-modules	V:945, I:1000	892	Pluggable Authentication Modules (grundlegende Dienste)
libpam-ldapd	V:8, I:13	82	Pluggable Authentication Module, das LDAP-Schnittstellen erlaubt
libpam-systemd	V:788, I:972	849	Pluggable Authentication Module zur Registrierung von Nutzersitzungen für logind
libpam-doc	I:7.7	1492	Pluggable Authentication Modules (Dokumentation in html- und Textform)
libc6	V:924, I:1000	5483	GNU-C-Bibliothek: Programmbibliothek, die auch einen ”Name Service Switch”-Dienst bereitstellt
glibc-doc	I:4.9	3927	GNU-C-Bibliothek: Handbuchseiten
glibc-doc-reference	I:3.0	14764	GNU-C-Bibliothek: Referenzhandbuch im info-, pdf- und html-Format (nicht-frei)
libnss-mdns	V:314, I:505	142	NSS-Modul für Multicast-DNS-Namensauflösung
libnss-ldapd	V:9, I:16	138	NSS-Modul, um LDAP als Namensdienst zu verwenden

Tabelle 4.5: Liste von PAM- und NSS-Systemen

4.5.1 Konfigurationsdateien, auf die PAM und NSS zugreifen

Hier einige erwähnenswerte Konfigurationsdateien, die von PAM und NSS genutzt werden:

Einschränkungen bei der Passwortauswahl sind über die PAM-Module pam_unix(8) und pam_cracklib(8) implementiert. Diese können über deren Argumente konfiguriert werden.

Tipp
PAM-Module verwenden den Anhang ”.so” für ihre Dateinamen.

4.5.2 Modernes zentralisiertes Systemmanagement

Ein Lightweight Directory Access Protocol (LDAP)-Server erlaubt modernes zentralisiertes Systemmanagement und somit die Administrierung vieler Unix-ähnlicher und nicht-Unix-Systeme über das Netzwerk. Die quelloffene Implementation des Lightweight Directory Access Protocol ist OpenLDAP.

Der LDAP-Server stellt die Konteninformationen auf Debian-Systemen durch die Nutzung von PAM und NSS über die libpam-ldapd- und libnss-ldapd-Pakete bereit. Verschiedene Aktionen sind nötig, um dies zu aktivieren. (Ich verwende dieses Setup nicht, daher stammen diese Informationen komplett aus zweiter Hand. Bitte beachten Sie das beim Lesen dieses Abschnitts.)

- Richten Sie einen zentralisierten LDAP-Server ein, indem Sie ein Programm wie den eigenständigen LDAP-Daemon slapd(8) starten.
- Ändern Sie die PAM-Konfigurationsdateien im Verzeichnis ”/etc/pam.d/”, um ”pam_ldap.so” statt dem Standard ”pam_unix.so” zu verwenden.
- Ändern Sie die NSS-Konfiguration in der Datei ”/etc/nsswitch.conf”, um ”ldap” statt dem Standard (”compat” oder ”file”) zu verwenden.
- Richten Sie es so ein, dass libpam-ldapd SSL- (oder TLS-)Verbindungen verwendet, um die Passwortsicherheit zu gewährleisten.
- Richten Sie es so ein, dass libnss-ldapd SSL- (oder TLS-)Verbindungen verwendet, um die Unversehrtheit der Daten sicherzustellen (auch wenn dies zusätzlichen LDAP-Netzwerk-Overhead verursacht).

Konfigurationsdatei	Funktion
`/etc/pam.d/`*`programmname`*	Einrichtung der PAM-Konfiguration für das Programm *`programmname`*; lesen Sie dazu pam(7) und pam.d(5)
`/etc/nsswitch.conf`	Einrichtung der NSS-Konfiguration mit Einträgen für die jeweiligen Dienste; lesen Sie dazu nsswitch.conf(5)
`/etc/nologin`	Einschränkung der Benutzeranmeldung über das pam_nologin(8)-Modul
`/etc/securetty`	Einschränkung des tty für den root-Zugriff durch das pam_securetty(8)-Modul
`/etc/security/access.conf`	Zugriffsbeschränkungen setzen über das pam_access(8)-Modul
`/etc/security/group.conf`	Einschränkungen für Gruppen setzen über das pam_group(8)-Modul
`/etc/security/pam_env.conf`	Umgebungsvariablen setzen über das pam_env(8)-Modul
`/etc/environment`	Zusätzliche Umgebungsvariablen setzen über das pam_env(8)-Modul mit dem Argument "`readenv=1`"
`/etc/default/locale`	Setzen der Locale (Gebietsschema) über das pam_env(8)-Modul mit dem Argument "`readenv=1 envfile=/etc/default/locale`" (Debian)
`/etc/security/limits.conf`	Beschränkungen für Ressourcen (ulimit, core, ...) setzen über das pam_linits(8)-Modul
`/etc/security/time.conf`	Zeitbeschränkungen setzen über das pam_time(8)-Modul
`/etc/systemd/logind.conf`	`systemd` Login-Manager-Konfiguration setzen (siehe logind.conf(5) and systemd-logind.service(8))

Tabelle 4.6: Liste von Konfigurationsdateien, auf die PAM und NSS zugreifen

- Sie sollten nscd(8) lokal ausführen, um alle Antworten auf LDAP-Anfragen zwischenzuspeichern; dies reduziert den LDAP-Netzwerkverkehr.

Lesen Sie die Dokumentation in nsswitch.conf(5), pam.conf(5), ldap.conf(5) und "`/usr/share/doc/libpam-doc/html/`" aus dem `libpam-doc`-Paket sowie "`info libc 'Name Service Switch'`", bereitgestellt durch das Paket `glibc-doc`.

Ähnlich zu diesem Verfahren können Sie auch alternative zentralisierte Systeme mit anderen Methoden einrichten:

- Verflechtung von Benutzer- und Gruppeninformationen mit Windows-Systemen
  - Zugriff auf Windows Domain-Dienste mittels der Pakete `winbind`- und `libpam_winbind`;
  - Lesen Sie dazu winbindd(8) und Integrating MS Windows Networks with Samba.
- Verflechtung von Benutzer- und Gruppeninformationen mit Unix-ähnlichen Systemen
  - Zugriff auf NIS (ursprünglich YP genannt) oder NIS+ mittels dem `nis`-Paket;
  - Lesen Sie dazu das Linux NIS(YP)/NYS/NIS+ HOWTO.

4.5.3 "Warum unterstützt GNU su nicht die wheel-Gruppe"

Dies ist eine bekannte Redewendung am Ende der alten "`info su`"-Seite von Richard M. Stallman. Aber keine Sorge: der aktuelle `su`-Befehl in Debian nutzt PAM, so dass man die Erlaubnis zur Nutzung von `su` auf die `root`-Gruppe beschränken kann, indem die Zeile mit "`pam_wheel.so`" in "`/etc/pam.d/su`" aktiviert wird.

4.5.4 Schärfere Passwortregeln

Die Installation des `libpam-pwquality`-Pakets ermöglicht Ihnen, schärfere Regeln für Passwörter vorzuschreiben.

Auf einem typischen GNOME-System, auf dem automatisch `libpam-gnome-keyring` installiert wird, sieht "`/etc/pam.d/`" wie folgt aus:

```
# here are the per-package modules (the "Primary" block)
password  requisite     pam_pwquality.so retry=3 minlen=8 difok=3
password  [success=1 default=ignore]  pam_unix.so obscure use_authtok try_first_pass  ↩
    yescrypt
# here's the fallback if no module succeeds
password  requisite     pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password  required      pam_permit.so
# and here are more per-package modules (the "Additional" block)
password  optional  pam_gnome_keyring.so
# end of pam-auth-update config
```

4.6 Sicherheit der Authentifizierung

Anmerkung
Die Informationen hier sind für Ihre Sicherheitsbedürfnisse **möglicherweise nicht passend**, aber sie sollten für den Anfang **eine gute Basis** sein.

4.6.1 Sicheres Passwort im Internet

Viele beliebte Transport-Layer-Dienste kommunizieren Nachrichten inklusive der Passwort-Authentifizierung im Klartext. Es ist eine sehr schlechte Idee, Passwörter im Klartext über das wilde Internet zu übertragen, wo es abgehört werden kann. Sie können diese Dienste über ”Transport Layer Security” (TLS) oder seinen Vorgänger ”Secure Sockets Layer” (SSL) betreiben, um die vollständige Kommunikation inklusive des Passworts über die Verschlüsselung zu schützen.

Name des unsicheren Dienstes	Port	Name des sicheren Dienstes	Port
`www (http)`	80	`https`	443
`smtp (mail)`	25	`ssmtp (smtps)`	465
`ftp-data`	20	`ftps-data`	989
`ftp`	21	`ftps`	990
`telnet`	23	`telnets`	992
`imap2`	143	`imaps`	993
`pop3`	110	`pop3s`	995
`ldap`	389	`ldaps`	636

Tabelle 4.7: Liste von unsicheren und sicheren Diensten und Ports

Die Verschlüsselung kostet CPU-Zeit. Als CPU-freundliche Alternative können Sie die Kommunikation im Klartext lassen und nur das Passwort verschlüsseln; verwenden Sie dazu ein sicheres Authentifizierungsprotokoll wie ”Authenticated Post Office Protocol” (APOP) für POP oder ”Challenge-Response Authentication Mechanism MD5” (CRAM-MD5) für SMTP und IMAP. (Um E-Mail-Nachrichten über das Internet von Ihrem Mail-Client-Programm zum Mail-Server zu übertragen, ist es derzeit verbreitet, den neuen Message-Submission-Port 587 zu nutzen statt dem traditionellen SMTP-Port 25, um zu vermeiden, dass Port 25 vom Netzwerk-Provider geblockt wird, während Sie sich über CRAM-MD5 authentifizieren.)

Die traditionelle Passwort-basierte Authentifizierung leidet nach wie vor unter immanenten Sicherheitsproblemen, wie der Wiederverwendung von identischen Passwörtern für verschiedene Websites. Andere Möglichkeiten der Authenti-

fizierung können eine sichere Lösung sein: Time-based one-time password (TOTP), Universal 2nd Factor (U2F) oder Web Authentication (WebAuthn, allgemein auch oft ”Passkey” genannt).

4.6.2 Secure Shell (sichere Shell)

Das Secure-Shell-(SSH-)Programm bietet sichere verschlüsselte Kommunikation zwischen zwei nicht vertrauenswürdigen Rechnern über ein unsicheres Netzwerk mittels sicherer Authentifizierung. Es besteht aus dem OpenSSH-Client (ssh(1)) und dem OpenSSH-Daemon (sshd(8)). SSH kann genutzt werden, um mittels der Port-Forwarding-Funktionalität (Port-Weiterleitung) eine unsichere Protokoll-Kommunikation wie POP oder X gesichert durch das Internet zu tunneln.

Der Client versucht, sich selbst über eine Host-basierte Authentifizierung gegenüber dem Server zu identifizieren; dazu können verschiedene Verfahren angewandt werden: Public Key Authentication (über einen öffentlichen Schlüssel), Challenge-Response Authentication (es wird eine Aufgabe gestellt, für die die andere Seite die Lösung liefern muss) oder Passwort-Authentifizierung. Die Nutzung der Public Key Authentication ermöglicht eine Anmeldung aus der Ferne ohne Passwort. Lesen Sie dazu Abschnitt 6.3.

4.6.3 Zusätzliche Sicherheitsmaßnahmen für das Internet

Sogar wenn Sie sichere Dienste wie Secure Shell (SSH) und Point-to-point Tunneling Protocol (PPTP) verwenden, bestehen trotzdem noch Chancen für die Einbrecher mittels Brute-Force-Atttacken zum Erraten von Passwörtern usw. über das Internet. Die Nutzung von Firewall-Richtlinien (mehr dazu in Abschnitt 5.7) zusammen mit den folgenden Sicherheitswerkzeugen kann die Situation weiter verbessern.

Paket	Popcon	Größe	Beschreibung
knockd	V:0.7, I:1.5	110	kleiner Port-Knock-Daemon knockd(1) (horcht auf spezielle Anklopf-Sequenzen für Ports) und -Client knock(1)
fail2ban	V:101, I:112	2275	IP-Adressen sperren, die vielfache Authentifizierungsfehler verursachen

Tabelle 4.8: Liste von Werkzeugen, die zusätzliche Sicherheitsmaßnahmen ermöglichen

4.6.4 Sichern des root-Passworts

Um zu verhindern, dass Leute auf Ihre Maschine mit root-Privilegien zugreifen, müssen Sie folgendes erledigen:

- den physischen Zugriff auf den System-Datenspeicher (HDD / SSD / ...) verhindern;
- das UEFI/BIOS abriegeln und verhindern, dass von Wechseldatenträgern gebootet wird;
- ein Passwort für interaktive Sitzungen von GRUB vergeben;
- das Editieren des GRUB-Menüs sperren.

Mit physischem Zugriff auf den Datenspeicher ist das Zurücksetzen des root-Passworts relativ leicht; dies erfordert folgende Schritte:

1. den Datenspeicher in einen PC mit UEFI/BIOS einbauen, der von USB gebootet werden kann;
2. das System mit einem Rettungssystem booten (siehe Abschnitt 3.2.2);
3. die root-Partition mit Lese-/Schreibberechtigung einbinden;

4. ”/etc/passwd” auf der root-Partition editieren, um den zweiten Eintrag für den root-Zugang durch einen leeren Wert zu ersetzen.

Wenn Sie beim Booten Schreibzugriff auf den GRUB-Menüeintrag für grub-rescue-pc haben (lesen Sie dazu Abschnitt 3.1.2), ist es mit folgenden Schritten sogar noch einfacher:

1. das System booten, nachdem die Kernel-Parameter etwa wie folgt geändert wurden: ”root=/dev/sda6 rw init=/bin/sh”;
2. ”/etc/passwd” editieren, um den zweiten Eintrag für den root-Zugang durch einen leeren Wert zu ersetzen;
3. Neustart des Systems.

Die root-Shell des Systems ist jetzt ohne Passwort zugänglich.

Anmerkung
Sobald jemand Zugriff auf die root-Shell hat, kann er auf alles auf dem System zugreifen und jegliche Passwörter auf dem System zurücksetzen. Mehr noch, er könnte die Passwörter für alle Benutzerkonten kompromittieren, indem Brute-Force-Werkzeuge zum Knacken von Passwörtern wie john oder crack eingesetzt werden (Näheres in Abschnitt 9.5.10). Diese geknackten Passwörter könnten dann dazu verwendet werden, um andere Systeme zu kompromittieren.

Die einzige vernünftige Software-Lösung, um all diese Bedenken auszuräumen, ist die Verwendung einer Software-verschlüsselten root-Partition (oder ”/etc”-Partition) mittels dm-crypt und initramfs (lesen Sie dazu Abschnitt 9.9). Sie benötigen dann allerdings immer ein Passwort, um das System zu booten.

4.7 Andere Möglichkeiten zur Zugriffskontrolle

Es gibt Zugriffsbeschränkungen auf dem System, die nicht auf passwort-basierter Authentifizierung und Dateiberechtigungen beruhen.

Anmerkung
Abschnitt 9.4.16 enthält Informationen, wie Sie die Kernel-Funktion Secure attention key (SAK) einschränken.

4.7.1 Zugriffssteuerungslisten (Access control lists/ACLs)

ACLs sind eine Obermenge der regulären Berechtigungen, (beschrieben in Abschnitt 1.2.3).

ACLs in Aktion finden Sie in modernen Arbeitsplatzumgebungen. Wenn ein formatierter USB-Speicher automatisch eingebunden wird, z.B. als ”/media/penguin/USBSTICK”, kann ein normaler Benutzer penguin folgendes ausführen:

```
 $ cd /media/penguin
 $ ls -la
total 16
drwxr-x---+ 1 root    root    16 Jan 17 22:55 .
drwxr-xr-x  1 root    root    28 Sep 17 19:03 ..
drwxr-xr-x  1 penguin penguin 18 Jan  6 07:05 USBSTICK
```

Das ”+” an der 11. Stelle zeigt an, dass ACLs aktiv sind. Ohne ACLs sollte es einem normalen Benutzer nicht erlaubt sein, diesen ls-Befehl auszuführen, da penguin nicht in der root-Gruppe ist. Sie können die ACLs auflisten mit:

```
 $ getfacl .
# file: .
# owner: root
# group: root
user::rwx
user:penguin:r-x
group::---
mask::r-x
other::---
```

Hierbei gilt:

- "`user::rwx`", "`group::---`" und "`other::---`" stehen für regulären Eigentümer, Gruppe und Berechtigungen anderer.
- Die ACL "`user:penguin:r-x`" erlaubt es einem normalen Benutzer `penguin`, die Rechte "`r-x`" zu erhalten. Damit ist es diesem möglich, mit "`ls -la`" den Verzeichnisinhalt aufzulisten.
- Die ACL "`mask::r-x`" setzt eine Obergrenze für die Berechtigungen.

Lesen Sie dazu "POSIX Access Control Lists on Linux", acl(5), getfacl(1) und setfacl(1).

4.7.2 sudo

Das Programm sudo(8) wurde entwickelt, um einem Systemadministrator die Möglichkeit zu geben, Benutzern eingeschränkte root-Privilegien zu gewähren sowie um die Aktivitäten rund um das root-Konto zu protokollieren. `sudo` benötigt nur das Passwort eines normalen Benutzers. Installieren Sie das `sudo`-Paket und aktivieren Sie es, indem Sie passende Optionen in "`/etc/sudoers`" setzen. Konfigurationsbeispiele finden Sie unter "`/usr/share/doc/sudo/examples`" und in Abschnitt 1.1.12.

Die Art, wie ich `sudo` auf meinem Einzelbenutzersystem verwende (lesen Sie Abschnitt 1.1.12), soll mich selbst vor meiner eigenen Dummheit schützen. Ich persönlich denke, dass die Verwendung von `sudo` eine bessere Alternative zur dauerhaften Nutzung des root-Kontos ist. Der folgende Befehl ändert zum Beispiel den Eigentümer von "*`irgendeine_datei`*" in "*`mein_name`*":

```
$ sudo chown my_name some_file
```

Wenn Sie das root-Passwort kennen (was bei Benutzern, die sich ihr Debian-System selbst installieren, immer der Fall ist), können Sie natürlich jeden Befehl von jeglichem Benutzerkonto aus ausführen, indem Sie "`su -c`" verwenden.

4.7.3 PolicyKit

PolicyKit ist eine Komponente des Betriebssystems zur Kontrolle von systemweiten Privilegien auf Unix-ähnlichen Systemen.

Neuere GUI-Anwendungen sind nicht dafür entwickelt, als privilegierte Prozesse zu laufen. Sie kommunizieren mit privilegierten Prozessen über PolicyKit, um administrative Operationen durchführen zu können.

Auf Debian-Systemen beschränkt PolicyKit solche Operationen auf Benutzerkonten, die der `sudo`-Gruppe angehören.

Lesen Sie dazu polkit(8).

4.7.4 Den Zugriff auf einige Server-Dienste einschränken

Aus Gründen der Systemsicherheit ist es eine gute Idee, so viele Server-Programme wie möglich zu deaktivieren. Dies ist besonders für Netzwerk-Server kritisch. Ungenutzte Server, die entweder direkt als Daemon oder über den Super-Server aktiviert sind, müssen als Sicherheitsrisiko angesehen werden.

Viele Programme wie sshd(8) verwenden PAM-basierte Zugriffskontrollen. Es gibt viele Möglichkeiten, um den Zugriff auf Server-Dienste einzuschränken:

- Konfigurationsdateien: "`/etc/default/`*`programmname`*";
- Systemd Dienste-Unit-Konfiguration für den Daemon;
- PAM (Pluggable Authentication Modules);
- "`/etc/inetd.conf`" für den Super-Server;
- "`/etc/hosts.deny`" und "`/etc/hosts.allow`" für TCP-Wrapper (tcpd(8));
- "`/etc/rpc.conf`" für Sun RPC;
- "`/etc/at.allow`" und "`/etc/at.deny`" für atd(8);
- "`/etc/cron.allow`" und "`/etc/cron.deny`" für crontab(1);
- Netzwerk-Firewall aus der netfilter-Infrastruktur.

Lesen Sie dazu Abschnitt 3.6, Abschnitt 4.5.1 und Abschnitt 5.7.

Tipp
Sun-RPC-Dienste müssen für NFS und andere RPC-basierte Programme aktiviert sein.

Tipp
Falls Sie Probleme mit dem Zugriff von außen auf ein aktuelles Debian-System haben, kommentieren Sie blockierende Konfigurationselemente wie "`ALL: PARANOID`" in "`/etc/hosts.deny`" aus, falls solche existieren. (Aber sie müssen vorsichtig sein bezüglich der Sicherheitsrisiken, die durch solche Aktionen entstehen.)

4.7.5 Linux Sicherheits-Funktionalitäten

Der Linux-Kernel hat Sicherheitsfunktionen entwickelt und unterstützt sie immer noch, die in traditionellen UNIX-Implementierungen nicht zu finden sind.

Linux unterstützt erweiterte Dateiattribute, die über die traditionellen UNIX-Attribute hinausgehen (siehe xattr(7)).

Linux teilt die Privilegien, die dem Superuser traditionell zugewiesen wurden, in verschiedene Bereiche auf, bekannt als Capabilities(7), die unabhängig voneinander aktiviert oder deaktiviert werden können. Capatilities sind Attribute, die pro Prozess definiert werden können, und werden seit der Kernel-Version 2.2 unterstützt.

Das Linux Security Module (LSM) Framework stellt einen Mechanismus für verschiedene Sicherheits-Checks bereit, die von neuen Kernel-Erweiterungen abhängig sind. Zum Beispiel:

- AppArmor
- Security-Enhanced Linux (SELinux)
- Smack (Simplified Mandatory Access Control Kernel)
- Tomoyo Linux

Da diese Erweiterungen die Privilegien grundsätzlich weiter einschränken, als dies im normalen UNIX-artigen Sicherheitsmodell der Fall ist, können unter Umständen sogar die Superuser-Rechte beschnitten sein. Sie werden aufgefordert, das Dokument zum Linux Security Module (LSM) Framework auf kernel.org zu lesen.

Das Konzept der Linux Namespaces hüllt eine globale Systemressource in eine Abstraktion, die dazu führt, dass es für einen Prozess innerhalb des Namespace so erscheint, als hätte er eine eigene isolierte Instanz dieser globalen Ressource. Änderungen an der globalen Ressource sind für alle Prozesse innerhalb des Namespace sichtbar, nicht aber für andere Prozesse. Seit Kernel-Version 5.6 gibt es 8 Arten von Namespaces (lesen Sie dazu namespaces(7), unshare(1), nsenter(1)).

Mit Stand Debian 11 Bullseye (2021) nutzt Debian die Funktionalität ”unified cgroup hierarchy” (a.k.a. cgroups-v2).

Anwendungsbeispiele für Namespaces mit cgroups, die die Isolation von Prozessen sowie die Zugriffskontrolle ermöglichen, sind:

- Systemd. Siehe Abschnitt 3.3.1.
- Sandbox-Umgebungen. Näheres dazu in Abschnitt 7.7.
- Linux-Container wie Docker oder LXC. Lesen Sie hierzu Abschnitt 9.11.

Diese Funktionalitäten können nicht über das Konzept in Abschnitt 4.1 realisiert werden! Solche fortgeschrittenen Themen sind allerdings nicht Thema dieses einführenden Dokuments.

Kapitel 5

Netzwerkkonfiguration

Tipp
Bezüglich einer aktuellen Anleitung für Debian zum Thema Netzwerk lesen Sie Debian Administratorhandbuch — Konfigurieren des Netzwerks.

Tipp
Unter systemd kann networkd für die Netzwerkverwaltung genutzt werden; lesen Sie dazu systemd-networkd(8).

5.1 Die elementare Netzwerkinfrastruktur

Lassen Sie uns einen Blick auf die elementare Netzwerkinfrastruktur eines modernen Debian-Systems werfen:

5.1.1 Die Auflösung des Rechnernamens

Die Auflösung des Rechnernamens (hostname) wird derzeit auch durch den NSS-(Name-Service-Switch-)Mechanismus unterstützt. Die Auflösung läuft wie folgt ab:

1. Die "`/etc/nsswitch.conf`"-Datei mit Einträgen wie "`hosts: files dns`" bestimmt die Reihenfolge der Rechnernamenauflösung. (Dies ersetzt die alte Funktionalität der "`order`"-Einträge in "`/etc/host.conf`".)
2. Als erstes wird in diesem Beispiel die `files`-Methode aufgerufen. Wenn der Rechnername in der "`/etc/hosts`"-Datei gefunden wird, werden alle gültigen Adressen für den Rechner ausgegeben und die Abfrage wird beendet. (Die "`/etc/host.conf`"-Datei enthält "`multi on`".)
3. Dann wird die `dns`-Methode wird aufgerufen. Wenn der Rechnername über das Internet Domain Name System (DNS) (definiert über die Datei "`/etc/resolv.conf`") gefunden wird, werden alle dafür gültigen Adressen ausgegeben und die Abfrage wird beendet.

Eine typische Arbeitsplatzstation könnte z.B. installiert werden mit einem Rechnernamen wie "*host_name*" und der optionalen Domain-Namen bleibt leer. Dann sähe die "`/etc/hosts`" wie folgt aus:

```
127.0.0.1 localhost
127.0.1.1 host_name

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Pakete	Popcon	Größe	Art	Beschreibung
network-manager	V:429, I:472	7366	config::NM	NetworkManager (Daemon): das Netzwerk automatisch verwalten
network-manager-gnome	V:33, I:136	19	config::NM	NetworkManager (GNOME-Frontend)
netplan.io	V:2.4, I:8.7	340	config::NM+networkd	Netplan (generator): Vereinheitlichte, deklarative Schnittstelle zu NetworkManager und systemd-networkd-Backends
ifupdown	V:686, I:973	204	config::ifupdown	standardisiertes Werkzeug zum Aktivieren und Deaktivieren des Netzwerks (Debian-spezifisch)
pppoeconf	V:0.2, I:3.6	176	config::helper	Konfigurations-Hilfswerkzeug für PPPoE-Verbindungen
wpasupplicant	V:431, I:520	4337	config::helper	clientseitige Unterstützung für WPA und WPA2 (IEEE 802.11i)
wpagui	V:0.2, I:1.2	780	config::helper	Qt-GUI-Programm für wpa_supplicant
wireless-tools	V:196, I:263	232	config::helper	Werkzeuge zum Bearbeiten der Linux Wireless Extensions
iw	V:42, I:478	332	config::helper	Werkzeug zum Konfigurieren von Drahtlos-Netzwerkgeräten unter Linux
iproute2	V:802, I:988	4279	config::iproute2	iproute2, IPv6 und andere erweiterte Netzwerkkonfiguration: ip(8), tc(8) usw.
iptables	V:402, I:635	2408	config::Netfilter	Administrationswerkzeuge für Paketfilterung und NAT (Netfilter)
nftables	V:301, I:889	189	config::Netfilter	Administrationswerkzeuge für Paketfilterung und NAT (Netfilter) (Nachfolger von {ip,ip6,arp,eb}tables)
iputils-ping	V:222, I:998	188	Test	Erreichbarkeit eines fernen Rechners über das Netzwerk testen, entweder mittels Rechnername oder IP-Addresse (iproute2)
iputils-arping	V:2, I:17	53	Test	Erreichbarkeit eines fernen Rechners über das Netzwerk mittels seiner ARP-Addresse testen
iputils-tracepath	V:2, I:19	50	Test	Netzwerkpfad zu einem fernen Rechner verfolgen
ethtool	V:103, I:252	1106	Test	Eigenschaften von Ethernet-Geräten anzeigen oder ändern
mtr-tiny	V:5, I:38	182	test::low-level	Netzwerkpfad zu einem fernen Rechner verfolgen (Curses-basiert)
mtr	V:4, I:39	231	test::low-level	Netzwerkpfad zu einem fernen Rechner verfolgen (Curses- und GTK-basiert)
gnome-nettool	V:0.5, I:7.9	2480	test::low-level	Werkzeuge für allgemeine Netzwerkinformations-Operationen (GNOME)
nmap	V:27, I:180	4946	test::low-level	Netzwerk-Mapper/Port-Scanner (Nmap, konsolen-basiert)
tcpdump	V:24, I:181	1418	test::low-level	Netzwerkverkehr-Analysator (Tcpdump, konsolen-basiert)
wireshark	V:4, I:38	17442	test::low-level	Netzwerkverkehr-Analysator (Wireshark, GTK-basiert)
tshark	V:3, I:23	615	test::low-level	Netzwerkverkehr-Analysator (konsolen-basiert)
tcptrace	V:0.2, I:1.5	407	test::low-level	eine Zusammenfassung von Verbindungen auf Basis der `tcpdump`-Ausgabe erstellen
dnsutils	V:4, I:137	23	test::low-level	Netzwerk-Clients, die mit BIND bereitgestellt werden: nslookup(1), nsupdate(1), dig(1)
dlint	V:0.1, I:1.9	52	test::low-level	DNS-Zoneninformationen mittels Nameserver-Abfragen überprüfen
dnstracer	V:0.1, I:1.0	63	test::low-level	eine Verkettung von DNS-Servern zu ihrer Quelle verfolgen

Tabelle 5.1: Liste von Werkzeugen zur Netzwerkkonfiguration

Jede Zeile beginnt mit einer IP-Addresse und dahinter steht jeweils der zugeordnete Rechnername.

Die IP-Adresse `127.0.1.1` in der zweiten Zeile dieses Beispiels ist auf einigen anderen Unix-ähnlichen Systemen möglicherweise nicht vorhanden. Der Debian Installer erstellt diesen Eintrag für Systeme ohne feste IP-Adresse als provisorische Lösung für einige Software-Produkte (z.B. GNOME), wie in Fehler #719621 dokumentiert.

Der Eintrag *rechnername* entspricht dem in"`/etc/hostname`" festgelegten Rechnernamen (Näheres in Abschnitt 3.8.1).

Auf Systemen mit einer festen IP-Adresse sollte allerdings diese feste IP-Adresse statt der `127.0.1.1` verwendet werden.

Bei Systemen mit einer festen IP-Adresse und einem voll qualifizierten Domain-Namen (FQDN), bereitgestellt durch das Domain Name System (DNS), sollte *rechnername*.*domain-name* verwendet werden statt nur *rechnername*.

Die Datei "`/etc/resolv.conf`" ist eine statische Datei, falls das Paket `resolvconf` nicht installiert ist. Falls das Paket installiert ist, ist dies ein symbolischer Link. In beiden Fällen enthält es Informationen zur Initialisierung der Namensauflösungs-Routinen. Wenn das DNS zum Beispiel über die IP "`192.168.11.1`" erreichbar ist, enthält sie Folgendes:

```
nameserver 192.168.11.1
```

Das `resolvconf`-Paket macht "`/etc/resolv.conf`" zu einem symbolischen Link und verwaltet ihren Inhalt automatisch über die Hook-Skripte.

Bei PC-Arbeitsplatzrechnern in einer typischen LAN-Umgebung kann der Rechnername zusätzlich zu den grundlegenden `files`- und `dns`-Methoden auch über Multicast DNS (mDNS) aufgelöst werden.

- Avahi stellt ein Rahmenwerk für Multicast-DNS-Diensteabfragen auf Debian-Systemen bereit.
- Es ist ein Äquivalent zu Apple Bonjour / Apple Rendezvous.
- Das `libnss-mdns`-Plugin-Paket bietet Rechnernamensauflösung via mDNS für die GNU Name-Service-Switch-(NSS-)Funktionalität der GNU C-Bibliothek (glibc).
- Die Datei "`/etc/nsswitch.conf`" sollte einen Eintrag wie "`hosts: files mdns4_minimal [NOTFOUND=return] dns`" enthalten (andere Konfigurationen finden Sie in `/usr/share/doc/libnss-mdns/README.Debian`).
- Ein Rechnername mit der Pseudo-Top-Level Domain ".local" als Suffix wird aufgelöst, indem eine mDNS-Anfrage in einem Multicast-UDP-Paket gesendet wird (unter Verwendung entweder der IPv4-Adresse "`224.0.0.251`" oder der IPv6-Adresse "`FF02::FB`").

Anmerkung
Die Ausweitung generischer Top-Level-Domains (gTLD) im Domain-Name-System ist in Arbeit. Achten Sie bei Auswahl von Domain-Namen, die nur im lokalen Netzwerk verwendet werden sollen, auf Namenskollisionen.

Anmerkung
Die Verwendung von Paketen wie `libnss-resolve` zusammen mit `systemd-resolved`, oder `libnss-myhostname`, oder `libnss-mymachine`, mit entsprechenden Einträgen in der "`hosts`"-Zeile von "`/etc/nsswitch.conf`" könnte die oben erwähnte traditionelle Netzwerkkonfiguration überschreiben. Näheres dazu in nss-resolve(8), systemd-resolved(8), nss-myhostname(8) und nss-mymachines(8).

5.1.2 Der Netzwerkschnittstellenname

Systemd verwendet "verlässlich vorhersagbare Namen (Predictable Network Interface Names)" wie "`enp0s25`".

5.1.3 Der Netzwerkadressbereich für das LAN

Wir wollen uns an die IPv4 32-Bit-Adressbereiche erinnern, die durch die rfc1918 für jede Klasse zur Verwendung in Local Area Networks (LANs) reserviert sind. Diese Adressen werden bestimmt nicht mit irgendwelchen Adressen im Internet kollidieren.

Anmerkung
IP-Adressen mit Doppelpunkten sind IPv6-Adressen, z.B. "`::1`" für `localhost`.

Klasse	Netzwerkadressen	Netzmaske	Netzmaske /Bits	Anzahl der Subnetze
A	10.x.x.x	255.0.0.0	/8	1
B	172.16.x.x — 172.31.x.x	255.255.0.0	/16	16
C	192.168.0.x — 192.168.255.x	255.255.255.0	/24	256

Tabelle 5.2: Liste der Netzwerkadressbereiche

Anmerkung
Wenn eine dieser Adressen einem Rechner zugewiesen ist, kann dieser Rechner das Internet nicht direkt erreichen, sondern muss ein Gateway verwenden, der als Proxy für verschiedene Dienste dient, oder er nutzt Network Address Translation (NAT). Ein Breitband-Router nutzt üblicherweise NAT für das Anwender-Netzwerk.

5.1.4 Unterstützung für Netzwerkgeräte

Der größte Teil verfügbarer Netzwerk-Hardware wird durch das Debian-System unterstützt; es gibt einige Geräte, die laut DFSG nicht-freie Firmware für den Betrieb erfordern. Lesen Sie dazu Abschnitt 9.10.5.

5.2 Moderne Netzwerkkonfiguration für Arbeitsplatzsysteme

Auf modernen Debian-Desktop-Systemen mit `systemd` erfolgt die Initialisierung von Netzwerkschnittstellen für die Loopback-Schnittstelle `lo` typischerweise durch "`networking.service`" und für andere Schnittstellen durch "`NetworkMar`

Debian-Systeme können Netzwerkverbindungen über Software-Daemons wie NetworkManager (NM) (network-manager und zugehörige Pakete) verwalten.

- Sie haben ihre eigenen grafischen GUI- und Befehlszeilen-Programme als Bedienoberfläche.
- Sie haben ihre eigenen Daemons als Unterbau.
- Sie erlauben eine einfache Verbindung Ihres Systems mit dem Internet.
- Sie ermöglichen eine problemlose Verwaltung von kabelgebundenen und kabellosen Netzwerkkonfigurationen.
- Sie erlauben uns, das Netzwerk unabhängig vom althergebrachten `ifupdown` zu konfigurieren.

Anmerkung
Verwenden Sie diese automatischen Netzwerkkonfigurations-Werkzeuge nicht für Server. Sie sind primär für die Nutzung auf Arbeitsplatzrechnern oder Laptops gedacht.

Diese modernen Werkzeuge müssen korrekt konfiguriert werden, um Konflikte mit dem `ifupdown`-Paket und seiner Konfigurationsdatei "`/etc/network/interfaces`" zu vermeiden.

5.2.1 Grafische Netzwerkkonfigurations-Werkzeuge

Offizielle Dokumentation für NM unter Debian ist in "`/usr/share/doc/network-manager/README.Debian`" verfügbar.

Grundsätzlich läuft die Netzwerkkonfiguration für Arbeitsplatzsysteme wie folgt ab:

1. Fügen Sie den Benutzer, der sich am Arbeitsplatz anmeldet, z.B. `foo`, mit folgendem Befehl zur Gruppe "`netdev`" hinzu (alternativ kann dies in modernen Arbeitsplatzumgebungen wie GNOME oder KDE auch automatisch über D-bus erledigt werden):

```
$ sudo usermod -a -G netdev foo
```

2. Halten Sie die Konfiguration in "`/etc/network/interfaces`" so einfach wie hier:

```
auto lo
iface lo inet loopback
```

3. Starten Sie NM mit folgendem Befehl neu:

```
$ sudo systemctl restart NetworkManager
```

4. Konfigurieren Sie Ihr Netzwerk über die grafische GUI-Oberfläche.

Anmerkung
Um Konflikte mit `ifupdown` zu vermeiden, werden nur Schnittstellen, die **nicht** in "`/etc/network/interfaces`" aufgelistet sind, von NM verwaltet.

Tipp
Wenn Sie die Fähigkeiten von NM erweitern möchten, suchen Sie nach entsprechenden Plugin-Modulen und zusätzlichen Paketen wie `network-manager-openconnect`, `network-manager-openvpn-gnome`, `network-manager-pptp-gnome`, `mobile-broadband-provider-info` usw.

5.3 Moderne Netzwerkkonfiguration ohne grafische Oberfläche

Unter systemd kann das Netzwerk stattdessen in `/etc/systemd/network/` konfiguriert werden. Lesen Sie dazu systemd-resolved(8), resolved.conf(5) und systemd-networkd(8).

Dies ermöglicht eine moderne Netzwerkkonfiguration auch ohne grafische Oberfläche.

Eine DHCP-Client-Konfiguration kann durch Erzeugen von "`/etc/systemd/network/dhcp.network`" eingerichtet werden, z.B. mit:

```
[Match]
Name=en*

[Network]
DHCP=yes
```

Eine statische Netzwerkkonfiguration richten Sie über "`/etc/systemd/network/static.network`" ein, wie hier:

```
[Match]
Name=en*

[Network]
Address=192.168.0.15/24
Gateway=192.168.0.1
```

5.4 Moderne Netzwerkkonfiguration für Cloud-Systeme

Eine moderne Netzwerkkonfiguration für Cloud-Systeme kann die Pakete `cloud-init` und `netplan.io` nutzen (siehe Abschnitt 3.8.4).

Das `netplan.io`-Paket unterstützt `systemd-networkd` und `NetworkManager` als Netzwerkkonfigurations-Backends und aktiviert eine deklarative Netzwerkkonfiguration mittels YAML-Daten. Nach Änderungen der YAML-Daten:

- Führen Sie "`netplan generate`" aus, um die benötigte Backend-Konfiguration aus YAML zu generieren.
- Führen Sie "`netplan apply`" aus, um die generierte Konfiguration bei den Backends anzuwenden.

Lesen Sie die "Netplan-Dokumentation", netplan(5), netplan-generate(8) und netplan-apply(8).

Lesen Sie auch in der "Cloud-init-Dokumentation" (speziell bezüglich "Configuration sources" und "Netplan Passthrough"), wie `cloud-init` die `netplan.io`-Konfiguration mit alternativen Datenquellen integrieren kann.

5.4.1 Moderne Netzwerkkonfiguration für Cloud-Systeme mit DHCP

Eine DHCP-Client-Konfiguration kann eingerichtet werden, indem eine Datei "`/etc/netplan/50-dhcp.yaml`" erstellt wird:

```
network:
  version: 2
  ethernets:
    all-en:
      match:
        name: "en*"
      dhcp4: true
      dhcp6: true
```

5.4.2 Moderne Netzwerkkonfiguration für Cloud-Systeme mit statischer IP-Adresse

Eine statische Netzwerkkonfiguration können Sie einrichten, indem Sie "`/etc/netplan/50-static.yaml`" erstellen:

```
network:
  version: 2
  ethernets:
    eth0:
      addresses:
        - 192.168.0.15/24
      routes:
        - to: default
          via: 192.168.0.1
```

5.4.3 Moderne Netzwerkkonfiguration für Cloud-Systeme mit NetworkManager

Eine Netzwerkkonfiguration über die NetworkManager-Infrastruktur richten Sie mittels "`/etc/netplan/00-network-mana`
ein:

```
network:
  version: 2
  renderer: NetworkManager
```

5.5 Netzwerkkonfiguration auf unterster Ebene

Für Netzwerkkonfiguration über die Konsole können Sie unter Linux die iproute2-Programme (ip(8), …) verwenden.

5.5.1 iproute2-Befehle

Die iproute2-Befehle bieten vollwertige Funktionalität auf der untersten Ebene der Netzwerkkonfiguration. Hier eine Tabelle zur Gegenüberstellung von veralteten net-tools-Befehlen und neuen iproute2- und anderen Befehlen.

net-tools (veraltet)	iproute2 usw. (neu)	Beeinflussung
ifconfig(8)	`ip addr`	Protokoll-Adresse (IP oder IPv6) eines Gerätes
route(8)	`ip route`	Eintrag in der Routing-Tabelle
arp(8)	`ip neigh`	ARP- oder NDISC-Cache-Eintrag
`ipmaddr`	`ip maddr`	Multicast-Adresse
`iptunnel`	`ip tunnel`	Tunnel über IP
nameif(8)	ifrename(8)	Netzwerkschnittstellen basierend auf MAC-Adressen benennen
mii-tool(8)	ethtool(8)	Einstellungen von Ethernet-Geräten

Tabelle 5.3: Gegenüberstellung von `net-tools`- und `iproute2`-Befehlen

Lesen Sie ip(8) und das Linux Advanced Routing & Traffic Control.

5.5.2 Sichere Basis-Netzwerkoperationen

Sie können die folgenden Netzwerkbefehle der untersten Ebene problemlos verwenden, da sie die Netzwerkkonfiguration nicht verändern:

Tipp
Einige dieser Basisbefehle zur Netzwerkkonfiguration sind in "`/usr/sbin/`" abgelegt. Sie müssen unter Umständen den vollständigen Pfad, wie z.B. "`/usr/sbin/ifconfig`" angeben oder "`/usr/sbin`" zur Variable "`$PATH`" in Ihrer "`~/.bashrc`"-Datei hinzufügen.

5.6 Netzwerkoptimierung

Die grundsätzliche Netzwerkoptimierung liegt außerhalb des Rahmens dieser Dokumentation. Ich erwähne hier nur Dinge, die für Anwender-typische Verbindungen passend sind.

5.6.1 Die optimale MTU finden

NM setzt den optimalen Wert für die Maximum Transmission Unit (MTU) normalerweise automatisch.

In speziellen Fällen möchten Sie die MTU jedoch vielleicht händisch setzen, nachdem Sie mit ping(8) und seiner Option "`-M do`" experimentiert haben; Sie haben damit die Möglichkeit, ein ICMP-Paket mit verschiedenen Paketgrößen zu verschicken. MTU ist die größte Paketgröße, bei der das Paket noch erfolgreich ohne Fragmentierung verschickt werden kann plus 28 Byte für die IPv4- bzw. 48 Byte für die IPv6-Adresse. In folgendem Beispiel wurde für eine IPv4-Verbindung eine MTU von 1460 ermittelt und für IPv6 eine MTU von 1500:

Befehl	Beschreibung
`ip addr show`	Verbindungs- und Adressstatus von aktiven Schnittstellen anzeigen
`route -n`	Vollständige Routing-Tabelle mit numerischen Adressen anzeigen
`ip route show`	Vollständige Routing-Tabelle mit numerischen Adressen anzeigen
`arp`	Aktuellen Inhalt der ARP-Cache-Tabellen anzeigen
`ip neigh`	Aktuellen Inhalt der ARP-Cache-Tabellen anzeigen
`plog`	Logdaten des PPP-Daemons anzeigen
`ping yahoo.com`	Internet-Verbindung zu "`yahoo.com`" überprüfen
`whois yahoo.com`	Überprüfen, wer "`yahoo.com`" in der Domain-Datenbank registriert hat
`traceroute yahoo.com`	Verbindung zu "`yahoo.com`" durch das Internet verfolgen
`tracepath yahoo.com`	Verbindung zu "`yahoo.com`" durch das Internet verfolgen
`mtr yahoo.com`	Verbindung zu "`yahoo.com`" durch das Internet verfolgen (wiederholt)
`dig [@dns-server.com] example.com [{a|mx|any}]`	DNS-Einträge von "`example.com`" laut den Daten von "`dns-server.com`" auf einen "a"-, "`mx`"- oder "`any`"-Eintrag überprüfen
`iptables -L -n`	Paketfilter überprüfen
`ss -a`	Alle offenen Ports finden
`ss -l`	Ports finden, die auf eine Verbindung warten
`ss -l -t -n`	TCP-Ports finden, die auf eine Verbindung warten (numerisch)
`dlint example.com`	DNS-Zonen-Informationen von "`example.com`" überprüfen

Tabelle 5.4: Liste von Basis-Netzwerkbefehlen

Pakete	Popcon	Größe	Beschreibung
`iftop`	V:6, I:83	93	Informationen zur Bandbreitennutzung einer Netzwerkschnittstelle anzeigen
`iperf`	V:2, I:31	23	Werkzeug zur IP-Bandbreiten-Messung
`ifstat`	V:0.5, I:5.1	52	InterFace STATistics Monitoring (Netzwerkschnittstellen-Statistik/-Überwachung)
`bmon`	V:2, I:19	154	Portierbarer Bandbreitenmonitor und Geschwindigkeitsrechner
`ethstatus`	V:0.2, I:2.3	41	Skript, das schnell den Durchsatz eines Netzwerkgerätes messen kann
`bwm-ng`	V:1.2, I:8.8	95	Kleiner und einfacher konsolenbasierter Bandbreitenmonitor
`ethstats`	V:0.06, I:0.35	21	Konsolenbasierter Ethernet-Statistikmonitor

Tabelle 5.5: Liste von Werkzeugen zur Netzwerkoptimierung

```
$ ping -4 -c 1 -s $((1500-28)) -M do www.debian.org
PING  (149.20.4.15) 1472(1500) bytes of data.
ping: local error: message too long, mtu=1460

---  ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

$ ping -4 -c 1 -s $((1460-28)) -M do www.debian.org
PING  (130.89.148.77) 1432(1460) bytes of data.
1440 bytes from klecker-misc.debian.org (130.89.148.77): icmp_seq=1 ttl=50 time=325 ms

---  ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 325.318/325.318/325.318/0.000 ms
$ ping -6 -c 1 -s $((1500-48)) -M do www.debian.org
PING www.debian.org(mirror-csail.debian.org (2603:400a:ffff:bb8::801f:3e)) 1452 data bytes
1460 bytes from mirror-csail.debian.org (2603:400a:ffff:bb8::801f:3e): icmp_seq=1 ttl=47  ↩
    time=191 ms

--- www.debian.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 191.332/191.332/191.332/0.000 ms
```

Dies ist das Path MTU (PMTU) Discovery-Verfahren (RFC1191) und der Befehl tracepath(8) kann dies automatisieren.

Netzwerkumgebung	MTU	Argumentation
Einwahlverbindung (IP: PPP)	576	Standard
Ethernet-Verbindung (IP: DHCP oder fest)	1500	Standard und vorgegeben

Tabelle 5.6: Wesentliche Grundregeln für den optimalen MTU-Wert

Zusätzlich zu diesen Grundregeln sollten Sie folgendes wissen:

- Jegliche Nutzung von Tunneling-Methoden (VPN usw.) kann aufgrund des Overheads den optimalen MTU-Wert reduzieren.
- Der MTU-Wert sollte den über die experimentelle Methode ermittelten PMTU-Wert nicht überschreiten.
- Ein größerer MTU-Wert ist grundsätzlich besser, wenn andere Einschränkungen greifen.

Die Maximum Segment Size (MSS) wird als alternative Messmethode für die Paketgröße verwendet. Der Zusammenhang zwischen MSS und MTU ist wie folgt:

- MSS = ”MTU - 40” bei IPv4
- MSS = ”MTU - 60” bei IPv6

Anmerkung
Bei Netzwerkoptimierung mittels iptables(8) (lesen Sie dazu auch Abschnitt 5.7) kann die Paketgröße über die MSS begrenzt werden; dies ist für einen Router nützlich. Lesen Sie den Abschnitt bezüglich ”TCPMSS” in iptables(8).

5.6.2 WAN-TCP-Optimierung

Der TCP-Durchsatz kann über die Anpassung von Parametern zur TCP-Puffergröße maximiert werden, wie in "TCP tuning"für modernes WAN mit hoher Bandbreite und hoher Latenz beschrieben. Das soll hierzu genügen; die aktuellen Debian-Standardeinstellungen funktionieren für mein LAN mit seiner Verbindung zum sehr schnellen 1G bps FFTP-Dienst sehr gut.

5.7 Die Netfilter-Infrastruktur

Netfilter stellt eine Infrastruktur für Stateful Packet Inspection (SPI, zustandsorientierte Paketüberprüfung) und Network Address Translation (NAT) über Module des Linux-Kernels (lesen Sie hierzu Abschnitt 3.10) zur Verfügung.

Pakete	Popcon	Größe	Beschreibung
`nftables`	V:301, I:889	189	Administrationswerkzeuge für Paketfilterung und NAT (Netfilter) (Nachfolger von {ip,ip6,arp,eb}tables)
`iptables`	V:402, I:635	2408	Administrationswerkzeuge für netfilter (iptables(8) für IPv4, ip6tables(8) für IPv6)
`arptables`	V:0.1, I:1.6	102	Administrationswerkzeuge für netfilter (arptables(8) für ARP)
`ebtables`	V:14, I:22	276	Administrationswerkzeuge für netfilter (ebtables(8) für Ethernet-Bridging-Betrieb)
`iptstate`	V:0.1, I:1.6	130	Fortlaufende Überwachung des netfilter-Status (ähnlich zu top(1))
`ufw`	V:87, I:110	862	Uncomplicated Firewall (UFW) ist ein Programm zur Verwaltung einer Netfilter-Firewall
`gufw`	V:5, I:10	3682	grafische Bedienoberfläche für Uncomplicated Firewall (UFW)
`firewalld`	V:16, I:22	2581	firewalld ist ein dynamisch verwaltetes Firewall-Programm mit Unterstützung für Netzwerkzonen
`firewall-config`	V:0.8, I:3.0	1096	grafische Bedienoberfläche für firewalld
`shorewall-init`	V:0.13, I:0.31	88	Initialisierung der Shoreline Firewall
`shorewall`	V:2.2, I:4.5	3090	Erzeugung von netfilter-Konfigurationsdateien für Shoreline Firewall
`shorewall-lite`	V:0.03, I:0.05	71	Erzeugung von netfilter-Konfigurationsdateien für Shoreline Firewall (abgespeckte Version)
`shorewall6`	V:0.7, I:1.2	1334	Erzeugung von netfilter-Konfigurationsdateien für Shoreline Firewall (IPv6-Version)
`shorewall6-lite`	V:0.01, I:0.01	71	Erzeugung von netfilter-Konfigurationsdateien für Shoreline Firewall (abgespeckte IPv6-Version)

Tabelle 5.7: Liste von Firewall-Werkzeugen

Siehe das netfilter.org "nftables"-Projekt für den neuen Userspace-Befehl nft(8) aus dem `nftables`-Paket (Linux kernel >= 3.13).

Siehe das netfilter.org "iptables"-Projekt für den älteren Userspace-Befehl iptables(8) und weitere aus dem `iptables`-Paket sowie aus anderen ähnlichen Paketen (Linux kernel >= 2.4.x).

Sie können netfilter von Hand interaktiv über die Shell konfigurieren, seinen Status mit "`nft list ruleset > file`" oder iptables-save(8) sichern und es beim Systemstart via Init-Skript über "`nft -f file`" oder über iptables-restore(8) wiederherstellen.

Konfigurations-Hilfsskripte wie shorewall vereinfachen diesen Prozess.

Sie finden Dokumentation unter Netfilter Documentation oder in "`/usr/share/doc/iptables/html/`":

- Linux Networking-concepts HOWTO
- Linux 2.4 Packet Filtering HOWTO
- Linux 2.4 NAT HOWTO
- nftables HOWTO

Kapitel 6

Netzwerkapplikationen

Nach dem Aufbau der Netzwerkverbindung (laut Kapitel 5) können Sie verschiedenste Netzwerkapplikationen nutzen.

Tipp
Bezüglich einer aktuellen Anleitung für Debian zum Thema Netzwerk-Infrastruktur lesen Sie das Debian Administratorhandbuch — Netzwerk-Infrastruktur.

Tipp
Falls Sie ”2-Schritt-Verifizierung” aktiviert haben, müssen Sie bei manchen Internet-Providern ein Anwendungspasswort beziehen, um aus Ihrem Anwendungsprogramm Zugriff auf POP- und SMTP-Dienste zu erhalten. Auch kann es erforderlich sein, dass Sie im voraus Ihre Host-IP-Adresse bestätigen müssen.

6.1 Webbrowser

Es gibt viele Webbrowser-Pakete, um über das Hypertext Transfer Protocol (HTTP-Protokoll) auf ferne Inhalte zuzugreifen:

Paket	Popcon	Größe	Art	Beschreibung des Webbrowsers
`chromium`	V:37, I:101	324490	X	Chromium (quelloffener Browser von Google)
`firefox`	V:15, I:21	317809	, ,	Firefox, (quelloffener Browser von Mozilla, nur in Debian Unstable verfügbar)
`firefox-esr`	V:188, I:423	299668	, ,	Firefox ESR, (Firefox-Variante mit erweitertem Support-Umfang, ESR)
`epiphany-browser`	V:2.0, I:9.7	6737	, ,	GNOME, HIG-konform, Epiphany
`konqueror`	V:26, I:110	8191	, ,	KDE, Konqueror
`dillo`	V:0.5, I:3.8	2203	, ,	Dillo (ressourcenschonender Browser, FLTK-basiert)
`w3m`	V:11, I:125	2853	Text	w3m
`lynx`	V:30, I:454	2031	, ,	Lynx
`elinks`	V:2, I:14	1839	, ,	ELinks
`links`	V:2, I:19	2344	, ,	Links (Nur-Text)
`links2`	V:0.8, I:9.2	5497	grafisch	Links (Konsolen-Grafik ohne X)

Tabelle 6.1: Liste der Webbrowser

6.1.1 Fälschen der User-Agent-Angabe

Um auf einige übermäßig beschränkte Webseiten zugreifen zu können, müssen Sie unter Umständen die User-Agent-Angabe fälschen, die Ihr Webbrowser zurück gibt. Näheres unter:

- MDN Web Docs: userAgent (englisch)
- Chrome-Entwickler: die User-Agent-Angabe überschreiben (englisch)
- Wie Sie Ihre User-Agent-Angabe ändern (englisch)
- Wie Sie die User-Agent-Angabe in Chrome, Firefox, Safari und weiteren ändern (englisch)
- Wie Sie die User-Agent-Angabe Ihres Browsers ändern, ohne irgendwelche Erweiterungen installieren zu müssen (englisch)
- Wie Sie die User-Agent-Angabe in Gnome Web (epiphany) ändern (englisch)

6.1.2 Browser-Erweiterung

Alle modernen grafischen Webbrowser unterstützen quellcode-basierte Browser-Erweiterungen; diese sollen als Web extensions standardisiert werden.

6.2 Das Mail-System

Dieser Abschnitt ist überwiegend für typische tragbare Arbeitsplatzrechner (Laptops) gedacht, die über Internetverbindungen verfügen, wie sie für normale Privatkunden üblich sind.



Achtung
Falls Sie vorhaben, den Mail-Server so einzurichten, dass Mails direkt über das Internet ausgetauscht werden, gibt es wahrscheinlich bessere Beschreibungen als dieses grundlegende Dokument.

6.2.1 Grundlagen des E-Mail-Systems

Eine E-Mail besteht aus drei Komponenten: den Absender- und Empfänger-Informationen, wie sie bei einem normalen Brief auf dem Umschlag stehen würden, den Nachrichten-Kopfzeilen (Header) und der eigentlichen Nachricht (dem "Nachrichtenkörper").

- Die "An"- und "Von"-Informationen ("To" und "From") auf dem "Umschlag" werden von SMTP genutzt, um die Nachricht zuzustellen. (Die "From"-Information auf dem Umschlag wird auch mit Bounce address/Envelope sender, From_ usw. bezeichnet).
- Die "An"- und "Von"-Informationen ("To" und "From") in den Nachrichten-Kopfzeilen werden vom E-Mail-Client (E-Mail-Programm) angezeigt. (Obwohl es sehr gängig ist, dass diese mit den Informationen vom "Umschlag" übereinstimmen, muss dies nicht immer der Fall sein.)
- Das E-Mail-Nachrichtenformat bestehend aus Kopfzeilen (Header) und Nachrichtenkörper (Body) wird erweitert durch die Multipurpose Internet Mail Extensions (MIME), so dass zusätzlich zum reinen ASCII-Text auch andere Zeichenkodierungen unterstützt werden sowie Anhänge mit Audio-, Video-, Grafik- oder Applikationsdaten.

Voll ausgestattete GUI-basierte E-Mail-Clients bieten all diese Funktionen über eine grafische intuitive Konfiguration.

- Sie erstellen und interpretieren die Nachrichten-Kopfzeilen und den Nachrichtenkörper unter Verwendung der Multipurpose Internet Mail Extensions (MIME), um den Datentyp und die Kodierung des Nachrichteninhalts korrekt zu bestimmen.
- Sie authentifizieren sich selbst gegenüber den SMTP- und IMAP-Servern des ISP mittels der althergebrachten Basic Access Authentication oder der modernen Variante OAuth 2.0. (Um OAuth 2.0 zu nutzen, legen Sie dies über die Einstellungen Ihrer Arbeitsplatzumgebung fest; z.B. über ”Einstellungen” -> ”Online-Konten”.)
- Sie versenden die Nachrichten an den SMTP-Server des ISP (über den Message Submission Port 587).
- Sie empfangen die auf dem Mail-Server des ISP vorhandenen Nachrichten über den TSL/IMAP4-Port (993).
- Sie können Mails über deren Attribute filtern.
- Teilweise bieten sie auch weitere optionale Funktionalitäten: Kontakte, Kalender, Aufgaben, Notizen.

Paket	Popcon	Größe	Art
`evolution`	V:27, I:227	521	Programm mit grafischer X-Oberfläche (GNOME, Groupware-Suite)
`thunderbird`	V:39, I:100	304278	Programm mit grafischer X-Oberfläche (GTK, Mozilla Thunderbird)
`kmail`	V:43, I:101	25384	Programm mit grafischer X-Oberfläche (KDE)
`mutt`	V:11, I:78	7378	Programm für textbasiertes Terminal, zu benutzen u.a. mit `vim`
`mew`	V:0.01, I:0.14	2319	Programm für textbasiertes Terminal, zu benutzen mit `(x)emacs`

Tabelle 6.2: Liste der Mail User Agents (MUA)

6.2.2 Einschränkungen moderner E-Mail-Dienste

Moderne E-Mail-Dienste unterliegen einigen Einschränkungen, um die Belästigung/Schädigung durch Spam (unerwünschte und unverlangte E-Mails) zu minimieren.

- Sie können nicht davon ausgehen, dass das Betreiben eines SMTP-Servers an einem Endverbraucher-Anschluß (und somit das Senden von Nachrichten direkt an einen fernen Zielrechner) zuverlässig funktioniert.
- Eine E-Mail kann an jeglicher Stelle auf der Route zum Ziel ohne weiteren Hinweis verworfen werden, außer sie ist als so authentisch wie möglich erkennbar.
- Sie können nicht davon ausgehen, dass das Versenden von Mails an ferne Rechner mit einer zufälligen Absender-Mail-Adresse über einen eigenen Smarthost zuverlässig funktioniert.

Die Gründe hierfür sind:

- Kommunikation über den SMTP-Port (25) von Rechnern an einem Endverbraucher-Anschluß wird geblockt.
- Kommunikation über den SMTP-Port (25) zu Rechnern an einem Endverbraucher-Anschluß wird geblockt.
- Ausgehende Nachrichten von Rechnern an einem Endverbraucher-Anschluß ins Internet können nur über den Message Submission Port (587) versandt werden.
- Zur E-Mail-Filterung werden großflächig Anti-Spam-Techniken wie DomainKeys Identified Mail (DKIM), Sender Policy Framework (SPF) und Domain-based Message Authentication, Reporting and Conformance (DMARC) eingesetzt.
- Unter Umständen können Sie zum Versand von Mails über einen Smarthost den DomainKeys Identified Mail-Dienst verwenden.
- Der Smarthost-Dienst überschreibt unter Umständen die Absenderadresse in den Nachrichten-Kopfzeilen mit der Ihres Mail-Accounts, um zu verhindern, dass die E-Mail-Adresse gefälscht wird.

6.2.3 Historische Erwartungen an E-Mail-Dienste

Einige Programme im Debian-System erwarten, dass auf den `/usr/sbin/sendmail`-Befehl zugegriffen werden kann, um in der Standardeinstellung (oder auch in benutzerspezifisch angepassten Konfigurationen) E-Mails verschicken zu können. Der Grund hierfür ist, dass der Mail-Dienst auf UNIX-Systemen historisch gesehen wie folgt funktioniert:

- Eine E-Mail wird als Textdatei erstellt.
- Die E-Mail wird an den `/usr/sbin/sendmail`-Befehl übergeben.
- Wenn die Zieladresse auf dem gleichen Rechner liegt, stellt `/usr/sbin/sendmail` die Mail per lokaler Auslieferung zu, indem sie an die Datei `/var/mail/$username` angehängt wird.
  - Befehle, die diese Funktionalität nutzen: `apt-listchanges`, `cron`, `at`, ...
- Liegt die Zieladresse auf einem fernen Rechner, sendet `/usr/sbin/sendmail` die Nachricht per SMTP an den (per DNS-MX-Eintrag ermittelten) Zielrechner.
  - Befehle, die diese Funktionalität nutzen: `popcon`, `reportbug`, `bts`, ...

6.2.4 Mail Transfer Agent (MTA)

Mobile Arbeitsplatzrechner unter Debian können seit Debian 12 Bookworm so konfiguriert werden, dass ein voll ausgestatteter grafischer E-Mail-Client ohne installierten Mail Transfer Agent (MTA) trotzdem korrekt funktioniert.

Traditionell hat Debian immer ein MTA-Programm installiert, um Mail-Programme zu unterstützen, die den `/usr/sbin/send` erwarten. Solche MTA müssen auf mobilen Arbeitsplatzrechnern mit Abschnitt 6.2.2 und Abschnitt 6.2.3 zurechtkommen.

Bei mobilen Arbeitsplatzrechnern ist die typische Wahl für den MTA entweder `exim4-daemon-light` oder `postfix` (bei diesem wird die Installationsoption ”E-Mails direkt mittels SMTP oder über fetchmail empfangen; ausgehende E-Mails über einen Smarthost versenden.” ausgewählt). Dies sind ressourcen-schonende MTAs, die auch ”`/etc/aliases`” respektieren.

Tipp

Die Konfiguration von `exim4` zum Versenden von E-Mails über mehrere Absender-Mail-Adressen und mehrere dazugehörige Smarthosts ist alles andere als einfach. Wenn Sie solche Funktionalitäten für einige Programme benötigen, richten Sie diese so ein, dass sie `msmtp` benutzen; damit ist es auf einfache Art möglich, mehrere Absender-Adressen einzurichten. Den Haupt-MTA können Sie dann für nur eine einzige Mail-Adresse eingerichtet lassen.

6.2.4.1 Die Konfiguration von exim4

Für das Verschicken von Mails über das Internet via Smarthost (re-)konfigurieren Sie die `exim4-*`-Pakete wie folgt:

```
$ sudo systemctl stop exim4
$ sudo dpkg-reconfigure exim4-config
```

Wählen Sie bei ”Generelle E-Mail-Einstellungen” den Eintrag ”Versand über Sendezentrale (Smarthost); Empfang mit SMTP oder Fetchmail”.

Setzen Sie den ”E-Mail-Name des Systems” auf seinen Standardwert, den FQDN (Näheres dazu in Abschnitt 5.1.1).

Bei ”IP-Adressen, an denen eingehende SMTP-Verbindungen erwartet werden” wählen Sie den Standardwert ”127.0.0.1 ; ::1”.

Paket	Popcon	Größe	Beschreibung
exim4-daemon-light	V:197, I:206	1777	Exim4 Mail Transfer Agent (MTA: Debian-Standard)
exim4-daemon-heavy	V:4.5, I:4.8	1955	Exim4 Mail Transfer Agent (MTA: flexible Alternative)
exim4-base	V:201, I:211	1687	Exim4-Dokumentation (Text-Variante) und allgemeine Dateien
exim4-doc-html	I:0.94	3809	Exim4-Dokumentation (html-Variante)
exim4-doc-info	I:0.51	649	Exim4-Dokumentation (info-Variante)
postfix	V:99, I:104	4277	Postfix Mail Transfer Agent (MTA: sichere Alternative)
postfix-doc	I:4.0	5030	Postfix-Dokumentation (html- + Text-Variante)
sasl2-bin	V:4.5, I:9.4	370	Cyrus SASL API Implementation (Ergänzung zu Postfix für SMTP AUTH)
cyrus-sasl2-doc	I:0.68	2140	Cyrus SASL - Dokumentation
msmtp	V:10, I:15	822	Ressourcen-schonender MTA
msmtp-mta	V:7.8, I:9.8	139	Ressourcen-schonender MTA (Sendmail-Kompatibilitätserweiterung für `msmtp`)
nullmailer	V:7.2, I:7.6	483	eingeschränkter MTA, kein lokaler Mail-Transfer
ssmtp	V:3.8, I:5.6	134	eingeschränkter MTA, kein lokaler Mail-Transfer
sendmail-bin	V:9.4, I:9.7	1954	vollständig ausgestatteter MTA (nur, wenn Sie sich bereits damit auskennen)
git-email	V:0.4, I:8.6	1229	git-send-email(1) Programm zum Versenden von Patch-Serien per E-Mail

Tabelle 6.3: Liste von Paketen für grundlegende Mail Transfer Agents

Löschen Sie alle eingetragenen Werte bei der Abfrage von ”Weitere Ziele, für die E-Mails angenommen werden sollen”.

Löschen Sie alle eingetragenen Werte bei der Abfrage von ”Rechner, für die E-Mails weitergeleitet werden (Relay)”.

Setzen Sie ”IP-Adresse oder Rechnername der Sendezentrale für ausgehende E-Mails” auf ”smtp.hostname.dom:587”.

Wählen Sie ”Nein” bei der Frage ”Lokalen E-Mail-Namen in ausgehenden E-Mails verbergen?”. (Nutzen Sie stattdessen ”`/etc/email-addresses`” wie in Abschnitt 6.2.4.3 beschrieben.)

Bei der Abfrage ”DNS-Anfragen minimieren (Automatische Einwahl, Dial-on-Demand)?” gehen Sie wie folgt vor:

- Wählen Sie ”Nein”, wenn Ihr System während des Rechnerstarts mit dem Internet verbunden ist.
- Wählen Sie ”Ja”, wenn Ihr System während des Rechnerstarts **nicht** (oder nicht immer) mit dem Internet verbunden ist.

Setzen Sie den Wert für ”Versandart bei lokaler E-Mail-Zustellung:” auf ”Mbox-Format in /var/mail/”.

Bei der Frage ”Einstellungen auf kleine Dateien aufteilen?” wählen Sie ”Ja”.

Erzeugen Sie Passworteinträge für den Smarthost, indem Sie ”`/etc/exim4/passwd.client`” editieren:

```
$ sudo vim /etc/exim4/passwd.client
 ...
$ cat /etc/exim4/passwd.client
^smtp.*\.hostname\.dom:username@hostname.dom:password
```

Konfigurieren Sie exim4(8) mit ”`QUEUERUNNER='queueonly'`”, ”`QUEUERUNNER='nodaemon'`” usw. in ”`/etc/default/e`” um die Nutzung von Systemressourcen zu minimieren (optional).

Starten Sie `exim4` mit folgendem Befehl:

```
$ sudo systemctl start exim4
```

Der Rechnername in "`/etc/exim4/passwd.client`" sollte nicht der Alias-Name sein. Sie können den echten Rechnernamen wie folgt herausfinden:

```
$ host smtp.hostname.dom
smtp.hostname.dom is an alias for smtp99.hostname.dom.
smtp99.hostname.dom has address 123.234.123.89
```

Ich verwende reguläre Ausdrücke in "`/etc/exim4/passwd.client`", um das Alias-Problem zu umgehen. SMTP AUTH funktioniert unter Umständen sogar, wenn der Internet-Diensteanbieter den Host (Rechner), auf den der Alias zeigt, verändert.

Sie können die `exim4`-Konfiguration händisch aktualisieren, indem Sie wie folgt vorgehen:

- Aktualisieren der `exim4`-Konfigurationsdateien in "`/etc/exim4/`".
  - Erzeugen von "`/etc/exim4/exim4.conf.localmacros`", um MACROs zu aktivieren und Editieren von "`/etc/exim` (bei Installation ohne gesplittete Konfigurationsdateien)
  - Erzeugen neuer oder Editieren vorhandener Dateien in den Unterverzeichnissen von "`/etc/exim4/exim4.conf.d`" (bei Installation mit gesplitteten Konfigurationsdateien)
- Führen Sie "`systemctl reload exim4`" aus.



Achtung
Das Starten von `exim4` dauert ziemlich lange, wenn die Frage "DNS-Anfragen minimieren (Automatische Einwahl, Dial-on-Demand)?" mit "Nein" (dem Standardwert) beantwortet wurde und das System während dem Rechnerstart **nicht** mit dem Internet verbunden ist.

Lesen Sie bitte die offizielle Anleitung unter "`/usr/share/doc/exim4-base/README.Debian.gz`" und update-exim4.conf(8).



Warnung
Aus praktischen Gründen sollten Sie SMTP mit STARTTLS auf Port 587 verwenden oder SMTPS (SMTP over SSL) auf Port 465, stattreinem SMTP auf Port 25.

6.2.4.2 Die Konfiguration von Postfix mit SASL

Für das Verschicken von Mails über das Internet via Smarthost sollten Sie zunächst die Postfix-Dokumentation und einschlägige dazugehörige Handbuchseiten lesen.

Befehl	Funktion
postfix(1)	Postfix-Steuerprogramm
postconf(1)	Postfix-Konfigurationswerkzeug
postconf(5)	Postfix-Konfigurationsparameter
postmap(1)	Wartung von Postfix-Lookup-Tabellen
postalias(1)	Wartung der Postfix-Alias-Datenbank

Tabelle 6.4: Liste von wichtigen Postfix-Handbuchseiten

Sie (re-)konfigurieren die `postfix`- und `sasl2-bin`-Pakete wie folgt:

```
$ sudo systemctl stop postfix
$ sudo dpkg-reconfigure postfix
```

Wählen Sie ”Internet mit Smarthost”.

Setzen Sie ”SMTP-Relay-Server (leere Eingabe: keiner)” auf ”`[smtp.rechnername.dom]:587`” und konfigurieren Sie ihn mit folgenden Befehlen:

```
$ sudo postconf -e 'smtp_sender_dependent_authentication = yes'
$ sudo postconf -e 'smtp_sasl_auth_enable = yes'
$ sudo postconf -e 'smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd'
$ sudo postconf -e 'smtp_sasl_type = cyrus'
$ sudo vim /etc/postfix/sasl_passwd
```

Erzeugen Sie Passworteinträge für den Smarthost:

```
$ cat /etc/postfix/sasl_passwd
[smtp.hostname.dom]:587     username:password
$ sudo postmap hush:/etc/postfix/sasl_passwd
```

Starten Sie `postfix` wie folgt:

```
$ sudo systemctl start postfix
```

Hierbei stellt die Verwendung von ”`[`” und ”`]`” im `dpkg-reconfigure`-Dialog und in ”`/etc/postfix/sasl_passwd`” sicher, dass keine MX-Einträge abgefragt werden, sondern direkt der angegebene Rechnername genutzt wird. Lesen Sie dazu ”Enabling SASL authentication in the Postfix SMTP client” in ”`/usr/share/doc/postfix/html/SASL_README`

6.2.4.3 Die Mail-Adress-Konfiguration

Es gibt mehrere Dateien zur Konfiguration der Mail-Adresse für Mail Transfer Agents, Mail Delivery Agents und Mail User Agents.

Datei	Funktion	Applikation
`/etc/mailname`	Standard-Rechnername für (ausgehende) Mails	Debian-spezifisch, mailname(5)
`/etc/email-addresses`	Änderung des Rechnernamens für ausgehende Mails	exim(8)-spezifisch, exim4-config_files(5)
`/etc/postfix/generic`	Änderung des Rechnernamens für ausgehende Mails	postfix(1)-spezifisch, aktiviert nach Ausführung des postmap(1)-Befehls
`/etc/aliases`	Kontonamen-Alias für eingehende Mails	Grundlegend, aktiviert nach Ausführung des newaliases(1)-Befehls

Tabelle 6.5: Liste von Konfigurationsdateien für Mail-Adressen

Der **mailname** in der Datei ”`/etc/mailname`” ist normalerweise ein voll-qualifizierter Domain-Name (FQDN), der sich auf eine der IP-Adressen des Rechners auflösen lässt. Bei Privatkundenrechnern, die keinen solchen FQDN-Namen mit einer aufzulösenden IP-Adresse haben, setzen Sie diesen **mailname** auf den Wert von ”`hostname -f`”. (Dies ist eine sichere Wahl und funktioniert sowohl für `exim4-*` wie auch für `postfix`.)

Tipp
Der Inhalt von ”`/etc/mailname`” wird von vielen nicht-MTA-Programmen für deren Standardverhalten genutzt. Für `mutt` setzen Sie die ”`hostname`”- und ”`from`”-Variablen in der Datei `~/muttrc`, um den **mailname**-Wert zu überschreiben. Für Programme im `devscripts`-Paket, wie bts(1) und dch(1), exportieren Sie die Umgebungsvariablen ”`$DEBFULLNAME`” und ”`$DEBEMAIL`”, um ihn zu überschreiben.

Tipp
Das Paket `popularity-contest` sendet normalerweise Mails vom root-Konto über den FQDN. Sie müssen `MAILFROM` in `/etc/popularity-contest.conf` setzen, wie in der Datei `/usr/share/popularity-contest/default.conf` beschrieben. Andernfalls wird Ihre Mail vom Smarthost-SMTP-Server abgelehnt werden. Obwohl lästig, ist dieser Ansatz sicherer als das Neuschreiben der Ursprungsadresse für alle Mails von root über den MTA und sollte für andere Daemons und cron-Skripte ebenfalls genutzt werden.

Wenn Sie den **mailname** auf den Wert von "`hostname -f`" setzen, kann die Änderung der Mail-Ursprungsadresse via MTA wie folgt realisiert werden:

- für exim4(8) über die Datei "`/etc/email-addresses`", wie in exim4-config_files(5) beschrieben;
- für postfix(1) über die Datei "`/etc/postfix/generic`", wie in generic(5) beschrieben.

Für `postfix` sind die folgenden zusätzlichen Schritte nötig:

```
# postmap hash:/etc/postfix/generic
# postconf -e 'smtp_generic_maps = hash:/etc/postfix/generic'
# postfix reload
```

Sie können die Mailadressen-Konfiguration wie folgt testen:

- exim(8) mit den Optionen `-brw, -bf, -bF, -bV, …`;
- postmap(1) mit der Option `-q`.

Tipp
Exim wird zusammen mit mehreren Hilfsprogrammen wie exiqgrep(8) und exipick(8) ausgeliefert. Informationen über verfügbare Befehle finden Sie mittels "`dpkg -L exim4-base|grep man8/`".

6.2.4.4 Grundlegende MTA-Operationen

Es gibt mehrere grundlegende MTA-Operationen. Einige könnten über die sendmail(1)-Kompatibilitätsschnittstelle ausgeführt werden.

Tipp
Es könnte eine gute Idee sein, alle Mails durch ein Skript in "`/etc/ppp/ip-up.d/*`" zustellen zu lassen.

6.3 Der Server für Fernzugriff (SSH) und Hilfsprogramme

Die Secure SHell (SSH) ist der **sichere** Weg für Verbindungen über das Internet. Eine freie Version von SSH namens OpenSSH ist in Debian über die `openssh-client`- und `openssh-server`-Pakete verfügbar.

Für den Benutzer fungiert ssh(1) als clevere und sichere Alternative zu telnet(1). Anders als der `telnet`-Befehl scheitert `ssh` nicht am `telnet`-Maskierungszeichen (escape character; Standardeinstellung STRG-]).

Obwohl `shellinabox` kein SSH-Programm ist, ist es hier als interessante Alternative für fernen Terminal-Zugriff aufgelistet.

Lesen Sie auch Abschnitt 7.9 für Infos zur Verbindung mit fernen X-Client-Programmen.

exim-Befehl	postfix-Befehl	Beschreibung
`sendmail`	`sendmail`	Mails von der Standardeingabe lesen und für die Zustellung vorbereiten (`-bm`)
`mailq`	`mailq`	Die Mail-Warteschlange auflisten mit Status und Queue-ID (`-bp`)
`newaliases`	`newaliases`	Initialisieren der Alias-Datenbank (`-I`)
`exim4 -q`	`postqueue -f`	Wartende Mails zustellen (`-q`)
`exim4 -qf`	`postsuper -r ALL deferred; postqueue -f`	Alle Mails zustellen
`exim4 -qff`	`postsuper -r ALL; postqueue -f`	Zurückgehaltene Mails ebenfalls zustellen
`exim4 -Mg queue_id`	`postsuper -h queue_id`	Eine durch ihre Queue-ID spezifizierte Nachricht zurückhalten
`exim4 -Mrm queue_id`	`postsuper -d queue_id`	Eine durch ihre Queue-ID spezifizierte Nachricht löschen
Nicht verfügbar	`postsuper -d ALL`	Alle Nachrichten löschen

Tabelle 6.6: Liste grundlegender MTA-Operationen

Paket	Popcon	Größe	Werkzeug	Beschreibung
`openssh-client`	V:709, I:997	3611	ssh(1)	Secure-Shell-Client
`openssh-server`	V:781, I:824	3615	sshd(8)	Secure-Shell-Server
`ssh-askpass`	I:15	103	ssh-askpass(1)	Fragt den Benutzer nach einer Passphrase für ssh-add (reines X)
`ssh-askpass-gnome`	V:0.4, I:2.9	46	gnome-ssh-askpass(1)	Fragt den Benutzer nach einer Passphrase für ssh-add (GNOME)
`ssh-askpass-fullscreen`	V:0.08, I:0.36	41	ssh-askpass-fullscreen(1)	Fragt den Benutzer nach einer Passphrase für ssh-add (GNOME) - etwas hübscher
`shellinabox`	V:0.60, I:0.88	528	shellinaboxd(1)	Webserver für den VT100-Terminal-Emulator mit Browser-Zugriff

Tabelle 6.7: Liste von Servern für Fernzugriff und Hilfsprogrammen



Achtung
Lesen Sie Abschnitt 4.6.3, falls Ihr SSH über das Internet erreichbar ist.

Tipp
Bitte nutzen Sie das Programm screen(1), um dem Remote-Shell-Prozess die Chance zu geben, Verbindungsunterbrechungen zu überstehen (Weiteres dazu finden Sie in Abschnitt 9.1.2).

6.3.1 Grundlagen von SSH

Der OpenSSH-Daemon unterstützt lediglich das SSH-Protokoll 2.

Bitte lesen Sie "`/usr/share/doc/openssh-client/README.Debian.gz`", ssh(1), sshd(8), ssh-keygen(1), ssh-add(1) und ssh-agent(1).



Warnung
Wenn Sie den OpenSSH-Server laufen lassen möchten, darf "`/etc/ssh/sshd_not_to_be_run`" nicht vorhanden sein.
Aktivieren Sie NICHT rhost-basierte Authentifizierung (`HostbasedAuthentication` in `/etc/ssh/sshd_config`).

Konfigurationsdatei	Beschreibung
`/etc/ssh/ssh_config`	SSH-Client-Standardeinstellungen, lesen Sie ssh_config(5)
`/etc/ssh/sshd_config`	SSH-Server-Standardeinstellungen, lesen Sie sshd_config(5)
`~/.ssh/authorized_keys`	öffentliche Standard-SSH-Schlüssel, die Clients verwenden, um sich mit diesem Konto auf diesem SSH-Server zu verbinden
`~/.ssh/id_rsa`	geheimer SSH-2 RSA-Schlüssel des Benutzers
`~/.ssh/id_`*`key-type-name`*	geheimer SSH-2 *key-type-name*-Schlüssel wie z.B. `ecdsa`, `ed25519`, ... des Benutzers

Tabelle 6.8: Liste von SSH-Konfigurationsdateien

Mit folgenden Befehlen starten Sie eine ssh(1)-Verbindung von einem Client:

Befehl	Beschreibung
`ssh benutzername@rechnername.domaene.ext`	im Standardmodus verbinden
`ssh -v benutzername@rechnername.domaene.ext`	im Standardmodus verbinden mit aktivierten Debugging-Meldungen (zur Fehlersuche)
`ssh -o PreferredAuthentications=password username@hostname.domain.ext`	Verwendung des Passworts erzwingen mit SSH Version 2
`ssh -t username@hostname.domain.ext passwd`	führen Sie `passwd` aus, um das Passwort auf einem fernen Rechner zu ändern

Tabelle 6.9: Liste von Beispielen zum Start einer SSH-Verbindung auf einem Client

6.3.2 Benutzername auf dem fernen Rechner

Wenn Sie auf dem lokalen und dem fernen Rechner den gleichen Benutzernamen verwenden, können Sie "`username@`" weglassen.

Sogar wenn Sie unterschiedliche Benutzernamen nutzen, können Sie sie weglassen, sofern Sie "`~/.ssh/config`" verwenden. Für den Debian Salsa-Service mit dem Kontoname "`foo-guest`" muss "`~/.ssh/config`" z.B. folgendes enthalten:

```
Host salsa.debian.org people.debian.org
User foo-guest
```

6.3.3 Verbindungen ohne Passwörter für die ferne Seite

Man kann es vermeiden, sich Passwörter für ferne Systeme merken zu müssen, indem man "`PubkeyAuthentication`" (SSH-2-Protokoll) nutzt.

Setzen Sie dazu auf dem fernen System die entsprechenden Einträge: "`PubkeyAuthentication yes`" in "`/etc/ssh/ssh`

Erzeugen Sie die Authentifizierungs-Schlüssel lokal und installieren Sie sie wie folgt auf dem fernen System:

```
$ ssh-keygen -t rsa
$ cat .ssh/id_rsa.pub | ssh user1@remote "cat - >>.ssh/authorized_keys"
```

Sie können weitere Optionen zu den Einträgen in "`~/.ssh/authorized_keys`" hinzufügen, um die erlaubten Rechner einzuschränken und spezifische Befehle ausführen. Lesen Sie dazu sshd(8) - "AUTHORIZED_KEYS FILE FORMAT".

6.3.4 Der Umgang mit fremden SSH-Clients

Es sind einige freie SSH-Clients für andere Plattformen verfügbar:

Umgebung	freies SSH-Programm
Windows	puTTY (PuTTY: ein freier SSH- und Telnet-Client) (GPL)
Windows (Cygwin)	SSH in cygwin (Cygwin: das Linux-Feeling unter Windows bekommen) (GPL)
Mac OS X	OpenSSH; verwenden Sie `ssh` in der Terminal-Applikation (GPL)

Tabelle 6.10: Liste freier SSH-Clients für andere Plattformen

6.3.5 Einrichten von ssh-agent

Es ist sicherer, Ihren geheimen SSH-Authentifizierungs-Schlüssel mit einer Passphrase zu schützen. Falls Sie noch keine Passphrase vergeben haben, verwenden Sie dazu "`ssh-keygen -p`".

Legen Sie Ihren öffentlichen SSH-Schlüssel (z.B. "`~/.ssh/id_rsa.pub`") wie folgt in "`~/.ssh/authorized_keys`" auf dem fernen Server ab, und zwar über eine passwort-basierte Verbindung zum Server, wie oben beschrieben.

```
$ ssh-agent bash
$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/username/.ssh/id_rsa:
Identity added: /home/username/.ssh/id_rsa (/home/username/.ssh/id_rsa)
```

Es ist kein Passwort für das ferne System zur Ausführung des nächsten Befehls mehr nötig:

```
$ scp foo username@remote.host:foo
```

Drücken Sie Strg-D, um die ssh-agent-Sitzung zu beenden.

Für den X-Server führt das normale Debian-Start-Skript `ssh-agent` als Eltern-Prozess aus. Sie müssen `ssh-add` daher nur einmal ausführen. Für weitere Details lesen Sie ssh-agent(1) und ssh-add(1).

6.3.6 Eine Mail versenden auf einem fernen Rechner

Wenn Sie ein SSH-Shell-Konto auf einem Server mit vernünftigen DNS-Einstellungen haben, können Sie eine Mail, die Sie auf Ihrem Arbeitsplatzrechner vorbereitet haben, von dem fernen Rechner aus als E-Mail verschicken:

```
$ ssh username@example.org /usr/sbin/sendmail -bm -ti -f "username@example.org" < mail_data ↩
    .txt
```

6.3.7 Portweiterleitung für SMTP-/POP3-Tunnelung

Um für eine Verbindung von Port 4025 auf `localhost` zu Port 25 auf `remote-server` (einem fernen Server) oder für eine Verbindung von Port 4110 auf `localhost` zu Port 110 auf `remote-server` via `ssh` eine Weiterleitung aufzubauen, führen Sie auf dem lokalen Rechner (localhost) folgendes aus:

```
# ssh -q -L 4025:remote-server:25 4110:remote-server:110 username@remote-server
```

Dies ist ein sicherer Weg für Verbindungen zu SMTP-/POP3-Servern über das Internet. Setzen Sie in "`/etc/ssh/sshd_con` auf dem fernen Server den Wert für "`AllowTcpForwarding`" auf "`yes`".

6.3.8 Wie Sie das ferne System über SSH herunterfahren

Sie müssen den Prozess, der das "`shutdown -h now`" ausführt (lesen Sie Abschnitt 1.1.8), davor schützen, durch SSH beendet zu werden. Verwenden Sie dazu wie folgt den at(1)-Befehl (weiteres zu at in Abschnitt 9.4.13):

```
# echo "shutdown -h now" | at now
```

"`shutdown -h now`" in einer screen(1)-Sitzung auszuführen (weitere Infos in Abschnitt 9.1.2) ist ein anderer möglicher Weg.

6.3.9 Fehlersuche bei SSH

Falls Sie Probleme haben, kontrollieren Sie die Zugriffsrechte der Konfigurationsdateien und starten Sie `ssh` mit der Option "`-v`".

Nutzen Sie die Option "`-p`", wenn Sie root sind und Probleme mit einer Firewall haben; dadurch wird die Verwendung der Ports 1 - 1023 auf dem Server vermieden.

Wenn `ssh`-Verbindungen zu einem fernen Rechner plötzlich nicht mehr funktionieren, könnte dies auf Spielereien des Systemadministrators zurückzuführen sein, höchstwahrscheinlich Änderungen am "`host_key`" im Rahmen von Systemwartungsarbeiten. Nachdem Sie sich versichert haben, dass dies der Fall ist und niemand versucht, den fernen Rechner über einen cleveren Hack zu imitieren, können Sie die Verbindung wiedererlangen, indem Sie den "`host_key`"-Eintrag in der "`~/.ssh/known_hosts`"-Datei auf dem lokalen Rechner entfernen.

6.4 Der Print-Server und Hilfsprogramme

Im alten Unix-ähnlichen System war der BSD Line Printer Daemon (lpd) Standard und das Standard-Druckausgabe-Format von freier Software auf Unix-artigen Systemen war PostScript (PS). Ein Filtersystem wurde zusammen mit Ghostscript verwendet, um das Drucken auf nicht-PostScript-Druckern zu ermöglichen. Siehe Abschnitt 11.4.1.

Im modernen Debian-System ist das Common UNIX Printing System (CUPS) der De-Facto-Standard und das Standard-Druckausgabe-Format von moderner freier Software ist das Portable Document Format (PDF).

CUPS nutzt das Internet Printing Protocol (IPP). IPP ist der plattform-übergreifende De-Facto-Standard für das Drucken über Netzwerk; es unterstützt bi-direktionale Kommunikation.

Dank der Dateiformat-abhängigen automatischen Konvertierungsfunktion des CUPS-Systems sollte die einfache Übergabe jeglicher Daten an den `lpr`-Befehl zur gewünschten Druckausgabe führen. (In CUPS kann `lpr` aktiviert werden, indem das `cups-bsd`-Paket installiert wird.)

Das Debian-System enthält einige erwähnenswerte Pakete für Print-Server und deren Hilfsprogramme:

Paket	Popcon	Größe	Port	Beschreibung
`lpr`	V:1.9, I:2.2	378	printer (515)	BSD lpr/lpd (Line Printer Daemon)
`cups`	V:141, I:444	1102	IPP (631)	Internet-Printing CUPS-Server
`cups-client`	V:149, I:455	432	, ,	System-V-Druckerbefehle für CUPS: lp(1), lpstat(1), lpoptions(1), cancel(1), lpmove(8), lpinfo(8), lpadmin(8), …
`cups-bsd`	V:33, I:178	131	, ,	BSD-Druckbefehle für CUPS: lpr(1), lpq(1), lprm(1), lpc(8)
`printer-driver-gutenprint`	V:10, I:49	1117	Nicht anwendbar	Druckertreiber für CUPS

Tabelle 6.11: Liste von Print-Servern und Hilfsprogrammen

Tipp
Sie können das CUPS-System konfigurieren, indem Sie in Ihrem Browser ”http://localhost:631/” eingeben.

6.5 Weitere Netzwerkanwendungs-Server

Hier einige weitere Netzwerkanwendungs-Server:

Das Common Internet File System - Protokoll (CIFS) ist das gleiche Protokoll wie Server Message Block (SMB) und wird von Microsoft Windows ausgiebig genutzt.

Tipp
In Abschnitt 4.5.2 finden Sie Informationen zur Integration von Server-Systemen.

Tipp
Die Rechnernamenauflösung wird normalerweise über den DNS-Server realisiert. Für IP-Adressen, die dynamisch über einen DHCP-Server zugewiesen werden, kann Dynamic DNS eingerichtet werden; dazu können `bind9` und `kea` genutzt werden, wie auf der DDNS-Seite im Debian-Wiki beschrieben.

Paket	Popcon	Größe	Protokoll	Beschreibung
inetutils-telnetd	V:0.2, I:1.3	200	TELNET	TELNET-Server
nfs-kernel-server	V:46, I:53	886	NFS	Datei-Netzwerkfreigabe unter Unix
samba	V:112, I:123	5190	SMB	Datei- und Drucker-Netzwerkfreigabe unter Windows
netatalk	V:0.57, I:0.79	933	ATP	Datei- und Drucker-Netzwerkfreigabe unter Apple/Mac (AppleTalk)
proftpd-core	V:7.6, I:8.4	5041	FTP	Grundlegender Datei-Download
apache2	V:168, I:205	590	HTTP	Grundlegender Web-Server
squid	V:8.8, I:9.3	9585	, ,	Grundlegender Web-Proxy-Server
bind9	V:32, I:35	905	DNS	IP-Adresse für andere Rechner
kea	I:0.51	245	DHCP	IP-Adresse des Clients selbst

Tabelle 6.12: Liste von weiteren Netzwerkanwendungs-Servern

Tipp
Die Verwendung eines Proxy-Servers wie `squid` ist viel effizienter, um Bandbreite zu sparen, als ein lokaler Archiv-Spiegel mit dem vollständigen Inhalt des Debian-Archivs.

6.6 Weitere Netzwerkanwendungs-Clients

Hier einige weitere Netzwerkanwendungs-Clients:

6.7 Diagnose von System-Daemons

Das `telnet`-Programm ermöglicht die manuelle Verbindung zu den System-Daemons und ihren Diagnosefunktionen.

Um einen reinen POP3-Dienst zu testen, probieren Sie folgendes:

```
$ telnet mail.ispname.net pop3
```

Um einen POP3-Dienst mit aktiviertem TLS/SSL (wie sie bei manchen Providern vorkommen) zu testen, benötigen Sie einen `telnet`-Client mit aktiviertem TLS/SSL, z.B. aus dem `telnet-ssl`- oder `openssl`-Paket:

```
$ telnet -z ssl pop.gmail.com 995
```

```
$ openssl s_client -connect pop.gmail.com:995
```

Die folgenden RFCs enthalten das zur Diagnose erforderliche Wissen für jeden System-Daemon.

Die Verwendung der Ports ist in "`/etc/services`" beschrieben.

Paket	Popcon	Größe	Protokoll	Beschreibung
netcat-traditional	V:61, I:905	139	TCP/IP	TCP/IP-Alleskönner
netcat-openbsd	V:23, I:118	105	TCP/IP	TCP/IP-Alleskönner mit Unterstützung für IPv6, Proxys und Unix-Sockets
openssl	V:853, I:996	2564	SSL	Secure-Socket-Layer-(SSL-)Binärdatei und dazugehörige Kryptografie-Werkzeuge
stunnel4	V:7, I:10	24	, ,	universeller SSL-Wrapper
inetutils-telnet	V:48, I:778	282	TELNET	TELNET-Client
nfs-common	V:151, I:188	1259	NFS	Datei-Netzwerkfreigabe unter Unix
smbclient	V:29, I:204	2117	SMB	Datei- und Druckerfreigabe-Client für MS Windows
cifs-utils	V:34, I:117	369	, ,	Befehle zum Einbinden und Trennen von fern abgelegten MS-Windows-Dateien
wget	V:186, I:982	3784	HTTP und FTP	Programm zum Herunterladen von Dateien aus dem Web
curl	V:296, I:739	526	, ,	, ,
transmission-gtk	V:10, I:135	6639	BitTorrent	BitTorrent-Client (GTK)
transmission-qt	V:0.6, I:2.5	6522	, ,	BitTorrent-Client (Qt)
ktorrent	V:1.3, I:4.8	5031	, ,	BitTorrent-Client (Qt)
qbittorrent	V:8, I:23	16666	, ,	BitTorrent-Client (Qt)
bind9-host	V:143, I:940	137	DNS	host(1) von bind9, "Priorität: standard"
bind9-dnsutils	V:116, I:836	521	, ,	dig(1) von bind, "Priorität: standard"
ldap-utils	V:11, I:56	791	LDAP	Daten von einem LDAP-Server beziehen

Tabelle 6.13: Liste von Netzwerkanwendungs-Clients

RFC	Beschreibung
rfc1939 und rfc2449	POP3-Dienst
rfc3501	IMAP4-Dienst
rfc2821 (rfc821)	SMTP-Dienst
rfc2822 (rfc822)	Mail-Datei-Format
rfc2045	Multipurpose Internet Mail Extensions (MIME)
rfc819	DNS-Dienst
rfc2616	HTTP-Dienst
rfc2396	URI-Definition

Tabelle 6.14: Liste populärer RFCs

Kapitel 7

GUI-System

7.1 GUI-Arbeitsplatzumgebung

Im Debian-System stehen mehrere voll ausgestattete grafische GUI-Arbeitsplatzumgebungen zur Auswahl:

Programmgruppen-Paket	Popcon	Größe	Beschreibung
task-gnome-desktop	I:191	9	GNOME-Arbeitsplatzumgebung
task-xfce-desktop	I:88	9	Xfce-Arbeitsplatzumgebung
task-kde-desktop	I:93	6	KDE Plasma-Arbeitsplatzumgebung
task-mate-desktop	I:30	9	MATE-Arbeitsplatzumgebung
task-cinnamon-desktop	I:35	9	Cinnamon-Arbeitsplatzumgebung
task-lxde-desktop	I:18	9	LXDE-Arbeitsplatzumgebung
task-lxqt-desktop	I:16	9	LXQt-Arbeitsplatzumgebung
task-gnome-flashback-desktop	I:9.4	6	GNOME Flashback-Arbeitsplatzumgebung

Tabelle 7.1: Liste der Arbeitsplatzumgebungen

Tipp
Paketabhängigkeiten, die durch ein Programmgruppen-Metapaket (task-*) ausgewählt wurden, könnten nicht mehr synchron sein zu den letzten Versionsübergängen in Debian `Unstable`/`Testing`. Für `task-gnome-desktop` beispielsweise müssen Sie eventuell die Paketauswahl wie folgt anpassen:

- Starten Sie aptitude(8) mit `sudo aptitude -u`.
- Wählen Sie ”Tasks” aus und drücken Sie ”Enter”.
- Wählen Sie ”Endbenutzer” aus und drücken Sie ”Enter”.
- Wählen Sie ”GNOME” aus und drücken Sie ”Enter”.
- Wählen Sie `task-gnome-desktop` aus und drücken Sie ”Enter”.
- Wählen Sie ”Hängt ab von” aus und drücken Sie ”m” (manuell installiert).
- Wählen Sie ”Empfiehlt” aus und drücken Sie ”m” (manuell installiert).
- Wählen Sie ”`task-gnome-desktop`” aus und drücken Sie ”-” (entfernen).
- Anpassen ausgewählter Pakete bei gleichzeitigem Entfernen von problematischen Paketen, die Abhängigkeitskonflikte auslösen
- Drücken Sie ”g”, um die Installation/Aktualisierung zu starten.

Dieser Abschnitt behandelt die Standard-Arbeitsplatzumgebung von Debian: `task-gnome-desktop`, die GNOME unter wayland bereitstellt.

7.2 GUI-Kommunikationsprotokoll

Beim GNOME-Desktop verfügbare GUI-Kommunikationsprotokolle:

- Wayland (Display-Server-Protokoll) (nativ)
- X-Window-System Core-Protokoll (über `xwayland`)

Bitte besuchen Sie die freedesktop.org-Seite für Infos zu Unterschieden zwischen Wayland und der X-Window-Architektur.

Aus Sicht des Benutzers können die Unterschiede wie folgt zusammengefasst werden:

- Wayland ist ein rechner-internes GUI-Kommunikationsprotokoll: neu, einfacher, schneller, keine setuid-root-Binärdatei;
- X-Window ist ein netzwerkfähiges GUI-Kommunikationsprotokoll: traditionell, komplex, langsamer, hat eine setuid-root-Binärdatei.

Betreffend Anwendungen, die das Wayland-Protokoll verwenden: der Zugriff auf deren Bildschirminhalte von einem fernen Rechner wird unterstützt durch VNC und RDP. Siehe Abschnitt 7.8.

Moderne X-Server enthalten die MIT Shared Memory Extension und kommunizieren über gemeinsam genutzten Speicher (shared memory) mit den lokalen X-Clients. So wird der netzwerk-transparente Xlib-Interprozess-Kommunikationskanal umgangen und die Performance erhöht. Diese Situation war der Hintergrund für die Erstellung von Wayland als ausschließlich lokales GUI-Kommunikationsprotokoll.

Über das `xeyes`-Programm - vom GNOME-Terminal aus gestartet - können Sie das GUI-Kommunikationsprotokoll für jede GUI-Anwendung getrennt kontrollieren.

```
$ xeyes
```

- Wenn der Mauscursor auf einer Anwendung steht, die das Wayland Display-Server-Protokoll verwendet - wie z.B. das ”GNOME-Terminal”, bewegen sich die Augen nicht zusammen mit dem Mauscursor.
- Befindet sich der Mauscursor auf einer Anwendung, die das X-Window-System Core-Protokoll nutzt - wie ”xterm”, bewegen sich die Augen synchron zum Mauscursor; hier zeigt sich die nicht-isolierte Natur der X-Window-Architektur.

Mit Stand April 2021 wurden viele populäre GUI-Anwendungen wie GNOME- und LibreOffice (LO)-Anwendungen auf das Wayland Display-Server-Protokoll migriert. Soweit ich das sehe, verwenden z.B. `xterm`, `gitk`, `chromium`, `firefox`, `gimp`, `dia` und KDE-Anwendungen noch das X-Window-System Core-Protokoll.

Anmerkung
Sowohl für xwayland unter Wayland wie auch für das native X-Window-System sollte die alte X-Server-Konfigurationsdatei ”`/etc/X11/xorg.conf`” nicht auf dem System vorhanden sein. Die Grafikausgabe- und Eingabegeräte werden jetzt vom Kernel über DRM, KMS und udev konfiguriert. Der native X-Server wurde entsprechend angepasst, um diese zu verwenden. Weiteres dazu unter ”modedb default video mode support” in der Dokumentation des Linux-Kernels.

7.3 GUI-Infrastruktur

Hier einige erwähnenswerte GUI-Infrastruktur-Pakete für die GNOME-Umgebung unter Wayland:

Paket	Popcon	Paketgröße	Beschreibung
`mutter`	V:1, I:21	218	GNOMEs Fenster-Manager mutter **[auto]**
`xwayland`	V:256, I:332	2653	Ein X-Server, der auf wayland läuft **[auto]**
`gnome-remote-desktop`	V:154, I:239	2743	Remote-Desktop-Daemon für GNOME, der PipeWire verwendet **[auto]**
`gnome-tweaks`	V:17, I:230	1145	Fortgeschrittene Konfigurationsoptionen für GNOME
`gnome-shell-extension-prefs`	V:7, I:109	33	Werkzeug, um die GNOME-Shell-Extensions zu aktivieren/deaktivieren

Tabelle 7.2: Liste erwähnenswerter GUI-Infrastruktur-Pakete

Hierbei bedeutet ”**[auto]**”, dass diese Pakete automatisch installiert werden, wenn Sie bei der Installation die Programmgruppe `task-gnome-desktop` auswählen.

Tipp
`gnome-tweaks` ist das unverzichtbare Konfigurationswerkzeug für GNOME. Zum Beispiel:

- können Sie eine ”Über-Verstärkung” der Sound-Lautstärke erzwingen (unter ”Allgemein”);
- können Sie erzwingen, dass die Feststelltaste (”Caps-Lock”) zu ”Esc” wird (unter ”Tastatur & Maus” -> ”Tastatur” -> ”Zusätzliche Layout-Optionen”).

Tipp
Detail-Funktionen der GNOME-Desktop-Umgebung können mit Hilfsprogrammen konfiguriert werden, die Sie erreichen, indem Sie nach dem Drücken der `Super`-Taste ”`settings`”, ”`tweaks`” oder ”`extensions`” eingeben.

7.4 GUI-Anwendungen

Viele nützliche GUI-Anwendungen sind jetzt in Debian verfügbar. Die Installation von Paketen wie `scribus` (aus KDE) innerhalb der GNOME-Arbeitsplatzumgebung ist absolut akzeptabel, da eine entsprechende Funktionalität in GNOME nicht enthalten ist. Allerdings könnte die Installation von vielen Paketen mit identischer Funktion Ihr System aufblähen.

Hier eine Liste von GUI-Anwendungen, die mir ins Auge gestochen sind:

7.5 Benutzerverzeichnisse

Die Standardnamen für Benutzerverzeichnisse wie "`~/Desktop`", "`~/Documents`", ... sind innerhalb der Desktop-Umgebung abhängig von der Locale, die für die Systeminstallation verwendet wurde, können also in eine andere Sprache als Englisch übersetzt sein. Sie können sie auf Englisch zurücksetzen mit:

```
$ LANGUAGE=C xdg-user-dirs-update --force
```

Anschließend verschieben Sie alle Daten in die neuen Verzeichnisse. Siehe dazu xdg-user-dirs-update(1).

Sie können auch frei wählbare eigene Namen vergeben, indem Sie "`~/.config/user-dirs.dirs`" editieren. Näheres dazu in user-dirs.dirs(5).

7.6 Schriften

Viele nützliche skalierbare Schriftarten sind für Debian-Nutzer verfügbar. Anliegen der Benutzer hierbei sind, wie sie Redundanzen (doppelt vorhandene Schriften) vermeiden können und wie sie Teile von installierten Schriftarten deaktivieren können. Ansonsten könnten unnötigerweise verfügbare Schriften die Menüs der GUI-Anwendungen aufblähen.

Das Debian-System verwendet die FreeType 2.0-Bibliothek zur Rasterung vieler skalierbarer Schriftformate für Bildschirmanzeige und Druck:

- Type 1 (PostScript)-Schriften, die kubische Bézierkurven verwenden (nahezu veraltetes Format);
- TrueType-Schriften, die quadratische Bézierkurven verwenden (gute Wahl);
- OpenType-Schriften, die kubische Bézierkurven verwenden (beste Wahl).

7.6.1 Basis-Schriftarten

Die folgende Tabelle wurde erstellt in der Hoffnung, dass sie Nutzern hilft, passende skalierbare Schriftarten auszuwählen durch ein klares Verständnis der metrischen Kompatibilität und der Glyphenabdeckung. Die meisten Schriftenarten decken alle lateinischen, griechischen und kyrillischen Zeichen ab. Auch ästhetische Gründe könnten Einfluß darauf haben, welche Schriften Sie letztlich aktivieren. Diese Schriften können sowohl für die Bildschirmanzeige wie auch für den Druck genutzt werden.

Hierbei gilt:

- steht "MCM" für "metrisch kompatibel mit durch Microsoft bereitgestellte Schriften";
- steht "MCMATC" für "metrisch kompatibel mit durch Microsoft bereitgestellte Schriften: Arial, Times New Roman, Courier New";
- steht "MCAHTC" für "metrisch kompatibel mit durch Adobe bereitgestellte Schriften: Helvetica, Times, Courier";

Paket	Popcon	Paketgröße	GUI	Beschreibung
evolution	V:27, I:227	521	GNOME	persönliches Informationsmanagement (Groupware und E-Mail)
thunderbird	V:39, I:100	304278	GTK	E-Mail-Client (Mozilla Thunderbird)
kontact	V:1.0, I:9.7	2416	KDE	persönliches Informationsmanagement (Groupware und E-Mail)
libreoffice-writer	V:92, I:417	34243	LO	Schreibprogramm
abiword	V:0.8, I:4.5	3596	GNOME	Schreibprogramm
calligrawords	V:0.3, I:2.6	7144	KDE	Schreibprogramm
scribus	V:1, I:12	32942	KDE	Desktop-Publishing-Editor zum Ändern von PDF-Dateien
glabels	V:0.4, I:2.4	1295	GNOME	Aufkleber-Editor
libreoffice-calc	V:89, I:414	29622	LO	Tabellenkalkulation
gnumeric	V:2.8, I:9.7	10053	GNOME	Tabellenkalkulation
calligrasheets	V:0.2, I:1.6	14014	KDE	Tabellenkalkulation
libreoffice-impress	V:78, I:412	2501	LO	Präsentation
calligrastage	V:0.1, I:1.6	6237	KDE	Präsentation
libreoffice-base	V:18, I:65	5136	LO	Datenbank-Verwaltung
kexi	V:0.04, I:0.73	7565	KDE	Datenbank-Verwaltung
libreoffice-draw	V:78, I:412	11424	LO	Vektorgrafik-Editor
inkscape	V:12, I:73	113357	GNOME	Vektorgrafik-Editor
karbon	V:0.2, I:2.0	4029	KDE	Vektorgrafik-Editor
dia	V:1, I:16	3846	GTK	Ablaufschema- und Diagramm-Editor
gimp	V:40, I:206	33386	GTK	Bitmapgrafik-Editor
shotwell	V:15, I:244	6334	GTK	Digitalfoto-Verwaltung
digikam	V:1.5, I:8.2	333	KDE	Digitalfoto-Verwaltung
darktable	V:3, I:10	36092	GTK	Leuchttisch und Dunkelkammer für Photographen
gnome-video-trimmer	V:0.16, I:0.97	1665	GNOME	verlustfreies Schneiden von Videodateien
kdenlive	V:3, I:21	22233	KDE	nicht-linearer Video-Editor
shotcut	V:1.2, I:7.5	8488	Qt-Appl.	nicht-linearer Video-Editor
openshot-qt	V:0.8, I:8.3	196007	Qt-Appl.	nicht-linearer Video-Editor
flowblade	V:0.2, I:2.1	42213	GTK	nicht-linearer Video-Editor
planner	V:0.2, I:4.3	1400	GNOME	Projektverwaltung
calligraplan	V:0.1, I:2.9	23545	KDE	Projektverwaltung
gnucash	V:2.0, I:6.4	29447	GNOME	Finanzverwaltung/Homebanking-Programm
homebank	V:0.3, I:1.5	3277	GTK	Finanzverwaltung/Homebanking-Programm
kmymoney	V:0.4, I:1.8	19313	KDE	Finanzverwaltung/Homebanking-Programm
frescobaldi	V:0.3, I:1.4	14150	Qt-Appl.	LilyPond Notenblatt-Texteditor
librecad	V:1, I:13	12361	Qt-Appl.	System für computerunterstützte Konstruktion (CAD, 2D)
freecad	V:1, I:20	112	Qt-Appl.	System für computerunterstützte Konstruktion (CAD, 3D)
kicad	V:3, I:14	226336	GTK	Software zum Entwurf von Schaltplänen und Platinen
xsane	V:9, I:127	1512	GTK	Scanner-Frontend
libreoffice-math	V:73, I:415	1980	LO	mathematischer Gleichungs-/Formel-Editor
calibre	V:7, I:24	76289	KDE	E-Book-Konvertierer und Bibliotheksverwaltung
fbreader	V:0.7, I:5.2	4204	GTK	E-Book-Reader
evince	V:76, I:280	953	GNOME	Dokumentenbetrachter (pdf)
okular	V:43, I:128	4544	KDE	Dokumentenbetrachter (pdf)
x11-apps	V:32, I:440	2461	reine X-Appl.	xeyes(1), usw.
x11-utils	V:233, I:553	651	reine X-Appl.	xev(1), xwininfo(1), usw.

Paket	Popcon	Größe	sans	serif	mono	Hinweise zur Schriftart
fonts-cantarell	V:173, I:284	224	59	-	-	Cantarell (GNOME 3; Bildschirmanzeige)
fonts-noto	I:146	30	61	63	40	Noto-Schriften (Google; mehrsprachig inkl. CJK)
fonts-dejavu	I:381	34	58	68	40	DejaVu (GNOME 2; MCM:Verdana, erweiterte Bitstream Vera)
fonts-liberation2	V:36, I:143	15	56	60	40	Liberation-Schriften für LibreOffice (Red Hat; MCMATC)
fonts-croscore	V:19, I:35	5260	56	60	40	Chrome OS: Arimo, Tinos und Cousine (Google; MCMATC)
fonts-crosextra-carlito	V:21, I:88	2696	57	-	-	Chrome OS: Carlito (Google; MCM:Calibri)
fonts-crosextra-caladea	V:13, I:81	347	-	55	-	Chrome OS: Caladea (Google; MCM:Cambria) (nur Latein)
fonts-freefont-ttf	V:77, I:200	14461	57	59	40	GNU FreeFont (erweiterte URW Nimbus)
fonts-quicksand	V:210, I:450	392	56	-	-	Debian task-desktop, Quicksand (Bildschirmanzeige, nur Latein)
fonts-hack	V:33, I:135	2509	-	-	40 P	Eine Schriftart, die für Quellcode-Hack entworfen wurde (Facebook)
fonts-sil-gentiumplus	V:1, I:27	14345	-	54	-	Gentium SIL
fonts-sil-charis	V:1, I:27	6704	-	59	-	Charis SIL
fonts-urw-base35	V:187, I:533	15558	56	60	40	URW Nimbus (Nimbus Sans, Roman No. 9 L, Mono L, MCAHTC)
fonts-ubuntu	V:2.1, I:4.4	4312	58	-	33 P	Ubuntu-Schriften (Bildschirmanzeige)
fonts-terminus	V:0.3, I:3.8	451	-	-	33	Cool retro terminal fonts
ttf-mscorefonts-installer	V:1, I:40	85	56?	60	40	Downloader für nicht-freie Microsoft-Schriften (siehe unten)

Tabelle 7.4: Liste erwähnenswerter TrueType- und OpenType-Schriftarten

- stehen Zahlen in den Schrifttyp-Spalten für die ungefähre Weite von ”M” der gleichen Punktgrößen-Schrift;
- steht ”P” in einer Mono-Schrifttyp-Spalte für die Eignung in der Programmierung (mit einem klaren Unterschied bei ”0”/”O” und ”1”/”l”/”I”).
- Das `ttf-mscorefonts-installer`-Paket lädt Microsofts ”Core fonts for the Web” herunter und installiert Arial, Times New Roman, Courier New, Verdana, ... Diese Schriftarten sind nicht-freie Daten.

Viele freie lateinische Schriftarten haben eine Abstammungslinie zur URW Nimbus-Familie oder zu Bitstream Vera.

Tipp
Wenn Ihr Gebietsschema Schriften erfordert, die von den oben genannten Schriften nicht gut abgedeckt werden, verwenden Sie bitte aptitude, um nach Tasks-Paketen zu suchen (unter ”Tasks” -> ”Lokalisierung”). Die Schriftpakete, die in dem zu Ihrem Gebietsschema gehörigen Lokalisierungs-Task zu finden sind (unter ”Hängt ab von” oder ”Empfiehlt”), sind die primär für Sie interessanten Kandidaten.

7.6.2 Schriftenrasterung

Debian verwendet FreeType für die Rasterung von Schriften. Deren Infrastruktur zur Schriftenauswahl wird von der Schriften-Konfigurationsbibliothek Fontconfig bereitgestellt.

Paket	Popcon	Größe	Beschreibung
`libfreetype6`	V:582, I:996	1098	FreeType-Bibliothek zur Schriftenrasterung
`libfontconfig1`	V:580, I:801	344	Fontconfig - Schriften-Konfigurationsbibliothek
`fontconfig`	V:477, I:688	415	`fc-*`: Konsolen-Befehle für Fontconfig
`font-manager`	V:2.2, I:6.3	1139	Font Manager: GUI-Programm für Fontconfig
`nautilus-font-manager`	V:0.09, I:0.28	43	Nautilus-Erweiterung für Font Manager

Tabelle 7.5: Liste erwähnenswerter Pakete für Schriftumgebungen

Tipp
Einige Schriftpakete wie `fonts-noto*` installieren zu viele Schriftarten. Auch möchten Sie vielleicht einige Schriftpakete installiert haben, aber die zugehörigen Schriftarten sollen normalerweise trotzdem nicht aktiv sein. Doppelte Glyphen sind für einige Unicode-Codepoints aufgrund der Han-Vereinheitlichung zu erwarten, und durch eine unkonfigurierte Fontconfig-Bibliothek könnte es dazu kommen, dass unerwünschte Glyphen ausgewählt werden. Sehr nervige Fälle sind dabei ”U+3001 IDEOGRAPHIC COMMA” und ”U+3002 IDEOGRAPHIC FULL STOP” in CJK-Ländern. Sie können solche Problematiken ganz einfach vermeiden, indem Sie die Verfügbarkeit von Schriftarten über den grafischen Schriften-Manager font-manager konfigurieren.

Der Schriftarten-Konfigurationsstatus kann auch über die Befehlszeile abgefragt werden:

- ”`fc-match(1)`” zeigt, welche fontconfig-Schriften Standardeinstellung sind;
- ”`fc-list(1)`” zeigt verfügbare fontconfig-Schriften.

Sie können den Schriftarten-Konfigurationsstatus auch über einen Texteditor konfigurieren, aber dies ist nicht trivial. Lesen Sie dazu fonts.conf(5).

7.7 Sandbox

Viele (meist grafische) Anwendungen unter Linux sind auch in binärer Form über Debian-fremde Quellen verfügbar:

- AppImage -- Linux apps that run anywhere
- FLATHUB -- Apps for Linux, right here
- snapcraft -- The app store for Linux



Warnung
Binärdateien aus diesen Quellen könnten proprietäre, nicht-freie Software-Pakete enthalten.

Es gibt eine Existenzberechtigung für diese Binärformat-Distributionen auch unter Liebhabern Freier Software, die Debian nutzen, da auf diesem Wege ein sauberer Satz von Bibliotheken für jede Anwendung vom jeweiligen Upstream-Entwickler bereitgestellt werden kann, unabhängig von den ansonsten in Debian verfügbaren Versionen.

Das dem Nutzen von externen Binärdateien anhängende Risiko kann reduziert werden, indem eine Sandbox-Umgebung genutzt wird, die moderne Linux-Sicherheitsfunktionalitäten zum Einsatz bringt (siehe Abschnitt 4.7.5):

- Binärdateien von AppImage und einigen weiteren Upstream-Sites sollten Sie in firejail laufen lassen (bei manueller Konfiguration).
- Binärdateien von FLATHUB sollten Sie in Flatpak laufen lassen. (Keine manuelle Konfiguration erforderlich.)
- Binärdateien von Snapcraft sollten Sie in Snap laufen lassen. (Keine manuelle Konfiguration erforderlich. Kompatibel mit Daemon-Programmen.)

Das `xdg-desktop-portal`-Paket bietet eine standardisierte API-Schnittstelle für allgemeine Desktop-Funktionalitäten. Näheres unter xdg-desktop-portal (Flatpak) und xdg-desktop-portal (Snap) .

Paket	Popcon	Größe	Beschreibung
`flatpak`	V:108, I:116	9297	Rahmenwerk zur Verteilung von Desktop-Anwendungen basierend auf Flatpak
`gnome-software-plugin-flatpak`	V:28, I:39	291	Flatpak-Unterstützung für GNOME Software
`snapd`	V:56, I:59	62738	Daemon und Werkzeuge, um Snap-Pakete zu aktivieren
`gnome-software-plugin-snap`	V:1.4, I:2.0	144	Snap-Support für GNOME Software
`xdg-desktop-portal`	V:376, I:438	2291	Desktop-Integrations-Portal für Flatpak und Snap
`xdg-desktop-portal-gtk`	V:340, I:436	715	xdg-desktop-portal-Backend für gtk (GNOME)
`xdg-desktop-portal-kde`	V:81, I:111	3333	xdg-desktop-portal-Backend für Qt (KDE)
`xdg-desktop-portal-wlr`	V:2.4, I:6.7	167	xdg-desktop-portal-Backend für wlroots (Wayland)
`firejail`	V:1.1, I:4.2	1725	das Sicherheits-Sandbox-Programm firejail zur Nutzung mit AppImage

Tabelle 7.6: Liste erwähnenswerter Pakete für Sandbox-Umgebungen

Diese Sandbox-Technologie ähnelt sehr den Apps auf Smartphone-Betriebssystemen, wo Apps mit eingeschränktem Ressourcenzugriff ausgeführt werden.

Einige große GUI-Anwendungen in Debian wie Webbrowser nutzen intern ebenfalls die Sandbox-Technologie, um sicherer zu werden.

7.8 Arbeitsplatz-Fernzugriff (Remote Desktop)

Paket	Popcon	Größe	Protokolle	Beschreibung
gnome-remote-desktop	V:154, I:239	2743	RDP	GNOME Remote Desktop-Server
xrdp	V:29, I:31	4757	RDP	xrdp, Remote-Desktop-Protocol (RDP) Server
x11vnc	V:10, I:48	1863	RFB (VNC)	x11vnc, Remote-Framebuffer-Protocol (VNC) Server
tigervnc-standalone-server	V:6, I:15	3176	RFB (VNC)	TigerVNC, Remote-Framebuffer-Protocol (VNC) Server
gnome-connections	V:7, I:154	1726	RDP, RFB (VNC)	GNOME Remote-Desktop-Client
remmina	V:16, I:59	1046	RDP, RFB (VNC), SPICE, SSH, ...	Remmina: GTK-Client für Arbeitsplatz-Fernzugriff
krdc	V:2, I:15	4144	RDP, RFB (VNC)	KRDC: KDE-Client für Arbeitsplatz-Fernzugriff
virt-viewer	V:5, I:36	1278	RFB (VNC), SPICE	Virtual Machine Manager: grafischer Display-Client von Gastbetriebssystemen

Tabelle 7.7: Liste erwähnenswerter Server für Fernzugriff

7.9 X-Server-Verbindungen

Es gibt verschiedene Wege, um eine Anwendung auf einem fernen Rechner mit dem X-Server zu verbinden; dazu gehört auch `xwayland` auf dem lokalen Rechner.

Paket	Popcon	Größe	Befehl	Beschreibung
openssh-server	V:781, I:824	3615	`sshd` mit der Option `X11-forwarding`	SSH-Server (sicher)
openssh-client	V:709, I:997	3611	`ssh -X`	SSH-Client (sicher)
xauth	V:197, I:975	81	`xauth`	Werkzeug für die X-authority-Datei
x11-xserver-utils	V:376, I:532	559	`xhost`	Server-Zugriffskontrolle für X

Tabelle 7.8: Liste der Verbindungsmethoden zum X-Server

7.9.1 Lokale Verbindung zum X-Server

Ein Zugriff auf den lokalen X-Server durch lokale Anwendungen, die das X-Core-Protokoll nutzen, kann über einen lokalen UNIX-Domain-Socket realisiert werden. Die Authentifizierung hierfür kann über die authority-Datei mittels Access-Cookies erfolgen. Der Speicherort der authority-Datei wird über die Umgebungsvariable "`$XAUTHORITY`" definiert, und das X-Display über die "`$DISPLAY`"-Umgebungsvariable. Da diese normalerweise automatisch gesetzt werden, ist keine spezielle Aktion dafür nötig; für "`gitk`" starten Sie z.B. einfach:

```
username $ gitk
```

Anmerkung
Für `xwayland` enthält `XAUTHORITY` einen Wert wie "`/run/user/1000/.mutter-Xwaylandauth.YVSU30`".

7.9.2 Fernverbindung auf den X-Server

Der Zugriff von fern auf den lokalen X-Server mittels einer Anwendung, die das X-Core-Protokoll verwendet, wird über die Nutzung der X11-forwarding-Funktionalität unterstützt:

- Öffnen Sie `gnome-terminal` auf dem lokalen Rechner.
- Führen Sie ssh(1) mit der Option `-X` wie folgt aus, um eine Verbindung mit der fernen Seite aufzubauen:

```
localname @ localhost $ ssh -q -X loginname@remotehost.domain
Password:
```

- Starten Sie auf der fernen Seite einen X-Anwendungsbefehl, z.B. "`gitk`":

```
loginname @ remotehost $ gitk
```

Mit dieser Methode kann die Anzeige eines fernen X-Clients dargestellt werden, als wäre sie lokal über einen UNIX-Domain-Socket verbunden.

Lesen Sie für Informationen zu SSH/SSHD Abschnitt 6.3.



Warnung
Eine TCP/IP-Verbindung zu einem fernen X-Server ist in Debian aus Sicherheitsgründen deaktiviert. Wenn Sie es vermeiden können, sollten Sie dies auch nicht aktivieren (z.B. indem Sie einfach "`xhost +`" setzen oder XDMCP-Verbindungen gestatten).

7.9.3 Chroot-Verbindung zum X-Server

Der Zugriff auf den X-Server über Anwendungen, die das X-Core-Protokoll verwenden und auf dem gleichen Rechner laufen, jedoch in einer Umgebung (wie einem chroot), in der die authority-Datei nicht erreichbar ist, kann sicher über `xhost` authentifiziert werden, indem User-based access verwendet wird, z.B. wie hier für "`gitk`":

```
username $ xhost + si:localuser:root ; sudo chroot /path/to
# cd /src
# gitk
# exit
username $ xhost -
```

7.10 Zwischenablage

Um Text in die Zwischenablage zu kopieren, lesen Sie Abschnitt 1.4.4.

Um Grafiken in die Zwischenablage zu kopieren, gehen Sie zu Abschnitt 11.6.

Auch einige CLI-Befehle können die Zwischenablage beeinflussen:

Paket	Popcon	Paketgröße	Ziel	Beschreibung
xsel	V:8, I:42	55	X	Befehlszeilenprogramm für die X-Auswahl (Zwischenablage)
xclip	V:19, I:78	62	X	Befehlszeilenprogramm für die X-Auswahl (Zwischenablage)
wl-clipboard	V:11, I:28	174	Wayland	wl-copy / wl-paste: Befehlszeilenschnittstelle für die Wayland-Zwischenablage
gpm	V:8.0, I:8.7	524	Linux-Konsole	ein Daemon, der Mausereignisse auf der Linux-Konsole verarbeitet

Tabelle 7.9: Liste von Programmen zum Beeinflussen der Zwischenablage

Kapitel 8

I18N und L10N

Multilingualisation (M17N) oder native Sprachunterstützung (die Anpassung eines Software-Produkts zur Unterstützung mehrerer Sprachen) besteht aus zwei Schritten:

- Internationalisierung (I18N): eine Software so einrichten, dass sie grundsätzlich verschiedene Gebietsschemata (Locales) unterstützt.
- Lokalisierung (L10N): eine Software für die Unterstützung eines bestimmten Gebietsschemas anpassen.

Tipp
Bei den englischen Begriffen Multilingualization, Internationalization und Localization befinden sich 17, 18 bzw. 10 Buchstaben zwischen dem ersten und letzten Buchstaben (”m” und ”n”, ”i” und ”n” bzw. ”l” und ”n”), was M17N, I18N und L10N ergibt. Weitere Details finden Sie unter Internationalization and localization (englisch).

8.1 Das Gebietsschema (Locale)

Das Verhalten von Programmen, die Internationalisierung unterstützen, wird bestimmt durch die Umgebungsvariable ”$LANG”. Die Unterstützung von Locale-abhängigen Funktionalitäten erfordert die Installation von `locales`-Paketen oder dem Paket `locales-all`. Die `locales`-Pakete müssen korrekt eingerichtet sein.

Wenn weder irgendwelche `locales`-Pakete noch `locales-all` installiert sind, gibt es keine Locale-spezifischen Funktionalitäten und das System nutzt Meldungen in amerikanischem Englisch; Daten werden als **ASCII** behandelt. Dieses Verhalten ist dasselbe, wie wenn ”$LANG” auf ”LANG=”, ”LANG=C” oder ”LANG=POSIX” gesetzt ist.

Moderne Software wie GNOME oder KDE ist für die Unterstützung verschiedener Sprachen ausgerüstet. Sie ist mittels UTF-8 internationalisiert und über übersetzte Texte mittels gettext(1)-Infrastruktur lokalisiert. Übersetzte Texte werden teilweise als separate Lokalisierungspakete bereitgestellt.

Das aktuelle grafische Arbeitsplatzsystem in Debian setzt das Gebietsschema normalerweise in grafischen GUI-Umgebungen auf ”LANG=xx_YY.UTF-8”. Hierbei entspricht ”xx” dem Sprachcode gemäß ISO 639 und ”YY” dem Ländercode gemäß ISO 3166. Diese Werte werden über den Konfigurationsdialog der jeweiligen Arbeitsplatzumgebung gesetzt und haben entsprechenden Einfluß auf die Programme. Näheres dazu finden Sie in Abschnitt 1.5.2.

8.1.1 Argumentation für UTF-8-Gebietsschemata

Die einfachste Art von Textdaten ist **ASCII**, was passend ist für Englisch und weniger als 127 Zeichen enthält (welche in 7 Bits dargestellt werden können).

Sogar reiner Text in Englisch kann nicht-ASCII-Zeichen enthalten, so sind z.B. die leicht geschweiften rechten und linken Anführungszeichen in ASCII nicht enthalten:

```
b''“b''double quoted textb''”b'' is not "double quoted ASCII"
b''‘b''single quoted textb''’b'' is not 'single quoted ASCII'
```

Um mehr Zeichen zu unterstützen, wurden weitere Zeichensätze und Zeichenkodierungen entwickelt, um eine Vielzahl von Sprachen unterstützen zu können (siehe Tabelle 11.2).

Der Unicode-Zeichensatz kann nahezu alle dem Menschen bekannten Zeichen mit Codepunkten aus einem Bereich von 21 Bits (dies entspricht 0 bis 10FFFF in hexadezimaler Darstellung) abbilden.

Das Zeichenkodierungssystem UTF-8 passt Unicode-Codepunkte in einen 8 Bit breiten Datenstrom ein, der überwiegend zum ASCII-Datenverarbeitungssystem kompatibel ist. Das macht **UTF-8** zur modernen und bevorzugten Zeichenkodierung. **UTF** steht dabei für Unicode Transformation Format. Wenn reiner ASCII-Text in UTF-8 konvertiert wird, hat er exakt den gleichen Inhalt und die gleiche Größe wie der originale ASCII-Text. Sie verlieren also nichts, wenn Sie ein UTF-8-Gebietsschema nutzen.

Mit einem UTF-8-Gebietsschema und einer dazu kompatiblen Anwendung können Sie Textdaten in jeglicher fremden Sprache anzeigen und editieren, solange die entsprechenden Schriftarten und Eingabemethoden installiert und aktiviert sind. Wenn beispielsweise das Gebietsschema auf ”`LANG=fr_FR.UTF-8`” gesetzt ist, kann mit gedit(1) (dem Texteditor der GNOME Arbeitsplatzumgebung) chinesischer Text dargestellt und bearbeitet werden, während das Menü in Französisch angezeigt wird.

Tipp
Sowohl das neue (standardmäßige) ”`en_US.UTF-8`” wie auch das alte ”C”/”`POSIX`”-Gebietsschema nutzen US-Englisch für Meldungen, es gibt nur minimale Unterschiede bei der Sortierreihenfolge etc. Wenn Sie nicht nur ASCII-Zeichen anzeigen möchten, sondern auch UTF-8-kodierte Zeichen, und trotzdem das alte Verhalten der ”C”-Locale bevorzugen, verwenden Sie das Gebietsschema ”`C.UTF-8`” (kein Standard).

Anmerkung
Einige Programme verbrauchen mehr Speicher, wenn sie I18N unterstützen. Das kommt daher, weil sie für die Unicode-Unterstützung intern UTF-32(UCS4) verwenden (zwecks Geschwindigkeitsoptimierung; dabei werden 4 Bytes für jedes ASCII-Zeichen genutzt, unabhängig vom gewählten Gebietsschema). Nochmals: Sie verlieren nichts, wenn Sie ein UTF-8-Gebietsschema einsetzen.

8.1.2 Die Neukonfiguration des Gebietsschemas (Locale)

Damit das System ein bestimmtes Gebietsschema verwenden kann, müssen die entsprechenden Locale-Daten aus der Datenbank kompiliert werden.

Das `locales`-Paket enthält all die verfügbaren Locale-Daten **nicht** bereits vorkompiliert. Sie müssen Sie also konfigurieren mittels:

```
# dpkg-reconfigure locales
```

Dieser Prozess besteht aus zwei Schritten:

1. Auswahl der benötigten Locale-Daten, die in die binäre Form kompiliert werden sollen. (Stellen Sie bitte sicher, dass Sie mindestens ein UTF-8-Gebietsschema auswählen.)
2. Setzen des Werts für das systemweite Standard-Gebietsschema (mittels ”`/etc/default/locale`”); dies wird von PAM genutzt (Näheres in Abschnitt 4.5).

Der Wert für das systemweite Standard-Gebietsschema (in ”`/etc/default/locale`”) kann von der Konfiguration für grafische GUI-Anwendungen überschrieben werden.

Anmerkung
Das effektiv genutzte Kodierungssystem kann über "`/usr/share/i18n/SUPPORTED`" identifiziert werden. So ist "`LANG=en_US`" letztlich "`LANG=en_US.ISO-8859-1`".

Das Paket `locales-all` enthält alle Locale-Daten in vorkompilierter Form. Da es aber nicht "`/etc/default/locale`" erzeugt, müssen Sie eventuell trotzdem auch noch das `locales`-Paket installieren.

Tipp
Das `locales`-Paket einiger Debian-Derivate enthält vorkompilierte Locale-Daten für alle Locales. Auf einem Debian-System müssen Sie sowohl das `locales`- wie auch das `locales-all`-Paket installieren, um solch eine Systemumgebung zu emulieren.

8.1.3 Dateinamenkodierung

Für den Datenaustausch über verschiedene Plattformen hinweg (Näheres in Abschnitt 10.1.7) müssen Sie unter Umständen beim Einbinden einiger Dateisysteme bestimmte Kodierungen vorwählen. Zum Beispiel wird beim Einbinden eines vfat-Dateisystems durch mount(8) die CP437-Kodierung genutzt, wenn nichts anderes angegeben ist. Sie müssen eine explizite mount-Option angeben, wenn UTF-8 oder CP932 als Kodierung für die Dateinamen genutzt werden soll.

Anmerkung
Wenn ein während des Betrieb eingesteckter USB-Flash-Speicher in einer modernen Arbeitsplatzumgebung wie GNOME automatisch eingebunden wird, können Sie solche mount-Optionen eventuell durch einen Rechtsklick auf das entsprechende Icon auf der Arbeitsfläche auswählen. Klicken Sie auf die Lasche "Laufwerk", klappen Sie per Mausklick die Liste der "Einstellungen" auf und geben Sie dann "utf8" bei den "Mount-Optionen" ein. Wenn der USB-Speicher das nächste Mal eingesteckt wird, ist das Einbinden mit UTF-8 automatisch aktiviert.

Anmerkung
Beim Hochrüsten eines Systems oder wenn Sie Laufwerke von älteren nicht-UTF-8-Systemen verwenden, könnten Dateinamen mit nicht-ASCII-Zeichen in historischen und überholten Kodierungen wie ISO-8859-1 oder eucJP kodiert sein. Bitte suchen Sie die Hilfe von Textkonvertierungswerkzeugen, um diese in UTF-8 zu konvertieren. Details hierzu finden Sie in Abschnitt 11.1.

Samba verwendet für neuere Clients (Windows NT, 200x, XP und später) standardmäßig Unicode, aber bei älteren für DOS und Windows 9x/Me wird per Voreinstellung CP850 eingesetzt. Dieser Standard für ältere Clients kann mittels "`dos charset`" in der Datei "`/etc/samba/smb.conf`" geändert werden, CP932 zum Beispiel für Japanisch.

8.1.4 Lokalisierte Meldungen und übersetzte Dokumentation

Für viele Textmeldungen und Dokumente, die im Debian-System angezeigt werden, wie z.B. Fehlermeldungen, Standard-Programmausgaben, Menüs und Handbuchseiten, existieren Übersetzungen. Für die meisten Übersetzungsaktivitäten werden die GNU-gettext(1)-Werkzeuge als Backend-Programme im Hintergrund verwendet.

aptitude(8) bietet unter "Tasks" → "Lokalisierung" eine ausführliche Liste nützlicher Binärpakete, die lokalisierte Meldungen für Anwendungen und übersetzte Dokumentation enthalten.

So können Sie übersetzte Handbuchseiten (manpages) in Deutsch erhalten, indem Sie das `manpages-de`-Paket installieren. Um andererseits die italienische Handbuchseite für *programmname* aus "`/usr/share/man/it/`" zu lesen, führen Sie folgendes aus:

```
LANG=it_IT.UTF-8 man programname
```

GNU gettext kann über die Umgebungsvariable $LANGUAGE eine priorisierte Liste von Übersetzungssprachen erstellen. Nutzen Sie dafür zum Beispiel:

```
 $ export LANGUAGE="pt:pt_BR:es:it:fr"
```

Bezüglich weiterer Infos lesen Sie `info gettext` und dort speziell den Abschnitt ”The LANGUAGE variable”.

8.1.5 Auswirkungen des Gebietsschemas

Die Sortierreihenfolge der Zeichen bei Befehlen wie sort(1) und ls(1) ist abhängig von der Locale. Das Exportieren der Variablen `LANG=en_US.UTF-8` führt zur Reihenfolge wie in einem Lexikon (`A->a->B->b...->Z->z`), während `LANG=C.UTF-8` eine Reihenfolge in ASCII-Binärform bewirkt (`A->B->...->Z->a->b...`).

Das Gebietsschema wirkt sich auch auf das Datumsformat von ls(1) aus (Näheres in Abschnitt 9.3.4).

Das Datumsformat von date(1) ist abhängig von der Locale. Zum Beispiel:

```
 $ unset LC_ALL
 $ LANG=en_US.UTF-8 date
Thu Dec 24 08:30:00 PM JST 2023
 $ LANG=en_GB.UTF-8 date
Thu 24 Dec 20:30:10 JST 2023
 $ LANG=es_ES.UTF-8 date
jue 24 dic 2023 20:30:20 JST
 $ LC_TIME=en_DK.UTF-8 date
2023-12-24T20:30:30 JST
```

Die Zahlen-Interpunktion unterscheidet sich ebenfalls abhängig vom Gebietsschema. Zum Beispiel wird ein-tausend-komma-eins im englischen Gebietsschema ”`1,000.1`” geschrieben, im deutschen hingegen ”`1.000,1`”. Sie können den Unterschied in einem Tabellenkalkulationsprogramm sehen.

Jegliches Detail der ”`$LANG`”-Umgebungsvariable kann über das Setzen einer der ”`$LC_*`”-Variablen überschrieben werden. Diese wiederum können über die ”`$LC_ALL`”-Variable überschrieben werden. Details hierzu finden Sie in der locale(7)-Handbuchseite. Bitte versuchen Sie, dies alles zu vermeiden und nutzen Sie nur die Variable ”`$LANG`”, um eine der UTF-8-Locales zu setzen (außer Sie haben gute Gründe für die Erstellung einer komplizierten Konfiguration).

8.2 Die Tastatureingabe

Das Tastatursystem kann in verschiedenen Ebenen des Systems konfiguriert werden:

- Linux-Kernel: keyboard(5)
- X-Server: setxkbmap(1), xkeyboard-config(5), Umgebungsvariable `XMODIFIERS`
- Grafische Desktop-Umgebung: Eingabemethoden-Rahmenwerk: `ibus`, `fcitx5`
- Anwendung: Umgebungsvariablen, um die Eingabequelle festzulegen: `GTK_IM_MODULE`, `QT_IM_MODULE`, `QT_IM_MODUL` ...

Eingabemethoden-Rahmenwerk (IM), bestehend aus:

- Eingabemethoden-Engine (IME): die eigentliche Eingabenmethodik
- Konfiguration: erledigt die Konfiguration für IBus und andere Dienste wie IME-Plugins

- Panel/Leiste: Benutzerschnittstelle wie eine Sprachenleiste und Auswahltabellen

Mehrsprachige Eingaben für Anwendungen werden im groben wie hier dargestellt verarbeitet:

```
Keyboard        UI panel    Configuration          Application
|                  ^ |           |                       ^  ^
v                  | v           v                       |  |
Linux kernel -> Input method engine (IME) -+-> Gtk, Qt ---+  |
                   | ^                     |                 |
                   | |                     +-> X11, Wayland -+
                   v |
                 IME plugin (ibus-mozc, ...)
```

8.2.1 Tastatureingabe für die Linux-Konsole und X-Window

Das Debian-System kann mittels des Pakets `keyboard-configuration` so konfiguriert werden, dass es mit vielen internationalen Tastaturkonfigurationen funktioniert:

```
# dpkg-reconfigure keyboard-configuration
```

Für die Linux-Konsole und das X-Window-System werden damit Parameter in "`/etc/default/keyboard`" angepasst. Viele nicht-ASCII-Zeichen, inklusive akzentuierte Zeichen, die in europäischen Sprachen verwendet werden, sind über die Tottaste, AltGr-Taste und Compose-Taste zugänglich.

Anmerkung
Wenn ibus aktiv ist, könnte Ihre klassische X-Tastatur-Konfiguration (durchgeführt über `setxkbmap`) von `ibus` überschrieben werden, selbst wenn die klassische X-Arbeitsplatzumgebung läuft. Sie können `ibus` über `im-config` deaktivieren, indem Sie die Eingabemethode auf "Keine" einstellen. Weitere Informationen hierzu finden Sie im Debian-Wiki unter Keyboard.

8.2.2 Tastatureingabe für Wayland

Anders als das X-Window-Protkoll unterstützt das Wayland-Core-Protokoll nicht einmal die Eingabe von akzentuierten Zeichen. Populäre Wayland-Compositoren, wie Mutter für GNOME oder KWin für KDE, implementieren Erweiterungsprotokolle wie `text-input-unstable-v3` für die Texteingabe (hier eine Tabelle der "derzeitigen Wayland-Protokolle und deren Status").

Das `text-input-unstable-v3`-Protokoll funktioniert gut mit Eingabemethoden für Wayland (siehe auch "Wayland Eingabemethoden-Projekt post-mortem").

- Die meisten grafischen Anwendungen werden mit GUI-Bibliotheken wie GTK oder Qt gebaut, die `text-input-unstable` unterstützen.
- Populäre Eingabemethoden-Engines (IME), wie IBus oder Fcitx (Version 5), arbeiten gut mit `text-input-unstable-v3` zusammen.
- IMEs unterstützen die Texteingabe für viele Sprachen über Plugins.
- Aktuelle IMEs integrieren Funktionalitäten der "X Keyboard Extensions (XKB)" zur Eingabe von akzentuierten Zeichen in europäischen Sprachen unter Wayland, ähnlich wie `setxkbmap` dies früher für X Window tat.

8.2.3 Unterstützung für Eingabemethoden mit IBus

Für GNOME ist "`ibus`" die Standard-IME, und wird über ihre Paketabhängigkeiten automatisch installiert.

- Die meisten Funktionalitäten für multilinguale Tastatureingaben können über "GNOME Settings" oder "GNOME Tweaks" konfiguriert werden.
- Einige wenige Features für multilinguale Tastatureingaben müssen eventuell über den Befehl ibus-setup(1) konfiguriert werden.
- Die emoji-Eingabemethode ist verfügbar, indem Sie "SUPER-`.`" eingeben (gleichzeitig die Windows- und die Punkt-Taste drücken), gefolgt von einem Schlüsselwort für das jeweilige Emoji und der Leertaste.

Hier eine Liste der IBus- und zugehöriger Plugin-Pakete:

Paket	Popcon	Größe	unterstütztes Gebietsschema
ibus	V:211, I:252	1884	Eingabemethoden-Rahmenwerk, das dbus verwendet
ibus-mozc	V:2.0, I:3.7	872	Japanisch
ibus-anthy	V:0.45, I:0.98	8740	Japanisch
ibus-skk	V:0.04, I:0.15	242	Japanisch
ibus-kkc	V:0.04, I:0.18	206	Japanisch
ibus-libpinyin	V:1.0, I:4.7	125	Chinesisch (für zh_CN)
ibus-chewing	V:0.11, I:0.73	298	Chinesisch (für zh_TW)
ibus-libzhuyin	I:0.08	41021	Chinesisch (für zh_TW)
ibus-rime	V:0.31, I:0.50	76	Chinesisch (für zh_CN/zh_TW)
ibus-cangjie	V:0.01, I:0.09	235	Chinesisch (für zh_HK)
ibus-hangul	V:0.3, I:1.9	261	Koreanisch
ibus-libthai	V:0.00, I:0.05	84	Thailändisch
ibus-table-thai	I:0.05	59	Thailändisch
ibus-unikey	V:0.21, I:0.42	286	Vietnamesisch
keyman	V:0.01, I:0.09	507	Multilingual: Keyman-Plugin für mehr als 2000 Sprachen
ibus-table	V:0.06, I:0.85	2482	Tabellen-Plugin für IBus
ibus-m17n	V:0.2, I:1.7	373	Multilingual: Indisch, Arabisch und andere

Tabelle 8.1: Liste von IBus- und dessen Plugin-Paketen

8.2.4 Unterstützung für Eingabemethoden mit Fcitx

Das Eingabemethode-Rahmenwerk Fcitx (Version 5) ist beliebt bei chinesischen Benutzern und kompatibel mit "`ibus`".

Hier eine Liste der fcitx5- und zugehöriger Plugin-Pakete:

Paket	Popcon	Größe	unterstütztes Gebietsschema
fcitx5	V:8, I:12	712	Eingabemethoden-Rahmenwerk, kompatibel mit ”`ibus`”
fcitx5-mozc	V:1.1, I:1.8	1578	Japanisch
fcitx5-anthy	V:0.06, I:0.24	853	Japanisch
fcitx5-skk	V:0.08, I:0.16	371	Japanisch
fcitx5-kkc	V:0.00, I:0.07	418	Japanisch
fcitx5-chinese-addons	I:8.7	17	Chinesisch (Metapaket für zh_*)
fcitx5-pinyin	V:4.2, I:9.4	1000	Chinesisch (für zh_CN)
fcitx5-chewing	V:0.29, I:0.92	228	Chinesisch (für zh_TW)
fcitx5-zhuyin	I:0.07	108535	Chinesisch (für zh_TW)
fcitx5-rime	V:0.57, I:0.99	393	Chinesisch (für zh_CN/zh_TW)
fcitx5-table-cangjie-large	I:0.12	1292	Chinesisch (für zh_HK)
fcitx5-hangul	V:0.13, I:0.28	239	Koreanisch
fcitx5-libthai	I:0.04	119	Thailändisch
fcitx5-table-thai	I:0.07	34	Thailändisch
fcitx5-unikey	V:0.13, I:0.26	624	Vietnamesisch
fcitx5-m17n	V:0.41, I:0.85	171	Multilingual: Indisch, Arabisch und andere
fcitx5-table	V:0.3, I:9.2	428	Tabellen-Plugin für fcitx5
fcitx5-keyman	V:0.04, I:0.05	241	Multilingual: Keyman-Plugin für mehr als 2000 Sprachen

Tabelle 8.2: Liste von Fcitx5- und dessen Plugin-Paketen

8.2.5 Ein Beispiel für Japanisch

Ich finde die japanische Eingabemethode, gestartet in einer englischen Umgebung (”en_US.UTF-8”), sehr nützlich. Hier ein Beispiel, wie ich dies mit IBus eingerichtet habe:

1. Installieren Sie das japanische Eingabewerkzeug über `ibus-mozc` (oder `ibus-anthy`).
2. • Allgemein: ibus-setup(1) ausführen → ”Eingabemethode” auswählen → klicken auf ”Hinzufügen” → ”Japanisch” → ”Mozc (oder Anthy)” → Klicken auf ”Hinzufügen”
   • GNOME: ”Einstellungen” wählen → ”Tastatur” → ”Eingabequellen”, bei den ”Eingabequellen” klicken auf ”+” → ”Japanisch” → ”Mozc (oder Anthy)” → klicken auf ”Hinzufügen”
3. Sie können weitere Eingabequellen wählen.
4. Melden Sie sich am Benutzerkonto neu an.
5. Richten Sie jede Eingabequelle durch einen Rechtsklick auf das Icon in der GUI-Werkzeugleiste ein.
6. Wählen Sie zwischen den installierten Eingabequellen mittels SUPERTASTE+LEERTASTE (die SUPERTASTE ist normalerweise die Windows-Taste).

Tipp
Wenn Sie mit Ihrer japanischen Tastatur Zugriff auf eine Tastaturumgebung mit alphabetischen Zeichen wünschen, bei der Umschalt-2 das doppelte Anführungszeichen (") ausgibt, sollten Sie in der obigen Prozedur ”Japanisch” auswählen. Auch können Sie Japanisch über ”Japanisch mozc (oder anthy)” mit einer physikalischen ”US”-Tastatur eingeben, dabei wird dann über Umschalt-2 das @-Zeichen ausgegeben.

- Für Wayland:
  - Das `im-config`-Paket hat keinen Zweck und kann entfernt werden.
  - Möglicherweise müssen Sie keine Umgebungsvariablen mehr setzen, ausser zum Zwecke der Rückwärtskompatibilität.
  - Falls Sie Umgebungsvariablen setzen müssen, erzeugen Sie eine Datei wie ”`~/.config/environment.d/50-input`” um sie festzulegen.
- Für X-Window:
  - Installieren Sie das `im-config`-Paket.
  - Der Eintrag für im-config(8) im GUI-Menü ist ”Eingabemethode”; starten Sie es jetzt.
  - Alternativ führen Sie jetzt ”`im-config`” über eine Benutzer-Shell aus.
  - im-config(8) verhält sich unterschiedlich abhängig davon, ob es von root ausgeführt wurde oder nicht.
  - im-config(8) aktiviert standardmäßig ohne weiteres Zutun des Benutzers die beste Eingabemethode auf dem System.

8.3 Die Bildschirmausgabe

Die Linux-Konsole kann nur einen beschränkten Bereich von Zeichen anzeigen. (Sie benötigen ein spezielles Terminal-Programm wie jfbterm(1), um nicht-europäische Sprachen in der Konsole außerhalb der grafischen Umgebung anzuzeigen.)

Die grafische Umgebung (Kapitel 7) kann jegliche Zeichen in UTF-8 anzeigen, sofern die benötigten Schriften installiert und aktiviert sind. (Die Kodierung der Original-Schriftdaten wird für den Benutzer transparent erledigt.)

8.3.1 Die Terminal-Konfiguration

Das Debian-System kann über das Paket `console-setup` so eingerichtet werden, dass es mit vielen internationalen Tastaturkonfigurationen funktioniert:

```
# dpkg-reconfigure console-setup
```

Für die Linux-Konsole und das X-Window-System werden damit Parameter in "`/etc/default/console-setup`" angepasst, um viele nicht-ASCII-Zeichen, inklusive akzentuierte Zeichen, die in europäischen Sprachen verwendet werden, anzeigen zu können.

Es existieren mehrere Komponenten zur Konfiguration von Systemfunktionalitäten für textbasierte Konsolen und ncurses(3):

- die Datei "`/etc/terminfo/*/*`" (terminfo(5));
- die Umgebungsvariable "`$TERM`" (term(7));
- setterm(1), stty(1), tic(1) und toe(1).

Falls bei Anmeldung von fern auf einem Debian-System mit einem Debian-fremden `xterm`der `terminfo`-Eintrag für `xterm` nicht funktioniert, ändern Sie den Terminaltyp ("`$TERM`") von "`xterm`" auf eine der funktionseingeschränkten Versionen wie "`xterm-r6`". Näheres dazu finden Sie in "`/usr/share/doc/ncurses-base/FAQ`". "`dumb`" ist der kleinste gemeinsame Nenner für "`$TERM`".

8.3.2 Unbekannte Zeichenbreite bei ostasiatischen Zeichen

Bei Verwendung des ostasiatischen Gebietsschemas können die Zeichnung der Box sowie griechische und kyrillische Schriftzeichen breiter als beabsichtigt dargestellt werden, was zu einer ungleichmäßigen, schlecht ausgerichteten Terminalausgabe führt (näheres unter Unicode Standard Annex #11, 4.2 Ambiguous Characters).

Sie können dies Problem jedoch umgehen:

- `gnome-terminal`: Einstellungen → Profile → *Profilname* → Kompatibilität → Zeichen mit unbekannter Breite → Schmal
- `ncurses`: Setzen einer Umgebungsvariablen mit `export NCURSES_NO_UTF8_ACS=0`.

Kapitel 9

Systemtipps

Hier beschreibe ich einige grundlegende Tipps zur Konfiguration und Verwaltung des Systems, überwiegend für die Konsole.

9.1 Tipps für die Konsole

Es gibt einige Hilfsprogramme, die Sie bei Ihren Konsolenaktivitäten unterstützen können:

Paket	Popcon	Größe	Beschreibung
mc	V:42, I:172	1590	siehe Abschnitt 1.3
bsdutils	V:445, I:1000	338	script(1)-Befehl, um eine Aufzeichnung von einer Terminal-Sitzung zu erstellen
screen	V:49, I:187	1007	Terminal-Multiplexer mit VT100-/ANSI-Terminal-Emulation
tmux	V:114, I:172	1653	Terminal-Multiplexer-Alternative (verwenden Sie ”Strg-B”)
ripgrep	V:17, I:49	5284	schnelle rekursive Textsuche für Quellcode-Bäume mit automatischer Filterung
fzf	V:10, I:37	5692	Unschärfe-Textsuch-Programm
fzy	V:0.05, I:0.31	59	Unschärfe-Textsuch-Programm
rlwrap	V:1, I:11	328	Befehlszeilen-Wrapper mit readline-Funktionalität
ledit	V:0.4, I:7.1	401 401	Befehlszeilen-Wrapper mit readline-Funktionalität
rlfe	V:0.03, I:0.49	45	Befehlszeilen-Wrapper mit readline-Funktionalität

Tabelle 9.1: Liste von Programmen für Konsolenaktivitäten

9.1.1 Shell-Aktivitäten sauber aufzeichnen

Die schlichte Verwendung von script(1) wie in Abschnitt 1.4.9 erzeugt eine Datei mit Steuerzeichen. Sie können dies durch Nutzung von col(1) wie folgt beheben:

```
$ script
Script started, file is typescript
```

Machen Sie irgendetwas … und drücken Sie `Strg-D`, um `script` zu beenden.

```
$ col -bx < typescript > cleanedfile
$ vim cleanedfile
```

Es gibt alternative Methoden, um Shell-Aktivitäten aufzuzeichnen:

- Verwendung von `tee` (auch verwendbar während des Boot-Vorgangs in der initramfs):

```
$ sh -i 2>&1 | tee typescript
```

- Verwendung von `gnome-terminal` erweitertem Zeilenpuffer, um zurückscrollen zu können.
- Verwendung von `screen` mit "`^A H`" (siehe Abschnitt 9.1.2), um eine Konsolensitzung aufzuzeichnen.
- Verwendung von `vim` mit "`:terminal`", um in den Terminal-Modus zu wechseln. (Mit "`Ctrl-W N`" kommen Sie vom Terminal-Modus zurück in den Normalmodus.) Geben Sie "`:w typescript`" ein, um den Puffer in eine Datei zu schreiben.
- Verwendung von `emacs` mit "`M-x shell`", "`M-x eshell`" oder "`M-x term`", um eine Konsolensitzung aufzuzeichnen. Um den Puffer in eine Datei zu schreiben, verwenden Sie "`C-x C-w`".

9.1.2 Das Programm screen

screen(1) bietet nicht nur die Möglichkeit, in einem Terminal-Fenster mehrere Prozesse laufen zu lassen, sondern erlaubt auch **einem Remote-Shell-Prozess, Verbindungsunterbrechungen zu überstehen**. Hier ein typisches Anwendungsszenario für screen(1):

1. Sie melden sich auf einer fernen Maschine an.
2. Sie starten `screen` auf einer einfachen Konsole.
3. Sie führen mehrere Programme in `screen`-Fenstern aus, die Sie über `^A c` ("Strg-A" gefolgt von "c") erzeugen.
4. Sie können mittels `^A n` ("Strg-A" gefolgt von "n") zwischen den verschiedenen `screen`-Fenstern hin- und herschalten.
5. Plötzlich müssen Sie Ihr Terminal verlassen, aber Sie möchten Ihre aktuelle Arbeit nicht verlieren und deshalb die Verbindung erhalten.
6. Sie können die Verbindung zur `screen`-Sitzung durch eine der folgenden Methoden **lösen**:
   - Ganz brutal: die Netzwerkverbindung trennen;
   - Tippen Sie `^A d` ("Strg-A" gefolgt von "d") und melden Sie sich händisch von der Fernverbindung ab;
   - Tippen Sie `^A DD` ("Strg-A" gefolgt von "DD"), damit `screen` die Verbindung löst und Sie abmeldet.
7. Sie melden Sich am gleichen fernen Rechner erneut an (funktioniert sogar bei Anmeldung über ein anderes Terminal).
8. Sie starten `screen` über "`screen -r`".
9. `screen` **verbindet** ganz von selbst alle vorherigen `screen`-Fenster mit allen aktuell laufenden Programmen.

Tipp
Bei Verbindungen, die über Zeit oder Volumen abgerechnet werden, können Sie mit `screen` Kosten sparen, indem Sie einen Prozess aktiv laufen lassen, während die Verbindung aber unterbrochen ist, und die Sitzung später erneut verbinden, wenn Sie sich wieder einwählt haben.

In einer `screen`-Sitzung werden alle Tastatureingaben zu Ihrem aktuell laufenden screen-Fenster gesendet, außer Befehlseingaben. Alle `screen`-Befehlseingaben werden mittels ^A ("Strg-A") plus einer einzelnen Taste [plus eventuellen Parametern] eingegeben. Hier einige wichtige zur Erinnerung:

Lesen Sie screen(1) bezüglich weiterer Details.

Lesen Sie tmux(1) für Funktionalitäten einer Alternative zum Befehl screen.

Tastaturbefehl	Bedeutung
^A ?	eine Hilfe anzeigen (Tastaturkürzel anzeigen)
^A c	ein neues Fenster erstellen und dorthin wechseln
^A n	zu nächstem Fenster wechseln
^A p	zu vorherigem Fenster wechseln
^A 0	zu Fenster Nr. 0 wechseln
^A 1	zu Fenster Nr. 1 wechseln
^A w	eine Liste vorhandener Fenster anzeigen
^A a	ein Strg-A als Tastatureingabe an das aktuelle Fenster senden
^A h	Hardcopy des aktuellen Fensters in eine Datei schreiben
^A H	Starten/stoppen des Aufzeichnens vom aktuellen Fenster in eine Datei
^A ^X	Terminal sperren (Passwort-geschützt)
^A d	Verbindung der screen-Sitzung zum Terminal lösen
^A DD	Verbindung der screen-Sitzung lösen und abmelden

Tabelle 9.2: Liste von Tastaturkürzeln für screen

9.1.3 Durch Verzeichnisse navigieren

In Abschnitt 1.4.2 finden Sie 2 Tipps, die eine schnelle Navigation durch Verzeichnisse ermöglichen: `$CDPATH` und `mc`.

Wenn Sie einen Unschärfe-Textfilter verwenden, müssen Sie dabei nicht einmal den exakten Pfad angeben. Für `fzf` geben Sie dazu folgendes in `~/.bashrc` ein:

```
FZF_KEYBINDINGS_PATH=/usr/share/doc/fzf/examples/key-bindings.bash
FZF_COMPLETION_PATH=/usr/share/doc/fzf/examples/completion.bash
if [ -f $FZF_KEYBINDINGS_PATH ] && [ -f $FZF_COMPLETION_PATH ]; then
  . $FZF_KEYBINDINGS_PATH
  . $FZF_COMPLETION_PATH
fi
```

Beispiele:

- Sie können mit minimalem Aufwand in ein sehr tief verschachteltes Unterverzeichnis springen. Zunächst geben Sie ”`cd **`” ein und drücken `Tab`. Es werden Ihnen jetzt die in Frage kommenden Pfade angezeigt. Durch Eingabe von Teilen des Pfads, z.B. `s/d/b foo`, schränken Sie die Kandidaten weiter ein. Den gewünschten Pfad wählen Sie mittels `cd` und Pfeil-/Entertasten aus.
- Sie können ganz einfach und effizient einen Befehl aus der Befehlshistorie aufrufen. Drücken Sie dazu am Eingabe-Prompt `Strg-R`. Es werden Ihnen die in Frage kommenden Befehle angezeigt. Durch Eingabe von Teilen des Befehls, z.B. `vim d`, schränken Sie die Kandidaten weiter ein. Den gewünschten Befehl wählen Sie mittels Pfeil-/Entertasten aus.

9.1.4 Readline-Wrapper

Einige Befehle wie z.B. `/usr/bin/dash`, die nicht die Fähigkeit zum Editieren der Befehlshistorie haben, können diese Fähigkeit trotzdem erlangen, indem sie mittels `rlwrap` (oder seinen Äquivalenten) ausgeführt werden.

```
$ rlwrap dash -i
```

Dies bietet eine komfortable Plattform, um gewisse Dinge mit `dash` zu testen mit dem Komfort einer `bash`-ähnlichen Umgebung.

9.1.5 Durchsuchen eines Quellcode-Baums

Der Befehl rg(1) aus dem `ripgrep`-Paket ist eine schnellere Alternative zu grep(1) zum Durchsuchen eines Quellcode-Baums nach typischen Situationen. Er nutzt die Vorteile moderner Multi-Kern-CPUs und wendet bei einigen Dateien automatisch passende Filter an.

9.2 Anpassen von vim

Nachdem Sie die Grundlagen von vim(1) in Abschnitt 1.4.8 kennengelernt haben, lesen Sie bitte Bram Moolenaars "Seven habits of effective text editing (2000)", um zu verstehen, wie `vim` genutzt werden sollte.

9.2.1 Anpassen von vim mit internen Funktionalitäten

Das Verhalten von `vim` kann signifikant verändert werden, indem über Ex-Modus-Befehle interne Funktionalitäten aktiviert werden, wie z.B. "`set ...`" zum Setzen von vim-Optionen.

Diese Ex-Modus-Befehle können in der vimrc-Datei des Benutzers (traditionell "`~/.vimrc`", oder auch git-freundlich "`~/.vim/vimrc`") eingefügt werden. Hier ein einfaches Beispiel 1:

```
""" Generic baseline Vim and Neovim configuration (~/.vimrc)
"""   - For NeoVim, use "nvim -u ~/.vimrc [filename]"
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
let mapleader = ' '             " :h mapleader
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
set nocompatible                " :h 'cp -- sensible (n)vim mode
syntax on                       " :h :syn-on
filetype plugin indent on       " :h :filetype-overview
set encoding=utf-8              " :h 'enc (default: latin1) -- sensible encoding
""" current vim option value can be verified by :set encoding?
set backspace=indent,eol,start  " :h 'bs (default: nobs) -- sensible BS
set statusline=%<%f%m%r%h%w%=%y[U+%04B]%2l/%2L=%P,%2c%V
set listchars=eol:¶,tab:b''▫b''\ ,extends:b''▫b'',precedes:b''▫b'',nbsp:b''▫b''
set viminfo=!,'100,<5000,s100,h " :h 'vi -- bigger copy buffer etc.
""" Pick "colorscheme" from blue darkblue default delek desert elflord evening
""" habamax industry koehler lunaperche morning murphy pablo peachpuff quiet ron
""" shine slate torte zellner
colorscheme industry
""" don't pick "colorscheme" as "default" which may kill SpellUnderline settings
set scrolloff=5                 " :h 'scr -- show 5 lines around cursor
set laststatus=2                " :h 'ls (default 1)  k
""" boolean options can be unset by prefixing "no"
set ignorecase                  " :h 'ic
set smartcase                   " :h 'scs
set autoindent                  " :h 'ai
set smartindent                 " :h 'si
set nowrap                      " :h 'wrap
"set list                        " :h 'list (default nolist)
set noerrorbells                " :h 'eb
set novisualbell                " :h 'vb
set t_vb=                       " :h 't_vb -- termcap visual bell
set spell                       " :h 'spell
set spelllang=en_us,cjk         " :h 'spl -- english spell, ignore CJK
set clipboard=unnamedplus       " :h 'cb -- cut/copy/paste with other app
set hidden                      " :h 'hid
set autowrite                   " :h 'aw
set timeoutlen=300              " :h 'tm
```

1Tiefergehende Anpassungsbeispiele: "Vim Galore", "sensible.vim", ...

Die Tastenbelegung von `vim` kann in der vimrc-Datei des Benutzers geändert werden. Z.B.:



Achtung
Versuchen Sie nicht ohne guten Grund, die Standard-Tastaturkürzel zu ändern.

```
""" Popular mappings (imitating LazyVim etc.)
""" Window moves without using CTRL-W which is dangerous in INSERT mode
nnoremap <C-H> <C-W>h
nnoremap <C-J> <C-W>j
nnoremap <C-K> <C-W>k
silent! nnoremap <C-L> <C-W>l
""" Window resize
nnoremap <C-LEFT> <CMD>vertical resize -2<CR>
nnoremap <C-DOWN> <CMD>resize -2<CR>
nnoremap <C-UP> <CMD>resize +2<CR>
nnoremap <C-RIGHT> <CMD>vertical resize +2<CR>
""" Clear hlsearch with <ESC> (<C-L> is mapped as above)
nnoremap <ESC> <CMD>noh<CR><ESC>
inoremap <ESC> <CMD>noh<CR><ESC>
""" center after jump next
nnoremap n nzz
nnoremap N Nzz
""" fast "jk" to get out of INSERT mode (<ESC>)
inoremap  jk <CMD>noh<CR><ESC>
""" fast "<ESC><ESC>" to get out of TERM mode (CTRL-\ CTRL-N)
tnoremap <ESC><ESC> <C-\><C-N>
""" fast "jk" to get out of TERM mode (CTRL-\ CTRL-N)
tnoremap jk <C-\><C-N>
""" previous/next trouble/quickfix item
nnoremap [q <CMD>cprevious<CR>
nnoremap ]q <CMD>cnext<CR>
""" buffers
nnoremap <S-H> <CMD>bprevious<CR>
nnoremap <S-L> <CMD>bnext<CR>
nnoremap [b <CMD>bprevious<CR>
nnoremap ]b <CMD>bnext<CR>
""" Add undo break-points
inoremap  , ,<C-G>u
inoremap  . .<C-G>u
inoremap  ; ;<C-G>u
""" save file
inoremap <C-S> <CMD>w<CR><ESC>
xnoremap <C-S> <CMD>w<CR><ESC>
nnoremap <C-S> <CMD>w<CR><ESC>
snoremap <C-S> <CMD>w<CR><ESC>
""" better indenting
vnoremap < <gv
vnoremap > >gv
""" terminal (Somehow under Linux, <C-/> becomes <C-_> in Vim)
nnoremap <C-_> <CMD>terminal<CR>
"nnoremap <C-/> <CMD>terminal<CR>
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
if ! has('nvim')
""" Toggle paste mode with <SPACE>p for Vim (no need for Nvim)
set pastetoggle=<leader>p
""" nvim default mappings for Vim.  See :h default-mappings in nvim
""" copy to EOL (no delete) like D for d
noremap Y y$
```

```
""" sets a new undo point before deleting
inoremap <C-U> <C-G>u<C-U>
inoremap <C-W> <C-G>u<C-W>
""" <C-L> is re-purposed as above
""" execute the previous macro recorded with Q
nnoremap Q @@
""" repeat last substitute and *KEEP* flags
nnoremap & :&&<CR>
""" search visual selected string for visual mode
xnoremap * y/\V<C-R>"<CR>
xnoremap # y?\V<C-R>"<CR>
endif
```

Damit die oben aufgeführten Tastaturkürzel korrekt funktionieren, muss das Terminal-Programm so konfiguriert sein, dass es ”ASCII DEL” für die `Backspace`-Taste und ”Escape sequence” für die `Entf`-Taste generiert.

Weitere Konfigurationsparameter können in der vimrc-Datei des Benutzers angepasst werden. Z.B.:

```
""" Use faster 'rg' (ripgrep package) for :grep
if executable("rg")
  set grepprg=rg\ --vimgrep\ --smart-case
  set grepformat=%f:%l:%c:%m
endif
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""" Retain last cursor position :h '"
augroup RetainLastCursorPosition
  autocmd!
  autocmd BufReadPost *
    \ if line("'\"") > 0 && line ("'\"") <= line("$") |
    \   exe "normal! g'\"" |
    \ endif
augroup END
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""" Force to use underline for spell check results
augroup SpellUnderline
  autocmd!
  autocmd ColorScheme * highlight SpellBad term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellCap term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellLocal term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellRare term=Underline gui=Undercurl
augroup END
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""" highlight tailing spaces except when typing as red (set after colorscheme)
highlight TailingWhitespaces ctermbg=red guibg=red
""" \s\+     1 or more whitespace character: <Space> and <Tab>
""" \%#\@<!  Matches with zero width if the cursor position does NOT match.
match TailingWhitespaces /\s\+\%#\@<!$/
```

9.2.2 Anpassen von vim mit externen Paketen

Interessante externe Plugin-Pakete finden Sie unter:

- Vim - the ubiquitous text editor -- Die offizielle Upstream-Seite von Vim und Vim-Scripten
- VimAwesome -- Auflistung von Vim-Plugins
- vim-scripts -- Debian-Paket: eine Sammlung von Vim-Scripten

Plugin-Pakete aus dem vim-scripts-Paket können über die vimrc-Datei des Benutzers aktiviert werden. Z.B.:

```
packadd! secure-modelines
packadd! winmanager
" IDE-like UI for files and buffers with <space>w
nnoremap <leader>w         :WMToggle<CR>
```

Das neue native Vim-Paketsystem funktioniert sehr gut mit "`git`" und "`git submodule`". Eine solche Beispielkonfiguration finden Sie in meinem git-Repository: dot-vim. Was sie bewirkt, ist letztlich:

- Durch Verwenden von "`git`" und "`git submodule` werden die aktuellsten externen Pakete in `~/.vim/pack/*/opt/` abgelegt, z.B. für "*name*" in `~/.vim/pack/*/opt/`*name*, und entsprechend für andere Pakete.
- Indem eine `:packadd!` *name*-Zeile zur vimrc-Datei des Benutzers hinzugefügt wird, werden die entsprechenden Pakete im `runtimepath` abgelegt.
- Vim lädt diese Pakete während der Initialisierung aus dem `runtimepath`.
- Am Ende der Initialisierung werden die Marker (Tags) für die installierten Dokumente mit "`helptags ALL`" aktualisiert.

Wenn Sie es genauer wissen möchten, starten Sie `vim` mit "`vim --startuptime vimstart.log`", um die aktuell ausgeführte Startsequenz sowie die für jeden Schritt benötigte Zeit kontrollieren zu können.

Es kann sehr verwirrend sein, wenn es zu viele Wege 2 für die Verwaltung und das Herunterladen externer `vim`-Pakete gibt. Sich an die Original-Informationen zu halten, ist dann die beste Hilfe.

Tastatureingabe	Information
`:help package`	Erklärung zum Vim-Paket-Mechanismus
`:help runtimepath`	Erklärung zum `runtimepath`-Mechanismus
`:version`	interner Status inklusive Kandidaten für die vimrc-Datei
`:echo $VIM`	die Umgebungsvariable "`$VIM`", die verwendet wird, um die vimrc-Datei zu finden
`:set runtimepath?`	Liste der Verzeichnisse, die auf der Suche nach runtime-Support-Dateien durchsucht werden
`:echo $VIMRUNTIME`	die Umgebungsvariable "`$VIMRUNTIME`", die verwendet wird, um verschiedene (vom System bereitgestellte) Runtime-Support-Dateien zu finden

Tabelle 9.3: Informationen zur Initialisierung von `vim`

9.3 Datenaufzeichnung und -darstellung

9.3.1 Der log-Daemon

Viele traditionelle Programme zeichnen ihre Aktivitäten als Textdatei im "`/var/log/`"-Verzeichnis auf.

logrotate(8) wird verwendet, um die Administration der Logdateien auf Systemen zu vereinfachen, die viele solcher Logdateien erzeugen.

Viele neue Programme hingegen zeichnen ihre Aktivitäten in binärem Format mittels dem Journal-Service systemd-journald(8) im Verzeichnis "`/var/log/journal`" auf.

Sie können Logdaten von einem Shell-Skript zum systemd-journald(8)-Journal hinzufügen, indem Sie den systemd-cat(1)-Befehl nutzen.

Lesen Sie dazu Abschnitt 3.5 und Abschnitt 3.4.

2vim-pathogen war sehr populär.

9.3.2 Analyseprogramme für Logdateien

Hier eine Liste erwähnenswerter Analyseprogramme für Logdateien ("`~Gsecurity::log-analyzer`" in aptitude(8)):

Paket	Popcon	Größe	Beschreibung
`fail2ban`	V:101, I:112	2275	IP-Adressen sperren, die vielfache Authentifizierungsfehler verursachen
`logwatch`	V:9, I:10	2440	Log-Analysator mit netter Ausgabe, geschrieben in Perl
`awstats`	V:4.7, I:7.5	6934	leistungsfähiger und mit vielen Funktionen ausgestatteter Webserver-Log-Analysator
`analog`	V:3, I:81	3739	Webserver-Log-Analysator
`sarg`	V:0.61, I:0.67	863	Analysereportgenerator für Squid
`pflogsumm`	V:1.4, I:3.4	173	Programm zur Zusammenfassung von Postfix-Logeinträgen
`fwlogwatch`	V:0.11, I:0.15	487	Firewall-Log-Analysator
`squidview`	V:0.04, I:0.38	224	Programm zum Beobachten und Analysieren von Squid-access.log-Dateien
`swatch`	V:0.07, I:0.28	99	Logdatei-Betrachter mit Unterstützung für reguläre Ausdrücke, Hervorhebung und Einsprungstellen
`crm114`	V:0.05, I:0.30	1371	Programm zur flexiblen Einordnung von E-Mails und anderen Daten (CRM114)
`icmpinfo`	V:0.01, I:0.34	42	Programm zur Interpretation von ICMP-Nachrichten

Tabelle 9.4: Liste von System-Log-Analyzern

Anmerkung
CRM114 bietet eine Sprachinfrastruktur, um **Unschärfe**-Filter (fuzzy filters) mittels der TRE-regex-Bibliothek zu erstellen. Ein populärer Anwendungsfall ist der Spam-Mail-Filter, aber es kann auch als Log-Analysator verwendet werden.

9.3.3 Angepasste Anzeige von Textdaten

Obwohl Pager wie more(1) und less(1) (lesen Sie dazu Abschnitt 1.4.5) sowie spezielle Werkzeuge für Hervorhebung und Formatierung (Näheres dazu in Abschnitt 11.1.8) Text gut anzeigen können, sind normale Texteditoren (Details in Abschnitt 1.4.6) vielseitiger und besser anzupassen.

Tipp
Bei vim(1) und seinem Pager-Modus alias view(1) wird über "`:set hls`" die Suche mit Hervorhebung aktiviert.

9.3.4 Angepasste Anzeige von Datum und Uhrzeit

Das Standard-Anzeigeformat von Datum und Uhrzeit bei dem Befehl "`ls -l`" hängt vom gewählten **Gebietsschema** (locale) ab (siehe dazu auch Abschnitt 1.2.6). Die Variable "`$LANG`" wird als erstes abgefragt, der Wert kann jedoch mittels der "`$LC_TIME`"- oder "`$LC_ALL`"-Variable überschrieben werden.

Das letztlich verwendete Anzeigeformat für das jeweilige Gebietsschema hängt von der verwendeten Standard-C-Bibliothek (aus dem `libc6`-Paket) ab. Aufgrunddessen hatten verschiedene Debian-Veröffentlichungen unterschiedliche Standardeinstellungen. Bezüglich der ISO-Formate besuchen Sie ISO 8601.

Wenn Sie das Anzeigeformat von Datum und Uhrzeit wirklich über die **locale**-Einstellung hinaus anpassen möchten, sollten Sie die Option **time style value** setzen, entweder über das "`--time-style`"-Argument oder über den Wert der Variable "`$TIME_STYLE`" (Näheres unter ls(1), date(1) und "`info coreutils 'ls invocation'`").

time style value	Locale	Anzeige von Datum und Uhrzeit
`iso`	alle	`01-19 00:15`
`long-iso`	alle	`2009-01-19 00:15`
`full-iso`	alle	`2009-01-19 00:15:16.000000000 +0900`
`locale`	`C`	`Jan 19 00:15`
`locale`	`en_US.UTF-8`	`Jan 19 00:15`
`locale`	`es_ES.UTF-8`	`ene 19 00:15`
`+%d.%m.%y %H:%M`	alle	`19.01.09 00:15`
`+%d.%b.%y %H:%M`	`C` oder `en_US.UTF-8`	`19.Jan.09 00:15`
`+%d.%b.%y %H:%M`	`es_ES.UTF-8`	`19.ene.09 00:15`

Tabelle 9.5: Anzeigebeispiele von Datum und Uhrzeit für den Befehl "`ls -l`" mit **time style value**-Einstellung

Tipp
Sie können die Eingabe langer Optionsparameter auf der Befehlszeile vermeiden, indem sie einen Befehls-Alias verwenden (siehe Abschnitt 1.5.9):

```
alias ls='ls --time-style=+%d.%m.%y %H:%M'
```

9.3.5 Farbige Shell-Ausgabe

Die Shell-Ausgabe der meisten modernen Terminals kann über ANSI-Escape-Codes farbig gestaltet werden (lesen Sie dazu auch "`/usr/share/doc/xterm/ctlseqs.txt.gz`").

Probieren Sie zum Beispiel folgendes:

```
$ RED=$(printf "\x1b[31m")
$ NORMAL=$(printf "\x1b[0m")
$ REVERSE=$(printf "\x1b[7m")
$ echo "${RED}RED-TEXT${NORMAL} ${REVERSE}REVERSE-TEXT${NORMAL}"
```

9.3.6 Farbige Befehle

Farbige Befehle sind praktisch, um deren Ausgabe in einer interaktiven Umgebung zu kontrollieren. Ich habe Folgendes in meiner "`~/.bashrc`":

```
if [ "$TERM" != "dumb" ]; then
    eval "`dircolors -b`"
    alias ls='ls --color=always'
    alias ll='ls --color=always -l'
    alias la='ls --color=always -A'
    alias less='less -R'
    alias ls='ls --color=always'
    alias grep='grep --color=always'
    alias egrep='egrep --color=always'
    alias fgrep='fgrep --color=always'
    alias zgrep='zgrep --color=always'
else
```

```
    alias ll='ls -l'
    alias la='ls -A'
fi
```

Die Verwendung von Befehls-Alias beschränkt die Farbeffekte auf interaktive Befehlseingaben. Dies ist ein Vorteil gegenüber dem Exportieren der Umgebungsvariablen (”`export GREP_OPTIONS='--color=auto'`”), da die Farben auch in Pager-Programmen wie less(1) sichtbar sind. Falls Sie die Farbeffekte bei Weiterleitung zu anderen Programmen deaktivieren möchten, nutzen Sie stattdessen ”`--color=auto`” in dem obigen Beispiel für die ”`~/.bashrc`”.

Tipp
Sie können die Farb-Alias in der interaktiven Umgebung ausschalten, indem Sie die Shell mit ”`TERM=dumb bash`” starten.

9.3.7 Aufzeichnen von Editor-Aktivitäten für komplexe Wiederholungen

Sie können Editor-Aktivitäten aufzeichnen, um sie später für komplexe Wiederholungen wiederzuverwenden.

Bei Vim wie folgt:

- ”qa”: Aufnahme der eingegebenen Zeichen in das Register ”a” starten;
- … Editor-Aktivitäten;
- ”q”: Aufnahme der eingegebenen Zeichen beenden;
- ”@a”: Ausführen des Inhalts von Register ”a”.

Bei Emacs wie folgt:

- ”`C-x (`”: Definition eines Tastaturmakros starten;
- … Editor-Aktivitäten;
- ”`C-x )`”: Definition des Tastaturmakros beenden;
- ”`C-x e`”: Ausführen des Tastaturmakros.

9.3.8 Die Bildschirmanzeige einer X-Anwendung aufzeichnen

Es gibt mehrere Wege, die Anzeige einer X-Anwendung aufzuzeichnen, inklusive eines `xterm`-Displays:

Paket	Popcon	Größe	Umgebung	Befehl
`gnome-screenshot`	V:12, I:91	1115	Wayland	Bildschirmfoto-Applikation für GNOME
`flameshot`	V:7, I:17	3542	Wayland	Bildschirmfoto-Applikation auf Steroid
`gimp`	V:40, I:206	33386	Wayland + X	Bildschirmfoto im GUI-Menü
`x11-apps`	V:32, I:440	2461	X	xwd(1)
`imagemagick`	V:7, I:266	82	X	import(1)
`scrot`	V:4, I:47	144	X	scrot(1)

Tabelle 9.6: Liste von Werkzeugen zur Bildbearbeitung

Paket	Popcon	Größe	Beschreibung
etckeeper	V:23, I:26	156	Konfigurationsdateien und deren Metadaten mit Git (Standard), Mercurial oder GNU Bazaar abspeichern
timeshift	V:7, I:13	4700	Werkzeug zur Systemwiederherstellung mittels rsync oder BTRFS-Schnappschüssen
snapper	V:6.3, I:8.4	2461	Verwaltungswerkzeug für Linux-Dateisystem-Schnappschüsse

Tabelle 9.7: Liste von Paketen zur Aufzeichnung der Konfigurations-Historie

9.3.9 Aufzeichnen von Änderungen in Konfigurationsdateien

Es gibt spezielle Werkzeuge, um Änderungen in Konfigurationsdateien mit Hilfe von DVCS-Systemen aufzuzeichnen sowie System-Schnappschüsse mittels Btrfs zu erstellen:

Sie könnten auch den Ansatz über ein lokales Skript wie in Abschnitt 10.2.3 in Erwägung ziehen.

9.4 Überwachen, Steuern und Starten von Programmaktivitäten

Programmaktivitäten können mittels spezieller Werkzeuge überwacht und kontrolliert werden:

Paket	Popcon	Größe	Beschreibung
coreutils	V:914, I:1000	19751	nice(1): ein Programm mit veränderter Ablaufpriorität ausführen
bsdutils	V:445, I:1000	338	renice(1): die Ablaufpriorität eines laufenden Prozess verändern
procps	V:845, I:998	2692	Werkzeuge für das "`/proc`"-Dateisystem: ps(1), top(1), kill(1), watch(1), …
psmisc	V:404, I:705	950	Werkzeuge für das "`/proc`"-Dateisystem: killall(1), fuser(1), peekfd(1), pstree(1)
time	V:7, I:73	148	time(1): ein Programm ausführen und die Zeit ausgeben, während der Ressourcen verbraucht wurden
sysstat	V:112, I:163	1904	sar(1), iostat(1), mpstat(1), …: Werkzeuge zur Messung der System-Performance unter Linux
isag	V:0.1, I:3.1	109	Interaktive grafische Darstellung der Systemaktivität, basierend auf sysstat
lsof	V:463, I:949	492	lsof(8): mit der Option "`-p`" diejenigen Dateien auflisten, die von einem gerade laufenden Prozess geöffnet wurden
strace	V:11, I:99	3250	strace(1): Systemaufrufe und -signale verfolgen
ltrace	V:1, I:11	420	ltrace(1): Bibliotheksaufrufe verfolgen
xtrace	V:0.03, I:0.65	342	xtrace(1): Kommunikation zwischen X11-Client und -Server verfolgen
powertop	V:38, I:231	696	powertop(1): Informationen über verbrauchte Systemleistung
cron	V:929, I:997	248	Prozesse laut dem Ablaufplan des cron(8)-Daemons im Hintergrund ausführen
anacron	V:424, I:480	114	cron-ähnlicher Programmablauf-Planer für Systeme, die nicht 24 Stunden am Tag laufen
at	V:68, I:87	158	at(1) oder batch(1): einen Prozess zu einer bestimmten Zeit oder unterhalb einer bestimmten Systembelastung ausführen

Tabelle 9.8: Liste von Werkzeugen zur Überwachung und Steuerung von Programmaktivitäten

Tipp
Das `procps`-Paket enthält sehr grundlegende Werkzeuge zum Überwachen, Steuern und Starten von Programmaktivitäten. Sie sollten sich mit diesen vertraut machen.

9.4.1 Zeitmessung für einen Prozess

Zeit anzeigen, die der von dem Befehl angestossene Prozess benötigt hat:

```
# time some_command >/dev/null
real    0m0.035s       # time on wall clock (elapsed real time)
user    0m0.000s       # time in user mode
sys     0m0.020s       # time in kernel mode
```

9.4.2 Die Ablaufpriorität

Der nice-Wert wird verwendet, um die Ablaufpriorität von Prozessen zu steuern.

nice-Wert	Ablaufpriorität
19	niedrigste Prozesspriorität (sparsam)
0	sehr hohe Prozesspriorität für Benutzer
-20	sehr hohe Prozesspriorität für root (nicht sparsam)

Tabelle 9.9: Liste der nice-Werte für die Ablaufpriorität

```
# nice  -19 top                                        # very nice
# nice --20 wodim -v -eject speed=2 dev=0,0 disk.img # very fast
```

Manchmal schadet ein extremer nice-Wert dem System mehr als er ihm nützt. Nutzen Sie diesen Befehl mit Vorsicht.

9.4.3 Der Befehl ps

Der ps(1)-Befehl im Debian-System unterstützt sowohl BSD- wie auch SystemV-Funktionalitäten und hilft dabei, die Prozessaktivitäten (in statischem Zustand) zu beurteilen.

Art	typischer Befehl	Funktionalität
BSD	`ps aux`	anzeigen von %CPU und %MEM
System V	`ps -efH`	anzeigen der PPID

Tabelle 9.10: Liste von ps-Befehlen

Nicht mehr vorhandene Zombie-Kindprozesse können Sie über die Prozess-ID des Eltern-Prozesses in dem ”`PPID`”-Feld beenden.

Der Befehl pstree(1) zeigt alle Prozesses in Form eines Hierarchie-Baums an.

9.4.4 Der Befehl top

top(1) auf einem Debian-System hat reichhaltige Funktionalitäten und hilft dabei, Prozesse zu identifizieren, die sich sehr auffällig dynamisch verhalten.

Es ist ein interaktives Programm mit Vollbildschirmmodus. Sie erhalten eine Hilfe zur Nutzung, indem Sie die Taste ”h” drücken; mit ”q” beenden Sie das Programm.

9.4.5 Dateien auflisten, die von einem Prozess geöffnet wurden

Sie können wie folgt alle Dateien auflisten, die von einem Prozess mit einer bestimmten Prozess-ID (PID), z.B. 1, geöffnet wurden:

```
$ sudo lsof -p 1
```

PID 1 ist normalerweise das `init`-Programm.

9.4.6 Programmaktivitäten verfolgen

Sie können Programmaktivitäten mit strace(1) (für Systemaufrufe und -signale), ltrace(1) (für Bibliotheksaufrufe) oder xtrace(1) (für die Kommunikation zwischen X11-Client und -Server) verfolgen.

Systemaufrufe des `ls`-Befehls verfolgen Sie z.B. wie folgt:

```
$ sudo strace ls
```

Tipp
Nutzen Sie das Skript **strace-graph** aus **/usr/share/doc/strace/examples/**, um eine hübsche Baumansicht zu erstellen.

9.4.7 Identifikation von Prozessen, die Dateien oder Sockets verwenden

Auch können Sie Prozesse, die bestimmte Dateien nutzen, mit fuser(1) identifizieren, hier z.B. "`/var/log/mail.log`":

```
$ sudo fuser -v /var/log/mail.log
                     USER        PID ACCESS COMMAND
/var/log/mail.log:   root       2946 F.... rsyslogd
```

Sie sehen, dass die Datei "`/var/log/mail.log`" von dem Befehl rsyslogd(8) zum Schreiben geöffnet wurde.

Prozessse, die bestimmte Sockets verwenden, können Sie ebenfalls mittels fuser(1) identifizieren, für "`smtp/tcp`" z.B. wie hier:

```
$ sudo fuser -v smtp/tcp
                     USER        PID ACCESS COMMAND
smtp/tcp:            Debian-exim   3379 F.... exim4
```

Jetzt wissen Sie, dass auf Ihrem System exim4(8) läuft, um TCP-Verbindungen zum SMTP-Port (25) zu bedienen.

9.4.8 Einen Befehl mit festem Intervall wiederholt ausführen

watch(1) führt ein Programm wiederholt mit einem festen Intervall aus und zeigt dessen Ausgabe im Vollbildschirmmodus an.

```
$ watch w
```

Hiermit wird alle 2 Sekunden neu angezeigt, wer auf dem System angemeldet ist.

9.4.9 Einen Befehl wiederholt mit einer Schleife über verschiedene Dateien ausführen

Es gibt verschiedene Wege, um einen Befehl zu wiederholen und jedes Mal eine andere Datei aus einer bestimmten Menge von Dateien (hier z.B. ausgewählt über ein glob-Suchmuster auf "`*.ext`") als Argument zu verwenden.

- Methode mit einer for-Schleife in der Shell (Näheres in Abschnitt 12.1.4):

```
for x in *.ext; do if [ -f "$x"]; then command "$x" ; fi; done
```

- Kombination aus find(1) und xargs(1):

```
find . -type f -maxdepth 1 -name '*.ext' -print0 | xargs -0 -n 1 command
```

- find(1) mit "`-exec`"-Option und einem Befehl:

```
find . -type f -maxdepth 1 -name '*.ext' -exec command '{}' \;
```

- find(1) mit "`-exec`"-Option und einem kurzen Shell-Skript:

```
find . -type f -maxdepth 1 -name '*.ext' -exec sh -c "command '{}' && echo 'successful'" \;
```

Die obigen Beispiele wurden geschrieben, um auch eine korrekte Behandlung von komischen Dateinamen, wie z.B. solche mit Leerzeichen, sicherzustellen. Lesen Sie dazu auch Abschnitt 10.1.5; dort finden Sie außerdem weitergehende Verwendungen des Befehls find(1).

9.4.10 Ein Programm aus der grafischen Oberfläche heraus starten

Auf der Befehlszeile (command-line interface, CLI) wird das erste Programm mit dem passenden Namen ausgeführt, das in den durch die Umgebungsvariable `$PATH` definierten Verzeichnissen gefunden wird. Lesen Sie dazu auch Abschnitt 1.5.3.

Bei einer zum freedesktop.org-Standard konformen grafischen Oberfläche (graphical user interface, GUI) enthalten die `*.desktop`-Dateien in dem Verzeichnis `/usr/share/applications/` alle nötigen Attribute für die Anzeige der Programmeinträge im grafischen Menü. Jedes Paket, das mit dem xdg-Menüsystem von Freedesktop.org konform ist, installiert seine Menüdaten (für die "*.desktop"-Dateien) unter "/usr/share/applications/". Moderne Arbeitsplatzumgebungen gemäß Freedesktop.org-Standard verwenden diese Daten, um mittels dem xdg-utils-Paket das Programmmenü zu erstellen. Näheres dazu in "/usr/share/doc/xdg-utils/README".

Im folgenden Beispiel definiert die Datei `chromium.desktop` die Attribute für den "Chromium-Webbrowser", z.B. "Name" für den Programmnamen, "Exec" für den Pfad zum auszuführenden Programm mit zugehörigen Argumenten, "Icon" für das verwendete Icon-Symbol usw. (Details finden Sie unter Desktop Entry Specification):

```
[Desktop Entry]
Version=1.0
Name=Chromium Web Browser
GenericName=Web Browser
Comment=Access the Internet
Comment[fr]=Explorer le Web
Exec=/usr/bin/chromium %U
Terminal=false
X-MultipleArgs=false
Type=Application
Icon=chromium
Categories=Network;WebBrowser;
MimeType=text/html;text/xml;application/xhtml_xml;x-scheme-handler/http;x-scheme-handler/ ↩
    https;
StartupWMClass=Chromium
StartupNotify=true
```

Dies ist eine vereinfachte Beschreibung. Die `*.desktop`-Dateien werden wie folgt eingelesen:

Die Desktop-Umgebung setzt die Umgebungsvariablen `$XDG_DATA_HOME` und `$XDG_DATA_DIR`. Unter GNOME z.B.:

- `$XDG_DATA_HOME` wird nicht genutzt (der Standardwert aus `$HOME/.local/share` wird verwendet).
- `$XDG_DATA_DIRS` wird auf `/usr/share/gnome:/usr/local/share/:/usr/share/` gesetzt.

Daher werden die Basisverzeichnisse (lesen Sie XDG Base Directory Specification) und `applications`-Verzeichnisse wie folgt gesetzt:

- `$HOME/.local/share/` → `$HOME/.local/share/applications/`
- `/usr/share/gnome/` → `/usr/share/gnome/applications/`
- `/usr/local/share/` → `/usr/local/share/applications/`
- `/usr/share/` → `/usr/share/applications/`

Die `*.desktop`-Dateien aus diesen `applications`-Verzeichnissen werden in obiger Reihenfolge eingelesen.

Tipp
Ein benutzerdefinierter Menüeintrag für die grafische Oberfläche kann erstellt werden, indem eine `*.desktop`-Datei zum Verzeichnis `$HOME/.local/share/applications/` hinzugefügt wird.

Tipp
Die "`Exec=...`"-Zeile wird nicht durch die Shell ausgewertet. Nutzen Sie den env(1)-Befehl, wenn Umgebungsvariablen gesetzt werden müssen.

Tipp
Ähnlich dazu verhält es sich, wenn eine `*.desktop`-Datei zum `autostart`-Verzeichnis in diesen Basisverzeichnissen hinzugefügt wird: das in der `*.desktop`-Datei definierte Programm wird beim Start der grafischen Benutzeroberfläche automatisch ausgeführt (z.B. nach dem Anmelden). Lesen Sie dazu Desktop Application Autostart Specification.

Tipp
Ähnlich ist es auch, wenn eine `*.desktop`-Datei im `$HOME/Desktop`-Verzeichnis erstellt wird und die grafische Umgebung die Funktionalität der Programmstarter-Icons (desktop icon launcher) unterstützt: das definierte Programm wird bei Anklicken des Icons ausgeführt. Bitte beachten Sie, dass der eigentliche Name des `$HOME/Desktop`-Verzeichnisses letztlich vom Gebietsschema (Locale) abhängt; Näheres dazu finden Sie in xdg-user-dirs-update(1).

9.4.11 Anpassen des zu startenden Programms

Einige Programme starten automatisch weitere Programme. Hier einige Punkte bezüglich der Anpassung dieses Prozesses.

- Menü zur Konfiguration von bevorzugten Anwendungen:
  - GNOME-Desktop: "Einstellungen" → "Anwendungen" → "Vorgabe-Anwendungen";

  - KDE-Desktop: ”Anwendungsstarter” → ”Systemeinstellungen” → ”Standardanwendungen”
  - Iceweasel-Browser: ”Bearbeiten” → ”Einstellungen” → ”Anwendungen”;
  - mc(1): ”`/etc/mc/mc.ext`”;
- Umgebungsvariablen wie ”`$BROWSER`”, ”`$EDITOR`”, ”`$VISUAL`” und ”`$PAGER`” (Näheres in environ(7));
- das update-alternatives(1)-System für Programme wie ”`editor`”, ”`view`”, ”`x-www-browser`”, ”`gnome-www-browser`” und ”`www-browser`” (Näheres in Abschnitt 1.4.7);
- die Inhalte der Dateien ”`~/.mailcap`” und ”`/etc/mailcap`”, über die die MIME-Types bestimmten Programmen zugeordnet werden (Näheres in mailcap(5));
- die Inhalte der Dateien ”`~/.mime.types`” und ”`/etc/mime.types`”, über die Dateinamenerweiterungen entsprechenden MIME-Types zugeordnet werden (Näheres in run-mailcap(1)).

Tipp
update-mime(8) aktualisiert die ”`/etc/mailcap`”-Datei unter Verwendung von ”`/etc/mailcap.order`” (Näheres in mailcap.order(5)).

Tipp
Das `debianutils`-Paket stellt sensible-browser(1), sensible-editor(1) und sensible-pager(1) bereit; diese treffen eine sinnvolle Auswahl, welcher Editor, Pager bzw. Webbrowser aufgerufen wird. Ich empfehle, dass Sie diese Shell-Skripte lesen.

Tipp
Um eine Konsolen-Applikation (hier `mutt`) im GUI als bevorzugte Anwendung festzulegen, sollten Sie wie folgt ein Startskript für die Applikation erzeugen (in diesem Fall ”`/usr/local/bin/mutt-term`”) und dieses als bevorzugt zu startende Anwendung konfigurieren:

```
# cat /usr/local/bin/mutt-term <<EOF
#!/bin/sh
gnome-terminal -e "mutt \$@"
EOF
# chmod 755 /usr/local/bin/mutt-term
```

9.4.12 Einen Prozess beenden (kill)

Verwenden Sie kill(1), um über die Prozess-ID einen Prozess zu beenden (oder diesem ein Signal zu senden).

Nutzen Sie killall(1) oder pkill(1), um das gleiche über den Befehlsnamen und andere Attribute des Prozesses zu erledigen.

9.4.13 Einmalige Aufgaben planen

Führen Sie den at(1)-Befehl wie folgt aus, um eine einmalige Aufgabe zu planen:

```
$ echo 'command -args'| at 3:40 monday
```

Signalwert	Signalname	Aktion	Hinweis
0	---	es wird kein Signal gesendet (siehe kill(2))	Überprüfung, ob der Prozess läuft
1	SIGHUP	den Prozess beenden	nicht mehr verbundenes Terminal (Signal aufgehängt)
2	SIGINT	den Prozess beenden	Unterbrechen durch die Tastatur (`Strg-C`)
3	SIGQUIT	den Prozess beenden und Core dump erstellen	Beenden durch die Tastatur (`Strg-\`)
9	SIGKILL	den Prozess beenden	Signal zum unmittelbaren Beenden
15	SIGTERM	den Prozess beenden	Anfrage zum Beenden

Tabelle 9.11: Liste von häufig verwendeten Signalen für den kill-Befehl

9.4.14 Regelmäßige Aufgaben planen

Verwenden Sie cron(8), um regelmäßig wiederkehrende Aufgaben zu planen. Lesen Sie dazu crontab(1) and crontab(5).

Sie können Prozesse zur Ausführung durch einen normalen Benutzer, z.B. den Benutzer `foo`, einplanen, indem Sie mit dem Befehl "`crontab -e`" eine crontab(5)-Datei namens "`/var/spool/cron/crontabs/foo`" erstellen.

Hier ein Beispiel einer crontab(5)-Datei:

```
# use /usr/bin/sh to run commands, no matter what /etc/passwd says
SHELL=/bin/sh
# mail any output to paul, no matter whose crontab this is
MAILTO=paul
# Min Hour DayOfMonth Month DayOfWeek command (Day... are OR'ed)
# run at 00:05, every day
5  0  *  * *   $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 14:15 on the first of every month -- output mailed to paul
15 14 1  * *   $HOME/bin/monthly
# run at 22:00 on weekdays(1-5), annoy Joe. % for newline, last % for cc:
0 22 *   * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%.%%
23 */2 1 2 *   echo "run 23 minutes after 0am, 2am, 4am ..., on Feb 1"
5  4 *   * sun echo "run at 04:05 every Sunday"
# run at 03:40 on the first Monday of each month
40 3 1-7 * *   [ "$(date +%a)" == "Mon" ] && command -args
```

Tipp
Auf Systemen, die nicht ständig laufen, installieren Sie das `anacron`-Paket, um regelmäßige Aufgaben auszuführen, sobald nach entsprechender Maschinen-Einschaltdauer ein festgelegter Intervall abgelaufen ist. Näheres unter anacron(8) und anacrontab(5).

Tipp
Geplante Skripte zur Systemwartung können Sie vom root-Konto aus wiederholt ausführen, indem Sie solche Skripte in "`/etc/cron.hourly/`", "`/etc/cron.daily/`", "`/etc/cron.weekly/`" oder "`/etc/cron.monthly/`" einfügen. Der Ausführungszeitpunkt dieser Skripte kann über "`/etc/crontab`" und "`/etc/anacrontab`" angepasst werden.

Systemd gewisse Fähigkeiten, um die Ausführung von Programmen ohne den `cron`-Daemon zu planen. Zum Beispiel werden über `/lib/systemd/system/apt-daily.timer` und `/lib/systemd/system/apt-daily.service` tägliche Download-Aktivitäten von apt geplant. Näheres in systemd.timer(5) .

9.4.15 Aufgaben ereignisgesteuert planen

Systemd kann Programme nicht nur über das timer-Ereignis gesteuert ausführen, sondern auch über das mount-Ereignis. Beispiele finden Sie in Abschnitt 10.2.3.3 und Abschnitt 10.2.3.2.

9.4.16 Die Alt-SysRq-Tastenkombination

Alt-SysRq (auf deutschen Tastaturen das Drücken von Alt+Druck) gefolgt von einer weiteren Taste löst verschiedene, fast magische Aktionen aus, um im Notfall die Kontrolle über das System wiederzuerlangen:

Weitere Details finden Sie im Linux kernel user's and administrator's guide » Linux Magic System Request Key Hacks.

Taste nach Alt-SysRq	Beschreibung der Aktion
k	alle Prozesse auf der **aktuellen virtuellen Konsole** durch ein **k**ill zum Beenden zwingen (SAK)
s	alle eingebundenen Dateisysteme **s**ynchronisieren (Dateisystempuffer auf Platte schreiben), um Datenverluste zu vermeiden
u	alle eingebundenen Dateisysteme neu einbinden mit Nur-Lese-Berechtigung (**u**mount)
r	Tastatur aus dem Roh-Modus (**r**aw) wiederherstellen nach X-Abstürzen

Tabelle 9.12: Liste erwähnenswerter SAK-Befehlstasten

Tipp
Von einem SSH-Terminal u.ä. können Sie die Alt-SysRq-Funktionalität nutzen, indem Sie in "`/proc/sysrq-trigger`" schreiben. Zum Beispiel werden durch "`echo s > /proc/sysrq-trigger; echo u > /proc/sysrq-trigger`" alle eingebundenen Dateisysteme **s**ynchronisiert und mittels **u**mount aus der Einbindung gelöst.

Der aktuelle Debian amd64 Linux-Kernel (Stand 2021) hat `/proc/sys/kernel/sysrq=438=0b110110110`:

- 2 = 0x2 - Aktivieren der Kontrolle über das Konsolen-Log-Level (AN)
- 4 = 0x4 - Aktivieren der Kontrolle über die Tastatur (SAK, unraw) (AN)
- 8 = 0x8 - Aktivieren von Speicher-Dumps der Prozesse zwecks Fehlersuche (Debugging) usw. (AUS)
- 16 = 0x10 - Aktivieren des sync-Befehls (AN)
- 32 = 0x20 - Aktivieren von Neueinbinden mit Nur-Lese-Berechtigung (remount read-only) (AN)
- 64 = 0x40 - Aktivieren des Sendens von Signalen an Prozesse (term, kill, oom-kill) (AUS)
- 128 = 0x80 - Erlauben von Neustart/Ausschalten (AN)
- 256 = 0x100 - Erlauben von nice-Prioritäten für all Echtzeit-(RT-)Prozesse (AN)

9.5 Tipps zur Systempflege

9.5.1 Wer ist/war im System aktiv?

Sie können wie folgt überprüfen, wer derzeit im System aktiv ist:

- who(1) zeigt, wer angemeldet ist.
- w(1) zeigt, wer angemeldet ist und was derjenige jeweils tut.
- last(1) zeigt eine Auflistung der zuletzt angemeldeten Benutzer.
- lastb(1) zeigt eine Auflistung der letzten Benutzer mit fehlgeschlagenen Anmeldeversuchen.

Tipp
Diese Informationen werden in "`/var/run/utmp`" und "`/var/log/wtmp`" zur Verfügung gehalten. Lesen Sie dazu login(1) und utmp(5).

9.5.2 Allen eine Warnung schicken

Mit wall(1) können Sie wie folgt eine Nachricht an alle am System angemeldeten Benutzer schicken:

```
$ echo "We are shutting down in 1 hour" | wall
```

9.5.3 Hardware-Identifikation

Bei PCI-basierten Geräten (AGP, PCI-Express, CardBus, ExpressCard usw.) ist lspci(8) (eventuell mit der Option "`-nn`") ein guter Anfang zur Hardware-Identifikation.

Alternativ können Sie die Hardware auch über das Auslesen der Inhalte von "`/proc/bus/pci/devices`" identifizieren oder indem Sie die Verzeichnisse unterhalb von "`/sys/bus/pci`" durchsuchen (Näheres in Abschnitt 1.2.12).

Paket	Popcon	Größe	Beschreibung
`pciutils`	V:279, I:994	289	Hilfsprogramme für PCI unter Linux: lspci(8)
`usbutils`	V:146, I:894	322	Hilfsprogramme für USB unter Linux: lsusb(8)
`nvme-cli`	V:30, I:39	2222	Hilfsprogramme für NVME unter Linux: nvme(1)
`pcmciautils`	V:3.8, I:5.7	92	Hilfsprogramme für PCMCIA unter Linux: pccardctl(8)
`scsitools`	V:0.2, I:2.0	261	Sammlung von Werkzeugen für das SCSI-Hardware-Management: lsscsi(8)
`procinfo`	V:0.3, I:5.7	149	Systeminformationen, gewonnen aus "`/proc`": lsdev(8)
`lshw`	V:12, I:87	971	Informationen über die Hardware-Konfiguration: lshw(1)
`discover`	V:26, I:527	81	System zur Hardware-Identifikation: discover(8)

Tabelle 9.13: Liste von Werkzeugen zur Hardware-Identifikation

9.5.4 Hardware-Konfiguration

Obwohl der größte Teil der Hardware-Konfiguration auf modernen GUI-Arbeitsplatzsystemen wie GNOME und KDE über entsprechende GUI-Werkzeuge verwaltet werden kann, ist es eine gute Idee, zumindest einige grundlegende Methoden für deren Konfiguration zu kennen.

Für Infos zur Tastatur- und Terminalkonfiguration lesen Sie Abschnitt 8.2 und Abschnitt 8.3.

ACPI ist ein neueres Rahmenwerk für das Power Management und der Nachfolger für das ältere APM.

Tipp
Die Skalierung der CPU-Frequenz wird auf modernen Systemen durch Kernel-Module wie `acpi_cpufreq` abgewickelt.

9.5.5 System- und Hardware-Zeit

Über folgende Befehle wird die System- und Hardware-Zeit auf MM/DD hh:mm, CCYY (MM - Monat, DD - Tag, hh - Stunde, mm - Minute, CCYY - Jahr) gesetzt:

```
# date MMDDhhmmCCYY
# hwclock --utc --systohc
# hwclock --show
```

Paket	Popcon	Größe	Beschreibung
console-setup	V:87, I:977	422	Werkzeuge für Schriftarten und Tastaturbelegungen auf der Linux-Konsole
keyd	V:1.7, I:1.8	1437	Daemon zur Neubelegung von Tastaturtasten, der auf Eingabeprimitive der Kernelebene (evdev, uinput) zurückgreift
keyd-application-mapper	V:0.02, I:0.07	34	Daemon zur Neubelegung von Tastaturtasten – anwendungsspezifisches Neubelegungswerkzeug
x11-xserver-utils	V:376, I:532	559	X-Server-Hilfsprogamme: xset(1), xmodmap(1)
acpid	V:51, I:74	158	Daemon zur Verwaltung von Ereignissen der Advanced-Configuration-and-Power-Interface-(ACPI-)Schnittstelle
acpi	V:6, I:69	49	Werkzeug zur Anzeige von Informationen über ACPI-Geräte
sleepd	V:0.08, I:0.10	84	Daemon, mit dem man einen Laptop bei Inaktivität in Ruhemodus schicken kann
hdparm	V:111, I:192	246	Optimierung des Festplattenzugriffs (lesen Sie dazu Abschnitt 9.6.9)
smartmontools	V:249, I:274	2455	Speichersysteme mittels S.M.A.R.T. kontrollieren und überwachen
setserial	V:3.0, I:4.7	104	Werkzeugsammlung zur Verwaltung von seriellen Ports
memtest86+	V:1, I:19	12483	Werkzeugsammlung zur Speicher-Hardware-Verwaltung
scsitools	V:0.2, I:2.0	261	Werkzeugsammlung zur SCSI-Hardware-Verwaltung
setcd	V:0.06, I:0.59	33	Optimierung des Zugriffs auf CD-Laufwerke
big-cursor	I:0.57	26	größere Maus-Cursor für X

Tabelle 9.14: Liste von Werkzeugen zur Hardware-Konfiguration

Zeiten werden auf einem Debian-System normalerweise in der Lokalzeit angezeigt, aber die Hardware- und Systemzeit verwenden üblicherweise UTC (GMT).

Wenn die Hardware-Zeit (teilweise auch als BIOS-/UEFI- oder CMOS-Uhr bezeichnet) auf UTC eingestellt ist, ändern Sie die Einstellung in "`/etc/default/rcS`" auf "`UTC=yes`".

Mit folgendem Befehl konfigurieren Sie die vom Debian-System verwendete Zeitzone neu:

```
# dpkg-reconfigure tzdata
```

Falls Sie die Systemzeit über das Netzwerk aktualisieren möchten, sollten Sie die Verwendung des NTP-Dienstes in Erwägung ziehen; dazugehörige Pakete sind `systemd-imesyncd`, `ntpsec`, `ntpsec-ntpdate` und `chrony`.

Tipp
Unter systemd sollten Sie stattdessen `systemd-timesyncd` für die Netzwerk-Zeitsynchronisation verwenden. Lesen Sie dazu systemd-timesyncd(8).

Hier finden Sie weitere Informationen:

- Managing Accurate Date and Time HOWTO;
- NTP Public Services Project;
- im `ntpsec-doc`-Paket.

Tipp
ntptrace(1) aus dem `ntpsec`-Paket kann eine Hintereinanderschaltung mehrerer NTP-Server zu deren primärer Quelle zurückverfolgen.

9.5.6 Die Audio-Infrastruktur

Gerätetreiber für Soundkarten werden in aktuellen Linux-Systemen von Advanced Linux Sound Architecture (ALSA) bereitgestellt. ALSA enthält zwecks Kompatibilität auch einen Emulationsmodus für das ältere Open Sound System (OSS).

Anwendungssoftware kann konfiguriert sein, nicht die Soundkarte direkt anzusprechen, sondern über standardisierte Sound-Server-Systeme darauf zuzugreifen. Derzeit werden PulseAudio, JACK und PipeWire als Sound-Server-Systeme genutzt. Schauen Sie im Debian-Wiki unter Sound, wie die aktuelle Situation ist.

Es gibt normalerweise für jede populäre Arbeitsplatzumgebung eine Sound-Engine. Für jede Sound-Engine, die von einer Anwendung verwendet wird, kann ausgewählt werden, mit welchem der verschiedenen Sound-Server sie sich verbindet.

Tipp
Verwenden Sie "`cat /dev/urandom > /dev/audio`" oder speaker-test(1), um die Lautsprecher zu testen (^C zum stoppen).

Tipp
Falls Sie keinen Ton hören, sind Ihre Lautsprecher möglicherweise an einem stummgeschalteten Ausgang angeschlossen. Moderne Sound-Systeme haben oft mehrere Ausgänge. alsamixer(1) aus dem `alsa-utils`-Paket ist nützlich für die Einstellung von Lautstärke und Stummschaltung.

Paket	Popcon	Größe	Beschreibung
`alsa-utils`	V:349, I:457	2700	Werkzeuge zur Konfiguration und Nutzung von ALSA
`oss-compat`	V:0.5, I:8.7	21	Paket für OSS-Kompatibilität, verhindert unter ALSA das Auftreten von "`/dev/dsp not found`"-Fehlern
`pipewire`	V:330, I:370	140	Multimedia-Server - Engine für Audio- und Videoverarbeitung - Metapaket
`pipewire-bin`	V:336, I:370	2254	Multimedia-Server - Engine für Audio- und Videoverarbeitung - Audio-Server und CLI-Programme
`pipewire-alsa`	V:175, I:243	197	Multimedia-Server - Engine für Audio- und Videoverarbeitung - Audio-Server, um ALSA zu ersetzen
`pipewire-pulse`	V:304, I:350	56	Multimedia-Server - Engine für Audio- und Videoverarbeitung - Audio-Server, um PulseAudio zu ersetzen
`pulseaudio`	V:144, I:166	6616	PulseAudio-Server
`libpulse0`	V:445, I:578	984	PulseAudio-Client-Bibliothek
`jackd`	V:2, I:13	8	JACK Audio Connection Kit (JACK)-Server (niedrige Latenzzeiten)
`libjack0`	V:1.7, I:8.3	320	JACK Audio Connection Kit (JACK)-Bibliothek (niedrige Latenzzeiten)
`libphonon4qt5-4`	V:16, I:38	572	Phonon: KDE-Sound-Engine

Tabelle 9.15: Liste von Audio-Paketen

9.5.7 Deaktivieren des Bildschirmschoners

Verwenden Sie folgende Befehle, um den Bildschirmschoner zu deaktivieren:

Umgebung	Befehl
Linux-Konsole	`setterm -powersave off`
X-Window (Bildschirmschoner ausschalten)	`xset s off`
X-Window (dpms deaktivieren)	`xset -dpms`
X-Window (grafische Oberfläche zur Konfiguration des Bildschirmschoners)	`xscreensaver-command -prefs`

Tabelle 9.16: Liste von Befehlen zur Deaktivierung des Bildschirmschoners

9.5.8 Ausschalten von Pieptönen

Man kann immer den Stecker des internen PC-Lautsprechers abziehen, um jegliche Pieptöne loszuwerden, aber auch über das Entfernen des `pcspkr`-Kernel-Moduls kann dies erreicht werden.

Folgender Befehl verhindert, dass das readline(3)-Programm, das von bash(1) genutzt wird, einen Piepton ausgibt, wenn es ein Alarm-Zeichen (ASCII=7) empfängt:

```
$ echo "set bell-style none">> ~/.inputrc
```

9.5.9 Arbeitsspeichernutzung

Es gibt zwei Ressourcen, über die Sie die aktuelle Situation zur Arbeitsspeichernutzung abfragen können:

- Die Boot-Meldungen des Kernels in "`/var/log/dmesg`" enthalten die exakte Größe des verfügbaren Arbeitsspeichers.
- free(1) und top(1) zeigen Informationen über Arbeitsspeicher-Ressourcen auf dem laufenden System an.

Hier ein Beispiel:

```
# grep '\] Memory' /var/log/dmesg
[    0.004000] Memory: 990528k/1016784k available (1975k kernel code, 25868k reserved, 931k ↩
    data, 296k init)
$ free -k
             total       used       free     shared    buffers     cached
Mem:        997184     976928      20256          0     129592     171932
-/+ buffers/cache:     675404     321780
Swap:      4545576          4    4545572
```

Sie fragen sich vielleicht: "Warum sagt dmesg mir, dass 990 MB frei sind, während free -k sagt, 320 MB seien frei. Da fehlen mehr als 600 MB …"

Sorgen Sie sich nicht über den hohen Wert von "`used`" und die kleine Größe von "`free`" in der "`Mem:`"-Zeile; schauen Sie sich stattdessen die Werte darunter an (inklusive Puffer/Cache, 675404 und 321780 in obigem Beispiel) und entspannen Sie sich.

Auf meinem MacBook mit 1GB=1048576k DRAM (das Grafik-System stiehlt einiges davon) sehe ich Folgendes:

9.5.10 Systemsicherheits- und Integritätsüberprüfung

Schlechte Systemwartung könnte Ihr System für Angriffe von extern anfällig machen.

Um eine Systemsicherheits- und Integritätsüberprüfung durchzuführen, sollten Sie mit folgendem beginnen:

- dem `debsums`-Paket, lesen Sie dazu debsums(1) und Abschnitt 2.5.2;

gemeldet	Größe
absolute Größe in dmesg	1016784k = 1GB - 31792k
frei in dmesg	990528k
absolut auf der Shell	997184k
frei auf der Shell	20256k (aber effektiv 321780k)

Tabelle 9.17: Liste der gemeldeten Arbeitsspeichergrößen

- dem `chkrootkit`-Paket, lesen Sie dazu chkrootkit(1);
- der `clamav`-Paketfamilie, lesen Sie dazu clamscan(1) und freshclam(1);
- der Debian Sicherheits-FAQ;
- der Anleitung zum Absichern von Debian.

Paket	Popcon	Größe	Beschreibung
`logcheck`	V:4.2, I:5.2	120	Daemon, um Anomalien in den System-Logdateien per Mail an den Administrator zu melden
`debsums`	V:4, I:29	108	Hilfsprogramm, um installierte Paketdateien anhand von MD5-Prüfsummen zu verifizieren
`chkrootkit`	V:10, I:14	988	rootkit-Erkennungsprogramm
`clamav`	V:8, I:37	20551	Antiviren-Programm für Unix - Befehlszeilenschnittstelle
`tiger`	V:1.0, I:1.3	7800	Sicherheitslücken im System melden
`tripwire`	V:1.1, I:1.4	5060	Integritätsüberprüfung für Dateien und Verzeichnisse
`john`	V:1.1, I:7.1	469	Programm zum aktiven Knacken von Passwörtern
`aide`	V:2.1, I:2.7	328	Fortschrittliche Intrusion-Detection-Umgebung - statische Binärdatei
`integrit`	V:0.07, I:0.16	2759	Dateiintegritäts-Verifizierungsprogramm
`crack`	V:0.04, I:0.52	151	Programm zum Erraten von Passwörtern

Tabelle 9.18: Liste von Werkzeugen für eine Systemsicherheits- und Integritätsüberprüfung

Hier ein einfaches Skript, um nach Dateien mit typischerweise falschen Dateiberechtigungen (schreibbar für alle) zu suchen:

```
# find / -perm 777 -a \! -type s -a \! -type l -a \! \( -type d -a -perm 1777 \)
```



Achtung
Da das `debsums`-Paket MD5-Prüfsummen verwendet, die lokal gespeichert sind, kann ihm als Sicherheits-Audit-Werkzeug gegen bösartige Angriffe nicht vollständig vertraut werden.

9.6 Tipps zur Speicherung von Daten

Das Booten eines Linux-Live-Systems (siehe Abschnitt 3.2.2 und Abschnitt 3.2.3) erlaubt es auf einfache Art, Datenspeicher auf dem installierten System zu rekonfigurieren.

Anmerkung
Aussagen zu Festplatten (HDD) sind auch für andere Speichergeräte, wie SSD / USB-Flash-Speicher / Speicherkarten / usw. anwendbar. Ersetzen Sie in den Beispielen einfach die Gerätenamen wie `/dev/sda` durch passende Namen wie `/dev/nvme0`, `/dev/mmcblk0`, ...

Teilweise werden Geräte vom grafischen Arbeitsplatzsystem automatisch eingebunden. Solche Geräte müssen Sie dann händisch über die Befehlszeile vom System trennen (mittels umount(8)), bevor Sie daran arbeiten können.

9.6.1 Verwendung des Plattenplatzes

Der verwendete Plattenplatz kann mit Programmen aus den Paketen `mount`, `coreutils` und `xdu` beurteilt werden:

- mount(8) gibt Informationen über alle eingebundenen Dateisysteme (= ”Laufwerke”) aus.
- df(1) gibt Informationen über den verwendeten Plattenplatz für das Dateisystem aus.
- du(1) gibt Informationen über den verwendeten Plattenplatz für den Verzeichnisbaum aus.

Tipp
Mit ”`du -k . |xdu`”, ”`sudo du -k -x / |xdu`” usw. können Sie die Ausgabe von du(8) an xdu(1) weiterleiten, um eine grafische und interaktive Präsentation zu erhalten.

9.6.2 Konfiguration der Festplattenpartitionen

Bei der Konfiguration von Laufwerkspartitionen können Sie erwägen, parted(8) zu verwenden (obwohl fdisk(8) lange als Standard angesehen wurde). Die Begriffe ”Partitionstabelle”, ”Disk partitioning data”, ”Partition map” und ”Disk label” werden alle gleichbedeutend verwendet.

Ältere PCs verwenden das klassische Schema des Master Boot Record (MBR) und legen die Partitionstabelle im ersten Sektor, also im LBA-Sektor 0 (512 Byte) ab.

Aktuelle PCs mit Unified Extensible Firmware Interface (UEFI), inklusive Intel-basierten Macs, verwenden teilweise das GUID Partition Table (GPT)-Schema, bei dem die Partitionstabelle nicht im ersten Sektor liegt.

Obwohl fdisk(8) lange Zeit das Standardwerkzeug zur Laufwerkspartitionierung war, ist parted(8) dabei, ihm den Rang abzulaufen.

Paket	Popcon	Größe	Beschreibung
`util-linux`	V:922, I:1000	4826	verschiedene System-Werkzeuge inklusive fdisk(8) und cfdisk(8)
`parted`	V:443, I:566	126	GNU Parted - Programm zum Anlegen / Ändern von Partitionen
`gparted`	V:12, I:81	2313	GNOME Partitionseditor, basierend auf `libparted`
`gdisk`	V:20, I:240	967	Partitionseditor für GPT/MBR-Hybrid-Laufwerke
`kpartx`	V:16, I:23	78	Programm zur Erzeugung von Gerätezuordnungen (device mappings) für Partitionen

Tabelle 9.19: Liste von Paketen für die Partitionierung



Achtung
Obwohl parted(8) vorgibt, Dateisysteme erzeugen und auch nachträglich in der Größe verändern zu können, ist es sicherer, für solche Dinge gut betreute und spezialisierte Werkzeuge wie mkfs(8) (mkfs.msdos(8), mkfs.ext2(8), mkfs.ext3(8), mkfs.ext4(8), …) und resize2fs(8) zu verwenden.

Anmerkung
Um zwischen GPT und MBR zu wechseln, müssen Sie die ersten paar Blöcke der Platte löschen (lesen Sie dazu Abschnitt 9.8.6) und "`parted /dev/sdx mklabel gpt`" oder "`parted /dev/sdx mklabel msdos`" nutzen, um die gewünschte Einstellung neu zu setzen. Bitte beachten Sie, dass hierbei "`msdos`" für MBR verwendet wird.

9.6.3 Zugriff auf Partitionen über die UUID-Kennung

Obwohl aufgrund der Neukonfiguration einer Partition oder der Aktivierungsreihenfolge von Wechseldatenträgern Laufwerke von einem zum anderen Mal unterschiedliche Laufwerksnamen haben können, gibt es auch eine Möglichkeit, konsistente Bezeichnungen für den Laufwerkszugriff zu verwenden. Dies ist ebenfalls hilfreich, wenn Sie mehrere Festplatten haben und Ihr BIOS/UEFI die Gerätenamen nicht jedes Mal identisch zuweist.

- mount(8) kann über die Option "`-U`" ein blockorientiertes Gerät mittels seiner UUID-Kennung einbinden, statt dessen Gerätedateinamen (wie z.B. "`/dev/sda3`") zu nutzen.
- "`/etc/fstab`" (lesen Sie hierzu auch fstab(5)) kann UUID verwenden.
- Auch Bootloader (Abschnitt 3.1.2) können unter Umständen die UUID benutzen.

Tipp
Sie können die UUID eines blockorientierten Gerätes mit blkid(8) abfragen.
Sie können die UUID und weitere Informationen auch mit "`lsblk -f`" abfragen.

9.6.4 LVM2

LVM2 ist ein Logical Volume Manager für den Linux-Kernel. Mit LVM2 können Partitionen auf logischen Volumes erzeugt werden statt auf physikalischen Festplatten.

LVM erfordert folgendes:

- device-mapper-Unterstützung im Linux-Kernel (Standardeinstellung für Debian-Kernel);
- die Userspace-Bibliothek zur Unterstützung von device-mapper (`libdevmapper*`-Paket);
- die Userspace-LVM2-Werkzeuge (`lvm2`-Paket).

Lehrreiche Informationen über LVM2 finden Sie mittels folgender Handbuchseiten:

- lvm(8): Grundlagen des LVM2-Mechanismus' (Liste aller LVM2-Befehle);
- lvm.conf(5): Konfigurationsdatei für LVM2;
- lvs(8): Informationen über logische Volumes ausgeben;
- vgs(8): Informationen über Volume-Gruppen ausgeben;
- pvs(8): Informationen über physikalische Volumes ausgeben.

9.6.5 Konfiguration von Dateisystemen

Für das ext4-Dateisystem enthält das `e2fsprogs`-Paket folgendes:

- mkfs.ext4(8) zur Erzeugung neuer ext4-Dateisysteme;
- fsck.ext4(8) zur Überprüfung und Reparatur vorhandener ext4-Dateisysteme;
- tune2fs(8) zur Konfiguration des Superblocks von ext4-Dateisystemen;
- debugfs(8) für interaktive Fehlersuche in ext4-Dateisystemen (es enthält den `undel`-Befehl, um gelöschte Dateien wiederherzustellen).

Die Befehle mkfs(8) und fsck(8) werden durch das `e2fsprogs`-Paket bereitgestellt und sind Frontends für Dateisystem-abhängige Programme (`mkfs.`*`dateisystemtyp`* und `fsck.`*`dateisystemtyp`*). Für ext4 sind das mkfs.ext4(8) und fsck.ext4(8) (dies sind symbolischer Link auf mke2fs(8) und e2fsck(8)).

Ähnliche Befehle sind für jedes von Linux unterstützte Dateisystem verfügbar.

Paket	Popcon	Größe	Beschreibung
`e2fsprogs`	V:842, I:998	1560	Hilfsprogramme für ext2/ext3/ext4-Dateisysteme
`btrfs-progs`	V:54, I:78	5291	Hilfsprogramme für das Btrfs-Dateisystem
`reiserfsprogs`	V:12, I:21	473	Hilfsprogramme für das Reiserfs-Dateisystem
`zfsutils-linux`	V:32, I:33	1905	Hilfsprogramme für das OpenZFS-Dateisystem
`dosfstools`	V:274, I:568	310	Hilfsprogramme für das FAT-Dateisystem (Microsoft: MS-DOS, Windows)
`exfatprogs`	V:36, I:466	624	Hilfsprogramme für das von Samsung betreute exFAT-Dateisystem
`exfat-fuse`	V:2, I:37	73	Treiber für das Lesen/Schreiben des exFAT-Dateisystems (Microsoft) über FUSE
`xfsprogs`	V:46, I:82	4651	Hilfsprogramme für das XFS-Dateisystem (SGI: IRIX)
`ntfs-3g`	V:197, I:510	1535	Treiber für das Lesen/Schreiben des NTFS-Dateisystems (Microsoft: Windows NT, ...) über FUSE
`jfsutils`	V:0.4, I:6.6	1104	Hilfsprogramme für das JFS-Dateisystem (IBM: AIX, OS/2)
`xorriso`	V:16, I:66	378	Hilfsprogramme für das ISO-9660-Dateisystem sowie zum CD/DVD-Brennen über libburnia
`wodim`	V:7, I:81	898	Befehlszeilen-Werkzeug zum Brennen von CD/DVD über cdrkit
`genisoimage`	V:17, I:150	1567	Befehlszeilen-Werkzeug für das ISO-9660-Dateisystem über cdrkit
`hfsprogs`	V:0.3, I:3.0	394	Hilfsprogramme für HFS- und HFS Plus-Dateisysteme (Apple: Mac OS)
`zerofree`	V:9, I:114	30	Programm, um freie Blöcke auf ext2/3/4-Dateisystemen mit Nullen zu überschreiben

Tabelle 9.20: Liste von Paketen für das Dateisystem-Management

Tipp
Ext4 ist das Standard-Dateisystem für Linux-Systeme und Sie sollten dies verwenden, außer Sie haben einen bestimmten Grund, ein anderes zu nutzen.
Den aktuellen Status zu Btrfs finden Sie im Debian Wiki und im kernel.org Wiki. Man geht davon aus, dass dies nach ext4 das neue Standard-Dateisystem wird.
Einige Werkzeuge erlauben den Zugriff auf Dateisysteme ohne entsprechende Unterstützung im Linux-Kernel (lesen Sie dazu Abschnitt 9.8.2).

9.6.6 Dateisystemerzeugung und Integritätsüberprüfung

Der mkfs(8)-Befehl erzeugt unter Linux ein Dateisystem. Mit fsck(8) führen Sie eine Integritätsüberprüfung oder Reparatur des Dateisystems durch.

Standardmäßig gibt es jetzt auf Debian-Systemen nach der Erzeugung des Dateisystems keinen periodischen Dateisystem-Check (`fsck`) mehr.



Achtung
Es ist grundsätzlich nicht sicher, `fsck` auf **eingebundenen Dateisystemen** laufen zu lassen.

Tipp
Sie können den fsck(8)-Befehl auf allen Dateisystemen inklusive dem root-Dateisystem (/) gefahrlos beim Reboot ausführen, indem Sie "`enable_periodic_fsck`" in "`/etc/mke2fs.conf`" setzen und den Max-Wert für die mount-Vorgänge mittels "`tune2fs -c0 /dev/partitionsname`" auf 0 setzen. Details finden Sie in mke2fs.conf(5) und tune2fs(8).
In "`/var/log/fsck/`" finden Sie Ergebnisse von dem fsck(8)-Lauf, der bei jedem Systemstart durch das Boot-Skript durchgeführt wird.

9.6.7 Optimierung von Dateisystemen über mount-Optionen

Die grundlegende statische Dateisystem-Konfiguration wird in "`/etc/fstab`" festgelegt. Zum Beispiel:

```
«file system»                    «mount point» «type» «options»    «dump» «pass»
proc                                     /proc proc   defaults          0 0
UUID=709cbe4c-80c1-56db-8ab1-dbce3146d2f7 /     ext4   errors=remount-ro 0 1
UUID=817bae6b-45d2-5aca-4d2a-1267ab46ac23 none  swap   sw                0 0
/dev/scd0                         /media/cdrom0  udf,iso9660 user,noauto  0 0
```

Tipp
Die UUID (Näheres in Abschnitt 9.6.3) kann statt der normalen Namen für blockorientierte Geräte wie "`/dev/sda1`", "`/dev/sda2`" … verwendet werden, um das Gerät zu identifizieren.

Seit Linux 2.6.30 ist das Standardverhalten die "`relatime`"-Option.

Lesen Sie fstab(5) und mount(8).

9.6.8 Optimierung von Dateisystemen über den Superblock

Die Charakteristik eines Dateisystems kann über seinen Superblock optimiert werden; verwenden Sie dazu den Befehl tune2fs(8):

- "`sudo tune2fs -l /dev/sda1`" zeigt den Inhalt des Dateisystem-Superblocks auf "`/dev/sda1`" an.
- "`sudo tune2fs -c 50 /dev/sda1`" ändert den Intervall zur Überprüfung des Dateisystems (Ausführung von `fsck` während des Systemstarts) für "`/dev/sda1`" auf jeden 50. Start.
- "`sudo tune2fs -j /dev/sda1`" fügt Journalfunktionalität zum Dateisystem hinzu; das bedeutet, dass ein ext2-Dateisystem auf "`/dev/sda1`" nach ext3 konvertiert wird. (Führen Sie dies nur bei einem nicht eingebundenen Dateisystem durch.)

- "`sudo tune2fs -O extents,uninit_bg,dir_index /dev/sda1 && fsck -pf /dev/sda1`" konvertiert ein ext3-Dateisystem auf "`/dev/sda1`" nach ext4. (Führen Sie dies nur bei einem nicht eingebundenen Dateisystem durch.)

Tipp
Ungeachtet seines Namens funktioniert tune2fs(8) nicht nur für ext2-Dateisysteme, sondern auch für ext3 und ext4.

9.6.9 Optimierung der Festplatte



Warnung
Bitte überprüfen Sie Ihre Hardware und lesen Sie die Handbuchseite von hdparm(8), bevor Sie mit der Festplattenkonfiguration herumspielen, da dies ziemlich gefährlich für die Datenintegrität sein kann.

Sie können die Zugriffsgeschwindigkeit einer Festplatte testen, für "`/dev/sda`" z.B. mit "`hdparm -tT /dev/sda`". Bei einigen Festplatten, die über (E)IDE angeschlossen sind, kann diese über "`hdparm -q -c3 -d1 -u1 -m16 /dev/sda`" erhöht werden; dabei wird "(E)IDE 32-Bit I/O-Unterstützung" aktiviert, außerdem das "using_dma-Flag" und das "interrupt-unmask-Flag" gesetzt sowie die (gefährliche!) "Multiple 16 Sector I/O"-Einstellung aktiviert.

Sie können die Cache-Funktionalität für das Schreiben auf eine Festplatte testen, für "`/dev/sda`" z.B. mit "`hdparm -W /dev/sda`". Mit "`hdparm -W 0 /dev/sda`" deaktivieren Sie diese Funktion.

Im Falle von Problemen beim Lesen von schlecht gepressten CD-ROMs in modernen Hochgeschwindigkeits-CD-ROM-Laufwerken können Sie diese möglicherweise trotzdem lesen, indem Sie mit "`setcd -x 2`" die Geschwindigkeit herabsetzen.

9.6.10 Optimierung von Solid State Disks

Eine Solid State Disk (SSD) wird mittlerweile automatisch detektiert.

Reduzieren Sie unnötige Laufwerkszugriffe, um die Laufwerksabnutzung zu minimieren, indem Sie "`tmpfs`" für das Einbinden schnell veränderlicher Daten in Ihrer `/etc/fstab` verwenden.

9.6.11 SMART verwenden, um Festplattenausfälle vorherzusehen

Sie können Ihre Festplatte mit dem smartd(8)-Daemon überwachen und protokollieren, sofern diese mit dem SMART-Standard kompatibel ist:

1. Aktivieren Sie die SMART-Funktionalität im BIOS.
2. Installieren Sie das Paket `smartmontools`.
3. Identifizieren Sie Ihre Festplatten, indem Sie sie mit df(1) auflisten.
   - Wir gehen hier davon aus, dass die zu überwachende Festplatte als "`/dev/sda`" auftaucht.
4. Überprüfen Sie die Ausgabe von "`smartctl -a /dev/sda`", um festzustellen, ob die SMART-Funktionalität derzeit wirklich aktiv ist.
   - Falls nicht, aktivieren Sie sie mit "`smartctl -s on -a /dev/sda`".
5. Aktivieren Sie den smartd(8)-Daemon wie folgt:

- Entfernen Sie das Kommentarzeichen vor "`start_smartd=yes`" in der Datei "`/etc/default/smartmontools`".
- Führen Sie einen Neustart des smartd(8)-Daemons über "`sudo systemctl restart smartmontools`" durch.

Tipp
Die Einstellungen des smartd(8)-Daemons können über die `/etc/smartd.conf`-Datei angepasst werden; dazu gehört auch die Einstellung, auf welchem Wege Sie über Warnungen informiert werden möchten.

9.6.12 Angeben eines Verzeichnisses für temporäre Dateien über $TMPDIR

Anwendungen erzeugen temporäre Dateien normalerweise unterhalb des temporären Verzeichnisses "`/tmp`". Falls "`/tmp`" nicht genug freien Speicherplatz bietet, können Sie für Anwendungen, die sich diesbezüglich korrekt verhalten, auch mittels der `$TMPDIR`-Variable festlegen, welches Verzeichnis für solche temporären Daten genutzt werden soll.

9.6.13 Vergrößerung des nutzbaren Speicherplatzes mittels LVM

Partitionen, die bei der Installation über den Logical Volume Manager (LVM) (Linux-Funktionalität) erzeugt wurden, können einfach und ohne größere System-Neukonfiguration in der Größe verändert werden, indem Speicherplatz hinzugefügt oder entfernt wird, und zwar über die Grenzen einzelner Laufwerke hinweg.

9.6.14 Vergrößerung des nutzbaren Speicherplatzes über das Einbinden anderer Partitionen

Wenn Sie eine leere Partition (z.B. "`/dev/sdx`") haben, können Sie sie mit mkfs.ext4(1) formatieren und dann mit mount(8) in ein Verzeichnis einbinden, in dem Sie mehr Platz benötigen (Sie müssen die originalen Daten kopieren):

```
$ sudo mv work-dir old-dir
$ sudo mkfs.ext4 /dev/sdx
$ sudo mount -t ext4 /dev/sdx work-dir
$ sudo cp -a old-dir/* work-dir
$ sudo rm -rf old-dir
```

Tipp
Alternativ können Sie eine leere Festplatten-Image-Datei (lesen Sie dazu Abschnitt 9.7.5) als loop-device einbinden (Näheres dazu in Abschnitt 9.7.3). Die reelle Größe des Festplatten-Images wächst mit den wirklich darin abgelegten Daten.

9.6.15 Vergrößerung des nutzbaren Speicherplatzes, indem ein anderes Verzeichnis mit "mount --bind" eingebunden wird

Wenn Sie ein leeres Verzeichnis (z.B. "`/pfad/zu/leer`") auf einer anderen Partition haben, auf der noch Platz frei ist, können Sie dieses Verzeichnis mit der mount-Option "`--bind`" in ein anderes Verzeichnis (z.B. "`arbeit`") einbinden, in dem Sie mehr Speicherplatz benötigen:

```
$ sudo mount --bind /path/to/emp-dir work-dir
```

9.6.16 Vergrößerung des nutzbaren Speicherplatzes, indem ein anderes Verzeichnis mit ”Overlay-mounting” eingebunden wird

Wenn Sie freien Platz auf einer anderen Partition haben (mit 2 Verzeichnissen wie ”`/pfad/zu/leer`” und ”`/pfad/zu/arbe`
können Sie dort ein Verzeichnis erstellen und dieses einem anderen alten Verzeichnis (z.B. ”`/pfad/zu/alt`”) ”überstülpen”, in dem Sie mehr Platz benötigen. Dies wird ermöglicht durch die OverlayFS-Funktionalität im Linux-Kernel 3.18 oder neuer (ab Debian Stretch 9.0).

```
$ sudo mount -t overlay overlay \
  -olowerdir=/path/to/old-dir,upperdir=/path/to/empty,workdir=/path/to/work
```

Hierbei sollten ”`/pfad/zu/leer`” und ”`/pfad/zu/nutzverzeichnis`” auf einer schreibbaren (RW) Partition liegen, um ”`/pfad/zu/alt`” zu überlagern.

9.6.17 Vergrößerung des nutzbaren Speicherplatzes über einen symbolischen Link



Achtung
Diese Methode ist überholt. Manche Software könnte nicht korrekt funktionieren mit ”symbolischen Links auf ein Verzeichnis”. Verwenden Sie stattdessen einen der oben beschriebenen ”mounting”-basierten Ansätze.

Wenn Sie ein leeres Verzeichnis (z.B. ”`/pfad/zu/leer`”) auf einer anderen Partition haben, auf der noch Platz frei ist, können Sie mit ln(8) einen symbolischen Link zu einem anderen Verzeichnis erstellen:

```
$ sudo mv work-dir old-dir
$ sudo mkdir -p /path/to/emp-dir
$ sudo ln -sf /path/to/emp-dir work-dir
$ sudo cp -a old-dir/* work-dir
$ sudo rm -rf old-dir
```



Warnung
Verwenden Sie solch einen symbolischen Link auf ein Verzeichnis nicht für Verzeichnisse, die von dem System verwaltet werden, wie z.B. ”`/opt`”. Solch ein Link könnte bei einer Systemhochrüstung überschrieben werden.

9.7 Das Festplatten-Abbild

Hier wird die Veränderung eines Festplatten-Abbilds behandelt.

9.7.1 Erzeugung der Festplatten-Abbild-Datei

Ein Festplatten-Abbild (”`disk.img`”) eines nicht eingebundenen Gerätes, z.B. der zweiten SCSI- oder Serial-ATA-Festplatte ”`/dev/sdb`”, kann mit einem der folgenden Befehle erzeugt werden:

```
# dd if=/dev/sdb of=disk.img; sync
```

```
# cp /dev/sdb disk.img ; sync
```

```
# cat /dev/sdb > disk.img ; sync
```

Ein Abbild des Master Boot Record (MBR), wie er in traditionellen PCs verwendet wird (lesen Sie auch Abschnitt 9.6.2) und der im ersten Sektor der primären IDE-Festplatte abgelegt ist, kann mit dd(1) erstellt werden, wie hier:

```
# dd if=/dev/sda of=mbr.img bs=512 count=1
# dd if=/dev/sda of=mbr-nopart.img bs=446 count=1
# dd if=/dev/sda of=mbr-part.img skip=446 bs=1 count=66
```

- "`mbr.img`": der MBR mit der Partitionstabelle
- "`mbr-nopart.img`": der MBR ohne Partitionstabelle
- "`mbr-part.img`": die Partitionstabelle nur vom MBR

Falls Sie ein Abbild einer einzelnen Partition der Festplatte erstellen möchten, ersetzen Sie "`/dev/sda`" z.B. durch "`/dev/sda1`".

9.7.2 Direkt auf eine Festplatte schreiben

Das Festplatten-Abbild "`disk.img`" kann mit einem der folgenden Befehle auf ein nicht eingebundenes Laufwerk passender Größe (hier z.B. auf die zweite SCSI- oder Serial-ATA-Festplatte "`/dev/sdb`") geschrieben werden:

```
# dd if=disk.img of=/dev/sdb ; sync
```

```
# cp disk.img /dev/sdb ; sync
```

```
# cat disk.img >/dev/sdb ; sync
```

Ähnlich dazu kann mit folgendem Befehl die Abbild-Datei "`partition.img`" einer einzelnen Partition auf eine nicht eingebundene Partition passender Größe (hier z.B. die erste Partition der zweiten SCSI- oder Serial-ATA-Festplatte "`/dev/sdb1`") geschrieben werden:

```
# dd if=partition.img of=/dev/sdb1 ; sync
```

9.7.3 Einbinden der Festplatten-Abbild-Datei

Das Festplatten-Abbild "`partition.img`", welches ein einfaches Partitions-Abbild enthält, kann mittels einem loop device wie folgt eingebunden und anschließend wieder gelöst werden:

```
# losetup --show -f partition.img
/dev/loop0
# mkdir -p /mnt/loop0
# mount -t auto /dev/loop0 /mnt/loop0
...hack...hack...hack
# umount /dev/loop0
# losetup -d /dev/loop0
```

Das kann noch weiter vereinfacht werden:

```
# mkdir -p /mnt/loop0
# mount -t auto -o loop partition.img /mnt/loop0
...hack...hack...hack
# umount partition.img
```

Jede Partition des Festplatten-Abbilds "`disk.img`", das mehrere Partitionen enthält, kann mittels loop device eingebunden werden.

```
# losetup --show -f -P disk.img
/dev/loop0
# ls -l /dev/loop0*
brw-rw---- 1 root disk   7,  0 Apr  2 22:51 /dev/loop0
brw-rw---- 1 root disk 259, 12 Apr  2 22:51 /dev/loop0p1
brw-rw---- 1 root disk 259, 13 Apr  2 22:51 /dev/loop0p14
brw-rw---- 1 root disk 259, 14 Apr  2 22:51 /dev/loop0p15
# fdisk -l /dev/loop0
Disk /dev/loop0: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 6A1D9E28-C48C-2144-91F7-968B3CBC9BD1

Device         Start     End Sectors  Size Type
/dev/loop0p1  262144 4192255 3930112  1.9G Linux root (x86-64)
/dev/loop0p14   2048    8191    6144    3M BIOS boot
/dev/loop0p15   8192  262143  253952  124M EFI System

Partition table entries are not in disk order.
# mkdir -p /mnt/loop0p1
# mkdir -p /mnt/loop0p15
# mount -t auto /dev/loop0p1 /mnt/loop0p1
# mount -t auto /dev/loop0p15 /mnt/loop0p15
# mount |grep loop
/dev/loop0p1 on /mnt/loop0p1 type ext4 (rw,relatime)
/dev/loop0p15 on /mnt/loop0p15 type vfat (rw,relatime,fmask=0002,dmask=0002,allow_utime ↩
    =0020,codepage=437,iocharset=ascii,shortname=mixed,utf8,errors=remount-ro)
...hack...hack...hack
# umount /dev/loop0p1
# umount /dev/loop0p15
# losetup -d /dev/loop0
```

Alternativ können Sie ähnliche Funktionalitäten auch erreichen, indem Sie wie hier die device-mapper-Geräte nutzen, die von kpartx(8) aus dem `kpartx`-Paket erzeugt werden:

```
# kpartx -a -v disk.img
add map loop0p1 (253:0): 0 3930112 linear 7:0 262144
add map loop0p14 (253:1): 0 6144 linear 7:0 2048
add map loop0p15 (253:2): 0 253952 linear 7:0 8192
# fdisk -l /dev/loop0
Disk /dev/loop0: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 6A1D9E28-C48C-2144-91F7-968B3CBC9BD1

Device         Start     End Sectors  Size Type
/dev/loop0p1  262144 4192255 3930112  1.9G Linux root (x86-64)
/dev/loop0p14   2048    8191    6144    3M BIOS boot
/dev/loop0p15   8192  262143  253952  124M EFI System

Partition table entries are not in disk order.
# ls -l /dev/mapper/
total 0
crw------- 1 root root 10, 236 Apr  2 22:45 control
lrwxrwxrwx 1 root root       7 Apr  2 23:19 loop0p1 -> ../dm-0
lrwxrwxrwx 1 root root       7 Apr  2 23:19 loop0p14 -> ../dm-1
lrwxrwxrwx 1 root root       7 Apr  2 23:19 loop0p15 -> ../dm-2
```

```
# mkdir -p /mnt/loop0p1
# mkdir -p /mnt/loop0p15
# mount -t auto /dev/mapper/loop0p1 /mnt/loop0p1
# mount -t auto /dev/mapper/loop0p15 /mnt/loop0p15
# mount |grep loop
/dev/loop0p1 on /mnt/loop0p1 type ext4 (rw,relatime)
/dev/loop0p15 on /mnt/loop0p15 type vfat (rw,relatime,fmask=0002,dmask=0002,allow_utime ↩
    =0020,codepage=437,iocharset=ascii,shortname=mixed,utf8,errors=remount-ro)
...hack...hack...hack
# umount /dev/mapper/loop0p1
# umount /dev/mapper/loop0p15
# kpartx -d disk.img
```

9.7.4 Eine Festplatten-Abbild-Datei bereinigen

Eine Festplatten-Abbild-Datei (hier ”`disk.img`”) kann wie folgt von allen gelöschten Dateien bereinigt und in eine gesäuberte kompakte Abbild-Datei ”`new.img`” geschrieben werden:

```
# mkdir old; mkdir new
# mount -t auto -o loop disk.img old
# dd bs=1 count=0 if=/dev/zero of=new.img seek=5G
# mount -t auto -o loop new.img new
# cd old
# cp -a --sparse=always ./ ../new/
# cd ..
# umount new.img
# umount disk.img
```

Wenn ”`disk.img`” auf einem ext2-, ext3- oder ext4-Dateisystem liegt, können Sie auch zerofree(8) aus dem `zerofree`-Paket verwenden, wie hier gezeigt:

```
# losetup --show -f disk.img
/dev/loop0
# zerofree /dev/loop0
# cp --sparse=always disk.img new.img
# losetup -d /dev/loop0
```

9.7.5 Eine leere Abbild-Datei erstellen

Ein leeres Festplatten-Abbild ”`disk.img`”, das bis zu einer Größe von 5 GiB anwachsen kann, erzeugen Sie mit dd(1) wie folgt:

```
$ dd bs=1 count=0 if=/dev/zero of=disk.img seek=5G
```

Statt dd(1) kann hier auch das dafür spezialisierte fallocate(8) verwendet werden.

Mittels eines loop device erstellen Sie auf diesem Festplatten-Abbild ”`disk.img`” wie folgt ein ext4-Dateisystem:

```
# losetup --show -f disk.img
/dev/loop0
# mkfs.ext4 /dev/loop0
...hack...hack...hack
# losetup -d /dev/loop0
$ du  --apparent-size -h disk.img
5.0G  disk.img
$ du -h disk.img
83M disk.img
```

Die Dateigröße von ”disk.img” ist 5.0 GiB, aber der aktuell von ihm verwendete Speicherplatz ist lediglich 83 MiB. Diese Diskrepanz ist möglich, da ext4 die Verwendung von Sparse-Dateien unterstützt.

Tipp
Der wirklich genutzte Speicherplatz von Sparse-Dateien wächst mit den Daten, die in diese hineingeschrieben werden.

Die Nutzung von Befehlen ähnlich denen aus Abschnitt 9.7.3 auf Geräten, die von loop device oder device-mapper-Geräten erzeugt wurden, erlaubt Ihnen ebenfalls, das Abbild ”disk.img” über parted(8) oder fdisk(8) zu erzeugen, sowie Dateisysteme mit mkfs.ext4(8), mkswap(8) usw. zu erstellen.

9.7.6 Erstellen einer ISO9660-Abbild-Datei

Tipp
Die beiden Programme genisoimage(1) (aus dem cdrkit-Paket) und xorrisofs(1) (aus dem Libburnia-Paket) teilen die gleiche Befehlssyntax, abgesehen von dem Befehlsnamen.

Eine ISO9660-Abbild-Datei ”cd.iso” aus einem Quellverzeichnisbaum (hier in ”quell_verzeichnis”) kann mit genisoimage(1) aus dem cdrkit-Paket erstellt werden:

```
#  genisoimage -r -J -T -V volume_id -o cd.iso source_directory
```

Ähnlich dazu kann ein boot-fähiges ISO9660-Abbild ”cdboot.iso” aus einem Verzeichnisbaum (”quell_verzeichnis”) erstellt werden, der dem des debian-installer ähnelt:

```
#  genisoimage -r -o cdboot.iso -V volume_id \
   -b isolinux/isolinux.bin -c isolinux/boot.cat \
   -no-emul-boot -boot-load-size 4 -boot-info-table source_directory
```

Hier wird der Isolinux-Bootloader zum Booten verwendet (Näheres in Abschnitt 3.1.2).

Sie können die Berechnung der md5sum-Prüfsumme und die Erstellung eines ISO9660-Abbilds direkt von der CD-ROM durchführen, wie hier gezeigt:

```
$ isoinfo -d -i /dev/cdrom
CD-ROM is in ISO 9660 format
...
Logical block size is: 2048
Volume size is: 23150592
...
# dd if=/dev/cdrom bs=2048 count=23150592 conv=notrunc,noerror | md5sum
# dd if=/dev/cdrom bs=2048 count=23150592 conv=notrunc,noerror > cd.iso
```



Warnung
Sie müssen wie oben gezeigt dafür sorgen, dass Sie nicht von einem Fehler im Linux-Kernel (”ISO9660 filesystem read ahead bug”) betroffen werden; nur so können Sie korrekte Ergebnisse erhalten.

9.7.7 Direkt auf die CD/DVD-R/RW schreiben

Tipp
Eine DVD ist lediglich eine grosse CD für wodim(1) (aus dem cdrkit) und xorrecord(1) (aus dem Libburnia-Paket). Diese Befehle teilen die gleiche Befehlssyntax, abgesehen von dem Befehlsnamen.

Sie finden nutzbare Laufwerke wie folgt:

```
# wodim --devices
```

Dann wird die leere CD-R in das Laufwerk eingelegt und die ISO9660-Abbild-Datei "`cd.iso`" wird wie folgt mit wodim(1) auf das Laufwerk (hier "`/dev/sda`") geschrieben:

```
# wodim -v -eject dev=/dev/sda cd.iso
```

Falls statt der CD-R eine CD-RW genutzt wird, verwenden Sie dies:

```
# wodim -v -eject blank=fast dev=/dev/sda cd.iso
```

Tipp
Wenn Ihre Arbeitsplatzumgebung CDs automatisch einbindet, lösen Sie die Einbindung mit "`sudo umount /dev/sda`" über eine Konsole, bevor Sie den wodim(1)-Befehl ausführen.

9.7.8 Einbinden einer ISO9660-Abbild-Datei

Wenn "`cd.iso`" ein ISO9660-Abbild enthält, können Sie es wie folgt in "`/cdrom`" einbinden:

```
# mount -t iso9660 -o ro,loop cd.iso /cdrom
```

Tipp
Moderne Arbeitsplatzsysteme binden Wechseldatenträger (wie mit ISO9660 formatierte CDs) automatisch ein (lesen Sie dazu Abschnitt 10.1.7).

9.8 Binärdaten

Hier behandeln wir das Bearbeiten von Binärdaten auf einem Speichermedium.

9.8.1 Betrachten und Bearbeiten von Binärdaten

Die grundlegendste Methode zum Betrachten von Binärdaten ist die Verwendung des Befehls "`od -t x1`".

Tipp
HEX wird als Acronym für das Hexadezimal-Format mit einer Basis (Radix) von 16 verwendet. OKTAL steht für das Oktal-Format mit einer Basis von 8. ASCII ist der American Standard Code for Information Interchange, also für normalen englischsprachigen Text-Code. EBCDIC steht für den Extended Binary Coded Decimal Interchange Code, der auf IBM Mainframe-Betriebssystemen verwendet wird.

Paket	Popcon	Größe	Beschreibung
coreutils	V:914, I:1000	19751	grundlegendes Paket, das od(1) für die Ausgabe von Dateien (HEX, ASCII, OCTAL, …) enthält
bsdmainutils	V:4, I:112	17	Hilfspaket, das hd(1) für die Ausgabe von Dateien (HEX, ASCII, OCTAL, …) enthält
hexedit	V:0.7, I:7.4	70	Editor und Betrachter für Binärdateien (HEX, ASCII)
ghex	V:0.7, I:7.2	2525	voll ausgestatteter Hexadezimal-Editor (GNOME)
okteta	V:1, I:11	1591	voll ausgestatteter Hexadezimal-Editor (KDE4)
ncurses-hexedit	V:0.1, I:1.1	130	Editor und Betrachter für Binärdateien (HEX, ASCII, EBCDIC)

Tabelle 9.21: Liste von Paketen zum Betrachten und Bearbeiten von Binärdaten

9.8.2 Dateien bearbeiten, ohne das Dateisystem einzuhängen

Es gibt Werkzeuge, mit denen sich Dateien lesen und schreiben lassen, ohne die Dateisysteme einzubinden.

Paket	Popcon	Größe	Beschreibung
mtools	V:7, I:51	404	Dienstprogramme für MSDOS-Dateisysteme, ohne diese einzubinden
hfsplus	V:2, I:20	248	Dienstprogramme für HFS+-Dateisysteme, ohne diese einzubinden

Tabelle 9.22: Liste von Paketen zur Bearbeitung von Dateien, ohne die Dateisysteme einzuhängen

9.8.3 Datenredundanz

Software RAID-Systeme, bereitgestellt durch den Linux-Kernel, bieten Datenredundanz auf Ebene des Kernel-Dateisystems und erreichen so eine sehr hohe Zuverlässigkeit der Datenspeicherung.

Es gibt auch Werkzeuge, die Datenredundanz für Dateien auf Ebene der Anwendungen ermöglichen und so ebenfalls eine sehr zuverlässige Datenspeicherung erlauben:

Paket	Popcon	Größe	Beschreibung
par2	V:10, I:115	965	Parity Archive Volume Set, für die Dateiüberprüfung und -reparatur
dvdisaster	V:0.1, I:1.1	1422	Schutz vor Datenverlust/Kratzern/Alterung von CD-/DVD-Medien
dvbackup	I:0.04	413	Backup-Werkzeug, das MiniDV-Camcorder verwendet

Tabelle 9.23: Liste von Werkzeugen, um Redundanz für Dateien hinzuzufügen

9.8.4 Datenwiederherstellung und forensische Analyse

Hier einige Programme für Datenwiederherstellung und forensische Analysen:

Tipp
Sie können gelöschte Dateien auf einem ext2-Dateisystem wiederherstellen, indem Sie `list_deleted_inodes` und den `undel`-Befehl von debugfs(8) (aus dem `e2fsprogs`-Paket) verwenden.

Paket	Popcon	Größe	Beschreibung
testdisk	V:2, I:25	529	Hilfsprogramm zum Scannen von Partitionen und Wiederherstellen von Daten
magicrescue	V:0.2, I:1.6	257	Hilfsprogramm zur Dateiwiederherstellung mittels Suche nach ”magischen Bytes”
scalpel	V:0.2, I:2.2	89	ein sparsamer, sehr leistungsfähiger Datei-Carver
myrescue	V:0.2, I:1.9	81	Daten von beschädigten Festplatten retten
extundelete	V:0.4, I:7.1	152	Hilfsprogramm, um gelöschte Dateien von ext3-/ext4-Dateisystemen wiederherzustellen
ext4magic	V:0.3, I:3.5	235	Hilfsprogramm, um gelöschte Dateien von ext3-/ext4-Dateisystemen wiederherzustellen
ext3grep	V:0.1, I:1.7	299	Werkzeug, das bei der Wiederherstellung von gelöschten Dateien auf ext3-Dateisystemen hilft
scrounge-ntfs	V:0.1, I:1.6	49	Datenwiederherstellungsprogramm für NTFS-Dateisysteme
gzrt	V:0.02, I:0.43	34	Werkzeugsatz für die Wiederherstellung von gzip-Daten
sleuthkit	V:3, I:24	1130	Werkzeuge für forensische Analysen (SleuthKit)
autopsy	V:0.1, I:1.1	1026	grafische Oberfläche für SleuthKit
foremost	V:0.3, I:3.7	101	forensiche Anwendung zur Datenwiederherstellung
guymager	V:0.09, I:0.70	1049	forensisches Imaging-Werkzeug, basierend auf Qt
dcfldd	V:0.2, I:2.4	113	erweitertes dd für Forensik und Sicherheit

Tabelle 9.24: Liste von Paketen für Datenwiederherstellung und forensische Analysen

9.8.5 Aufsplitten einer großen in mehrere kleine Dateien

Wenn eine Datei zu groß ist, um sie als einzelne Datei zu sichern, können Sie trotzdem ein Backup davon erstellen, nachdem Sie sie in kleinere, z.B. 2000 MiB große Stücke aufgeteilt haben; später können diese Stücke wieder zur Originaldatei zusammengesetzt werden.

```
$ split -b 2000m large_file
$ cat x* >large_file
```



Achtung
Stellen Sie sicher, dass Sie dort keine anderen Dateien haben, die mit einem ”x” beginnen, um Kollisionen bei den Dateinamen zu vermeiden.

9.8.6 Leeren von Dateiinhalten

Zum Leeren des Inhalt einer Datei, z.B. einer Logdatei, verwenden Sie nicht rm(1), um die Datei zu löschen und anschließend eine neue leere Datei zu erstellen, da in der Zeit zwischen den Befehlen möglicherweise Zugriffsversuche auf die Datei erfolgen könnten. Folgender Weg ist der sicherste, um Dateiinhalte zu leeren:

```
$ :>file_to_be_cleared
```

9.8.7 Dummy-Dateien

Folgende Befehle erzeugen Dummy- oder leere Dateien:

```
$ dd if=/dev/zero    of=5kb.file bs=1k count=5
$ dd if=/dev/urandom of=7mb.file bs=1M count=7
$ touch zero.file
$ : > alwayszero.file
```

Sie sollten folgende Dateien vorfinden:

- ”`5kb.file`” ist eine 5KB große Datei, die Nullen enthält.
- ”`7mb.file`” ist eine 7MB große Datei mit zufälligem Inhalt.
- ”`zero.file`” könnte eine 0 Byte große Datei sein. Falls sie bereits existierte, wurde ihr `mtime`-Attribut aktualisiert, aber der Inhalt wurde beibehalten.
- ”`alwayszero.file`” ist immer eine 0 Byte große Datei. Falls sie bereits existierte, wurde ihr `mtime`-Attribut aktualisiert und der Inhalt entfernt.

9.8.8 Eine vollständige Festplatte löschen

Es gibt mehrere Wege, um alle Daten von einer Festplatte oder einem ähnlichen Speichergerät zu entfernen, z.B. einem USB-Flash-Speicher unter ”`/dev/sda`”.



Achtung
Überprüfen Sie zuerst den Speicherort Ihres USB-Flash-Speichers mit mount(8), bevor Sie irgendwelche dieser Befehle ausführen. Das Gerät, auf das ”`/dev/sda`” verweist, könnte auch die SCSI- oder Serial-ATA-Festplatte sein, die Ihr komplettes Betriebssystem enthält.

Löschen Sie den kompletten Inhalt der Platte, indem Sie wie folgt die Daten auf 0 setzen:

```
# dd if=/dev/zero of=/dev/sda
```

Löschen Sie alles, indem Sie es wie folgt mit zufälligen Daten überschreiben:

```
# dd if=/dev/urandom of=/dev/sda
```

Löschen Sie alles, indem Sie es wie hier auf sehr effiziente Art mit zufälligen Daten überschreiben:

```
# shred -v -n 1 /dev/sda
```

Alternativ können Sie auch badblocks(8) mit der Option `-t random` verwenden.

Da dd(1) auf der Shell vieler boot-fähiger Linux-CDs (wie einer Debian-Installer-CD) verfügbar ist, können Sie Ihr installiertes Betriebssystem vollständig entfernen, indem Sie von solch einer CD einen Löschbefehl auf die System-Festplatte (z.B. ”`/dev/sda`”, ”`/dev/sda`” o.ä.) ausführen.

9.8.9 Einen ungenutzten Bereich einer Festplatte löschen

In einem nicht genutzten Bereich auf einer Festplatte (oder einem USB-Flash-Speicher), z.B. ”`/dev/sdb1`”, könnten gelöschte Daten noch immer vorhanden sein, da beim Löschen lediglich die Verbindung zum Dateisystem entfernt wird. Diese Daten könnten durch das Überschreiben dann wirklich entfernt werden.

```
# mount -t auto /dev/sdb1 /mnt/foo
# cd /mnt/foo
# dd if=/dev/zero of=junk
dd: writing to `junk': No space left on device
...
# sync
# umount /dev/sdb1
```



Warnung
Dies ist für Ihren USB-Flash-Speicher normalerweise ausreichend, aber es ist nicht perfekt. Die meisten Teile von gelöschten Dateinamen und deren Attribute könnten versteckt sein und im Dateisystem verbleiben.

9.8.10 Wiederherstellen von gelöschten, aber noch geöffneten Dateien

Wenn Sie versehentlich eine Datei gelöscht haben, die noch von einem Programm verwendet wird (lesend oder schreibend), ist es möglich, diese Datei wiederherzustellen.

Probieren Sie zum Beispiel folgendes:

```
$ echo foo > bar
$ less bar
$ ps aux | grep ' less[ ]'
bozo    4775  0.0  0.0  92200   884 pts/8    S+   00:18   0:00 less bar
$ rm bar
$ ls -l /proc/4775/fd | grep bar
lr-x------ 1 bozo bozo 64 2008-05-09 00:19 4 -> /home/bozo/bar (deleted)
$ cat /proc/4775/fd/4 >bar
$ ls -l
-rw-r--r-- 1 bozo bozo 4 2008-05-09 00:25 bar
$ cat bar
foo
```

Führen Sie auf einem anderen Terminal (wenn Sie das `lsof`-Paket installiert haben) folgendes aus:

```
$ ls -li bar
2228329 -rw-r--r-- 1 bozo bozo 4 2008-05-11 11:02 bar
$ lsof |grep bar|grep less
less 4775 bozo 4r REG 8,3 4 2228329 /home/bozo/bar
$ rm bar
$ lsof |grep bar|grep less
less 4775 bozo 4r REG 8,3 4 2228329 /home/bozo/bar (deleted)
$ cat /proc/4775/fd/4 >bar
$ ls -li bar
2228302 -rw-r--r-- 1 bozo bozo 4 2008-05-11 11:05 bar
$ cat bar
foo
```

9.8.11 Alle harten Links suchen

Dateien mit harten Links können mittels "`ls -li`" identifiziert werden:

```
$ ls -li
total 0
2738405 -rw-r--r-- 1 root root 0 2008-09-15 20:21 bar
2738404 -rw-r--r-- 2 root root 0 2008-09-15 20:21 baz
2738404 -rw-r--r-- 2 root root 0 2008-09-15 20:21 foo
```

Sowohl bei "`baz`" wie auch bei "`foo`" zeigt die Anzahl der Links von "2" (>1) an, dass für sie harte Links existieren. Sie haben beide die gemeinsame Inode-Nummer "2738404". Das bedeutet, dass dies beides die gleiche hart verlinkte Datei ist. Falls Sie nicht zufällig alle hart verlinkten Dateien finden, können Sie über die Inode (z.B. "2738404") danach suchen:

```
# find /path/to/mount/point -xdev -inum 2738404
```

9.8.12 Unsichtbarer Verbrauch von Festplattenplatz

Alle gelöschten, aber noch geöffneten Dateien verbrauchen Festplattenplatz, obwohl dies über einen normalen du(1)-Befehl nicht festgestellt werden kann. Sie können wie folgt über ihre Größe aufgelistet werden:

```
# lsof -s -X / |grep deleted
```

9.9 Tipps zur Datenverschlüsselung

Mit physikalischem Zugriff auf den Rechner kann jeder ganz einfach uneingeschränkte Rechte und Zugriffe auf alle Dateien auf Ihrem PC erlangen (lesen Sie dazu Abschnitt 4.6.4). Das passwortgeschützte Anmeldesystem ist nicht in der Lage, Ihre Privatsphäre und sensible Daten vor einem möglichen Diebstahl Ihres PCs zu schützen. Dies kann nur durch Verwendung einer Technologie zur Datenverschlüsselung erreicht werden. Obwohl GNU Privacy Guard (Näheres in Abschnitt 10.3) Dateien verschlüsseln kann, bedeutet es für den Benutzer einigen Aufwand.

dm-crypt ermöglicht eine automatische Datenverschlüsselung über native Linux-Kernel-Module mittels device-mapper bei minimalem Aufwand für den Benutzer.

Paket	Popcon	Größe	Beschreibung
`cryptsetup`	V:35, I:77	476	Werkzeug zur Verschlüsselung von blockorientierten Geräten (dm-crypt / LUKS)
`cryptmount`	V:1.8, I:2.2	236	Werkzeuge zur Verschlüsselung von blockorientierten Geräten (dm-crypt / LUKS) mit Fokus auf das Einbinden/Trennen durch normale Benutzer
`fscrypt`	V:0.4, I:1.1	6844	Werkzeuge für die Verschlüsselung von Linux-Dateisystemen (fscrypt)
`libpam-fscrypt`	V:0.32, I:0.35	6941	PAM-Modul für die Verschlüsselung von Linux-Dateisystemen (fscrypt)

Tabelle 9.25: Liste von Werkzeugen zur Datenverschlüsselung



Achtung
Datenverschlüsselung kostet CPU-Zeit usw. Verschlüsselte Daten sind nicht mehr zugänglich, wenn das zugehörige Passwort verloren ist. Bitte wägen Sie Kosten und Nutzen gegeneinander ab.

Anmerkung
Mit dem debian-installer (Lenny und später) kann ein vollständiges Debian-System auf einer verschlüsselten Festplatte installiert werden; dabei werden dm-crypt/LUKS und initramfs verwendet.

Tipp
In Abschnitt 10.3 finden Sie Infos über eine Verschlüsselungslösung, die komplett auf Benutzerebene abläuft: GNU Privacy Guard.

9.9.1 Verschlüsselung von Wechseldatenträgern mit dm-crypt/LUKS

Sie können die Inhalte eines Wechseldatenträgers, wie z.B. eines USB-Flash-Speichers auf "`/dev/sdx`", mit dm-crypt/LUKS verschlüsseln. Formatieren Sie ihn einfach wie folgt:

```
# fdisk /dev/sdx
... "n" "p" "1" "return" "return" "w"
# cryptsetup luksFormat /dev/sdx1
...
# cryptsetup open /dev/sdx1 secret
...
# ls -l /dev/mapper/
total 0
crw-rw---- 1 root root  10, 60 2021-10-04 18:44 control
lrwxrwxrwx 1 root root       7 2021-10-04 23:55 secret -> ../dm-0
# mkfs.vfat /dev/mapper/secret
...
# cryptsetup close secret
```

Dann kann er in einer modernen Arbeitsplatzumgebung wie ein normaler USB-Stick unter "`/media/benutzername/daten`" eingebunden werden (Näheres hierzu in Abschnitt 10.1.7), nur dass dabei nach dem Passwort gefragt wird; dazu wird das Paket `udisks2` genutzt. Der Unterschied ist, dass jegliche Daten, die auf den Stick geschrieben werden, verschlüsselt sind. Der Passworteintrag kann auch automatisch über einen Schlüsselbund erfolgen (Näheres dazu in Abschnitt 10.3.6).

Alternativ können Sie solche Medien auch mit anderen Dateisystemen formatieren, z.B. ext4 mit "`mkfs.ext4 /dev/mapper`". Falls btrfs genutzt wird, muss das Paket `udisks2-btrfs` installiert sein. Bei diesen Dateisystemen müssen unter Umständen der Eigentümer und die Berechtigungen der Dateien separat konfiguriert werden.

9.9.2 Einbinden verschlüsselter Laufwerke mit dm-crypt/LUKS

Eine verschlüsselte Plattenpartition zum Beispiel, die durch den Debian Installer mit dm-crypt/LUKS auf "`/dev/sdc5`" erstellt wurde, kann wie folgt unter "`/mnt`" eingebunden werden:

```
$ sudo cryptsetup open /dev/sdc5 ninja --type luks
Enter passphrase for /dev/sdc5: ****
$ sudo lvm
lvm> lvscan
  inactive          '/dev/ninja-vg/root' [13.52 GiB] inherit
  inactive          '/dev/ninja-vg/swap_1' [640.00 MiB] inherit
  ACTIVE            '/dev/goofy/root' [180.00 GiB] inherit
  ACTIVE            '/dev/goofy/swap' [9.70 GiB] inherit
lvm> lvchange -a y /dev/ninja-vg/root
lvm> exit
  Exiting.
$ sudo mount /dev/ninja-vg/root /mnt
```

9.10 Der Kernel

Debian stellt für unterstützte Architekturen modulare Linux-Kernel als Pakete bereit.

Wenn Sie diese Dokumentation lesen, müssen Sie vermutlich keinen eigenen Linux-Kernel kompilieren.

9.10.1 Kernel-Parameter

Viele Linux-Funktionalitäten sind wie folgt über Kernel-Parameter konfigurierbar:

- Kernel-Parameter, die durch den Bootloader initialisiert werden (Näheres in Abschnitt 3.1.2);

- Kernel-Parameter, die durch sysctl(8) zur Laufzeit geändert werden und die über sysfs erreichbar sind (Näheres in Abschnitt 1.2.12);
- Modul-Parameter, die über Argumente von modprobe(8) gesetzt werden, wenn ein Modul aktiviert wird (Näheres in Abschnitt 9.7.3).

Nähere Details finden Sie unter "The Linux kernel user's and administrator's guide » The kernel's command-line parameters" (Englisch).

9.10.2 Kernel-Header

Die meisten **normalen** Programme benötigen keine Kernel-Header und könnten im Gegenteil sogar gestört werden, wenn Sie diese direkt zum Kompilieren verwenden würden. Sie sollten stattdessen gegen die Header in "`/usr/include/li` und "`/usr/include/asm`" aus dem Paket `libc6-dev` kompiliert werden (diese werden auf einem Debian-System aus dem `glibc`-Quellpaket erzeugt).

Anmerkung
Um einige Kernel-spezifische Programme wie Kernel-Module aus externen Linux-Quellen oder den automounter-Daemon (`amd`) zu kompilieren, müssen Sie den Pfad zu den entsprechenden Kernel-Headern auf der Befehlszeile mit angeben.

9.10.3 Kompilieren des Kernels und dazugehöriger Module

Debian hat seine eigene Methode zur Kompilierung des Kernels und zugehöriger Module.

Paket	Popcon	Größe	Beschreibung
`build-essential`	V:20, I:508	17	Pakete, die zum Bauen von Debian-Paketen essentiell nötig sind: `make`, `gcc`, …
`bzip2`	V:163, I:973	113	Werkzeuge zum Komprimieren und Dekomprimieren von bz2-Dateien
`libncurses-dev`	V:16, I:148	2380	Entwickler-Bibliothek und Dokumentation für ncurses
`git`	V:442, I:630	52512	git: vom Linux-Kernel verwendetes verteiltes Versionskontrollsystem
`fakeroot`	V:34, I:508	225	bietet eine fakeroot-Umgebung, um Pakete als nicht-root-Benutzer zu bauen
`initramfs-tools`	V:525, I:989	49	Werkzeug zur Erzeugung eines initramfs (Debian-spezifisch)
`dkms`	V:81, I:147	235	Dynamic Kernel Module Support (DKMS) (generisch)
`module-assistant`	V:1, I:11	395	Hilfsprogramme zum Erstellen von Modulpaketen (Debian-spezifisch)
`devscripts`	V:5, I:30	2790	Helfer-Skripte für Debian-Paketbetreuer (Debian-spezifisch)

Tabelle 9.26: Liste von Schlüsselpaketen für die Neukompilierung des Kernels auf einem Debian-System

Wenn Sie eine `initrd` wie in Abschnitt 3.1.2 verwenden, lesen Sie unbedingt die entsprechenden Informationen in initramfs-tools(7), update-initramfs(8), mkinitramfs(8) und initramfs.conf(5).



Warnung
Setzen Sie in den Verzeichnissen Ihres Quellcode-Baums (z.B. "`/usr/src/linux*`") keine symbolischen Links auf "`/usr/include/linux`" and "`/usr/include/asm`" (einige veraltete Dokumentationen empfehlen dies).

Anmerkung
Um den aktuellsten Linux-Kernel auf einem Debian-`Stable`-System zu kompilieren, könnte die Verwendung von rückportierten aktuellen Werkzeugen aus `Unstable` nötig sein.
module-assistant(8) (oder seine Kurzform `m-a`) unterstützt Benutzer bei Bau und Installation von Modulpaketen für einen oder mehrere selbst angepasste Kernel.
Dynamic Kernel Module Support (DKMS) ist ein neues distributions-unabhängiges Rahmenwerk, das entwickelt wurde, um die Aktualisierung einzelner Kernel-Module ohne Austausch des kompletten Kernels zu ermöglichen. Dies wird verwendet für die Betreuung von Modulen außerhalb des Linux-Quellcode-Baums. Auch ist es damit sehr einfach, im Zuge der Hochrüstung des Kernels Module neu zu bauen.

9.10.4 Kompilieren des Kernel-Quellcodes: Empfehlung des Debian-Kernel-Teams

Um eigene Kernel-Binärpakete aus den Upstream-Kernel-Quelltexten zu erstellen, sollten Sie das angebotene "`deb-pkg`"-Target nutzen:

```
$ sudo apt-get build-dep linux
$ cd /usr/src
$ wget https://mirrors.edge.kernel.org/pub/linux/kernel/v7.x/linux-7.3.5.tar.xz
$ tar --xz -xvf linux-7.3.5.tar.xz
$ cd linux-7.3.5
$ cp /boot/config-7.3.5.config
$ make menuconfig
 ...
$ make deb-pkg
```

Tipp
Das Paket linux-source-*version* stellt den Linux-Kernel-Quelltext inklusive Debian-Patches als "`/usr/src/linux-`*`version`*`.tar.bz2`" bereit.

Um spezifische Binärpakete aus dem Debian-Kernel-Quellpaket zu bauen, sollten Sie die Targets "`binary-arch_`*`archite`* in "`debian/rules.gen`" verwenden:

```
$ sudo apt-get build-dep linux
$ apt-get source linux
$ cd linux-7.*
$ fakeroot make -f debian/rules.gen binary-arch_i386_none_686
```

Hier finden Sie weitere Informationen:

- Debian Wiki: KernelFAQ
- Debian Wiki: DebianKernel
- Debian Linux Kernel Handbook: https://kernel-handbook.debian.net

9.10.5 Hardware-Treiber und Firmware

Der Hardware-Treiber ist Code, der auf der Haupt-CPU des Zielsystems läuft. Die meisten Hardware-Treiber sind heutzutage als freie Software verfügbar und in den normalen Debian-Kernel-Paketen im `main`-Bereich des Debian-Archivs enthalten.

- GPU-Treiber

  - Intel GPU-Treiber (`main`)
  - AMD/ATI GPU-Treiber (`main`)
  - NVIDIA GPU-Treiber (`main` für den nouveau-Treiber, `non-free` für Treiber, die vom Hersteller als Nur-Binär-Variante bereitgestellt werden)

Die Firmware ist Code, der in das (mit dem Zielsystem verbundene) Gerät geladen wird (z.B. CPU-Mikrocode, Rendering-Code, der auf der GPU läuft, oder FPGA-/CPLD-Daten …). Einige Firmware-Pakete gibt es als freie Software, aber viele davon werden nicht als freie Software bereitgestellt, da sie Binärdaten ohne den zugehörigen Quelltext enthalten. Die Installation dieses Firmware-Codes ist erforderlich, damit das Gerät wie vorgesehen funktionieren kann.

- Firmware-Pakete, die Daten enthalten, welche in den flüchtigen Speicher des Zielgeräts geladen werden:
  - `firmware-linux-free` (`main`)
  - `firmware-linux-nonfree` (`non-free-firmware`)
  - `firmware-linux-*` (`non-free-firmware`)
  - `*-firmware` (`non-free-firmware`)
  - `intel-microcode` (`non-free-firmware`)
  - `amd64-microcode` (`non-free-firmware`)
- Pakete mit Firmware-Update-Programmen, die Daten in den nicht-flüchtigen Speicher des Zielgeräts laden:
  - fwupd (`main`): Firmware-Update-Daemon, der Firmware-Daten vom Linux Vendor Firmware Service herunterlädt
  - gnome-firmware (`main`): GTK-Frontend für fwupd
  - plasma-discover-backend-fwupd (`main`): Qt-Frontend für fwupd

Bitte beachten Sie, dass die offiziellen Installationsmedien seit Debian 12 Bookworm Pakete aus `non-free-firmware` bereithalten, um den Benutzern ein funktionales Installationserlebnis zu ermöglichen.Der Archivbereich `non-free-firmwa` ist in Abschnitt 2.1.5 beschrieben.

Bitte beachten Sie außerdem, dass die Firmware-Daten, die über fwupd vom Linux Vendor Firmware Service heruntergeladen und in den laufenden Linux-Kernel integriert werden, ebenfalls `nicht-frei (non-free)` sein könnten.

9.11 Virtualisierte Systeme

Die Verwendung eines virtualisierten Systems ermöglicht es uns, mehrere Instanzen eines Systems gleichzeitig auf einer einzigen Hardware laufen zu lassen.

Tipp
Lesen Sie die Debian-Wiki-Seite SystemVirtualization.

9.11.1 Virtualisierungs- und Emulationswerkzeuge

Es gibt verschiedene Virtualisierungs- und Emulations-Plattformen:

- Vollständige Hardware-Emulations-Pakete, wie die aus dem Metapaket games-emulator
- Plattformen, bei denen die CPU und einige Eingabe-/Ausgabegeräte überwiegend emuliert werden, wie z.B. QEMU
- Plattformen, bei denen die CPU und einige Eingabe-/Ausgabegeräte überwiegend virtualisiert werden, wie z.B. Kernel-based Virtual Machine (KVM) (Betriebssystemkern-basierte virtuelle Maschine)

- Container-Virtualisierung auf Betriebssytemebene mit Kernel-Level-Unterstützung, wie z.B. LXC (Linux Containers), Docker, systemd-nspawn(1), ...
- Dateisystemzugriff-Virtualisierung auf Betriebssystemebene, bei der der Aufruf der Systembibliotheken über den Dateipfad überschrieben/geändert wird, wie z.B. chroot
- Dateisystemzugriff-Virtualisierung auf Betriebssystemebene, bei der der Aufruf der Systembibliotheken über die Dateieigentümerschaft überschrieben/geändert wird, wie z.B.fakeroot
- Betriebssystem-API-Emulation, wie z.B. Wine
- Virtualisierung auf Interpreter-Ebene mit Überschreiben von Executable Selection und Runtime-Bibliotheken, wie z.B. virtualenv und venv für Python

Die Container-Virtualisierung nutzt Fähigkeiten aus Abschnitt 4.7.5 sowie die Backend-Technologie aus Abschnitt 7.7.

Hier einige Pakete, die Ihnen bei der Einrichtung eines virtualisierten Systems helfen:

Der Wikipedia-Artikel Comparison of platform virtual machines (Englisch) enthält detaillierte Gegenüberstellungen der verschiedenen Plattform-Virtualisierungslösungen.

9.11.2 Arbeitsablauf bei Virtualisierung

Anmerkung
Standard-Debian-Kernel unterstützen KVM seit `Lenny`.

Ein typischer Arbeitsablauf für eine Virtualisierung enthält folgende Schritte:

- Erzeugen eines leeren Dateisystems (ein Verzeichnisbaum oder ein Festplatten-Image);
  - Ein Verzeichnisbaum kann über "`mkdir -p /path/to/chroot`" erzeugt werden.
  - Eine rohe (leere) Image-Datei kann mittels dd(1) erstellt werden (lesen Sie dazu Abschnitt 9.7.1 und Abschnitt 9.7.5).
  - qemu-img(1) kann verwendet werden, um zu QEMU kompatible Image-Dateien zu erzeugen.
  - Rohe Image-Dateien und solche im VMDK-Format sind weit verbreitet und können bei verschiedenen Virtualisierungslösungen eingesetzt werden.
- Einbinden des Festplatten-Images in das Dateisystem mit mount(8) (optional);
  - Bei einer rohen Image-Datei verwenden Sie zum Einbinden ein loop device oder Device Mapper-Geräte (Näheres in Abschnitt 9.7.3).
  - Festplatten-Images, die von QEMU unterstützt werden, binden Sie als Network Block Device ein (lesen Sie dazu Abschnitt 9.11.3).
- Bestücken des Zieldateisystems mit den benötigten Systemdaten;
  - Die Nutzung von Programmen wie `debootstrap` und `cdebootstrap` hilft Ihnen bei diesem Schritt (Details dazu in Abschnitt 9.11.4).
  - Verwenden Sie die Installationsroutinen anderer Betriebssysteme für das in der Emulation laufende System.
- Ausführen eines Programms in der virtualisierten Umgebung;
  - chroot bietet eine grundlegende virtualisierte Umgebung, die zur Kompilierung von Programmen, Ausführung von Konsolenanwendungen sowie Daemons ausreichende Funktionalitäten hat.
  - QEMU stellt eine CPU-Emulation quer über verschiedene Plattformen zur Verfügung.
  - QEMU mit KVM bietet eine vollständige Systememulation mit hardware-unterstützter Virtualisierung.
  - VirtualBox bietet eine vollständige System-Emulation auf i386 und amd64 mit oder ohne hardware-unterstützter Virtualisierung.

Paket	Popcon	Größe	Beschreibung
coreutils	V:914, I:1000	19751	GNU core-Hilfsprogramme, die chroot(8) enthalten
util-linux	V:922, I:1000	4826	Paket mit Systemwerkzeugen, das u.a. unshare(1) enthält
systemd-container	V:74, I:77	2964	systemd container/nspawn-Werkzeuge, die systemd-nspawn(1) enthalten
schroot	V:7.1, I:8.9	2241	spezialisiertes Werkzeug, um Debian-Binärpakete in chroot-Umgebungen auszuführen
sbuild	V:3.8, I:6.8	160	Werkzeug, um Debian-Binärpakete aus Debian-Quellen zu bauen
debootstrap	V:4, I:43	332	Programm zum Bootstrap eines grundlegenden Debian-Systems (geschrieben in sh)
mmdebstrap	V:8, I:12	613	Programm zum Bootstrap eines Debian-Systems (geschrieben in Perl)
cdebootstrap	V:0.1, I:1.2	113	Programm zum Bootstrap eines Debian-Systems (geschrieben in C)
cloud-image-utils	V:1, I:14	52	Management-Werkzeuge für Cloud-Images
cloud-guest-utils	V:5, I:25	67	Cloud-Gast-Werkzeuge
virt-manager	V:12, I:47	2309	Virtual Machine Manager: grafische Arbeitsplatzanwendung zur Verwaltung von virtuellen Maschinen
libvirt-clients	V:49, I:70	1168	Programme für die libvirt-Bibliothek
docker.io	V:51, I:55	80705	docker: Linux-Container-Runtime
podman	V:36, I:40	86285	podman: Engine für OCI-basierte Container in Pods
podman-docker	V:2.9, I:3.4	271	Engine für OCI-basierte Container in Pods - Wrapper für docker
incus	V:1.0, I:3.7	24	Incus: System-Container und Manager für virtuelle Maschinen
games-emulator	I:0.18	21	games-emulator: Debians Emulatoren für Spiele
bochs	V:0.03, I:0.53	8469	Bochs: IA-32 PC-Emulator
qemu-system	I:19	69	QEMU: Binärdateien zur Emulation eines vollständigen Systems
qemu-user	V:6.3, I:9.8	472221	QEMU: Binärdateien für User-Mode-Emulation
qemu-utils	V:16, I:117	12651	QEMU: Hilfsprogramme
qemu-system-x86	V:55, I:101	69265	KVM: vollständige Virtualisierungslösung auf x86-Hardware mit hardware-unterstützter Virtualisierung
virt-manager	V:12, I:47	2309	Desktop-Anwendung zur Verwaltung virtueller Maschinen – etwa KVM-basierte virtuelle Maschinen, Xen, LXC ...
gnome-boxes	V:1.1, I:5.8	7070	Boxes: einfache GNOME-App für den Zugriff auf virtuelle Systeme
xen-tools	V:0.1, I:1.3	719	Werkzeuge zur Verwaltung von virtuellen Debian-XEN-Servern
wine	V:13, I:54	209	Wine: Windows-API-Implementierung (Standard-Programm-Suite)
dosbox	V:1, I:11	3139	DOSBox: x86-Emulator mit Tandy-/Herc-/CGA-/EGA-/VGA-/SVGA-Grafik, Audioausgabe und DOS
lxc	V:11, I:13	1680	Linux-Container verwenden Werkzeuge im User-Space-Bereich
python3-venv	V:8, I:167	6	venv zur Erzeugung von virtuellen Umgebungen für Python (Systembibliothek)
python3-virtualenv	V:7, I:35	435	virtualenv zur Erzeugung isolierter virtueller Umgebungen für Python
pipx	V:8, I:52	3261	pipx für die Installation von Python-Applikationen in isolierten Umgebungen

Tabelle 9.27: Liste von Virtualisierungswerkzeugen

9.11.3 Einbinden des virtuellen Festplatten-Images

Wenn Sie eine rohe Image-Datei verwenden, finden Sie die nötigen Informationen in Abschnitt 9.7.

Bei anderen Dateiformaten für virtuelle Festplatten-Images können Sie qemu-nbd(8) verwenden, um diese über das Network Block Device-Protokoll zu exportieren; dann können sie mittels dem `nbd`-Kernelmodul in das Dateisystem eingebunden werden (mount).

qemu-nbd(8) unterstützt Festplattenformate, die auch von QEMU unterstützt werden. QEMU wiederum unterstützt folgende Formate: roh (raw), qcow2, qcow, vmdk, vdi, bochs, cow (User-Mode-Linux Copy-on-Write), parallels, dmg, cloop, vpc, vvfat (virtuelles VFAT) und host_device.

Das Network Block Device unterstützt Partitionen auf die gleiche Art wie das loop device (Näheres hierzu in Abschnitt 9.7.3). Sie können die erste Partition von "`disk.img`" wie folgt einbinden:

```
# modprobe nbd max_part=16
# qemu-nbd -v -c /dev/nbd0 disk.img
...
# mkdir /mnt/part1
# mount /dev/nbd0p1 /mnt/part1
```

Tipp
Eventuell möchten Sie lediglich die erste Partition von "`disk.img`" exportieren; geben Sie dazu für qemu-nbd(8) die Option "`-P 1`" an.

9.11.4 Chroot-System

Wenn Sie eine neue Debian-Umgebung im Terminal ausprobieren möchten, empfehle ich Ihnen, chroot zu verwenden. Es ermöglicht Ihnen, Konsolenapplikationen aus Debian `Unstable` und `Testing` laufen zu lassen ohne die üblichen Risiken, die dies sonst mit sich bringt, und ohne einen Reboot des Systems. chroot(8) ist der elementarste Weg für solch einen Ansatz.



Achtung
Die hier angegebenen Beispiele gehen davon aus, dass sowohl das Elternsystem wie auch das chroot-System die gleiche CPU-Architektur (`amd64`) haben.

Obwohl Sie sich eine chroot(8)-Umgebung auch händisch über debootstrap(8) erstellen können, erfordert dies einigen nicht ganz trivialen Aufwand.

Das sbuild-Paket zum Bauen von Debian-Paketen aus dem Quellcode nutzt eine chroot-Umgebung, die über das schroot-Paket verwaltet wird. Es enthält das Helfer-Skript sbuild-createchroot(8). Hier lernen Sie, wie dies funktioniert:

```
$ sudo mkdir -p /srv/chroot
$ sudo sbuild-createchroot -v --include=eatmydata,ccache unstable /srv/chroot/unstable- ↩
    amd64-sbuild http://deb.debian.org/debian
 ...
```

Sie sehen jetzt, wie debootstrap(8) Systemdaten für eine `unstable`-Umgebung in "`/srv/chroot/unstable-amd64-sbu` anlegt und so ein minimales Build-System erzeugt.

Sie können sich über schroot(1) in dieser Umgebung anmelden.

```
$ sudo schroot -v -c chroot:unstable-amd64-sbuild
```

Hier sehen Sie nun, wie eine System-Shell in einer `unstable`-Umgebung erstellt wird.

Anmerkung
Die Datei "`/usr/sbin/policy-rc.d`", die immer mit dem Rückgabewert 101 beendet wird, verhindert, dass Daemon-Programme automatisch in dieser Debian-Umgebung gestartet werden. Lesen Sie hierzu "`/usr/share/doc/init-system-helpers/README.policy-rc.d.gz`".

Anmerkung
Einige Programme könnten unter chroot Zugriff auf mehr Dateien des Elternsystems erfordern, als `sbuild-createchroot` mit obiger Vorgehensweise bereitstellt. Zum Beispiel könnte es nötig sein, "`/sys`", "`/etc/passwd`", "`/etc/group`", "`/var/run/utmp`", "`/var/log/wtmp`" usw. über bind-mount einzubinden oder zu kopieren.

Tipp
Das `sbuild`-Paket hilft Ihnen, ein chroot-System zu erstellen mit `schroot` als Backend, und ein Paket innerhalb dieser Umgebung zu bauen. Dies ist ideal, um Paketabhängigkeiten zu testen. Weitere Informationen finden Sie im Debian-Wiki unter sbuild sowie im sbuild-Konfigurationsbeispiel des "Guide for Debian Maintainers".

Tipp
Der Befehl systemd-nspawn(1) hilft dabei, einen Befehl oder ein Betriebssystem in einem ressourcen-schonenden Container laufen zu lassen, ähnlich wie bei `chroot`. Es ist aber leistungsstärker, da es Namensräume verwendet, um Prozessbaum, IPC, Rechnername, Domainname und optional auch Netzwerk- und Benutzerdatenbanken vollständig zu virtualisieren. Siehe systemd-nspawn.

9.11.5 System mit mehreren Arbeitsplatzumgebungen

Wenn Sie eine neue grafische Arbeitsplatzumgebung von irgendeinem Betriebssystem ausprobieren möchten, empfehle ich Ihnen, QEMU oder KVM auf einem Debian-`Stable`-System zu verwenden, um mittels Virtualisierung mehrere Arbeitsplatzumgebungen (Desktops) sicher auf einem System laufen lassen zu können. So ist es möglich, Desktop-Anwendungen aus Debian `Unstable` und `Testing` ohne die üblichen damit verbundenen Risiken auszuführen sowie ohne einen Reboot des Systems.

Da reines QEMU sehr langsam ist, wird empfohlen, es mit KVM zu beschleunigen, falls das Host-System dies unterstützt.

Virtual Machine Manager, auch bekannt als `virt-manager`, ist ein praktisches grafisches Werkzeug für die Verwaltung von virtuellen KVM-Maschinen via libvirt.

Ein Image einer virtuellen Festplatte "`virtdisk.qcow2`", mit einem Debian-System für QEMU kann z.B. mittels Debian über das Internet installieren wie folgt erzeugt werden:

```
$ wget https://cdimage.debian.org/debian-cd/5.0.3/amd64/iso-cd/debian-503-amd64-netinst.iso
$ qemu-img create -f qcow2 virtdisk.qcow2 5G
$ qemu -hda virtdisk.qcow2 -cdrom debian-503-amd64-netinst.iso -boot d -m 256
...
```

Tipp
Andere GNU/Linux-Distributionen wie Ubuntu und Fedora mittels Virtualisierung laufen zu lassen ist eine tolle Möglichkeit, Tipps zur Konfiguration zu bekommen. Auch andere proprietäre Betriebssysteme können über diese GNU/Linux-Virtualisierung bequem zum Laufen gebracht werden.

Weitere Tipps finden Sie im Debian Wiki unter SystemVirtualization.

Kapitel 10

Datenmanagement

Hier werden Werkzeuge und Tipps zur Verwaltung von Binär- und Textdateien auf einem Debian-System beschrieben.

10.1 Austauschen, kopieren und archivieren von Dateien



Warnung
Nicht-koordinierte zeitgleiche Schreibzugriffe auf Geräte und Dateien durch mehrere Prozesse sind nicht erlaubt, da dadurch eine Race Condition (Wettlaufsituation um konkurrierende Zugriffe auf die Ressource) entstehen könnte. Dies kann über einen File locking-Mechanismus mittels flock(1) umgangen werden.

Die Sicherheit der Daten und deren kontrolliertes Austauschen mit anderen hat verschiedene Aspekte:

- Erzeugung von Datenarchiven;
- Fern-Speicherzugriff;
- Vervielfältigung;
- Nachverfolgung der Änderungshistorie;
- Erleichterung des Tauschens von Daten;
- Verhinderung von unerlaubten Dateizugriffen;
- Erkennung von unerlaubten Dateiveränderungen.

Diese können über die Verwendung einer Kombination von Werkzeugen realisiert werden:

- Archivierungs- und Kompressionswerkzeuge
- Kopier- und Synchronisationswerkzeuge
- Netzwerk-Dateisysteme;
- Wechseldatenträger;
- Secure Shell;
- Authentifizierungssysteme;
- Versionskontrollsysteme;
- Hash- und kryptographische Verschlüsselungswerkzeuge.

10.1.1 Archivierungs- und Kompressionswerkzeuge

Hier eine Zusammenfassung von Archivierungs- und Kompressionswerkzeugen im Debian-System:



Warnung
Setzen Sie nicht die "$TAPE"-Variable, außer Sie sind sich über die Folgen im klaren. Sie verändern dadurch das Verhalten von tar(1).

- Ein gezipptes tar(1)-Archiv verwendet die Dateierweiterung "`.tgz`" oder "`.tar.gz`".
- Ein xz-komprimiertes tar(1)-Archiv verwendet die Dateierweiterung "`.txz`" oder "`.tar.xz`".
- Die Popularität der Kompressionsmethoden in FOSS-Programmen wie tar(1) hat sich mit der Zeit wie folgt verschoben: `gzip` → `bzip2` → `xz`.
- cp(1), scp(1) und tar(1) könnten Einschränkungen bei speziellen Dateien haben. cpio(1) ist dabei erheblich vielseitiger.
- cpio(1) wurde entwickelt, um zusammen mit find(1) und anderen Befehlen verwendet zu werden und ist geeignet, Backup-Skripte zu erstellen, da der Teil des Skriptes zur Dateiauswahl eigenständig getestet werden kann.
- Die interne Struktur von Libreoffice-Dateien entspricht der von "`.jar`"-Dateien, die auch mit `unzip` geöffnet werden können.
- Das de-facto plattform-übergreifende Archivwerkzeug ist `zip`. Benutzen Sie es in der Art "`zip -rX`", um maximal mögliche Kompatibilität zu erreichen. Verwenden Sie auch die Option "`-s`", wenn die maximale Dateigröße bei Ihnen relevant ist.

10.1.2 Kopier- und Synchronisationswerkzeuge

Hier eine Zusammenfassung von einfachen Kopier- und Backup-Werkzeugen im Debian-System:

Das Kopieren von Dateien mit rsync(1) bietet mehr Funktionalitäten als mit anderen Werkzeugen:

- Delta-Transfer-Algorithmus, der nur die Unterschiede zwischen Quelle und Ziel überträgt;
- schneller Prüf-Algorithmus, der (standardmäßig) nach Dateien sucht, die sich in Größe oder Änderungs-Zeitstempel verändert haben;
- die Optionen "`--exclude`" und "`--exclude-from`", vergleichbar mit denen von tar(1);
- die Syntax des Schrägstrichs hinter dem Quellverzeichnis, die verhindert, dass eine zusätzliche Verzeichnisebene auf dem Zielort erstellt wird.

Tipp
Versionskontrollsysteme (VCS) wie die in Tabelle 10.14 können ebenfalls als Mehrwege-Kopier- und Synchronisationswerkzeuge dienen.

Paket	Popcon	Größe	Erweiterung	Befehl	Erläuterung
tar	V:911, I:1000	3101	.tar	tar(1)	Standard-Archivierungs-Programm (De-Facto-Standard)
cpio	V:388, I:999	1205	.cpio	cpio(1)	Unix-Archivier-Programm im System-V-Stil, zu verwenden mit find(1)
binutils	V:201, I:627	1132	.ar	ar(1)	Archivier-Programm zur Erstellung von statischen Bibliotheken
fastjar	V:0.9, I:9.0	182	.jar	fastjar(1)	Archivier-Programm für Java (zip-artig)
pax	V:4.9, I:8.5	167	.pax	pax(1)	neues POSIX-Archivier-Programm, Kompromiss zwischen `tar` und `cpio`
gzip	V:904, I:1000	262	.gz	gzip(1), zcat(1), …	GNU LZ77-Kompressionswerkzeug (De-Facto-Standard)
bzip2	V:163, I:973	113	.bz2	bzip2(1), bzcat(1), …	Burrows-Wheeler Block-Sorting Kompressions-Werkzeug mit höherem Kompressionsverhältnis als gzip(1) (langsamer als `gzip` mit ähnlicher Syntax)
lzma	V:0.7, I:9.4	349	.lzma	lzma(1)	LZMA-Kompressionswerkzeug mit höherem Kompressionsverhältnis als gzip(1) (überholt)
xz-utils	V:323, I:981	1551	.xz	xz(1), xzdec(1), …	XZ-Kompressionswerkzeug mit höherem Kompressionsverhältnis als bzip2(1) (langsamer als `gzip`, aber schneller als `bzip2`; ersetzt das LZMA-Kompressionswerkzeug)
zstd	V:346, I:850	2308	.zstd	zstd(1), zstdcat(1), …	Zstandard: schnelles verlustfreies Kompressionswerkzeug
7zip	V:86, I:339	7542	.7z	7zr(1), p7zip(1)	7-Zip-Dateiarchivier-Programm mit hohem Kompressionsverhältnis (LZMA- und andere Kompressionsalgorithmen)
lzop	V:13, I:128	164	.lzo	lzop(1)	LZO-Kompressionswerkzeug mit höherer Kompressions- und Dekompressionsgeschwindigkeit als gzip(1) (niedrigeres Kompressionsverhältnis als `gzip` mit ähnlicher Syntax)
zip	V:52, I:350	635	.zip	zip(1)	InfoZIP: DOS-Archivierungs- und Kompressionswerkzeug
unzip	V:126, I:738	399	.zip	unzip(1)	InfoZIP: DOS-Dearchivierungs- und Dekompressionswerkzeug

Tabelle 10.1: Liste von Archivierungs- und Kompressionswerkzeugen

Paket	Popcon	Größe	Werkzeug	Funktion
coreutils	V:914, I:1000	19751	GNU cp	Dateien und Verzeichnisse lokal kopieren (verwenden Sie "-a" für rekursive Operationen)
openssh-client	V:709, I:997	3611	scp	Dateien und Verzeichnisse von fern kopieren (Client, verwenden Sie "-r" für rekursive Operationen)
openssh-server	V:781, I:824	3615	sshd	Dateien und Verzeichnisse von fern kopieren (ferner Server)
rsync	V:191, I:538	985		1-Weg-Synchronisation und -Backup von fern
unison	V:3, I:12	14		2-Wege-Synchronisation und -Backup von fern

Tabelle 10.2: Liste von Kopier- und Synchronisationswerkzeugen

10.1.3 Aufrufe für Archivierungsoperationen

Hier finden Sie mehrere Wege, um den kompletten Inhalt des Verzeichnisses "./source" mit verschiedenen Werkzeugen zu archivieren oder dearchivieren.

GNU tar(1):

```
$ tar -cvJf archive.tar.xz ./source
$ tar -xvJf archive.tar.xz
```

Alternativ auch:

```
$ find ./source -xdev -print0 | tar -cvJf archive.tar.xz --null -T -
```

cpio(1):

```
$ find ./source -xdev -print0 | cpio -ov --null > archive.cpio; xz archive.cpio
$ zcat archive.cpio.xz | cpio -i
```

10.1.4 Aufrufe für Kopieroperationen

Hier finden Sie mehrere Wege, um den kompletten Inhalt des Verzeichnisses "./source" mit verschiedenen Werkzeugen zu kopieren.

- Lokale Kopie: "./source"-Verzeichnis → "/dest"-Verzeichnis
- Kopie auf einen fernen Rechner: "./source"-Verzeichnis auf dem lokalen Rechner → "/dest"-Verzeichnis auf dem Rechner "user@host.dom"

rsync(8):

```
# cd ./source; rsync -aHAXSv . /dest
# cd ./source; rsync -aHAXSv . user@host.dom:/dest
```

Sie können auch die Syntax mit Schrägstrich am Ende des Zielverzeichnisses nutzen:

```
# rsync -aHAXSv ./source/ /dest
# rsync -aHAXSv ./source/ user@host.dom:/dest
```

Alternativ auch:

```
# cd ./source; find . -print0 | rsync -aHAXSv0 --files-from=- . /dest
# cd ./source; find . -print0 | rsync -aHAXSv0 --files-from=- . user@host.dom:/dest
```

GNU cp(1) und openSSH scp(1):

```
# cd ./source; cp -a . /dest
# cd ./source; scp -pr . user@host.dom:/dest
```

GNU tar(1):

```
# (cd ./source && tar cf - . ) | (cd /dest && tar xvfp - )
# (cd ./source && tar cf - . ) | ssh user@host.dom '(cd /dest && tar xvfp - )'
```

cpio(1):

```
# cd ./source; find . -print0 | cpio -pvdm --null --sparse /dest
```

Sie können in allen Beispielen, die einen "`.`" enthalten, diesen "`.`" durch "`foo`" ersetzen, um Dateien aus dem Verzeichnis "`./source/foo`" in das Verzeichnis "`/dest/foo`" zu kopieren.

Auch kann in allen Beispielen, die einen "`.`" enthalten, dieser "`.`" durch einen absoluten Pfad wie "`/pfad/zu/source/foo`" ersetzt werden, damit kann dann auf "`cd ./source;`" verzichtet werden. Dadurch werden, abhängig von den verwendeten Werkzeugen, die Dateien an unterschiedliche Orte kopiert:

- bei rsync(8), GNU cp(1) und scp(1): nach "`/dest/foo`";
- bei GNU tar(1) und cpio(1): nach "`/dest/pfad/zu/source/foo`".

Tipp
rsync(8) und GNU cp(1) unterstützen die Option "`-u`", um Dateien zu überspringen, die am Zielort neuer sind als im Quellverzeichnis.

10.1.5 Aufrufe für die Auswahl von Dateien

find(1) wird verwendet, um Dateien für Archivierungs- und Kopierbefehle auszuwählen (lesen Sie Abschnitt 10.1.3 und Abschnitt 10.1.4) oder xargs(1) (Näheres in Abschnitt 9.4.9). Dies kann mit deren Befehlsargumenten noch erweitert werden.

Die grundsätzliche Syntax von find(1) kann wie folgt zusammengefasst werden:

- Seine bedingten Argumente werden von links nach rechts ausgewertet.
- Die Auswertung wird beendet, sobald ihr Resultat ermittelt wurde.
- Ein "logisches **ODER**" (definiert über "`-o`" zwischen den Bedingungen) hat eine niedrigere Priorität als ein "logisches **UND**" (das über "`-a`" oder nichts zwischen den Bedingungen definiert wird).
- Ein "logisches **NICHT**" (definiert über "`!`" vor der Bedingung) hat eine höhere Priorität als ein "logisches **UND**".
- "`-prune`" liefert immer ein logisches **WAHR** zurück und, falls es ein Verzeichnis ist, wird die Suche nach Dateien an diesem Punkt beendet.
- "`-name`" findet Dateien über den Anfang des Dateinamens mittels Shell-Glob-Suchmuster (lesen Sie dazu Abschnitt 1.5.6), findet sie aber über Metazeichen wie "`*`" und "`?`" auch bei einem führenden "`.`" (neue POSIX-Funktionalität).
- "`-regex`" findet Dateien mit vollständigem Pfad standardmäßig über Emacs-artige reguläre Ausdrücke (**BRE**, nähere Infos finden Sie in Abschnitt 1.6.2).
- "`-size`" findet Dateien basierend auf der Dateigröße (mit einem "+" vor dem Wert für größer, mit einem "`-`" für kleiner).
- "`-newer`" findet Dateien, die neuer sind als in dem Argument angegeben.

- ”`-print0`” liefert immer ein logisches **WAHR** zurück und gibt den kompletten Dateinamen (abgeschlossen durch ein Nullzeichen) auf der Standardausgabe aus.

find(1) wird oft in einem idiomatischen Stil verwendet, wie hier:

```
# find /path/to \
    -xdev -regextype posix-extended \
    -type f -regex ".*\.cpio|.*~" -prune -o \
    -type d -regex ".*/\.git" -prune -o \
    -type f -size +99M -prune -o \
    -type f -newer /path/to/timestamp -print0
```

Das bedeutet folgendes:

1. nach allen Dateien suchen, beginnend in ”`/pfad/zu`”;
2. die Suche global auf das Dateisystem beschränken, in dem sie begonnen wurde, und stattdessen reguläre Ausdrücke (**ERE**, lesen Sie dazu Abschnitt 1.6.2) verwenden;
3. Dateien ausschließen, auf die die regulären Ausdrücke ”`.*\.cpio`” oder ”`.*~`” zutreffen, indem die Suche abgebrochen wird;
4. Verzeichnisse ausschließen, auf die der reguläre Ausdruck ”`.*/\.git`” zutrifft, indem die Suche abgebrochen wird;
5. Dateien ausschließen, die größer als 99 Megabyte (1048576 Byte) sind, indem die Suche abgebrochen wird;
6. Dateinamen ausgeben, die obige Suchkriterien erfüllen und neuer als ”`/path/to/timestamp`” sind.

Bitte beachten Sie die idiomatische Verwendung von ”`-prune -o`”, um in obigen Beispielen Dateien auszuschließen.

Anmerkung
Auf Unix-artigen Nicht-Debian-Systemen werden einige Optionen von find(1) unter Umständen nicht unterstützt. Versuchen Sie in diesem Fall, die Suchmethoden anzupassen und ”`-print0`” durch ”`-print`” zu ersetzen. Unter Umständen müssen Sie auch zugehörige Befehle anpassen.

10.1.6 Archivierungsmedien

Wenn Sie Speichermedien für die Sicherung wichtiger Daten suchen, sollten Sie sorgfältig deren Einschränkungen abwägen. Für eine kleine persönliche Datensicherung verwende ich CD-R und DVD-R von einem Markenhersteller und lagere die Disks in einer kühlen, dunklen, trockenen und sauberen Umgebung. (Für die professionelle Nutzung scheinen Tapes (Speicherbänder) zur Archivierung sehr beliebt zu sein.)

Anmerkung
Ein feuerbeständiger Tresor ist gedacht für Dokumente in Papierform. Jedoch haben die meisten Speichermedien eine niedrigere Temperaturtoleranz als Papier. Ich baue auf mehrere sicher verschlüsselte Kopien, die an verschiedenen sicheren Orten aufbewahrt werden.

Optimistische Angabe der Lebensdauer von Speichermedien, gefunden im Internet (überwiegend Herstellerangaben):

- 100 Jahre und mehr: säurefreies Papier mit Tinte;
- 100 Jahre: Optische Speichermedien (CD/DVD, CD-R/DVD-R);
- 30 Jahre: Magnetische Speichermedien (Bänder, Disketten);

- 20 Jahre: Optische Medien basierend auf Phasenänderung (CD-RW).

Hierbei sind keine mechanischen Ausfälle durch Handhabung usw. berücksichtigt.

Optimistische Angabe von Schreibzyklen der Speichermedien, gefunden im Internet (überwiegend Herstellerangaben):

- 250000 Zyklen und mehr: Festplattenlaufwerk;
- 10000 Zyklen und mehr: Flash-Speicher;
- 1000 Zyklen: CD-RW/DVD-RW;
- 1 Zyklus: CD-R/DVD-R, Papier.



Achtung
Die obigen Angaben zu Lebensdauer und Schreibzyklen sollten nicht für Entscheidungen bezüglich kritischer Datenspeicherung herangezogen werden. Bitte konsultieren Sie in diesem Fall die spezifischen Produktinformationen des Herstellers.

Tipp
Da CD-R/DVD-R und Papier nur einmal beschrieben werden können, schützen sie von Natur aus vor dem versehentlichen Datenverlust durch Überschreiben. Dies ist ein Vorteil!

Tipp
Wenn Sie eine schnelle und wiederholte Sicherung großer Datenmengen benötigen, könnte eine Festplatte in einem fernen Rechner, verbunden über eine schnelle Internetverbindung, die einzige realistische Option sein.

Tipp
Wenn Sie wiederbeschreibbare Medien für Ihre Backups verwenden, könnte es eine gute Idee sein, Dateisysteme wie btrfs oder zfs zu nutzen, die nur-lesbare Schnappschüsse unterstützen.

10.1.7 Wechseldatenträger

Wechseldatenträger können folgende Geräte sein:

- USB-Flash-Speicher (USB-Stick);
- Festplattenlaufwerk;
- Optisches Laufwerk (CD/DVD/BD);
- Digitalkamera;
- Digitaler Mediaplayer.

Sie können über eine dieser Möglichkeiten verbunden sein:

- USB
- IEEE 1394 / FireWire

Moderne Arbeitsplatzumgebungen wie GNOME und KDE können diese Wechseldatenträger auch ohne einen entsprechenden "`/etc/fstab`"-Eintrag automatisiert einbinden.

- Das `udisks2`-Paket enthält einen Daemon und dazugehörige Hilfsprogramme, um diese Datenträger automatisch einzubinden und zu trennen.
- D-bus erzeugt Ereignisse, um automatische Prozesse anzustoßen.
- PolicyKit stellt die erforderlichen Berechtigungen bereit.

Tipp
Automatisch eingebundene Geräte haben eventuell die mount-Option "`uhelper=`", die von umount(8) genutzt wird.

Tipp
In modernen Arbeitsplatzumgebungen funktioniert das automatische Einbinden von Laufwerken nur, wenn diese Geräte nicht in "`/etc/fstab`" aufgelistet sind.

Der Einbindungspunkt wird in modernen Umgebungen in der Form "`/media/`*`benutzername/laufwerksname`*" abgebildet; die Laufwerksbezeichnung (manchmal auch als Disk-Label bezeichnet) kann wie folgt angepasst werden:

- mlabel(1) bei FAT-Dateisystemen;
- genisoimage(1) mit der Option "`-V`" bei ISO9660-Dateisystemen;
- tune2fs(8) mit der Option "`-L`" bei ext2-/ext3-/ext4-Dateisystemen.

Tipp
Die verwendete Zeichenkodierung muss unter Umständen als mount-Option angegeben werden (lesen Sie dazu Abschnitt 8.1.3).

Tipp
Die Verwendung des grafischen GUI-Menüs zur Trennung eines eingebundenen Dateisystems könnte auch dessen dynamisch erzeugte Gerätedatei (z.B. "`/dev/sdc`" entfernen. Falls Sie diese Gerätedatei erhalten möchten, trennen Sie die Einbindung mit dem umount(8)-Befehl von einem Shell-Prompt.

10.1.8 Dateisystemauswahl für den Datenaustausch

Um Daten mit anderen Systemen über Wechseldatenträger auszutauschen, sollten Sie diese mit einem Dateisystem formatieren, das von beiden Systemen unterstützt wird.

Anmerkung
Aussagen zu Festplatten (HDD) sind auch für andere Speichergeräte, wie SSD / USB-Flash-Speicher / Speicherkarten / usw. anwendbar. Ersetzen Sie in den Beispielen einfach die Gerätenamen wie `/dev/sda` durch passende Namen wie `/dev/nvme0`, `/dev/mmcblk0`, ...

Tipp
Details zum plattformübergreifenden Datenaustausch mit Verschlüsselung auf Geräteebene finden Sie in Abschnitt 9.9.1.

Dateisystemname	Typisches Anwendungsszenario
FAT12	Plattformübergreifender Datenaustausch mittels Diskette (<32 MiB)
FAT16	Plattformübergreifender Datenaustausch mittels kleiner festplatten-ähnlicher Geräte (<2 GiB)
FAT32	Plattformübergreifender Datenaustausch mittels großer festplatten-ähnlicher Geräte (<8 TiB, unterstützt von Systemen neuer als MS Windows95 OSR2)
exFAT	Plattformübergreifender Datenaustausch mittels großer festplatten-ähnlicher Geräte (<512 TiB, unterstützt von WindowsXP, Mac OS X Snow Leopard 10.6.5 und Linux-Kernel ab 5.4)
NTFS	Plattformübergreifender Datenaustausch mittels großer festplatten-ähnlicher Geräte (nativ unterstützt von MS Windows NT und späteren Versionen; auch unterstützt durch NTFS-3G via FUSE unter Linux)
ISO9660	Plattformübergreifender Austausch statischer Daten mittels CD-R und DVD+/-R
UDF	inkrementelles Schreiben von Daten auf CD-R und DVD+/-R (neu)
MINIX	platzsparendes Speichern von Unix-Daten auf Diskette
ext2	Datenaustausch mit älteren Linux-Systemen auf festplatten-ähnlichen Geräten
ext3	Datenaustausch mit älteren Linux-Systemen auf festplatten-ähnlichen Geräten
ext4	Datenaustausch mit aktuellen Linux-Systemen auf festplatten-ähnlichen Geräten
btrfs	Datenaustausch mit aktuellen Linux-Systemen auf festplatten-ähnlichen Geräten mit nur-lesbaren Schnappschüssen

Tabelle 10.3: Liste von Dateisystemen für Wechseldatenträger mit typischen Anwendungsszenarien

Das FAT-Dateisystem wird von nahezu allen modernen Dateisystemen unterstützt und ist für den Datenaustausch über Wechseldatenträger sehr nützlich.

Wenn Sie Geräte wie externe Festplatten für den plattformübergreifenden Datenaustausch mit dem FAT-Dateisystem formatieren, sollten die folgenden Varianten eine sichere Wahl sein:

- Partitionieren Sie das Medium mit fdisk(8), cfdisk(8) oder parted(8) (lesen Sie dazu Abschnitt 9.6.2) mit einer einzigen primären Partition und markieren Sie sie wie folgt:
  - Typ ”6” (FAT16) für Medien kleiner als 2 GB;
  - Typ ”c” (FAT32, LBA) für größere Medien.
- Formatieren der primären Partition mit mkfs.vfat(8) wie folgt:
  - einfach über den Gerätenamen, z.B. ”`/dev/sda1`” für FAT16;
  - über die explizite Option und den Gerätenamen, z.B. ”`-F 32 /dev/sda1`” für FAT32.

Wenn das FAT- oder ISO9660-Dateisystem für den Dateiaustausch verwendet wird, sollte folgendes eine sichere Variante sein:

- archivieren der Dateien in eine Archivdatei mittels tar(1) oder cpio(1), um die langen Dateinamen, symbolischen Links, originalen Unix-Dateiberechtigungen und Benutzerinformationen zu erhalten;
- splitten der Archivdatei in Stücke kleiner als 2 GiB mittels split(1), um so die Beschränkung der Dateigröße umgehen zu können;
- verschlüsseln der Archivdatei, um den Inhalt vor unberechtigtem Zugriff zu schützen.

Anmerkung
Bei dem FAT-Dateisystem liegt - begründet durch sein Design - die maximale Dateigröße bei `(2^32 - 1) Byte = (4 GiB - 1 Byte)`. Bei einigen Anwendungen auf älteren 32-Bit-Betriebssystemen war die maximale Dateigröße sogar noch kleiner: `(2^31 - 1) Byte = (2 GiB - 1 Byte)`. Debian ist von letzterem nicht betroffen.

Anmerkung
Microsoft selbst empfiehlt FAT nicht für Laufwerke oder Partitionen über 200 MB Größe. Microsoft hebt die Nachteile wie ineffiziente Speicherplatznutzung in seiner "Übersicht über die Dateisysteme FAT, HPFS und NTFS" hervor. Natürlich sollten wir für Linux normalerweise das ext4-Dateisystem nutzen.

Tipp
Mehr Informationen über Dateisysteme und Dateisystemzugriffe finden Sie im "Filesystems HOWTO" (Englisch).

10.1.9 Datenaustausch über das Netzwerk

Wenn Sie Daten mit anderen Systemen über das Netzwerk austauschen, sollten Sie allgemein gängige Dienste verwenden. Hier einige Hinweise:

Netzwerkdienst	Beschreibung des typischen Anwendungsszenarios
SMB/CIFS - über Netzwerk mit Samba eingebundenes Dateisystem	Datenaustausch über "Microsoft Windows Network", Näheres in smb.conf(5) und The Official Samba 3.x.x HOWTO and Reference Guide
NFS - über Netzwerk mittels Linux-Kernel eingebundenes Dateisystem	Datenaustausch über "Unix/Linux Network", Näheres in exports(5) und im Linux NFS-HOWTO
HTTP-Dienst	Datenaustausch zwischen Web-Server/-Client
HTTPS-Dienst	Datenaustausch zwischen Web-Server/-Client mit verschlüsseltem Secure Sockets Layer (SSL) oder Transport Layer Security (TLS)
FTP-Dienst	Datenaustausch zwischen FTP-Server/-Client

Tabelle 10.4: Liste von Netzwerkdiensten mit typischen Anwendungsszenarien

Obwohl Dateisysteme, die über solche Netzwerk-gestützten Transfermethoden eingebunden sind, für den Datenaustausch sehr praktisch sind, könnten Sie unsicher sein. Die genutzte Netzwerkverbindung muss wie folgt abgesichert sein:

- verschlüsseln Sie sie mit SSL/TLS;
- tunneln Sie sie via SSH;
- tunneln Sie sie via VPN;
- schränken Sie den Zugriff über eine Firewall ein.

Lesen Sie auch Abschnitt 6.5 und Abschnitt 6.6.

10.2 Datensicherung und -wiederherstellung

Wir alle wissen, dass Computer manchmal defekt sein können oder menschliche Fehler System- und Datenausfälle verursachen. Aktionen zur Datensicherung und -wiederherstellung sind unverzichtbarer Teil einer erfolgreichen Systemadministration. Alle möglichen Fehlervarianten werden Sie eines Tages ereilen.

Tipp
Halten Sie Ihr Datensicherungssystem einfach und führen Sie häufig Sicherungen durch. Aktuelle Sicherungen von seinen Daten zu haben ist wichtiger als die technische Qualität der Sicherungsmethodik.

10.2.1 Richtlinien für Datensicherung und -wiederherstellung

Es gibt drei Schlüsselfaktoren, die letztendig die Sicherungs- und Wiederherstellungsstrategie bestimmen:

1. Zu wissen, was man sichern und wiederherstellen muss:
   - Daten, die direkt von Ihnen selbst erstellt wurden: Daten in "`~/`";
   - Daten, die von Anwendungen, die Sie verwenden, erstellt wurden: Daten in "`/var/`" (außer "`/var/cache/`", "`/var/run/`" und "`/var/tmp/`");
   - Systemkonfigurationsdateien: Daten in "`/etc/`";
   - Lokale Programme: Daten in "`/usr/local/`" oder "`/opt/`";
   - Informationen zur Systeminstallation: Aufzeichnungen über wichtige Schritte (Partitionierung, …) in einfacher Textform;
   - Daten, von denen Sie wissen, dass sie wichtig sind, bestätigt durch im Vornherein versuchsweise durchgeführte Wiederherstellungsoperationen.
     - Cron-Job als Benutzerprozess: Dateien im Verzeichnis "`/var/spool/cron/crontabs`"; anschließend cron(8) neu starten. Informationen über cron(8) und crontab(1) in Abschnitt 9.4.14.
     - Systemd Timer-Jobs als Benutzerprozess: Dateien im Verzeichnis "`~/.config/systemd/user`". Näheres in systemd.timer(5) und systemd.service(5).
     - Autostart-Jobs als Benutzerprozess: Dateien im Verzeichnis "`~/.config/autostart`". Siehe Desktop Application Autostart Specification (Englisch).
2. Wissen, wie Sie Daten sichern und wiederherstellen:
   - Sicheres Speichern von Daten: geschützt vor Überschreiben und Systemausfällen;
   - Häufige Datensicherungen: Sicherungen planen;
   - Redundante Datensicherungen: Spiegeln der Sicherungsdateien;
   - Idiotensicheres Vorgehen: Sicherung durch einen einfachen Befehl.
3. Bewertung der entstehenden Risiken und Kosten:
   - Risiko, falls die Daten verloren gehen;
     - Daten sollten zumindest auf verschiedenen Festplattenpartitionen, besser sogar auf unterschiedlichen Laufwerken und Maschinen liegen, um unanfällig gegen Dateisystembeschädigungen zu sein. Wichtige Daten sollten am besten auf einem schreibgeschützten Dateisystem abgelegt werden. 1
   - Risiko, falls die Daten öffentlich werden;
     - Sensible Identitätsdaten wie "`/etc/ssh/ssh_host_*_key`", "`~/.gnupg/*`", "`~/.ssh/*`", "`~/.local/share/`", "`/etc/passwd`", "`/etc/shadow`", "`popularity-contest.conf`", "`/etc/ppp/pap-secrets`" oder "`/etc/ex`" sollten nur verschlüsselt gesichert werden. 2 (Siehe auch Abschnitt 9.9.)
     - Fügen Sie niemals System-Login-Passwörter oder Verschlüsselungs-Passphrasen in ein Skript ein, nicht einmal auf einem System, dem Sie vertrauen. (Siehe dazu Abschnitt 10.3.6.)
   - Mögliche Ausfälle und deren Wahrscheinlichkeit.
     - Hardware-Defekte (speziell bei Festplatten) sind wahrscheinlich
     - Dateisysteme könnten beschädigt werden und enthaltene Daten gehen verloren
     - Fernen Speichersystemen (Cloud, ...) können Sie aufgrund von Sicherheitslücken nicht trauen
     - Ein schwacher Passwortschutz kann leicht kompromittiert werden
     - Systeme zur Festlegung von Dateiberechtigungen können kompromittiert werden
   - Für die Sicherung benötigte Ressourcen: Personen, Hardware, Software, …;

1 Medien zu verwenden, die nur einmal beschrieben werden können (wie CD-R/DVD-R), schützt vor versehentlichem Überschreiben. (In Abschnitt 9.8 lesen Sie, wie Sie von der Befehlszeile aus auf die Speichermedien schreiben können. Die grafische Oberfläche der GNOME-Arbeitsplatzumgebung bietet Ihnen auch einen einfachen Weg über das Menü: "Orte → CD/DVD-Ersteller".)

2 Einige dieser Daten kann nicht wieder hergestellt werden, selbst wenn die exakt gleichen Eingaben erneut getätigt werden.

– Automatisch ablaufende Backups über cron-Jobs oder systemd-timer-Jobs

Tipp
Sie können die debconf-Konfigurationsdaten mit "`debconf-set-selections debconf-selections`" wiederherstellen und die dpkg-Paketauswahl mit "`dpkg --set-selection <dpkg-selections.list`".

Anmerkung
Machen Sie keine Sicherungen von den Pseudo-Dateisystemen in `/proc`, `/sys`, `/tmp` und `/run` (Näheres dazu in Abschnitt 1.2.12 und Abschnitt 1.2.13). Dies sind absolut nutzlose Daten, außer Sie genau wissen, was Sie tun.

Anmerkung
Sie sollten eventuell einige Anwendungs-Daemons wie den MTA (lesen Sie dazu Abschnitt 6.2.4) beenden, während Sie die Datensicherung durchführen.

10.2.2 Programmsammlungen für Datensicherungsaufgaben

Hier eine Auswahl erwähnenswerter, im Debian-System verfügbarer Datensicherungsprogramme:

Datensicherungs-Werkzeuge haben alle ihren speziellen Fokus:

- Mondo Rescue ist ein Backup-System, das die schnelle Wiederherstellung eines vollständigen Systems von CD/DVD ermöglicht, ohne dass dabei die normalen Systeminstallations-Prozesse durchlaufen werden müssen.
- Bacula, Amanda und BackupPC sind voll ausgestattete Backup-Lösungen, die auf die regelmäßige Datensicherung über Netzwerk fokussiert sind.
- Duplicity und Borg sind einfache Backup-Werkzeuge für typische Arbeitsplatzrechner.

10.2.3 Backup-Tipps

Für einen Arbeitsplatzrechner könnte eine voll ausgestattete, für Server-Umgebungen entwickelte Backup-Lösung unpassend sein. Gleichzeitig könnten aber vorhandene Backup-Werkzeuge für Arbeitsplatzrechner auch Schwächen haben.

Hier einige Tipps, um Backups zu erleichtern, bei gleichzeitig mimimalem Aufwand. Diese Techniken können bei allen Backup-Programmen Anwendung finden.

Für Demonstrationszwecke gehen wir hier davon aus, dass der primäre Benutzer- und Gruppenname `penguin` ist. Wir erstellen ein Beipiel für ein Backup- und Schnappschuss-Skript namens "`/usr/local/bin/bkss.sh`" wie folgt:

```
#!/bin/sh -e
SRC="$1" # source data path
DSTFS="$2" # backup destination filesystem path
DSTSV="$3" # backup destination subvolume name
DSTSS="${DSTFS}/${DSTSV}-snapshot" # snapshot destination path
if [ "$(stat -f -c %T "$DSTFS")" != "btrfs" ]; then
  echo "E: $DESTFS needs to be formatted to btrfs" >&2 ; exit 1
fi
MSGID=$(notify-send -p "bkup.sh $DSTSV" "in progress ...")
if [ ! -d "$DSTFS/$DSTSV" ]; then
  btrfs subvolume create "$DSTFS/$DSTSV"
  mkdir -p "$DSTSS"
fi
```

Paket	Popcon	Größe	Beschreibung
bacula-common	V:5.7, I:6.4	2501	Bacula: Datensicherung, -wiederherstellung und -verifizierung über das Netzwerk - gemeinsame Hilfsdateien
bacula-client	V:0.2, I:1.7	199	Bacula: Datensicherung, -wiederherstellung und -verifizierung über das Netzwerk - Client-Metapaket
bacula-console	V:0.6, I:2.0	112	Bacula: Datensicherung, -wiederherstellung und -verifizierung über das Netzwerk - Textkonsole
bacula-server	I:0.56	199	Bacula: Datensicherung, -wiederherstellung und -verifizierung über das Netzwerk - Server-Metapaket
amanda-common	V:0.5, I:2.0	9952	Amanda: Advanced Maryland Automatic Network Disk Archiver (Netzwerk-Backup-System - Bibliotheken)
amanda-client	V:0.5, I:2.0	1110	Amanda: Advanced Maryland Automatic Network Disk Archiver (Netzwerk-Backup-System - Client)
amanda-server	V:0.11, I:0.22	1096	Amanda: Advanced Maryland Automatic Network Disk Archiver (Netzwerk-Backup-System - Server)
backuppc	V:1.3, I:1.5	3088	BackupPC ist ein leistungsfähiges System der Enterprise-Klasse zur Datensicherung von PCs (festplatten-basiert)
duplicity	V:5, I:44	2679	Inkrementelles Backup (auch von fern)
deja-dup	V:28, I:41	5274	GUI-Frontend für duplicity
borgbackup	V:13, I:28	3717	deduplizierendes Backup (auch von fern)
borgmatic	V:3.4, I:4.4	946	borgbackup-Hilfsprogramm
rdiff-backup	V:2.1, I:6.0	1204	Inkrementelles Backup (auch von fern)
restic	V:10, I:17	44216	Inkrementelles Backup (auch von fern)
backupninja	V:2.0, I:2.4	360	ressourcenschonendes, erweiterbares **Meta-Backup**-System
slbackup	V:0.10, I:0.12	147	Inkrementelles Backup (auch von fern)
backup-manager	V:0.39, I:0.68	573	befehlszeilen-basiertes Datensicherungs-Werkzeug
backup2l	V:0.34, I:0.46	110	wartungsarmes Sicherungs-/Wiederherstellungsprogramm für mount-fähige Medien (festplatten-basiert)

Tabelle 10.5: Liste von Datensicherungsprogrammen

```
rsync -aHxS --delete --mkpath "${SRC}/" "${DSTFS}/${DSTSV}"
btrfs subvolume snapshot -r "${DSTFS}/${DSTSV}" ${DSTSS}/$(date -u --iso=min)
notify-send -r "$MSGID" "bkup.sh $DSTSV" "finished!"
```

Hier wird nur das Basiswerkzeug rsync(8) genutzt, um das System-Backup zu vereinfachen, und der Speicherplatz wird durch Btrfs effizient genutzt.

Tipp
Zu Ihrer Information: der Autor nutzt ein eigenes, ähnliches Shell-Skript "bss: Btrfs Subvolume Snapshot Utility" für seinen Arbeitsplatzrechner.

10.2.3.1 Grafische Backup-Software

Hier ein Beispiel zur Einrichtung einer grafischen Ein-Klick Backup-Lösung.

- Bereiten Sie einen USB-Massenspeicher vor, der für das Backup genutzt wird.
  - Formatieren Sie den USB-Speicher mit einer Partition als btrfs mit einem Label namens "BKUP". Dies kann auch verschlüsselt sein (siehe Abschnitt 9.9.1).
  - Schließen Sie ihn an Ihr System an. Das Arbeitsplatzsystem sollte ihn automatisch als "`/media/penguin/BKUP`" einbinden.
  - Führen Sie "`sudo chown penguin:penguin /media/penguin/BKUP`" aus, um ihn für den Benutzer schreibbar zu machen.
- Erstellen Sie "`~/.local/share/applications/BKUP.desktop`" (mit den Techniken unter Abschnitt 9.4.10), wie hier:

```
[Desktop Entry]
Name=bkss
Comment=Backup and snapshot of ~/Documents
Exec=/usr/local/bin/bkss.sh /home/penguin/Documents /media/penguin/BKUP Documents
Type=Application
```

Bei jedem Klick werden Ihre Daten aus "`~/Documents`" auf den USB-Speicher gesichert und es wird ein btrfs-Schnappschuss (nur lesbar) erzeugt.

10.2.3.2 Automatisches Backup, durch ein mount-Ereignis ausgelöst

Hier ist ein Beispiel, wie Sie ein automatisches Backup einrichten, das durch ein mount-Ereignnis ausgelöst wird.

- Bereiten Sie einen USB-Massenspeicher vor, der für das Backup genutzt wird (siehe Abschnitt 10.2.3.1).
- Erstellen Sie eine systemd Service-Unit-Datei "`~/.config/systemd/user/back-BKUP.service`" wie hier:

```
[Unit]
Description=USB Disk backup
Requires=media-%u-BKUP.mount
After=media-%u-BKUP.mount

[Service]
ExecStart=/usr/local/bin/bkss.sh %h/Documents /media/%u/BKUP Documents
StandardOutput=append:%h/.cache/systemd-snap.log
StandardError=append:%h/.cache/systemd-snap.log

[Install]
WantedBy=media-%u-BKUP.mount
```

- Aktivieren Sie diese systemd-Unit-Konfiguration mit:

```
$ systemctl --user enable bkup-BKUP.service
```

Bei jedem mount-Ereignis werden Ihre Daten aus "`~/Documents`" auf den USB-Speicher gesichert und ein btrfs-Schnappschuss (nur lesbar) wird erstellt.

Hierbei können Sie die Namen aller systemd-mount-Units, die systemd aktuell im Speicher hat, anzeigen lassen mit "`systemctl --user list-units --type=mount`".

10.2.3.3 Automatisches Backup, durch ein timer-Ereignis ausgelöst

Hier ein Beispiel für ein Backup, das automatisch durch ein timer-Ereignis ausgelöst wird.

- Bereiten Sie einen USB-Massenspeicher vor, der für das Backup genutzt wird (siehe Abschnitt 10.2.3.1).
- Erstellen Sie eine systemd timer-Unit-Datei "`~/.config/systemd/user/snap-Documents.timer`" wie hier:

```
[Unit]
Description=Run btrfs subvolume snapshot on timer
Documentation=man:btrfs(1)

[Timer]
OnStartupSec=30
OnUnitInactiveSec=900

[Install]
WantedBy=timers.target
```

- Erstellen Sie eine systemd Service-Unit-Datei "`~/.config/systemd/user/snap-Documents.service`":

```
[Unit]
Description=Run btrfs subvolume snapshot
Documentation=man:btrfs(1)

[Service]
Type=oneshot
Nice=15
ExecStart=/usr/local/bin/bkss.sh %h/Documents /media/%u/BKUP Documents
IOSchedulingClass=idle
CPUSchedulingPolicy=idle
StandardOutput=append:%h/.cache/systemd-snap.log
StandardError=append:%h/.cache/systemd-snap.log
```

- Aktivieren Sie diese systemd-Unit-Konfiguration mit:

```
$ systemctl --user enable snap-Documents.timer
```

Bei jedem timer-Ereignis werden Ihre Daten aus "`~/Documents`" auf den USB-Speicher gesichert und ein btrfs-Schnappschuss (nur lesbar) wird erstellt.

Hierbei können Sie die Namen aller systemd-timer-Units, die systemd aktuell im Speicher hat, anzeigen lassen mit "`systemctl --user list-units --type=timer`".

Für ein modernes Arbeitsplatzsystem kann dieser systemd-Ansatz einen feiner dosierten Ansatz bieten als die traditionellen Unix-Lösungen mit at(1), cron(8) oder anacron(8).

Paket	Popcon	Größe	Befehl	Beschreibung
gnupg	V:403, I:853	465	gpg(1)	GNU Privacy Guard - OpenPGP Verschlüsselungs- und Signaturwerkzeug
gpgv	V:206, I:911	576	gpgv(1)	GNU Privacy Guard - OpenPGP Werkzeug zur Verifizierung von Signaturen
sq	V:1, I:37	24191	sq(1)	Sequoia-PGP - OpenPGP Verschlüsselungs- und Signaturwerkzeug
sqv	V:517, I:595	1919	sqv(1)	Sequoia-PGP - OpenPGP - Werkzeug zur Verifizierung von Signaturen
paperkey	V:1, I:12	58	paperkey(1)	lediglich die geheimen Informationen aus geheimen OpenPGP-Schlüsseln extrahieren
cryptsetup	V:35, I:77	476	cryptsetup(8), …	Hilfsprogramme für die dm-crypt Blockgeräte-Verschlüsselung (enthält Unterstützung für LUKS)
coreutils	V:914, I:1000	19751	md5sum(1)	MD5-Message-Digest (Prüfsummen) berechnen und überprüfen
coreutils	V:914, I:1000	19751	sha1sum(1)	SHA1-Message-Digest (Prüfsummen) berechnen und überprüfen
openssl	V:853, I:996	2564	openssl(1ssl)	Message-Digest mit "`openssl dgst`" berechnen (OpenSSL)
libsecret-tools	V:2, I:14	44	secret-tool(1)	Speichern und Empfangen von Passwörtern (CLI)
seahorse	V:76, I:258	7999	seahorse(1)	Schlüsselverwaltungs-Werkzeug (GNOME)

Tabelle 10.6: Liste von Werkzeugen für die Datensicherheits-Infrastruktur

10.3 Datensicherheits-Infrastruktur

Die Sicherheitsinfrastruktur für Ihre Daten wird durch eine Kombination verschiedener Programme gewährleistet: Verschlüsselungswerkzeug, Message-Digest-Werkzeug und Signaturwerkzeug.

In Abschnitt 9.9 finden Sie Infos über dm-crypt und fscrypt, die automatische Datenverschlüsselungs-Infrastrukturen über Linux-Kernelmodule implementieren.

10.3.1 Schlüsselverwaltung für GnuPG

Hier einige Befehle für die grundlegende Schlüsselverwaltung mit GNU Privacy Guard:

Hier die Bedeutung des Vertrauenscodes:

Folgendes erzeugt meinen Schlüssel "`9563FC29932C409F1A77F9C1AB8A1522D8234C6A`".

```
$ gpg --gen-key
gpg (GnuPG) 2.4.7; Copyright (C) 2024 g10 Code GmbH
...
GnuPG needs to construct a user ID to identify your key.

Real name: Osamu Aoki
Email address: osamu@debian.org
You selected this USER-ID:
    "Osamu Aoki <osamu@debian.org>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
...
public and secret key created and signed.

pub   ed25519 2026-02-14 [SC] [expires: 2029-02-13]
```

Befehl	Beschreibung
`gpg --gen-key`	einen neuen Schlüssel generieren
`gpg --gen-revoke meine_Nutzer_ID`	einen Widerrufschlüssel (revoke key) für meine_Nutzer_ID generieren
`gpg --edit-key Nutzer_ID`	Schlüssel interaktiv editieren, ”help” eingeben für weitere Hilfe
`gpg -o datei --export`	alle öffentlichen Schlüssel in eine Datei exportieren
`gpg -o file --export-secret-keys`	alle privaten Schlüssel in eine Datei exportieren
`gpg --import datei`	alle Schlüssel aus datei importieren
`gpg --send-keys Nutzer_ID`	Schlüssel von Nutzer_ID zum Schlüsselserver übertragen
`gpg --recv-keys Nutzer_ID`	Schlüssel von Nutzer_ID vom Schlüsselserver empfangen
`gpg --list-keys Nutzer_ID`	Schlüssel von Nutzer_ID auflisten
`gpg --list-sigs user_ID`	Signaturen von Nutzer_ID auflisten
`gpg --check-sigs Nutzer_ID`	Signaturen von Nutzer_ID überprüfen
`gpg --fingerprint Nutzer_ID`	Fingerabdruck von Nutzer_ID überprüfen
`gpg --refresh-keys`	lokalen Schlüsselring aktualisieren

Tabelle 10.7: Liste von GNU-Privacy-Guard-Befehlen für die Schlüsselverwaltung

Code	Beschreibung des Vertrauenscodes
`-`	Kein Besitzervertrauen zugewiesen / noch nicht berechnet
`e`	Vertrauensberechnung fehlgeschlagen
`q`	nicht genügend Informationen für Berechnung
`n`	diesem Schlüssel niemals vertrauen
`m`	gerade noch vertrauenswert
`f`	voll vertrauenswert
`u`	absolut vertrauenswert

Tabelle 10.8: Liste der Bedeutungen des Vertrauenscodes

```
      9563FC29932C409F1A77F9C1AB8A1522D8234C6A
uid                      Osamu Aoki <osamu@debian.org>
sub   cv25519 2026-02-14 [E] [expires: 2029-02-13]
```

Folgendes lädt meinen Schlüssel "9563FC29932C409F1A77F9C1AB8A1522D8234C6A" auf den populären Schlüsselserver "hkp://keys.gnupg.net" hoch:

```
$ gpg --keyserver hkp://keys.gnupg.net --send-keys 9563FC29932C409F1A77F9C1AB8A1522D8234C6A
```

Ein gutes Standard-Schlüsselserver-Setup in "~/.gnupg/gpg.conf" (oder dem alten Ort "~/.gnupg/options") enthält folgendes:

```
keyserver hkp://keys.gnupg.net
```

Mit dem folgenden Befehl beziehen Sie unbekannte Schlüssel vom Schlüsselserver:

```
$ gpg --list-sigs --with-colons | grep '^sig.*\[User ID not found\]' |\
          cut -d ':' -f 5| sort | uniq | xargs gpg --recv-keys
```

Es gab einen Fehler in OpenPGP Public Key Server (Versionen vor 0.9.6), durch den Schlüssel mit mehr als zwei Unterschlüsseln korrumpiert wurden. Das neue gnupg-Paket (>1.2.1-2) kann mit diesen korrumpierten Unterschlüsseln umgehen. Lesen Sie in gpg(1) den Abschnitt zur Option "--repair-pks-subkey-bug".

Die Verwendung von SHA-1 für Hashs gilt als überholt. Wenn Ihr alter RSA-basierter openPGP-Schlüssel SHA-1 für Hashs verwendet, beheben Sie dies mit Hilfe von FixKeyWithSha1.

Tipp
Die Kommandos sq(1) und sqv(1) sind ein alternativer Satz von openPGP-Befehlen. Lesen Sie dazu die sq-Benutzerdokumentation sowie A Practical Introduction to using sq, Sequoia PGP's CLI.

10.3.2 Verwendung von GnuPG mit Dateien

Hier einige Beispiele für die Verwendung von GNU Privacy Guard mit Dateien:

10.3.3 Verwendung von GnuPG mit Mutt

Fügen Sie Folgendes zu "~/.muttrc" hinzu, um zu verhindern, dass das langsame GnuPG automatisch gestartet wird, während es durch Drücken von "S" im Indexmenü händisch gestartet werden kann:

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

10.3.4 Verwendung von GnuPG mit Vim

Das gnupg-Plugin ermöglicht Ihnen, GnuPG für Dateien mit den Endungen ".gpg", ".asc" und ".pgp" transparent laufen zu lassen: 3

```
$ sudo aptitude install vim-scripts
$ echo "packadd! gnupg" >> ~/.vim/vimrc
```

3Falls Sie "~/.vimrc" statt "~/.vim/vimrc" verwenden, passen Sie diese bitte entsprechend an.

Befehl	Beschreibung
`gpg -a -s datei`	Signieren von datei in die ASCII-bewehrte datei.asc
`gpg --armor --sign datei`	,,
`gpg --clearsign datei`	Signieren einer Nachricht mit Klartextsignatur
`gpg --clearsign datei|mail foo@example.org`	Versenden einer Nachricht mit Klartextsignatur an `foo@example.org`
`gpg --clearsign --not-dash-escaped patch-datei`	Klartext-signieren von patch-datei
`gpg --verify datei`	eine klartextsignierte datei verifizieren
`gpg -o datei.sig -b datei`	Erzeugen einer losgelösten Signatur
`gpg -o datei.sig --detach-sign datei`	,,
`gpg --verify datei.sig datei`	Verifizieren von datei mit datei.sig
`gpg -o crypt_datei.gpg -r name -e datei`	Verschlüsselung mit öffentlichem Schlüssel für name von datei in Binärdatei crypt_datei.pgp
`gpg -o crypt_datei.gpg --recipient name --encrypt datei`	,,
`gpg -o crypt_datei.asc -a -r name -e datei`	Verschlüsselung mit öffentlichem Schlüssel für name von datei in ASCII-bewehrte Datei crypt_datei.pgp
`gpg -o crypt_datei.gpg -c datei`	symmetrische Verschlüsselung von ”datei” in crypt_datei.gpg
`gpg -o crypt_datei.gpg --symmetric datei`	,,
`gpg -o crypt_datei.asc -a -c datei`	symmetrische Verschlüsselung für name von datei in ASCII-bewehrte Datei crypt_datei.asc
`gpg -o datei -d crypt_datei.gpg -r name`	Entschlüsselung
`gpg -o datei --decrypt crypt_datei.gpg`	,,

Tabelle 10.9: Liste von GNU-Privacy-Guard-Befehlen mit Dateien

10.3.5 Die MD5-Prüfsumme

md5sum(1) enthält ein Werkzeug, um über die in RFC1321 beschriebene Methode eine Digest-Datei zu erzeugen und jede Datei darüber zu verifizieren:

```
$ md5sum foo bar >baz.md5
$ cat baz.md5
d3b07384d113edec49eaa6238ad5ff00  foo
c157a79031e1c40f85931829bc5fc552  bar
$ md5sum -c baz.md5
foo: OK
bar: OK
```

Anmerkung
Die Berechnung der MD5-Prüfsumme ist weniger CPU-intensiv als die für die krypthographische Signatur durch GNU Privacy Guard (GnuPG). Üblicherweise ist nur die Digest-Datei der obersten Ebene (z.B. das Wurzelverzeichnis eines Verzeichnisbaums) kryptographisch signiert, um die Datenintegrität sicherzustellen.

10.3.6 Passwort-Schlüsselbund

Im GNOME-System verwaltet das grafische Hilfsprogramm seahorse(1) Passwörter und speichert sie sicher im Schlüsselbund `~/.local/share/keyrings/*`.

secret-tool(1) kann Passwörter über die Befehslzeile in diesem Schlüsselbund ablegen.

Lassen Sie uns die Passphrase für ein LUKS/dm-crypt-verschlüsseltes Festplatten-Image speichern:

```
$ secret-tool store --label='LUKS passphrase for disk.img' LUKS my_disk.img
Password: ********
```

Dieses gespeicherte Passwort kann ausgelesen und an andere Programme wie z.B. cryptsetup(8) übergegeen werden:

```
$ secret-tool lookup LUKS my_disk.img | \
  cryptsetup open disk.img disk_img --type luks --keyring -
$ sudo mount /dev/mapper/disk_img /mnt
```

Tipp
Wann immer Sie ein Passwort in einem Skript angeben müssen, verwenden Sie `secret-tool` und vermeiden Sie es, die Passphrase fest in dem Skript zu hinterlegen.

10.4 Werkzeuge zur Quellcode-Zusammenführung (merge)

Es gibt viele Werkzeuge für die Zusammenführung von Quellcode. Folgende Befehle haben meine Aufmerksamkeit erregt:

10.4.1 Unterschiede für Quelldateien extrahieren

Die folgenden Prozeduren extrahieren die Unterschiede zwischen zwei Quelldateien und erzeugen vereinheitlichte Diff-Dateien ”`file.patch0`” bzw. ”`file.patch1`”, abhängig vom Speicherort der Dateien:

```
$ diff -u file.old file.new > file.patch0
$ diff -u old/file new/file > file.patch1
```

Paket	Popcon	Größe	Befehl	Beschreibung
patch	V:100, I:729	243	patch(1)	eine diff-Datei auf ein Original anwenden
vim	V:73, I:330	4415	vimdiff(1)	zwei Dateien nebeneinander in vim vergleichen
imediff	V:0.02, I:0.21	329	imediff(1)	interaktives Werkzeug für Zwei-/Dreiwege-Zusammenführung mit Vollbildschirmmodus
meld	V:5, I:23	3996	meld(1)	vergleichen und zusammenführen von Dateien (GTK)
wiggle	V:0.01, I:0.12	191	wiggle(1)	zurückgewiesene Patches anwenden
diffutils	V:901, I:998	1768	diff(1)	Dateien Zeile für Zeile vergleichen
diffutils	V:901, I:998	1768	diff3(1)	drei Dateien Zeile für Zeile vergleichen und zusammenführen
quilt	V:2, I:16	880	quilt(1)	Serien von Patches verwalten
wdiff	V:5, I:37	580	wdiff(1)	Wort-für-Wort-Unterschiede zwischen Textdateien anzeigen
diffstat	V:8, I:91	86	diffstat(1)	eine Grafik der Veränderungen für eine diff-Datei erstellen
patchutils	V:10, I:91	269	combinediff(1)	einen kumulativen Patch aus zwei inkrementellen Patches erzeugen
patchutils	V:10, I:91	269	dehtmldiff(1)	ein Diff aus einer HTML-Seite extrahieren
patchutils	V:10, I:91	269	filterdiff(1)	Diffs aus einer Diff-Datei extrahieren oder entfernen
patchutils	V:10, I:91	269	fixcvsdiff(1)	von CVS erstellte Diff-Dateien, die patch(1) falsch interpretiert, reparieren
patchutils	V:10, I:91	269	flipdiff(1)	die Reihenfolge zweier Patches ändern
patchutils	V:10, I:91	269	grepdiff(1)	anzeigen, welche Dateien von einem Patch entsprechend einem regulären Ausdruck modifiziert werden
patchutils	V:10, I:91	269	interdiff(1)	Unterschiede zwischen zwei Unified-Diff-Dateien anzeigen
patchutils	V:10, I:91	269	lsdiff(1)	zeigen, welche Dateien von einem Patch verändert werden
patchutils	V:10, I:91	269	recountdiff(1)	Zähler und Offsets in vereinheitlichten Context-Diffs neu berechnen
patchutils	V:10, I:91	269	rediff(1)	Offsets und Zähler eines hand-editierten Diffs bereinigen
patchutils	V:10, I:91	269	splitdiff(1)	inkrementelle Patches aussortieren
patchutils	V:10, I:91	269	unwrapdiff(1)	Patches von fehlerhaften Zeilenumbrüchen befreien
dirdiff	V:0.1, I:1.3	167	dirdiff(1)	Unterschiede zwischen Verzeichnissen anzeigen und zusammenführen
docdiff	V:0.02, I:0.15	554	docdiff(1)	zwei Dateien Wort-für-Wort / Buchstabe-für-Buchstabe vergleichen
makepatch	V:0.01, I:0.14	99	makepatch(1)	erweiterte Patch-Dateien erzeugen
makepatch	V:0.01, I:0.14	99	applypatch(1)	erweiterte Patch-Dateien anwenden

Tabelle 10.10: Liste von Werkzeugen zur Quellcode-Zusammenführung

10.4.2 Aktualisierungen für Quelldateien zusammenführen

Die Diff-Datei (alternativ auch Patch-Datei genannt) wird genutzt, um eine Programmaktualisierung zu senden. Die Seite, die die Änderungen empfängt, wendet die Aktualisierung wie folgt an:

```
$ patch -p0 file < file.patch0
$ patch -p1 file < file.patch1
```

10.4.3 Interaktives Zusammenführen (merge)

Wenn Sie zwei Varianten eines Quelltextes haben, können Sie mittels imediff(1) eine interaktive Zweiwege-Zusammenführun durchführen:

```
$ imediff -o file.merged file.old file.new
```

Haben Sie drei Varianten eines Quelltextes, ist mit imediff(1) auch eine interaktive Dreiwege-Zusammenführung möglich:

```
$ imediff -o file.merged file.yours file.base file.theirs
```

10.5 Git

Git ist derzeit das Werkzeug der Wahl für Versionskontrollsysteme (VCS), da es alles Nötige sowohl für lokales wie auch für fernes Code-Management erledigen kann.

Debian stellt freie Git-Dienste über Debian Salsa bereit. Dokumentation finden Sie unter https://wiki.debian.org/Salsa.

Hier einige Git-betreffende Pakete:

10.5.1 Konfiguration eines Git-Clients

Sie möchten vielleicht wie folgt verschiedene globale Konfigurationsoptionen in "`~/.gitconfig`" zur Verwendung durch Git setzen, z.B. Name und E-Mail-Adresse:

```
$ git config --global user.name "Name Surname"
$ git config --global user.email yourname@example.com
```

Sie können das Standardverhalten von Git wie folgt auch noch weiter anpassen:

```
$ git config --global init.defaultBranch main
$ git config --global pull.rebase true
$ git config --global push.default current
```

Wenn Sie an CVS- oder Subversion-Befehle gewohnt sind, können Sie wie hier auch folgende Befehls-Alias setzen:

```
$ git config --global alias.ci "commit -a"
$ git config --global alias.co checkout
```

Sie können mit folgendem Befehl Ihre globale git-Konfiguration kontrollieren:

```
$ git config --global --list
```

Paket	Popcon	Größe	Befehl	Beschreibung
git	V:442, I:630	52512	git(1)	Git, das schnelle, skalierbare, verteilte Versionskontrollsystem
gitk	V:3, I:25	2045	gitk(1)	grafischer Browser mit Historie für Git-Depots
qgit	V:0.2, I:1.8	1431	qgit(1)	grafischer Browser mit Historie für Git-Depots
git-cola	V:0.7, I:4.4	5331	git-cola(1)	grafischer Browser mit Historie für Git-Depots
tig	V:2, I:11	1245	tig(1)	textbasierter Browser mit Historie für Git-Depots
lazygit	V:1.0, I:4.1	24167	lazygit(1)	textbasierter Browser mit Historie für Git-Depots
git-gui	V:1, I:15	2549	git-gui(1)	grafische Oberfläche (ohne Historie) für Git
git-email	V:0.4, I:8.6	1229	git-send-email(1)	eine Patch-Sammlung aus Git als E-Mail versenden
git-buildpackage	V:1.2, I:6.5	2027	git-buildpackage(1)	das Debian-Paketieren mit Git automatisieren
dgit	V:0.12, I:0.91	758	dgit(1)	Git-Interoperabilität mit dem Debian-Archiv
imediff	V:0.02, I:0.21	329	git-ime(1)	Interaktives Werkzeug, das hilft, umfangreiche Git-Commits weiter aufzusplitten
stgit	V:0.07, I:0.45	6005	stg(1)	Aufsatz für Git (Python)
git-doc	I:9.6	15192	Nicht verfügbar	offizielle Dokumentation für Git
gitmagic	I:0.43	721	Nicht verfügbar	”Git Magic”, eine einfacher verständliche Anleitung für Git

Tabelle 10.11: Liste von zu Git gehörigen Paketen und Befehlen

10.5.2 Grundlegende Git-Befehle

An Vorgängen mit Git sind verschiedene Datenbereiche beteiligt:

- Der Arbeitsbereich, der für den Benutzer sichtbare Dateien enthält und dort Änderungen erlaubt.
  - Änderungen müssen explizit ausgewählt und dem Index hingefügt werden, um aufgezeichnet zu werden. Dies erledigen die Befehle `git add` und `git rm`.
- Der Index, der hinzugefügte Dateien enthält.
  - Dem Index hinzugefügte Dateien werden auf Anfrage in das lokale Depot übernommen. Dies erledigt der Befehl `git commit`.
- Das lokale Depot, das übernommene Dateien enthält.
  - Git zeichnet die Historie der übernommenen Daten auf und bereitet diese in Form von Zweigen (Branches) im Depot auf.
  - Das lokale Depot kann über den `git push`-Befehl Daten an das ferne Depot senden.
  - Das lokale Depot kann über die Befehle `git fetch` und `git pull` Daten vom fernen Depot empfangen.
    - Der Befehl `git pull` führt `git merge` oder `git rebase` nach dem `git fetch`-Befehl aus.
    - Dabei führt `git merge` zwei separate Zweige (Branches) zu einem Stand zusammen. (Dies ist das Standard-verhalten von `git pull` - solange es nicht bewusst angepasst wurde - und sollte passend sein für Upstream-Entwickler, die Zweige (Branches) an viele Leute verteilen.)
    - Dabei erstellt `git rebase` einen einzigen Zweig (Branch) aus der aufeinander folgenden Historie des fernen Zweigs, dem der lokale Zweig folgt. (Dies entspricht der Anpassung `pull.rebase true`, und ist vermutlich passend für den Rest von uns.)
- Das ferne Depot, das übernommene Dateien enthält.
  - Für die Kommunikation zum fernen Depot wird ein sicheres Protokoll wie SSH oder HTTPS verwendet.

Als Arbeitsbereich gilt der Bereich außerhalb des `.git/`-Verzeichnisses. Zu den Dateien innerhalb von `.git/` gehören der Index, die lokalen Depotdaten und einige Textdateien für die Git-Konfiguration.

Hier ein Überblick über die wichtigsten Git-Befehle:

Git-Befehl	Funktion
`git init`	ein (lokales) Depot erstellen
`git clone URL`	duplizieren des fernen Depots in ein lokales Depot mit einem Arbeitsverzeichnis
`git pull origin main`	aktualisieren des lokalen `main`-Zweigs von dem fernen Depot `origin`
`git add .`	Änderungen in Dateien des Arbeitsbereichs zum Index hinzufügen - aber nur für Dateien, die bereits im Index existieren
`git add -A .`	Änderungen in Dateien des Arbeitsbereichs zum Index hinzufügen - für alle Dateien; betrifft auch Entfernungen von Dateien
`git rm dateiname`	Datei(en) aus dem Arbeitsbereich und dem Index entfernen
`git commit`	dem Index hinzugefügte Änderungen in das lokale Depot übernehmen (Commit)
`git commit -a`	alle Änderungen im Arbeitsbereich zum Index hinzufügen und in das lokale Depot übernehmen (vereint `git add` und `git commit`)
`git push -u origin zweig_name`	das ferne Depot `origin` vom lokalen Depot namens `zweig_name` aktualisieren (erstmaliger Aufruf)
`git push origin zweig_name`	das ferne Depot `origin` vom lokalen Depot namens `zweig_name` aktualisieren (nachfolgende Aufrufe)
`git diff treeish1 treeish2`	Unterschiede anzeigen zwischen Commit *baum1* und Commit *baum2*
`gitk`	grafische GUI-Anzeige der Historie eines Zweigs (Branch) von einem VCS-Depot

Tabelle 10.12: Wichtige Git-Befehle

10.5.3 Git-Tipps

Hier einige Tipps zu Git:



Warnung
Nutzen Sie in der Tag-Zeichenkette keine Leerzeichen, auch wenn Werkzeuge wie gitk(1) dies ermöglichen. Andere `git`-Befehle könnten dadurch ins Stocken geraten.



Achtung
Wenn für einen lokalen Zweig, der bereits über push in das ferne Depot übergeben wurde, ein rebase oder squash durchgeführt wird, könnte ein erneuter push gewisse Risiken beinhalten und die Option `--force` erfordern. Dies ist für den `main`-Zweig ein normalerweise unerwünschter Vorgang, könnte aber für einen topic-Zweig akzeptabel sein, bevor dieser in den `main`-Zweig integriert wird.



Achtung
Der direkte Aufruf eines `git`-Unterbefehls mittels ”`git-xyz`” über die Befehlszeile ist seit dem Frühjahr 2006 veraltet.

Git-Befehlszeile	Funktion
`gitk --all`	vollständige Git-Historie anzeigen und weiterverarbeiten, wie z.B. Zurücksetzen des HEAD auf einen anderen Commit, Herauspicken von einzelnen Änderungen, Erzeugen von Tags und Branches, ...
`git stash`	einen sauberen Arbeitsbereich erhalten, ohne Daten zu verlieren
`git remote -v`	Einstellung für ferne Depots anzeigen
`git branch -vv`	Einstellung für Zweig (Branch) anzeigen
`git status`	Status des Arbeitsbereichs anzeigen
`git config -l`	Git-Einstellungen auflisten
`git reset --hard HEAD; git clean -x -d -f`	alle Änderungen im Arbeitsbereich zurücksetzen und vollständig bereinigen
`git rm --cached dateiname`	bereits über `git add dateiname` zum Index hinzugefügte Änderungen zurücknehmen
`git reflog`	Git-Referenz-Log anzeigen (nützlich, um Commits von gelöschten Zweigen (Branches) wiederherzustellen)
`git branch neuer_zweig_name HEAD@{6}`	einen neuen Zweig aus reflog-Informationen erzeugen
`git remote add new_remote URL`	ein fernes Depot `new_remote` mit angegebener URL hinzufügen
`git remote rename origin upstream`	den Namen des fernen Depots `origin` in `upstream` umbenennen
`git branch -u upstream/branch_name`	das Remote-Tracking auf das ferne Depot `upstream` und seinen Zweig namens `branch_name` setzen.
`git remote set-url origin https://foo/bar.git`	URL von `origin` ändern
`git remote set-url --push upstream DISABLED`	push zu `upstream` deaktivieren (passen Sie `.git/config` an, um dies wieder zu aktivieren)
`git remote update upstream`	Updates aller fernen Zweige im `upstream`-Depot abrufen
`git fetch upstream foo:upstream-foo`	einen lokalen (eventuell verwaisten) Zweig `upstream-foo` als Kopie des `foo`-Zweigs im `upstream`-Depot erzeugen
`git checkout -b topic_branch ; git push -u topic_branch origin`	einen neuen Zweig namens `topic_branch` erzeugen und nach `origin` aktualisieren (push)
`git branch -m alter-name neuer-name`	den Namen eines lokalen Zweigs umbenennen
`git push -d origin branch_to_be_removed`	einen fernen Zweig (Branch) löschen (neue Methode)
`git push origin :branch_to_be_removed`	einen fernen Zweig (Branch) löschen (alte Methode)
`git checkout --orphan unconnected`	einen neuen `unconnected`-Zweig erzeugen
`git rebase -i origin/main`	Neuordnen/Löschen/Verdichten der Commits von `origin/main`, um eine saubere Branch-Historie zu bekommen
`git reset HEAD^; git commit --amend`	Verdichten der letzten 2 Commits in einen einzigen
`git checkout topic_branch ; git merge --squash topic_branch`	Verdichten eines kompletten Zweigs `topic_branch` in einen Commit
`git fetch --unshallow --update-head-ok origin '+refs/heads/*:refs/heads/*'`	einen shallow-Clone in einen Clone aller Zweige konvertieren
`git ime`	den letzten Commit in eine Serie von kleineren Datei-für-Datei-Commits konvertieren (`imediff`-Paket wird benötigt)
`git repack -a -d; git prune`	Neupacken des lokalen Depots in ein einzelnes Pack (dies könnte dazu führen, dass die Wiederherstellung verlorener Daten aus gelöschten Zweigen nicht mehr möglich ist, usw.)

Tabelle 10.13: Git-Tipps

Tipp
Wenn eine ausführbare Datei namens `git-foo` in dem durch `$PATH` definierten Pfad existiert, wird dieser `git-foo`-Befehl ausgeführt, indem man "`git foo`" (ohne Bindestrich!) auf der Befehlszeile aufruft. Dies ist eine Funktionalität des `git`-Befehls.

10.5.4 Weitere Referenzen zu Git

Hier finden Sie weitere Informationen:

- Handbuchseite: git(1) (`/usr/share/doc/git-doc/git.html`)
- git-Benutzerhandbuch (engl.) (`/usr/share/doc/git-doc/user-manual.html`)
- gittutorial: Einführung in git (engl.) (`/usr/share/doc/git-doc/gittutorial.html`)
- gittutorial-2: Einführung in git - Teil 2 (engl.) (`/usr/share/doc/git-doc/gittutorial-2.html`)
- Anleitung für die alltägliche git-Nutzung mit ungefähr 20 Befehlen (engl.) (`/usr/share/doc/git-doc/giteveryday.h`
- Git Magic (`/usr/share/doc/gitmagic/html/index.html`)

10.5.5 Andere Versionskontrollsysteme

Versionskontrollsysteme (VCS) sind auch teilweise bekannt als Revision Control System (RCS) oder Software Configuration Management (SCM).

Hier eine Zusammenfassung von VCS-Systemen im Debian-System abseits von Git:

Paket	Popcon	Größe	Werkzeug	VCS-Typ	Erläuterung
`mercurial`	V:3, I:24	2667	Mercurial	verteilt	DVCS in Python und ein bisschen C
`darcs`	V:0.2, I:2.9	38856	Darcs	verteilt	DVCS mit schlauer Algebra von Patches (langsam)
`tla`	V:0.03, I:0.60	1022	GNU arch	verteilt	DVCS hauptsächlich von Tom Lord (historisch)
`bzr`	V:0.1, I:3.8	28	GNU Bazaar	verteilt	von `tla` beeinflußtes DVCS, geschrieben in Python
`subversion`	V:9, I:53	5211	Subversion	fern	"CVS done right" (CVS richtig gemacht), neuerer Standard für fernes VCS (historisch)
`cvs`	V:3, I:22	4835	CVS	fern	früherer Standard für fernes VCS (historisch)
`tkcvs`	V:0.10, I:0.87	34	CVS, …	fern	GUI-Oberfläche von VCS-Depots (CVS, Subversion, RCS)
`rcs`	V:1.6, I:8.8	578	RCS	lokal	"Unix SCCS done right" (Unix SCCS richtig gemacht; historisch)
`cssc`	V:0.01, I:0.27	2044	CSSC	lokal	Klon von Unix SCCS (historisch)

Tabelle 10.14: Liste anderer Versionskontrollsystem-Werkzeuge

Kapitel 11

Datenkonvertierung

Hier werden Werkzeuge und Tipps für die Umwandlung von Dateien in andere Formate beschrieben.

Bei Werkzeugen für auf Standards basierende Formate ist die Situation sehr gut, aber bei proprietären Formaten sind die Möglichkeiten eingeschränkt.

11.1 Werkzeuge für Textkonvertierung

Folgende Pakete zur Textkonvertierung sind mir aufgefallen:

Paket	Popcon	Größe	Schlüsselwort	Beschreibung
libc6	V:924, I:1000	5483	charset	Konvertierung der Textkodierung zwischen verschiedenen Gebietsschemata (Locales) mit iconv(1) (elementar)
recode	V:2, I:12	534	charset+eol	Konvertierung der Textkodierung zwischen verschiedenen Gebietsschemata (Locales) (vielfältig, mehr Alias-Befehle und Funktionalitäten)
konwert	V:2, I:42	137	charset	Konvertierung der Textkodierung zwischen verschiedenen Gebietsschemata (Locales) (extravagant)
nkf	V:0.4, I:8.3	360	charset	Zeichensatzkonvertierer für Japanisch
tcs	V:0.01, I:0.12	518	charset	Zeichensatzkonvertierer
unaccent	V:0.03, I:0.26	35	charset	akzentuierte Buchstaben durch ihre nicht akzentuierten Pendants ersetzen
tofrodos	V:1, I:11	50	eol	Konvertiert Textformate zwischen DOS und Unix: fromdos(1) und todos(1)
macutils	V:0.03, I:0.38	319	eol	Konvertiert Textformate zwischen Macintosh und Unix: frommac(1) und tomac(1)

Tabelle 11.1: Liste von Textkonvertierungs-Werkzeugen

11.1.1 Konvertieren einer Textdatei mit iconv

Tipp
iconv(1) ist Teil des `libc6`-Pakets und immer auf nahezu allen Unix-artigen Systemen für die Änderung der Zeichenkodierung verfügbar.

Sie können die Kodierung einer Textdatei wie folgt mit iconv(1) ändern:

```
$ iconv -f encoding1 -t encoding2 input.txt >output.txt
```

Bei den Werten für die Kodierung ist die Groß-/Kleinschreibung nicht relevant, "`-`" und "_" werden ignoriert. Mit "`iconv -l`" können Sie überprüfen, welche Kodierungen unterstützt werden.

Wert für Zeichenkodierung	Verwendung
ASCII	American Standard Code for Information Interchange, 7-Bit-Code ohne akzentuierte Zeichen
UTF-8	aktueller multilingualer Standard für alle modernen Betriebssysteme
ISO-8859-1	alter Standard für westeuropäische Sprachen; ASCII + akzentuierte Zeichen
ISO-8859-2	alter Standard für osteuropäische Sprachen; ASCII + akzentuierte Zeichen
ISO-8859-15	alter Standard für westeuropäische Sprachen; ISO-8859-1 + Euro-Zeichen
CP850	Codepage 850, Microsoft-DOS-Zeichen mit Grafiken für westeuropäische Sprachen, Variante von ISO-8859-1
CP932	Codepage 932, Shift-JIS-Variante für Japanisch (angelehnt an Microsoft Windows)
CP936	Codepage 936, GB2312-, GBK- oder GB18030-Variante für vereinfachtes Chinesisch (angelehnt an Microsoft Windows)
CP949	Codepage 949, EUC-KR- oder Unified-Hangul-Code-Variante für Koreanisch (angelehnt an Microsoft Windows)
CP950	Codepage 950, Big5-Variante für traditionelles Chinesisch (angelehnt an Microsoft Windows)
CP1251	Codepage 1251, Kodierung für das kyrillische Alphabet (angelehnt an Microsoft Windows)
CP1252	Codepage 1252, ISO-8859-15-Variante für westeuropäische Sprachen (angelehnt an Microsoft Windows)
KOI8-R	alter russischer UNIX-Standard für das kyrillische Alphabet
ISO-2022-JP	Standard-Kodierung für japanische E-Mails, die nur 7-Bit-Codes verwenden
eucJP	alter japanischer UNIX-Standard-Code (8-Bit), völlig verschieden von Shift-JIS
Shift-JIS	Standard für Japanisch gemäß JIS X 0208 Anhang 1 (siehe auch CP932)

Tabelle 11.2: Liste von Werten für die Zeichenkodierung und deren Verwendung

Anmerkung
Einige der obigen Kodierungen werden nur für die Konvertierung unterstützt und nicht als Wert für das Gebietsschema (Locale) (Abschnitt 8.1).

Bei Zeichensätzen, die nur ein einziges Byte benötigen (wie ASCII und ISO-8859), entspricht die Zeichenkodierung nahezu dem Zeichensatz.

Bei Zeichensätzen mit vielen Zeichen (wie JIS X 0213 für Japanisch oder Universal Character Set (UCS, Unicode, ISO-10646-1) für praktisch alle Sprachen) gibt es viele Kodierungsschemata, die in die Sequenz der Byte-Daten eingepasst werden:

- EUC und ISO/IEC 2022 (auch bekannt als JIS X 0202) für Japanisch;
- UTF-8, UTF-16/UCS-2 und UTF-32/UCS-4 für Unicode.

Bei diesen gibt es klare Differenzierungen zwischen Zeichensatz und Zeichenkodierung.

Die Codepage wird als Synonym für einige hersteller-spezifische Zeichenkodierungstabellen verwendet.

Anmerkung
Bitte beachten Sie, dass die meisten Kodierungssysteme sich bei den 7-Bit-Zeichen identischen Code mit ASCII teilen, aber es gibt einige Ausnahmen. Wenn Sie alte japanische C-Programme und URLs aus dem Shift-JIS genannten Kodierungsformat nach UTF-8 konvertieren, müssen Sie "`CP932`" als Kodierungsname statt "`shift-JIS`" verwenden, um die erwarteten Resultate zu bekommen: `0x5C` → "\" und `0x7E` → "~". Andernfalls werden diese falsch konvertiert.

Tipp
recode(1) kann ebenfalls verwendet werden und bietet mehr als die kombinierte Funktionalität von iconv(1), fromdos(1), todos(1), frommac(1) und tomac(1). Weitere Informationen finden Sie unter "`info recode`".

11.1.2 Prüfen mit iconv, ob eine Datei UTF-8-kodiert ist

Sie können mit iconv(1) wie folgt überprüfen, ob eine Textdatei in UTF-8 kodiert ist:

```
$ iconv -f utf8 -t utf8 input.txt >/dev/null || echo "non-UTF-8 found"
```

Tipp
Verwenden Sie die Option "`--verbose`" in obigem Beispiel, um das erste nicht in UTF-8 kodierte Zeichen zu finden.

11.1.3 Dateinamen konvertieren mit iconv

Hier ein Beispielskript, um die Kodierung für alle Dateinamen in einem Verzeichnis von einer auf einem älteren Betriebssystem erzeugten Form in das moderne UTF-8 zu konvertieren:

```
#!/bin/sh
ENCDN=iso-8859-1
for x in *;
 do
 mv "$x" "$(echo "$x" | iconv -f $ENCDN -t utf-8)"
done
```

Die "`$ENCDN`"-Variable gibt dabei die Original-Kodierung (gemäß Tabelle 11.2) an, die in dem älteren Betriebssystem für die Dateinamen verwendet wurde.

In komplizierteren Fällen binden Sie bitte ein Dateisystem, das solche Dateinamen enthält (z.B. eine Festplattenpartition), mit korrekter Angabe der Dateinamenkodierung als Option zum mount(8)-Befehl ein (lesen Sie dazu Abschnitt 8.1.3), und kopieren Sie mit "`cp -a`" den vollständigen Inhalt der Partition in ein anderes Dateisystem, das als UTF-8 eingebunden ist.

11.1.4 EOL-Konvertierung

Das Format einer Textdatei, speziell der EOL-Code (end-of-line, Zeilenende), ist abhängig von der Systemplattform.

Plattform	EOL-Code	Steuerung	dezimal	hexadezimal
Debian (Unix)	LF	^J	10	0A
MSDOS und Windows	CR-LF	^M^J	13 10	0D 0A
Apples Macintosh	CR	^M	13	0D

Tabelle 11.3: Liste der EOL-Codes für verschiedene Plattformen

Die Programme zur Konvertierung von Zeilenumbruchformaten fromdos(1) beziehungsweise todos(1) (siehe Paket `tofrodos`) sowie frommac(1) beziehungsweise tomac(1) (siehe Paket `macutils`) sind recht nützlich. Auch recode(1) ist hilfreich.

Anmerkung
Die meisten Editoren (wie `vim`, `emacs`, `gedit`, ...) können mit Dateien mit EOL-Code im MSDOS-Stil transparent umgehen, ohne dass Sie es merken.

Tipp
Um eine Mischung aus MSDOS- und Unix-artigem EOL-Stil (z.B. nach der Zusammenführung von zwei Dateien im MSDOS-Stil mit diff3(1)) in einen einheitlichen MSDOS-Stil umzuwandeln, ist die Verwendung von "`sed -e '/\r$/!s/$/\r/'`" der von todos(1) vorzuziehen. Der Grund hierfür ist, dass `todos` jeder Zeile ein CR hinzufügt.

11.1.5 TAB-Konvertierung

Es gibt ein paar bekannte spezialisierte Programme für die Konvertierung der TAB-Codes:

Funktion	`bsdmainutils`	`coreutils`
TAB in Leerzeichen wandeln	"`col -x`"	`expand`
Leerzeichen in TAB wandeln	"`col -h`"	`unexpand`

Tabelle 11.4: Liste der Befehle zur TAB-Konvertierung aus den Paketen `bsdmainutils` und `coreutils`

indent(1) aus dem `indent`-Paket formatiert alle Whitespaces (Leerraumzeichen) in einem C-Programm neu.

Auch Editoren wie `vim` und `emacs` können zur TAB-Konvertierung genutzt werden. Bei `vim` z.B. verwenden Sie die Befehlssequenz "`:set expandtab`" + "`:%retab`", um ein TAB zum Leerzeichen zu expandieren. Den umgekehrten Fall erreichen Sie mit "`:set noexpandtab`" + "`:%retab!`".

11.1.6 Editoren mit automatischer Konvertierung

Moderne Editoren wie `vim` sind sehr clever und können mit jeglichen Kodierungssystemen und Dateiformaten umgehen. Für beste Kompatibilität sollten Sie diese Editoren mit einem UTF-8-Gebietsschema in einer UTF-8-tauglichen Konsole verwenden.

Eine alte westeuropäische Unix-Textdatei "`u-datei.txt`", gespeichert in der alten latin1-Kodierung (ISO-8859-1), kann mit `vim` einfach wie folgt bearbeitet werden:

```
$ vim u-file.txt
```

Dies ist möglich, da die automatische Erkennung der Dateikodierung in `vim` zunächst von einer UTF-8-Kodierung ausgeht und, falls dies fehlschlägt, latin1 verwendet.

Eine alte polnische Unix-Textdatei "`pu-datei.txt`", gespeichert in der alten latin2-Kodierung (ISO-8859-2), kann mit `vim` wie folgt bearbeitet werden:

```
$ vim '+e ++enc=latin2 pu-file.txt'
```

Eine alte japanische Unix-Textdatei "`ju-datei.txt`", gespeichert in der eucJP-Kodierung, kann mit `vim` wie folgt bearbeitet werden:

```
$ vim '+e ++enc=eucJP ju-file.txt'
```

Eine alte japanische MS-Windows-Textdatei "`jw-datei.txt`", gespeichert in der sogenannten Shift-JIS-Kodierung (präziser: CP932), kann mit `vim` wie folgt bearbeitet werden:

```
$ vim '+e ++enc=CP932 ++ff=dos jw-file.txt'
```

Wenn eine Datei mit den Optionen "`++enc`" und "`++ff`" geöffnet wird, speichert "`:w`" in der Vim-Befehlszeile sie im Originalformat ab und überschreibt die Originaldatei. Sie können auch das zum Speichern zu nutzende Format und den Dateinamen mit angeben, z.B. "`:w ++enc=utf8 neu.txt`".

Bitte lesen Sie den Abschnitt zum "Multi-byte text support" (mbyte.txt) in der `vim`-Online-Hilfe sowie Tabelle 11.2 für Infos, welche Locale-Werte mit der "`++enc`"-Option genutzt werden können.

Die `emacs`-Programmfamilie bietet ähnliche Funktionalitäten.

11.1.7 Extrahieren von reinem Text

Folgender Befehl liest eine Webseite in eine Textdatei ein. Das ist sehr nützlich, wenn Sie Konfigurationsbeispiele aus dem Web kopieren oder grundlegende Unix-Textbearbeitungswerkzeuge wie grep(1) auf den Inhalt der Webseite anwenden möchten.

```
$ w3m -dump https://www.remote-site.com/help-info.html >textfile
```

Ähnlich dazu können Sie reine Textdaten wie folgt aus anderen Formaten extrahieren:

11.1.8 Hervorheben und Formatieren von reinen Textdaten

Reine Textdaten können mit folgenden Befehlen hervorgehoben und formatiert werden:

11.2 XML-Daten

Die Extensible Markup Language (XML) ist eine Markup-Sprache für Dokumente mit Strukturinformationen.

Einführende Informationen finden Sie unter XML.COM:

- "Was ist XML? (englisch)"
- "Was ist XSLT? (englisch)"
- "Was ist XSL-FO? (englisch)"
- "Was ist XLink? (englisch)"

Paket	Popcon	Größe	Schlüsselwort	Funktion
w3m	V:11, I:125	2853	html→text	HTML-zu-Text-Konvertierung mit dem Befehl "w3m -dump"
html2text	V:4, I:66	298	html→text	fortgeschrittener HTML-zu-Text-Konvertierer (ISO 8859-1)
lynx	V:30, I:454	2031	html→text	HTML-zu-Text-Konvertierung mit dem Befehl "lynx -dump"
elinks	V:2, I:14	1839	html→text	HTML-zu-Text-Konvertierung mit dem Befehl "elinks -dump"
links	V:2, I:19	2344	html→text	HTML-zu-Text-Konvertierung mit dem Befehl "links -dump"
links2	V:0.8, I:9.2	5497	html→text	HTML-zu-Text-Konvertierung mit dem Befehl "links2 -dump"
catdoc	V:16, I:169	682	MSWord→text,TeX	Konvertierung von MSWord-Dateien in reinen Text oder TeX
antiword	V:0.9, I:6.1	596	MSWord→text,ps	Konvertierung von MSWord-Dateien in reinen Text oder PostScript
unhtml	V:0.04, I:0.50	40	html→text	Entfernen der Markups (Markierungen) aus einer HTML-Datei
odt2txt	V:1, I:17	60	odt→text	Konvertierung von OpenDocument-Text in reinen Text

Tabelle 11.5: Liste von Werkzeugen zum Extrahieren von reinen Textdaten

Paket	Popcon	Größe	Schlüsselwort	Beschreibung
vim-runtime	V:18, I:347	39473	Hervorheben	Vim-Macro zur Konvertierung von Quellcode nach HTML (mit ":source $VIMRUNTIME/syntax/html.vim")
cxref	V:0.02, I:0.20	1191	c→html	Konvertierung eines C-Programms nach Latex und HTML (C-Sprache)
src2tex	V:0.02, I:0.14	1799	Hervorheben	Konvertierung von vielen Quellcode-Formaten nach TeX (C-Sprache)
source-highlight	V:0.4, I:3.0	2131	Hervorheben	Konvertierung von vielen Quellcode-Formaten nach HTML, XHTML, LaTeX, Texinfo, ANSI-Color-Escape-Sequenzen und DocBook-Dateien mit Hervorhebung (C++)
highlight	V:0.3, I:2.7	1452	Hervorheben	Konvertierung von vielen Quellcode-Formaten nach HTML, XHTML, RTF, LaTeX, TeX oder XSL-FO-Dateien mit Hervorhebung (C++)
grc	V:0.9, I:5.4	208	text→color	grundlegender Einfärber für alles (Python)
pandoc	V:12, I:48	209559	text→any	grundlegender Markup-Konvertierer (Haskell)
python3-docutils	V:13, I:51	2012	text→any	reStructuredText-Dokument-Formatierer nach XML (Python)
markdown	V:0.4, I:5.5	56	text→html	Markdown: Textdokument-Konvertierer in (X)HTML (Perl)
asciidoctor	V:0.4, I:4.4	101	text→any	AsciiDoc: Textdokument-Konvertierer in XML/HTML (Ruby)
python3-sphinx	V:7, I:27	3245	text→any	auf reStructured Text basierendes Dokumenten-Publikationssystem (Python)
hugo	V:0.8, I:5.0	66919	text→html	auf Markdown basierendes Static-Site Publikationssystem (Go)

Tabelle 11.6: Liste von Werkzeugen für Hervorhebung/Formatierung von Textdaten

11.2.1 Grundlegende Hinweise für XML

Ein XML-Text sieht ein wenig wie HTML aus. XML ermöglicht es uns, verschiedene Ausgabeformate für ein und dasselbe Dokument zu verwalten. Ein einfaches XML-System ist `docbook-xsl`,das auch für dieses Dokument verwendet wird.

Jede XML-Datei beginnt mit einer Standard-XML-Deklaration wie der folgenden:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Die grundlegende Syntax für ein XML-Element ist wie folgt gekennzeichnet:

```
<name attribute="value">content</name>
```

Die Kurzform für ein XML-Element mit leerem Inhalt ist wie folgt:

```
<name attribute="value" />
```

Das "`attribute="Wert"`" in obigen Beispielen ist optional.

Ein Kommentar-Abschnitt wird in XML wie folgt gekennzeichnet:

```
<!-- comment -->
```

Anders als beim Hinzufügen von Markierungen (Markups) erfordert XML minimale Konvertierungen am Inhalt, um vordefinierte Entitäten für die folgenden Zeichen zu nutzen:

vordefinierte Entität	**Zeichen, in das konvertiert werden soll**
`"`	" (Anführungszeichen)
`'`	' (Apostroph)
`<`	< (kleiner-als)
`>`	> (größer-als)
`&`	& (kaufmännisches Und)

Tabelle 11.7: Liste von vordefinierten Entitäten für XML



Achtung
"<" oder "&" können nicht in Attributen oder Elementen verwendet werden.

Anmerkung
Wenn anwenderdefinierte Entitäten im SGML-Stil verwendet werden, wie z.B. "`&irgendein-begriff;`", wird die erste Definition gegenüber darauffolgenden bevorzugt. Die Entität wird in Form von "`<!ENTITY irgendein-begriff "Wert der Entität">`" definiert.

Anmerkung
Solange die XML-Markierungen konsistent mit einer bestimmten Art von Namen für diese Markierungen (entweder Daten als Inhalt oder Attributwert) ausgeführt sind, ist die Konvertierung in eine andere XML-Form mittels Extensible Stylesheet Language Transformations (XSLT) eine banale Aufgabe.

11.2.2 XML-Verarbeitung

Es gibt viele Werkzeuge zur Verarbeitung von XML-Dateien, wie z.B. die Extensible Stylesheet Language (XSL).

Grundsätzlich ist es so, dass Sie eine korrekt formatierte XML-Datei mittels Extensible Stylesheet Language Transformation (XSLT) in jegliches Format umwandeln können.

Extensible Stylesheet Language for Formatting Objects (XSL-FO) ist eine Lösung zur Formatierung. Das `fop`-Paket ist (aufgrund seiner Abhängigkeit zur Java-Programmiersprache) neu in Debians `main`-Archiv. Daher wird im Allgemeinen LaTEX-Code mittels XSLT aus XML erstellt und das LaTEX-System dann verwendet, um druckfähige Dateien wie DVI, PostScript oder PDF zu erzeugen.

Paket	Popcon	Größe	Schlüsselwort	Beschreibung
`docbook-xml`	V:15, I:403	2126	xml	XML-Dokumententyp-Definition (DTD) für DocBook
`docbook-xsl`	V:15, I:141	14823	xml/xslt	XSL-Stylesheets (Stilvorlagen), um DocBook-XML mittels XSLT in verschiedene Ausgabeformate umzuwandeln
`xsltproc`	V:15, I:69	83	xslt	XSLT-Befehlszeilen-Prozessor (XML → XML, HTML, reinen Text usw.)
`xmlto`	V:0.5, I:7.7	124	xml/xslt	XML-nach-alles-Konvertierer mit XSLT
`fop`	V:0.6, I:7.6	282	xml/xsl-fo	Docbook-XML-Dateien nach PDF konvertieren
`dblatex`	V:0.9, I:5.3	4635	xml/xslt	DocBook-Dateien mittels XSLT nach DVI, PostScript und PDF konvertieren
`dbtoepub`	V:0.04, I:0.52	37	xml/xslt	DocBook-XML-nach-epub-Konvertierer

Tabelle 11.8: Liste von XML-Werkzeugen

Da XML eine Untermenge der Standard Generalized Markup Language (SGML) ist, kann es mit den umfangreichen Werkzeugen, die für SGML verfügbar sind (wie der Document Style Semantics and Specification Language (DSSSL)) verarbeitet werden.

Paket	Popcon	Größe	Schlüsselwort	Beschreibung
`openjade`	V:1, I:20	1066	`dsssl`	ISO/IEC 10179:1996 - standardkonformer DSSSL-Prozessor (aktuell)
`docbook-dsssl`	V:0.5, I:7.2	2594	`xml/dsssl`	DSSSL-Stylesheets (Stilvorlagen), um DocBook-XML mittels DSSSL in verschiedene Ausgabeformate umzuwandeln
`docbook-utils`	V:0.3, I:5.1	287	`xml/dsssl`	Werkzeuge für DocBook-Dateien (inklusive Konvertierung in andere Formate (HTML, RTF, PS, man, PDF) mit DSSSL mittels `docbook2*`-Befehlen)

Tabelle 11.9: Liste von DSSSL-Werkzeugen

Tipp
GNOMEs `yelp` ist manchmal praktisch, da es DocBook-XML-Dateien ohne Konvertierung direkt vernünftig darstellen kann.

11.2.3 Extrahierung von XML-Daten

Sie können HTML- oder XML-Daten mit folgenden Programmen aus anderen Formaten extrahieren:

Paket	Popcon	Größe	Schlüsselwort	Beschreibung
man2html	V:0.1, I:1.2	142	manpage → html	Konvertierer von Handbuchseite (manpage) nach HTML (CGI-Unterstützung)
doclifter	V:0.00, I:0.05	487	troff → xml	Konvertierer von troff nach DocBook XML
texi2html	V:0.2, I:2.6	1847	texi → html	Konvertierer von Texinfo nach HTML
info2www	V:0.8, I:1.4	76	info → html	Konvertierer von GNU info nach HTML (CGI-Unterstützung)
wv	V:0.2, I:2.2	733	MSWord → any	Dokumentenkonvertierer von Microsoft Word nach HTML, LaTeX usw.
unrtf	V:0.3, I:2.7	158	rtf → html	Dokumentenkonvertierer von RTF nach HTML usw.
wp2x	V:0.00, I:0.07	200	WordPerfect → any	WordPerfect-5.0- und -5.1-Dateien nach TeX, LaTeX, troff, GML und HTML konvertieren

Tabelle 11.10: Liste von Werkzeugen zur Extrahierung von XML-Daten

11.2.4 XML Lint

HTML-Dateien (nicht-XML) können Sie nach XHTML konvertieren, was eine Instanz von korrekt formatiertem XML ist. XHTML kann von XML-Werkzeugen verarbeitet werden.

Die Syntax von XML-Dateien und die Gültigkeit von enthaltenen URLs können geprüft werden.

Paket	Popcon	Größe	Funktion	Beschreibung
libxml2-utils	V:64, I:204	214	xml ↔ html ↔ xhtml	Befehlszeilen-XML-Werkzeug mit xmllint(1) (Syntaxüberprüfung, Neuformatierung, …)
tidy	V:0.8, I:6.7	79	xml ↔ html ↔ xhtml	HTML-Syntaxüberprüfung und Neuformatierung
weblint-perl	V:0.05, I:0.85	32	lint	Ein Syntax- und (Minimal-)Stil-Prüfprogramm für HTML
linklint	V:0.05, I:0.45	343	Link-Prüfung	Schneller Linkchecker und Werkzeug für die Webseitenpflege

Tabelle 11.11: Liste von XML-Druck-Werkzeugen

Sobald eine saubere XML-Basis generiert wurde, können Sie die XSLT-Technologie nutzen, um Daten basierend auf dem Markup-Kontext zu extrahieren usw.

11.3 Textsatz

Das Unix-Programm troff, ursprünglich von AT&T entwickelt, kann für einfachen Textsatz verwendet werden. Es wird normalerweise genutzt, um Handbuchseiten (manpages) zu erzeugen.

TeX, entwickelt von Donald Knuth, ist ein sehr leistungsfähiges Textsatz-Werkzeug und der De-Facto-Standard. LaTeX (ursprünglich geschrieben von Leslie Lamport) ermöglicht einen sehr hochentwickelten Zugriff auf die Fähigkeiten von TeX.

11.3.1 roff-Textsatz

Traditionell ist roff das Haupt-Unix-System zur Textverarbeitung. Lesen Sie roff(7), groff(7), groff(1), grotty(1), troff(1), groff_mdoc(7), groff_man(7), groff_ms(7), groff_me(7), groff_mm(7) und "`info groff`".

Sie bekommen eine gute Einführung und Referenz zum "`-me`"-Makro in "`/usr/share/doc/groff/`", wenn Sie das `groff`-Paket installiert haben.

Paket	Popcon	Größe	Schlüsselwor	Beschreibung
texlive	V:1, I:26	55	(La)TeX	TeX-System für Textsatz, Vorschau und Druck
lilypond	V:0.7, I:5.1	18010	(La)TeX	Notensatzprogramm
groff	V:2, I:22	16868	troff	GNU troff Textformatierungs-System

Tabelle 11.12: Liste von Textsatz-Werkzeugen

Tipp
"`groff -Tascii -me -`" erzeugt reinen Text mit ANSI-Escape-Sequenzen. Wenn Sie eine Ausgabe ähnlich zu den Handbuchseiten mit vielen "^H" und "_" möchten, verwenden Sie stattdessen "`GROFF_NO_SGR=1 groff -Tascii -me -`".

Tipp
Um "^H" und "_" aus einer mit `groff` erzeugten Textdatei zu entfernen, filtern Sie diese mit "`col -b -x`".

11.3.2 TeX/LaTeX

Die TeX Live-Software-Distribution stellt ein vollständiges TeX-System bereit. Das `texlive`-Metapaket enthält eine sinnvolle Auswahl von TeX Live-Paketen, die für die meisten Aufgaben ausreichend sein sollten.

Es gibt viele Ressourcen für TeX und LaTeX:

- The teTeX HOWTO: The Linux-teTeX Local Guide;
- tex(1)
- latex(1)
- texdoc(1)
- texdoctk(1)
- "The TeXbook", von Donald E. Knuth (Addison-Wesley);
- "LaTeX - A Document Preparation System", von Leslie Lamport (Addison-Wesley);
- "The LaTeX Companion", von Goossens, Mittelbach, Samarin (Addison-Wesley).

TeX/LaTeX ist die leistungsfähigste Textsatz-Umgebung. Viele SGML-Prozessoren nutzen es im Hintergrund zur Textverarbeitung. Lyx aus dem `lyx`-Paket sowie GNU TeXmacs aus dem `texmacs`-Paket bieten eine nette WYSIWYG-Umgebung zum Editieren von LaTeX-Dokumenten, zu der viele Leute Emacs oder Vim als Quelltext-Editor wählen.

Es sind viele Online-Ressourcen verfügbar:

- The TEX Live Guide ("`/usr/share/doc/texlive-doc/texlive/texlive-en/texlive-en.html`" aus dem `texlive-base`-Paket);
- A Simple Guide to Latex/Lyx;
- Word Processing Using LaTeX;

Wenn die Dokumente größer werden, kann TeX eventuell Fehler verursachen. Sie müssen dann die Pool-Größe in "`/etc/texmf/texmf.cnf`" erhöhen (oder editieren Sie besser "`/etc/texmf/texmf.d/95NonPath`" und führen update-texmf(8) aus), um dieses Problem zu beheben.

Anmerkung
Der TeX-Quelltext von ”The TeXbook” ist auf der www.ctan.org Tex-Archiv-Seite für texbook.tex verfügbar. Diese Datei enthält die meisten der benötigten Makros. Ich habe gehört, dass Sie dieses Dokument mit tex(1) verarbeiten können, wenn Sie die Zeilen 7 - 10 auskommentieren und ”`\input manmac \proofmodefalse`” hinzufügen. Es wird dringend empfohlen, dass Sie dieses Buch (wie auch alle anderen Bücher von Donald E. Knuth) kaufen, statt die Online-Version zu lesen, aber der Quelltext ist ein tolles Beispiel für TeX-Eingaben!

11.3.3 Schöner Ausdruck einer Handbuchseite

Sie können eine Handbuchseite (manpage) in PostScript mit einem der folgenden Befehle schön ausdrucken:

```
$ man -Tps some_manpage | lpr
```

11.3.4 Erstellen einer Handbuchseite

Obwohl es möglich ist, eine Handbuchseite (manpage) in reinem troff-Format zu schreiben, gibt es auch einige Pakete mit Hilfsprogrammen zur Erstellung einer Handbuchseite:

Paket	Popcon	Größe	Schlüsselwort	Beschreibung
`docbook-to-man`	V:0.6, I:5.2	189	`SGML` → `manpage`	Konvertierer vom DocBook-SGML-Format in roff-man-Makros
`help2man`	V:0.5, I:6.0	542	`text` → `manpage`	Automatisch Handbuchseiten aus --help-Ausgabe generieren
`info2man`	V:0.01, I:0.18	134	`info` → `manpage`	Konvertierer von GNU-info-Dateien nach POD oder Manpage
`txt2man`	V:0.05, I:0.59	112	`text` → `manpage`	Einfachen ASCII-Text in das Manpage-Format umwandeln

Tabelle 11.13: Liste von Paketen, die bei der Erstellung einer Handbuchseite helfen

11.4 Druckfähige Daten

Druckfähige Daten werden im Debian-System im PostScript-Format dargestellt. CUPS (Common Unix Printing System) verwendet Ghostscript als Rasterizer-Backend-Programm für Drucker, die selbst kein PostScript interpretieren können.

Druckfähige Daten können im aktuellen Debian-System auch im PDF-Format angezeigt werden.

Zum Anzeigen von PDF-Dateien und Ausfüllen von Formulardaten in diesen Dateien können Werkzeuge wie Evince und Okular verwendet werden (siehe Abschnitt 7.4), sowie auch moderne Browser wie Chromium.

Mit grafischen Programmen wie LibreOffice, Scribus und Inkscape können Sie PDF-Dateien auch bearbeiten (siehe dazu Abschnitt 11.6).

Tipp
Sie können eine PDF-Datei mit GIMP einlesen und in das PNG-Format konvertieren; nutzen Sie dazu eine Auflösung größer als 300 dpi. Dies kann dann als Hintergrundbild für LibreOffice genutzt werden, um so mit minimalem Aufwand einen nach Wunsch angepassten Ausdruck zu erstellen.

11.4.1 Ghostscript

Der Kern der Verarbeitung von druckfähigen Daten ist ein Ghostscript-PostScript (PS)-Interpreter, der ein Raster-Image erzeugt.

Paket	Popcon	Größe	Beschreibung
ghostscript	V:180, I:554	177	der GPL Ghostscript-PostScript/PDF-Interpreter
ghostscript-x	V:0, I:12	88	GPL Ghostscript-PostScript/PDF-Interpreter - Unterstützung für Anzeige unter X
libpoppler162	V:16, I:24	5321	Bibliothek zur PDF-Darstellung, Abspaltung von dem PDF-Anzeigeprogramm xpdf
libpoppler-glib8t64	V:80, I:335	612	Bibliothek zur PDF-Darstellung (Laufzeitbibliothek auf Basis von GLib)
poppler-data	V:170, I:578	13086	CMaps für Bibliothek zur PDF-Darstellung (für CJK-Unterstützung: Adobe-*)

Tabelle 11.14: Liste von Ghostscript-PostScript-Interpretern

Tipp
"gs -h" zeigt die Konfiguration von Ghostscript an.

11.4.2 Zwei PS- oder PDF-Dateien zusammenführen

Sie können zwei PostScript (PS)- oder Portable Document Format (PDF)-Dateien mit dem gs(1)-Befehl von Ghostscript zusammenführen:

```
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pswrite -sOutputFile=bla.ps -f foo1.ps foo2.ps
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=bla.pdf -f foo1.pdf foo2.pdf
```

Anmerkung
PDF, ein plattform-unabhängiges Datenformat für druckfähige Daten, ist im Grunde ein komprimiertes PS-Format mit einigen zusätzlichen Funktionalitäten und Erweiterungen.

Tipp
Auf der Befehlszeile sind psmerge(1) und andere Befehle aus dem `psutils`-Paket sehr nützlich zur Bearbeitung von PostScript-Dokumenten. pdftk(1) aus dem `pdftk-java`-Paket ist ebenfalls praktisch, um PDF-Dokumente zu bearbeiten.

11.4.3 Werkzeuge für druckfähige Daten

Folgende Pakete mit Werkzeugen für druckfähige Daten sind mir ins Auge gestochen:

11.4.4 Drucken mit CUPS

Die beiden vom Common Unix Printing System (CUPS) angebotenen Befehle lp(1) und lpr(1) bieten Optionen, um das Drucken von druckfähigen Daten spezifisch anzupassen.

Mit einem der folgenden Befehle können Sie drei Kopien einer Datei auf einmal ausdrucken:

Paket	Popcon	Größe	Schlüsselwort	Beschreibung
poppler-utils	V:175, I:475	804	pdf → ps, text	PDF-Hilfsprogramme: pdftops, pdfinfo, pdfimages, pdftotext, pdffonts
psutils	V:4, I:47	34	ps → ps	Werkzeuge zur Konvertierung von PostScript-Dokumenten
poster	V:0.1, I:1.5	57	ps → ps	Erzeugen großer Poster aus PostScript-Seiten
enscript	V:1, I:12	2138	text → ps,html,rtf	Konvertieren von ASCII-Text nach PostScript, HTML, RTF oder Pretty-Print
a2ps	V:0.6, I:6.5	4109	text → ps	'Anything to PostScript'-Konvertierer (jegliches nach PostScript) und Pretty-Printer
pdftk	V:1, I:18	28	pdf → pdf	Werkzeug zur Konvertierung von PDF-Dokumenten: pdftk
html2ps	V:0.1, I:1.7	256	html → ps	Konvertierer von HTML nach PostScript
gnuhtml2latex	V:0.04, I:0.55	26	html → latex	Konvertierer von HTML nach LaTeX
latex2rtf	V:0.1, I:1.9	495	latex → rtf	Konvertieren von Dokumenten von LaTeX nach RTF, die dann von MS Word gelesen werden können
ps2eps	V:2, I:30	95	ps → eps	Konvertierer von PostScript nach EPS (Encapsulated PostScript)
e2ps	V:0.01, I:0.10	104	text → ps	Text-nach-PostScript-Konvertierer mit Unterstützung für japanische Zeichenkodierung
impose+	V:0.1, I:1.5	118	ps → ps	PostScript-Hilfsprogramme
trueprint	I:0.07	148	text → ps	Viele Quellformate (C, C++, Java, Pascal, Perl, Pike, Sh und Verilog) nach PostScript konvertieren mittels Pretty-Print (C-Sprache)
pdf2svg	V:0.2, I:2.8	33	pdf → svg	Konvertierer von PDF in das Scalable Vector Graphics-Format
pdftoipe	V:0.02, I:0.44	73	pdf → ipe	Konvertierer von PDF in IPE's XML-Format

Tabelle 11.15: Liste von Werkzeugen für druckfähige Daten

```
$ lp -n 3 -o Collate=True filename
```

```
$ lpr -#3 -o Collate=True filename
```

Druckoperationen können über Optionen für den Drucker noch weitgehender angepasst werden, z.B. mit "`-o number-up=2`" "`-o page-set=even`", "`-o page-set=odd`", "`-o scaling=200`", "`-o natural-scaling=200`" usw. Diese Optionen sind dokumentiert unter Command-Line Printing and Options (wenn Sie CUPS installiert haben).

11.5 Konvertierung von Mail-Daten

Folgende Pakete für die Konvertierung von Mail-Daten sind mir aufgefallen:

Paket	Popcon	Größe	Schlüsselwor	Beschreibung
`sharutils`	V:2, I:26	1436	mail	shar(1), unshar(1), uuencode(1), uudecode(1)
`mpack`	V:0.7, I:7.1	109	`MIME`	Kodieren und Dekodieren von MIME-Nachrichten: mpack(1) und munpack(1)
`tnef`	V:0.3, I:3.7	103	`ms-tnef`	Auspacken von MIME-Anhängen des Typs "application/ms-tnef" (ein nur von Microsoft verwendetes Format)
`uudeview`	V:0.2, I:1.6	105	`mail`	Kodieren und Dekodieren folgender Formate: uuencode, xxencode, BASE64, quoted printable und BinHex

Tabelle 11.16: Liste von Paketen zur Konvertierung von Mail-Daten

Tipp
Ein Internet Message Access Protocol-Server der Version 4 (IMAP4) kann verwendet werden, um Mails von proprietären Mail-Systemen zu exportieren; dazu muss der Mail-Client (z.B. unter Windows) so konfiguriert werden, dass er den auf dem Debian-System laufenden IMAP4-Server nutzt.

11.5.1 Grundlagen zu Mail-Daten

Mail-Daten (SMTP) sollten auf eine Abfolge von 7-Bit-Zeichen beschränkt werden. Daher werden Binärdaten und 8-Bit-Textdaten über Multipurpose Internet Mail Extensions (MIME) und die Auswahl des Zeichensatzes (lesen Sie dazu Tabelle 11.2) im 7-Bit-Format kodiert.

Das Standardformat zum Speichern von Mails ist mbox gemäß RFC2822 (aktualisiertes RFC822). Näheres dazu in mbox(5) (aus dem `mutt`-Paket).

Für europäische Sprachen wird normalerweise "`Content-Transfer-Encoding: quoted-printable`" mit dem ISO-8859-1-Zeichensatz bei Mails verwendet, da es dabei nicht viele 8-Bit-Zeichen gibt. Wenn europäischer Text in UTF-8 kodiert ist, wird dabei voraussichtlich "`Content-Transfer-Encoding: quoted-printable`" benutzt, da es überwiegend 7-Bit-Zeichen sind.

Für Japanisch wird gewöhnlich "`Content-Type: text/plain; charset=ISO-2022-JP`" bei Mails verwendet, um den Text im 7-Bit-Format zu halten. Ältere Microsoft-Systeme könnten jedoch Mail-Daten in Shift-JIS versenden, ohne dies korrekt zu deklarieren. Wenn japanischer Text in UTF-8 kodiert ist, wird dabei voraussichtlich Base64 benutzt, da dabei viele 8-Bit-Zeichen enthalten sind. Die Situation bei anderen asiatischen Sprachen ist ähnlich.

Anmerkung
Falls Sie mit einer Debian-fremden Client-Software auf Ihre nicht-Unix-Mail-Daten zugreifen können und dieser Client auch mit einem IMAP4-Server kommunizieren kann, können Sie die Mails von dem Fremdsystem herunterholen, indem Sie einen eigenen IMAP4-Server laufen lassen.

Anmerkung
Nutzen Sie andere Formate zur Speicherung Ihrer Mails, ist die Umstellung auf das mbox-Format ein guter erster Schritt. Ein vielseitiges Client-Programm wie z.B. mutt(1) kann dabei nützlich sein.

Sie können den Inhalt einer Mailbox auf einzelne Nachrichten aufsplitten, indem Sie procmail(1) und formail(1) verwenden.

Jede Mail kann mittels munpack(1) aus dem `mpack`-Paket (oder mit anderen spezialisierten Werkzeugen) entpackt werden, um die MIME-kodierten Inhalte zu erhalten.

11.6 Werkzeuge für Grafikdaten

Obwohl grafische GUI-Programme wie gimp(1) sehr leistungsfähig sind, können auch Befehlszeilenwerkzeuge wie imagemagick(1) nützlich sein, um automatisierte Bildbearbeitung über Skripte durchzuführen.

Der De-Facto-Standard für Bilddateien von Digitalkameras ist das Exchangeable Image File Format (EXIF); dies entspricht dem JPEG-Dateiformat mit zusätzlichen Metainformationen. Es kann auch Informationen wie Datum, Zeit und Kameraeinstellungen speichern.

Das Patent zur verlustlosen Datenkompression mit dem Lempel-Ziv-Welch-(LZW-)Algorithmus ist abgelaufen. Graphics Interchange Format (GIF)-Werkzeuge, die die LZW-Kompressionsmethode nutzen, sind jetzt frei im Debian-System verfügbar.

Tipp
Alle Digitalkameras oder Scanner mit Wechseldatenträgern als Speichermedium funktionieren unter Linux über USB-Speicher-Lesegeräte, da sie die Design-Regeln für Kamera-Dateisysteme befolgen und FAT als Dateisystem verwenden. Näheres finden Sie in Abschnitt 10.1.7.

11.6.1 Werkzeuge für Grafikdaten (Metapaket)

Die folgenden Metapakete sind ein guter Startpunkt, wenn Sie mit aptitude(8) nach Grafikprogrammen suchen. "Packages overview for Debian PhotoTools Maintainers" kann ein anderer Ansatz sein.

Paket	Popcon	Größe	Schlüsselwort	Beschreibung
`education-graphics`	I:0.30	26	`svg,jpeg,…`	Metapaket zum Lehren von Grafik- und Fotokunst
`open-font-design-toolkit`	I:0.04	9	`ttf,ps,…`	Metapaket für Open-font-Design

Tabelle 11.17: Liste von Werkzeugen für Grafikdaten (Metapaket)

Tipp
Weitere Werkzeuge zur Bildbearbeitung finden Sie mit dem regulären Ausdruck "`~Gworks-with::image`" in aptitude(8) (lesen Sie dazu Abschnitt 2.2.6).

11.6.2 Werkzeuge für Grafikdaten (GUI)

Folgende Pakete zur Konvertierung, Bearbeitung und Organisation von grafischen Daten via GUI sind mir aufgefallen:

Paket	Popcon	Größe	Schlüsselwort	Beschreibung
gimp	V:40, I:206	33386	image(bitmap)	GNU Image Manipulation Program (GIMP)
xsane	V:9, I:127	1512	image(bitmap)	GTK-basierte X11-Oberfläche für das Scanner-Programm SANE (Scanner Access Now Easy)
scribus	V:1, I:12	32942	ps/pdf/SVG/…	Scribus DTP-Editor
libreoffice-draw	V:78, I:412	11424	image(vector)	LibreOffice Office-Programm - Zeichnen
inkscape	V:12, I:73	113357	image(vector)	SVG (Scalable Vector Graphics)-Editor
dia	V:1, I:16	3846	image(vector)	Diagramm-Editor (Gtk)
xfig	V:0.5, I:8.4	7951	image(vector)	Programm zur interaktiven Erzeugung von Objekten in X11
gocr	V:0.4, I:3.4	549	image → text	freie OCR-Software
eog	V:20, I:121	10542	image(Exif)	Bildbetrachter Eye of GNOME
gthumb	V:3, I:11	5131	image(Exif)	Bildbetrachter und -browser (GNOME)
geeqie	V:3, I:10	3854	image(Exif)	Bildbetrachter, der GTK verwendet
shotwell	V:15, I:244	6334	image(Exif)	Digitalfoto-Organizer (GNOME)
gwenview	V:40, I:113	6051	image(Exif)	Bildbetrachter (KDE)
kamera	I:112	992	image(Exif)	Unterstützung von Digitalkameras für KDE-Anwendungen
digikam	V:1.5, I:8.2	333	image(Exif)	Digitalfoto-Verwaltung für KDE
darktable	V:3, I:10	36092	image(Exif)	virtueller Leuchttisch und Dunkelkammer für Photographen
hugin	V:0.5, I:5.2	6885	image(Exif)	Panorama-Foto-Ersteller
librecad	V:1, I:13	12361	DXF,…	2D-Editor für computerunterstützte Konstruktion (CAD)
freecad	V:1, I:20	112	DXF,…	3D-Editor für computerunterstützte Konstruktion (CAD)
blender	V:3, I:20	110904	blend,TIFF,VRML,…	Editor für 3D-Inhalte (Animationen usw.)
mm3d	V:0.03, I:0.23	4123	ms3d,obj,dxf,…	3D-Modell-Editor auf Basis von OpenGL
fontforge	V:0.5, I:5.2	4736	ttf,ps,…	Editor für PS-, TrueType- und OpenType-Schriften
xgridfit	V:0.02, I:0.07	878	ttf	Programm zum Gridfitting und Hinting von TrueType-Schriften

Tabelle 11.18: Liste von Werkzeugen für Grafikdaten (GUI)

11.6.3 Werkzeuge für Grafikdaten (Konsolen-Befehle)

Folgende Pakete zur Konvertierung, Bearbeitung und Organisation von grafischen Daten über Konsolen-Befehle sind mir aufgefallen:

11.7 Verschiedene Datenkonvertierungen

Es gibt viele andere Programme zum Konvertieren von Daten. Folgende Pakete habe ich über den regulären Ausdruck "`~Guse::converting`" in aptitude(8) gefunden (Näheres dazu in Abschnitt 2.2.6):

Sie können die Daten aus einem RPM-Archiv auch mit folgendem Befehl extrahieren:

Paket	Popcon	Größe	Schlüsselwort	Beschreibung
imagemagick	V:7, I:266	82	image(bitmap)	Programme zur Bildbearbeitung
graphicsmagick	V:1.0, I:7.8	5988	image(bitmap)	Programme zur Bildbearbeitung (Abspaltung von imagemagick)
netpbm	V:27, I:275	8470	image(bitmap)	Werkzeuge zur Grafikkonvertierung
libheif-examples	V:0.4, I:3.7	534	heif→jpeg(bitmap)	Konvertieren des High Efficiency Image File Formats (HEIF) in das JPEG-, PNG- oder Y4M-Format mit dem Befehl heif-convert(1)
icoutils	V:3, I:31	221	png↔ico(bitmap)	Konvertieren von MS-Windows-Icons und -Cursor in das und vom PNG-Format (Favicons)
pstoedit	V:2, I:36	1075	ps/pdf→image(vector)	PostScript- und PDF-Dateien in editierbare Vector-Grafiken (SVG) konvertieren
libwmf-bin	V:5, I:77	149	Windows/image(vector)	Konvertierungswerkzeuge für Windows-Metadaten (Vector-Grafiken)
fig2sxd	V:0.02, I:0.14	158	fig→sxd(vector)	XFig-Dateien in das OpenOffice.org-Draw-Format konvertieren
unpaper	V:2, I:16	417	image→image	Werkzeug zum Nachbearbeiten von eingescannten Seiten für OCR
tesseract-ocr	V:9, I:35	2411	image→text	freie OCR-Software basierend auf HPs kommerzieller OCR-Engine
tesseract-ocr-eng	V:9, I:35	4032	image→text	OCR-Engine-Daten: tesseract-ocr-Sprachdateien für englischen Text
ocrad	V:0.3, I:2.2	608	image→text	freie OCR-Software
exif	V:2, I:50	334	image(Exif)	Befehlszeilen-Werkzeug, um EXIF-Informationen von JPEG-Dateien anzuzeigen
exiv2	V:1, I:18	436	image(Exif)	Werkzeug zur Bearbeitung von Exif-/IPTC-Metadaten
exiftran	V:1, I:11	81	image(Exif)	Programm für die Umwandlung der JPEG-Bilder von Digitalkameras
exiftags	V:0.2, I:2.5	309	image(Exif)	Werkzeug, um Exif-Informationen aus Digitalkamera-JPEG-Dateien auszulesen
exifprobe	V:0.2, I:2.0	506	image(Exif)	Metadaten aus Digitalbildern auslesen
dcraw	V:0.7, I:7.0	428	image(Raw)→ppm	Dekodierer für Digitalkamerabilder im RAW-Format
findimagedupes	V:0.10, I:0.94	76	image→fingerprint	Visuell ähnliche oder doppelte Bilder finden
ale	V:0.01, I:0.14	850	image→image	Bilder zusammenfügen, um die Wiedergabetreue zu erhöhen oder Mosaike zu erzeugen
imageindex	V:0.1, I:1.1	143	image(Exif)→html	Erzeugen von statischen HTML-Galerien aus Bildern
outguess	V:0.1, I:1.0	230	jpeg,png	universelles Steganographie-Werkzeug
jpegoptim	V:0.5, I:5.6	59	jpeg	Optimieren von JPEG-Dateien
optipng	V:2, I:38	187	png	Optimieren von PNG-Dateien, verlustfreie Kompression
pngquant	V:1, I:11	62	png	Optimieren von PNG-Dateien, verlustbehaftete Kompression

Tabelle 11.19: Liste von Werkzeugen für Grafikdaten (Konsolen-Befehle)

Paket	Popcon	Größe	Schlüsselwort	Beschreibung
alien	V:1, I:12	150	rpm/tgz → deb	Programm zur Konvertierung von fremden Software-Paketen in das Debian-Paketformat
freepwing	I:0.01	447	EB → EPWING	Konvertierer von ”Electric Book” (beliebt in Japan) in ein separates JIS X 4081-Format (eine Untermenge von EPWING V1)
calibre	V:7, I:24	76289	any → EPUB	E-Book-Konvertierer und Bibliotheksverwaltung

Tabelle 11.20: Liste verschiedener Werkzeuge zur Datenkonvertierung

```
$ rpm2cpio file.src.rpm | cpio --extract
```

Kapitel 12

Programmierung

Ich gebe hier einige Startimpulse, um Programmierung im Debian-System zumindest so weit zu erlernen, dass der Programmcode aus Debian-Quellpaketen zwecks Fehlersuche nachverfolgt werden kann. Hier einige erwähnenswerte Pakete und dazugehörige Dokumentation:

Handbuchseiten (manpages) sind verfügbar, indem Sie nach Installation der `manpages`- und `manpages-dev`-Pakete ”`man programmname`” eingeben. Informationen über GNU-Werkzeuge erhalten Sie über ”`info programmname`” nach Installation der entsprechenden Dokumentationspakete. Sie müssen unter Umständen die Bereiche `contrib` und `non-free` des Archivs zusätzlich zu `main` freigeben, da einige GFDL-Dokumentation als nicht DFSG-frei angesehen wird.

Ziehen Sie bitte die Nutzung eines Versionskontrollsystems in Erwägung, siehe Abschnitt 10.5.



Warnung
Verwenden Sie nicht ”`test`” als Namen für eine ausführbare Testdatei. ”`test`” ist ein fest integrierter Shell-Builtin.



Achtung
Sie sollten Software, die direkt aus den Quellen kompiliert wurden, in ”`/usr/local`” oder ”`/opt`” installieren, um Kollisionen mit Systemprogrammen zu vermeiden.

12.1 Das Shell-Skript

Das Shell-Skript ist eine Textdatei mit gesetztem Ausführungsbit und enthält Befehle in der folgenden Form:

```
#!/bin/sh
 ... command lines
```

Die erste Zeile spezifiziert den Shell-Interpreter, der den Inhalt dieser Datei liest und ausführt.

Das Lesen von Shell-Skripten ist der **beste** Weg, um zu verstehen, wie ein Unix-artiges System arbeitet. Ich gebe hier einige Hinweise zur Shell-Programmierung. Lesen Sie ”Shell Mistakes” (https://www.greenend.org.uk/rjk/2001/-04/shell.html), um aus Fehlern zu lernen.

Anders als der interaktive Shell-Modus (lesen Sie Abschnitt 1.5 und Abschnitt 1.6) verwenden Shell-Skripte häufig Parameter, bedingte Ausdrücke und Schleifen.

12.1.1 POSIX-Shell-Kompatibilität

Viele Systemskripte können von jeder POSIX-konformen Shell (lesen Sie Tabelle 1.13) interpretiert werden.

- Die nicht-interaktive Standard-POSIX-Shell "`/usr/bin/sh`" ist ein symbolischer Link auf `/usr/bin/dash` und wird von vielen Systemprogrammen genutzt.
- Die interaktive Standard-POSIX-Shell ist `/usr/bin/bash`.

Vermeiden Sie, in Shell-Skripten **Bashisms** oder **Zshisms** (speziell für **Bash** oder **Zsh** angepasste Shell-Befehle) zu verwenden, damit sie auf alle POSIX-Shells portierbar sind. Sie können dies mittels checkbashisms(1) überprüfen.

Gut: POSIX	**Vermeiden: Bashism**
`if [ "$foo" = "$bar" ] ; then …`	`if [ "$foo" == "$bar" ] ; then …`
`diff -u datei.c.orig datei.c`	`diff -u datei.c{.orig,}`
`mkdir /foobar /foobaz`	`mkdir /foo{bar,baz}`
`funcname() { … }`	`function funcname() { … }`
Oktalformat: "`\377`"	Hexadezimalformat: "`\xff`"

Tabelle 12.1: Liste typischer Bashisms

Der Befehl "`echo`" muss mit Vorsicht verwendet werden, da seine Implementierung sich zwischen Shell-Builtins und externen Befehlen unterscheidet:

- vermeiden Sie die Verwendung jeglicher Befehlsoptionen außer "`-n`";
- vermeiden Sie die Verwendung von Escape-Sequenzen in Zeichenketten, da deren Handhabung unterschiedlich ist.

Anmerkung
Obwohl die Option "`-n`" **nicht** wirklich der POSIX-Syntax entspricht, ist sie grundsätzlich akzeptiert.

Tipp
Nutzen Sie den "`printf`"-Befehl statt "`echo`", wenn Sie Escape-Sequenzen in auszugebenen Zeichenketten einbetten möchten.

12.1.2 Shell-Parameter

Einige spezielle Shell-Parameter werden oft in Shell-Skripten verwendet:

Grundlegende **Parameterauswertungen**, die Sie sich einprägen sollten:

Hierbei ist der Doppelpunkt "`:`" in allen Operatoren optional:

- **mit** "`:`" = Operatortest für **existiert** und **nicht Null**;
- **ohne** "`:`" = Operatortest nur für **existiert**.

Shell-Parameter	Wert
`$0`	Name der Shell oder des Shell-Skripts
`$1`	erstes (1.) Shell-Argument
`$9`	neuntes (9.) Shell-Argument
`$#`	Anzahl der Parameter
`"$*"`	`"$1 $2 $3 $4 … "`
`"$@"`	`"$1" "$2" "$3" "$4" …`
`$?`	Beendigungs-Status des zuletzt ausgeführten Befehls
`$$`	Prozessnummer (PID) des Shell-Skripts
`$!`	PID des zuletzt ausgeführten Hintergrundbefehls

Tabelle 12.2: Liste von Shell-Parametern

Parameterauswertung	Wert, falls `var` gesetzt ist	Wert, falls `var` nicht gesetzt ist
`${var:-string}`	”`$var`”	”`string`”
`${var:+string}`	”`string`”	”`null`”
`${var:=string}`	”`$var`”	”`string`” (und ausführen von ”`var=string`”)
`${var:?string}`	”`$var`”	Ausgabe von ”`string`” auf **stderr** (und mit Fehlerstatus beenden)

Tabelle 12.3: Liste von Parameterauswertungen

Parameterersetzung	Ergebnis
`${var%suffix}`	entferne kleinstes Suffix-Muster
`${var%%suffix}`	entferne größtes Suffix-Muster
`${var#prefix}`	entferne kleinstes Prefix-Muster
`${var##prefix}`	entferne größtes Prefix-Muster

Tabelle 12.4: Liste von Shell-Parameterersetzungen

12.1.3 Bedingte Ausdrücke in der Shell

Jeder Befehl gibt einen **Beendigungs-Status** (Exit-Status) zurück, der für einen bedingten Ausdruck verwendet werden kann:

- Erfolg: 0 (”Wahr/True”)
- Fehler: nicht 0 (”Falsch/False”)

Anmerkung
”0” im Kontext eines bedingten Ausdrucks für die Shell bedeutet ”Wahr”, während ”0” im Kontext eines bedingten Ausdrucks für ein C-Programm ”Falsch” bedeutet.

Anmerkung
”`[`” ist das Äquivalent des `test`-Befehls; Argumente bis zum ”`]`” werden als bedingter Ausdruck gewertet.

Grundlegende **Ausdrucksformen für bedingte Ausdrücke**, die Sie sich einprägen sollten:

- ”*`befehl && bei_erfolg_auch_diesen_befehl_ausführen`* `|| true`”
- ”*`befehl || falls_kein_erfolg_auch_diesen_befehl_ausführen`* `|| true`”
- ein mehrzeiliger Skriptschnipsel wie dieser:

```
if [ conditional_expression ]; then
 if_success_run_this_command
else
 if_not_success_run_this_command
fi
```

Hierbei ist das ”`|| true`” am Ende erforderlich, um sicherzustellen, dass das Shell-Skript sich bei dieser Zeile nicht fälschlicherweise beendet, wenn die Shell mit der ”`-e`”-Option aufgerufen wird.

Gleichung	Wahr wenn …
`-e` *datei*	*datei* existiert
`-d` *datei*	*datei* existiert und ein Verzeichnis ist
`-f` *datei*	*datei* existiert und eine reguläre Datei ist
`-w` *datei*	*datei* existiert und schreibbar ist
`-x` *datei*	*datei* existiert und ausführbar ist
datei1 `-nt` *datei2*	*datei1* neuer als *datei2* ist (Änderungszeitpunkt)
datei1 `-ot` *datei2*	*datei1* älter als *datei2* ist (Änderungszeitpunkt)
datei1 `-ef` *datei2*	*datei1* und *datei2* die gleiche Device- und Inode-Nummer haben

Tabelle 12.5: Liste von Dateivergleichsoperatoren in bedingten Ausdrücken

Arithmetische Ganzzahlvergleicher in bedingten Ausdrücken sind ”`-eq`”, ”`-ne`”, ”`-lt`”, ”`-le`”, ”`-gt`” und ”`-ge`”.

12.1.4 Shell-Schleifen

Es gibt mehrere Ausdrucksweisen für Schleifen, die in POSIX-Shells genutzt werden können:

- ”`for x in foo1 foo2 … ; do befehl ; done`” führt Schleifen aus, indem Einträge aus der Liste ”`foo1 foo2 …`” der Variable ”x” zugewiesen werden und dann ”`befehl`” ausgeführt wird.

Gleichung	Wahr wenn …
`-z str`	die Länge von *str* gleich Null ist
`-n str`	die Länge von *str* nicht Null ist
`str1 = str2`	*str1* und *str2* gleich sind
`str1 != str2`	*str1* und *str2* ungleich sind
`str1 < str2`	*str1* in der Sortierung vor *str2* erscheint (Locale-abhängig)
`str1 > str2`	*str1* in der Sortierung hinter *str2* erscheint (Locale-abhängig)

Tabelle 12.6: Liste von String-Vergleichsoperatoren im bedingten Ausdruck

- "`while bedingung ; do befehl ; done`" wiederholt "`befehl`", solange "`bedingung`" wahr ist.
- "`until bedingung ; do befehl ; done`" wiederholt "`befehl`", solange "`bedingung`" nicht wahr ist.
- "`break`" ermöglicht, die Bearbeitung der Schleife zu beenden.
- "`continue`" ermöglicht, den nächsten Umlauf der Schleife fortzusetzen.

Tipp
Die der C-Sprache ähnliche nummerische Wiederholung (Iteration) kann realisiert werden, indem seq(1) für die Erzeugung der "`foo1 foo2 …`"-Liste genutzt wird.

Tipp
Lesen Sie auch Abschnitt 9.4.9.

12.1.5 Shell-Umgebungsvariablen

Einige populäre Umgebungsvariablen für den normalen Shell-Befehlsprompt könnten in der Ausführungsumgebung Ihres Skriptes nicht verfügbar ein.

- Statt "`$USER`" nutzen Sie besser "`$(id -un)`"
- Statt "`$UID`" nutzen Sie besser "`$(id -u)`"
- Statt "`$HOME`" nutzen Sie besser "`$(getent passwd "$(id -u)"|cut -d ":" -f 6)`" (dies funktioniert auch für Abschnitt 4.5.2).

12.1.6 Befehlsabfolge auf der Shell

Die Shell verarbeitet ein Skript im Prinzip in der folgenden Abfolge:

- Die Shell liest eine Zeile.
- Die Shell gruppiert Teile der Zeile zu **zusammengehörigen Ausdrücken (Token)** zusammen, wenn diese sich innerhalb von "…" oder '…' befinden.
- Die Shell splittet andere Teile der Zeile in **einzelne Ausdrücke (Token)** auf, wenn diese wie folgt von einander getrennt sind:
  - Whitespace-Zeichen: *Leerzeichen Tabulator newline*
  - Metazeichen: `< > | ; & ( )`

- Die Shell prüft jeden Ausdruck auf **Schlüsselworte** (wenn nicht innerhalb von "…" oder '…'), um deren Verhalten anzupassen.
  - **Schlüsselwörter** sind: `if then elif else fi for in while unless do done case esac`
- Die Shell expandiert **Alias**-Befehle (wenn nicht innerhalb von "…" oder '…').
- Die Shell expandiert eine **Tilde** (wenn nicht innerhalb von "…" oder '…'):
  - "~" → Heimatverzeichnis des aktuellen Benutzers
  - "~*benutzer*" → Heimatverzeichnis von *benutzer*
- Die Shell expandiert **Parameter** in deren Wert (wenn nicht innerhalb von '…'):
  - **Parameter**: "$PARAMETER" oder "${PARAMETER}"
- Die Shell expandiert **Befehlsersetzungen / command substitutions** (wenn nicht innerhalb von '…'):
  - "`$( befehl )`" → die Ausgabe von "`befehl`"
  - "`` ` befehl ` ``" → die Ausgabe von "`befehl`"
- Die Shell expandiert **Pfadnamenmuster** in die passenden Dateinamen (wenn nicht innerhalb von "…" oder '…'):
  - * → jegliche Zeichen (eins oder mehrere)
  - ? → irgendein (nur ein) Zeichen
  - […] → jegliche Zeichen von denen in "…"
- Die Shell sucht **befehl** in folgenden Definitionen und führt ihn aus:
  - **Funktions**-Definition
  - **Builtin** (integrierter Befehl)
  - **ausführbare Datei** in "$PATH"
- Die Shell geht zur nächsten Zeile und wiederholt diesen kompletten Ablauf vom Anfang.

Einfache Anführungszeichen innerhalb von doppelten Anführungszeichen haben keine Wirkung.

Das Ausführen von "`set -x`" in der Shell oder das Aufrufen einer Shell mit der Option "`-x`" veranlasst die Shell, alle ausgeführten Befehle auch auf dem Bildschirm auszugeben. Dies ist sehr nützlich zur Fehlersuche.

12.1.7 Hilfsprogramme für Shell-Skripte

Um Ihr Shell-Programm innerhalb des Debian-Systems möglichst weit portierbar zu machen, ist es eine gute Idee, die zu nutzenden Hilfsprogramme auf diejenigen einzuschränken, welche durch die **essential**-Pakete bereitgestellt werden:

- "`aptitude search ~E`" listet alle **essential**-Pakete auf;
- "`dpkg -L paketname |grep '/man/man.*/'`" listet Handbuchseiten (manpages) derjenigen Befehle auf, die von dem Paket *paketname* bereitgestellt werden.

Tipp
Obwohl `moreutils` außerhalb von Debian unter Umständen nicht verfügbar ist, bietet es interessante kleine Programme. Das erwähnenswerteste ist sponge(8); es ist sehr nützlich, wenn Sie eine Originaldatei überschreiben möchten.

Beispiele finden Sie in Abschnitt 1.6.

Paket	Popcon	Größe	Beschreibung
dash	V:933, I:999	211	kleine und schnelle, fast POSIX-kompatible Shell für sh
coreutils	V:914, I:1000	19751	Grundlegende GNU-Werkzeuge
grep	V:800, I:1000	1297	GNU grep, egrep und fgrep
sed	V:789, I:1000	987	GNU sed
mawk	V:517, I:998	299	kleine und schnelle Alternative zu awk
debianutils	V:940, I:997	228	verschiedene Hilfsprogramme speziell für Debian
bsdutils	V:445, I:1000	338	grundlegende Kommandos aus 4.4BSD-Lite
bsdextrautils	V:771, I:877	382	weitere Kommandos aus 4.4BSD-Lite
moreutils	V:18, I:38	232	zusätzliche Unix-Hilfsprogramme

Tabelle 12.7: Liste der Pakete, die kleine Hilfsprogramme für Shell-Skripte enthalten

Paket	Popcon	Größe	Dokumentation
dash	V:933, I:999	211	sh: kleine und schnelle POSIX-kompatible Shell für sh
bash	V:900, I:1000	7309	sh: "info bash" aus dem bash-doc-Paket
mawk	V:517, I:998	299	AWK: kleine und schnelle Alternative zu awk
gawk	V:246, I:288	3289	AWK: "info gawk" aus dem gawk-doc-Paket
perl	V:672, I:991	840	Perl: perl(1) bzw. die HTML-Seiten aus dem perl-doc- oder perl-doc-html-Paket
libterm-readline-gnu-perl	V:2, I:25	435	Perl-Erweiterung für die GNU ReadLine/History-Bibliothek: perlsh(1)
libreply-perl	I:0.08	171	REPL für Perl: reply(1)
libdevel-repl-perl	V:0.02, I:0.47	237	REPL für Perl: re.pl(1)
python3	V:748, I:977	81	Python: python3(1) bzw. die HTML-Seiten aus dem python3-doc-Paket
tcl	V:24, I:171	20	Tcl: tcl(3) und detaillierte Handbuchseiten aus dem tcl-doc-Paket
tk	V:18, I:165	20	Tk: tk(3) und detaillierte Handbuchseiten aus dem tk-doc-Paket
ruby	V:61, I:150	32	Ruby: ruby(1), erb(1), irb(1), rdoc(1), ri(1)

Tabelle 12.8: Liste von Interpreter-betreffenden Paketen

12.2 Skriptverarbeitung in Interpreter-Sprachen

Wenn Sie auf einem Debian-System eine Aufgabe automatisieren möchten, sollten Sie es zunächst mit einer Interpreter-Sprache versuchen. Eine Richtschnur für die Auswahl der passenden Sprache ist:

- Verwenden Sie `dash`, wenn die Aufgabe einfach ist und Konsolen-Befehle mit einem Shell-Programm kombiniert werden.
- Verwenden Sie `python3`, wenn es keine einfache Aufgabe ist und Sie alles von Grund auf neu schreiben.
- Verwenden Sie `perl`, `tcl`, `ruby`, ... falls bereits vorhandener Code in einer dieser Sprachen existiert, der übernommen werden muss, um die Aufgabe erledigen zu können.

Falls der resultierende Code zu langsam ist, können Sie den kritischen Teil zwecks Ausführung in einer kompilierten Sprache neu schreiben und diesen von der Interpreter-Sprache aus aufrufen.

12.2.1 Fehlersuche im Code für Interpreter-Sprachen

Die meisten Interpreter bieten eine grundlegende Syntaxprüfung und Funktionalitäten zur Code-Nachverfolgung.

- "**dash -n** *script.sh*" - Syntaxprüfung eines Shell-Skripts
- "**dash -x** *script.sh*" - Nachverfolgung (Trace) eines Shell-Skripts
- "**python3 -m py_compile** *script.py*" - Syntaxprüfung eines Python-Skripts
- "**python3 -mtrace --trace** *script.py*" - Nachverfolgung (Trace) eines Python-Skripts
- "**perl -I ../libpath -c** *script.pl*" - Syntaxprüfung eines Perl-Skripts
- "**perl -d:Trace** *script.pl*" - Nachverfolgung (Trace) eines Perl-Skripts

Um Code für `dash` zu testen, versuchen Sie Abschnitt 9.1.4, um eine `bash`-ähnliche interaktive Umgebung zu erhalten.

Um Code für `perl` zu testen, versuchen Sie eine REPL-Umgebung für Perl; sie bietet Ihnen eine Python-ähnliche REPL (=READ + EVAL + PRINT + LOOP)-Umgebung für Perl.

12.2.2 Grafisches GUI-Programm und Shell-Skripte

Ein Shell-Skript kann aufgewertet werden, um ein attraktives GUI-Programm zu erzeugen. Der Trick ist, eines der sogenannten Dialog-Programme zu verwenden, statt der stumpfen Interaktion über `echo`- und `read`-Befehle.

Hier ein Beispiel für ein GUI-Programm, um zu demonstrieren, wie einfach es sein kann mit nur einem Shell-Skript.

Dieses Skript nutzt `zenity`, um eine Datei auszuwählen (Standardvorwahl ist `/etc/motd`) und sie anzuzeigen.

Ein grafischer Starter für dieses Skript kann erstellt werden gemäß Abschnitt 9.4.10.

```
#!/bin/sh -e
# Copyright (C) 2021 Osamu Aoki <osamu@debian.org>, Public Domain
# vim:set sw=2 sts=2 et:
DATA_FILE=$(zenity --file-selection --filename="/etc/motd" --title="Select a file to check ↩
    ") || \
  ( echo "E: File selection error" >&2 ; exit 1 )
# Check size of archive
if ( file -ib "$DATA_FILE" | grep -qe '^text/' ) ; then
  zenity --info --title="Check file: $DATA_FILE" --width 640  --height 400 \
    --text="$(head -n 20 "$DATA_FILE")"
```

Paket	Popcon	Größe	Beschreibung
x11-utils	V:233, I:553	651	xmessage(1): eine Nachricht oder Abfrage in einem Fenster anzeigen (X)
whiptail	V:344, I:997	61	nutzerfreundliche Dialogboxen von Shell-Skripten anzeigen (newt)
dialog	V:8, I:75	535	nutzerfreundliche Dialogboxen von Shell-Skripten anzeigen (ncurses)
zenity	V:69, I:337	193	grafische Dialogboxen von Shell-Skripten anzeigen (GTK)
ssft	V:0.01, I:0.14	80	Werkzeug für Shell-Skript-Oberflächen (Aufrufprogramm für zenity, kdialog und dialog mit gettext)
gettext	V:49, I:206	20549	"/usr/bin/gettext.sh": Meldungen übersetzen

Tabelle 12.9: Liste von Dialog-Programmen

```
else
  zenity --info --title="Check file: $DATA_FILE" --width 640  --height 400 \
    --text="The data is MIME=$(file -ib "$DATA_FILE")"
fi
```

Dieser Ansatz eines grafischen Programms mit einem Shell-Skript ist nur für einfache Auswahlen sinnvoll. Wenn Sie komplexere Programme schreiben, sollten Sie die Nutzung von geeigneten Plattformen in Erwägung ziehen.

12.2.3 Eigene Aktionen im Dateimanager

Grafische Dateimanager können erweitert werden, um einige populäre Aktionen für ausgewählte Dateien ausführen zu können; dazu müssen eventuell zusätzliche Erweiterunspakete installiert werden. Auch können spezielle eigene Aktionen ausführt werden, indem Sie selbst erstellte Skripte hinzufügen.

- Für GNOME lesen Sie dazu NautilusScriptsHowto.
- Für KDE finden Sie weitere Details in Creating Dolphin Service Menus.
- Für Xfce sollten Sie Thunar - Custom Actions und https://help.ubuntu.com/community/ThunarCustomActions besuchen.
- Für LXDE schauen Sie unter Custom Actions.

12.2.4 Verrücktes bei kurzen Perl-Skripten

Um Prozessdaten zu verarbeiten, muss `sh` Unterprozesse generieren, um `cut`, `grep`, `sed` usw. aufrufen zu können; und es ist langsam. Auf der anderen Seite enthält `perl` interne Funktionalitäten zur Verarbeitung von Prozessdaten, und es ist schnell. Daher nutzen viele Wartungsskripte in Debian `perl`.

Denken wir an folgendes einzeilige AWK-Skript und sein Äquivalent in Perl:

```
awk '($2=="1957") { print $3 }' |
```

Es ist äquivalent zu jeder der folgenden Zeilen:

```
perl -ne '@f=split; if ($f[1] eq "1957") { print "$f[2]\n"}' |
```

```
perl -ne 'if ((@f=split)[1] eq "1957") { print "$f[2]\n"}' |
```

```
perl -ne '@f=split; print $f[2] if ( $f[1]==1957 )' |
```

```
perl -lane 'print $F[2] if $F[1] eq "1957"' |
```

```
perl -lane 'print$F[2]if$F[1]eq+1957' |
```

Das letzte ist eine Knobelaufgabe. Es nutzt die Vorteile folgender Perl-Funktionen:

- Der Whitespace ist optional.
- Es existiert eine automatische Konvertierung von Zahlen zu Zeichenketten.
- Perl's Befehlsausführungstricks über Befehlszeilen-Optionen: perlrun(1)
- Perl's spezielle Variablen: perlvar(1)

Diese Flexibilität ist die Stärke von Perl. Gleichzeitig erlaubt sie aber auch die Erstellung von kryptischem und verwinkeltem Code. Seien Sie vorsichtig.

12.3 Programmieren in kompilierten Sprachen

Paket	Popcon	Größe	Beschreibung
gcc	V:167, I:557	36	GNU-C-Compiler
libc6-dev	V:259, I:577	12863	GNU-C-Bibliothek: Entwicklungsbibliotheken und Header-Dateien
g++	V:64, I:527	13	GNU-C++-Compiler
libstdc++-16-dev	V:4, I:21	24416	GNU Standard-C++-Bibliothek Version 3 (Entwicklungsdateien)
cpp	V:346, I:717	18	GNU-C-Preprozessor
gettext	V:49, I:206	20549	GNU-Werkzeuge für die Internationalisierung
glade	V:0.2, I:2.5	1613	Builder für die GTK-Bedienoberfläche
valac	V:0.2, I:2.9	533	C#-ähnliche Sprache für das GObject-System
flex	V:7, I:66	1247	LEX-kompatibler schneller lexikalischer Analysegenerator
bison	V:7, I:70	3122	YACC-kompatibler Parser-Generator
susv2	I:0.02	16	die "Single UNIX Specifications v2"
susv3	I:0.04	16	die "Single UNIX Specifications v3"
susv4	I:0.03	16	die "Single UNIX Specifications v4"
golang	I:21	11	Compiler für die Programmiersprache Go
rustc	V:6, I:20	12745	Rust System-Programmiersprache
gfortran	V:4, I:47	15	GNU-Fortran-95-Compiler
fpc	I:2.3	101	Free Pascal

Tabelle 12.10: Liste von Compiler-betreffenden Paketen

Hierbei sind flex (siehe Abschnitt 12.3.3) und bison (siehe Abschnitt 12.3.4) mit aufgeführt, um zu zeigen, wie Compiler-ähnliche Programme in der Sprache C erstellt werden können, indem höherwertige Beschreibungssprachen in C kompiliert werden.

12.3.1 C

Sie können wie folgt eine korrekte Umgebung zum Kompilieren von in der C-Programmiersprache geschriebenen Programmen einrichten:

```
# apt-get install glibc-doc manpages-dev libc6-dev gcc build-essential
```

Das Paket `libc6-dev` (d.h. die GNU-C-Bibliothek) bietet als C-Standard-Bibliothek eine Sammlung von Header-Dateien und Bibliotheksroutinen, die von der C-Sprache genutzt werden.

Referenzen für C finden Sie über:

- "`info libc`" (Referenz für Funktionen der C-Bibliothek)
- gcc(1) und "`info gcc`"
- each_C_library_function_name(3)
- Kernighan & Ritchie, "The C Programming Language", 2. Ausgabe (Prentice Hall)

12.3.2 Ein einfaches C-Programm (gcc)

Hier ein einfaches Beispiel zum Kompilieren von "`example.c`" mit der Bibliothek "`libm`" in eine ausführbare Datei "`run_example`":

```
$ cat > example.c << EOF
#include <stdio.h>
#include <math.h>
#include <string.h>

int main(int argc, char **argv, char **envp){
        double x;
        char y[11];
        x=sqrt(argc+7.5);
        strncpy(y, argv[0], 10); /* prevent buffer overflow */
        y[10] = '\0'; /* fill to make sure string ends with '\0' */
        printf("%5i, %5.3f, %10s, %10s\n", argc, x, y, argv[1]);
        return 0;
}
EOF
$ gcc -Wall -g -o run_example example.c -lm
$ ./run_example
        1, 2.915, ./run_exam,     (null)
$ ./run_example 1234567890qwerty
        2, 3.082, ./run_exam, 1234567890qwerty
```

Hierbei wird "`-lm`" benötigt, um die Bibliothek "`/usr/lib/libm.so`" aus dem `libc6`-Paket für sqrt(3) zu verlinken. Die eigentliche Bibliothek liegt in "`/lib/`" und hat den Dateinamen "`libm.so.6`", was ein symbolischer Link auf "`libm-2.7.so`" ist.

Schauen Sie sich den letzten Parameter im ausgegebenen Text an. Dort werden mehr als 10 Zeichen ausgegeben, obwohl "`%10s`" angegeben wurde.

Die Verwendung von Funktionen, die Zeiger auf Speicherbereiche ohne Bereichsüberprüfung nutzen (wie sprintf(3) und strcpy(3)), wird missbilligt, um das Ausnutzen von Pufferüberläufen zu verhindern, die obige Überlaufeffekte in Gang bringen. Verwenden Sie stattdessen snprintf(3) und strncpy(3).

12.3.3 Flex - ein besseres Lex

Flex ist ein Lex-kompatibler schneller lexikalischer Analysegenerator.

Eine Einführung zu flex(1) finden Sie in "`info flex`".

Sie finden viele einfache Beispiele unter "`/usr/share/doc/flex/examples/`". 1

1Einige Kniffe könnten erforderlich sein, um sie auf dem aktuellen System zum Laufen zu bringen.

12.3.4 Bison - ein besseres Yacc

Einige Pakete stellen Yacc-kompatible LR-Parser- oder LALR-Parser-Generatoren in Debian bereit:

Paket	Popcon	Größe	Beschreibung
bison	V:7, I:70	3122	GNU LALR-Parser-Generator
byacc	V:0.1, I:2.6	263	Berkeley LALR-Parser-Generator
btyacc	V:0.01, I:0.04	247	rückverfolgender Parser-Generator basierend auf byacc

Tabelle 12.11: Liste Yacc-kompatibler LALR-Parser-Generatoren

Eine Einführung zu bison(1) finden Sie in "`info bison`".

Sie müssen Ihre eigenen "`main()`"- und "`yyerror()`"-Funktionen bereitstellen. "`main()`" ruft "`yyparse()`" auf, das wiederum "`yylex()`" aufruft, was gewöhnlich von FleX erzeugt wird.

Hier ein Beispiel für die Erstellung eines einfachen terminal-basierten Taschenrechner-Programms.

Lassen Sie uns `example.y` erstellen:

```
/* calculator source for bison */
%{
#include <stdio.h>
extern int yylex(void);
extern int yyerror(char *);
%}

/* declare tokens */
%token NUMBER
%token OP_ADD OP_SUB OP_MUL OP_RGT OP_LFT OP_EQU

%%
calc:
 | calc exp OP_EQU    { printf("Y: RESULT = %d\n", $2); }
 ;

exp: factor
 | exp OP_ADD factor  { $$ = $1 + $3; }
 | exp OP_SUB factor  { $$ = $1 - $3; }
 ;

factor: term
 | factor OP_MUL term { $$ = $1 * $3; }
 ;

term: NUMBER
 | OP_LFT exp OP_RGT  { $$ = $2; }
  ;
%%

int main(int argc, char **argv)
{
  yyparse();
}

int yyerror(char *s)
{
  fprintf(stderr, "error: '%s'\n", s);
}
```

Und dann `example.l`:

```
/* calculator source for flex */
%{
#include "example.tab.h"
%}

%%
[0-9]+ { printf("L: NUMBER = %s\n", yytext); yylval = atoi(yytext); return NUMBER; }
"+"    { printf("L: OP_ADD\n"); return OP_ADD; }
"-"    { printf("L: OP_SUB\n"); return OP_SUB; }
"*"    { printf("L: OP_MUL\n"); return OP_MUL; }
"("    { printf("L: OP_LFT\n"); return OP_LFT; }
")"    { printf("L: OP_RGT\n"); return OP_RGT; }
"="    { printf("L: OP_EQU\n"); return OP_EQU; }
"exit" { printf("L: exit\n");   return YYEOF; } /* YYEOF = 0 */
.      { /* ignore all other */ }
%%
```

Führen Sie dann auf der Shell folgendes aus:

```
$ bison -d example.y
$ flex example.l
$ gcc -lfl example.tab.c lex.yy.c -o example
$ ./example
1 + 2 * ( 3 + 1 ) =
L: NUMBER = 1
L: OP_ADD
L: NUMBER = 2
L: OP_MUL
L: OP_LFT
L: NUMBER = 3
L: OP_ADD
L: NUMBER = 1
L: OP_RGT
L: OP_EQU
Y: RESULT = 9

exit
L: exit
```

12.4 Werkzeuge zur statischen Code-Analyse

Lint-ähnliche Werkzeuge können bei der automatisierten statischen Code-Analyse helfen.

Werkzeuge wie Indent können Menschen bei Code-Überprüfungen helfen, indem Quellcode einheitlich formatiert wird.

Ctags-ähnliche Werkzeuge können Menschen bei Code-Überprüfungen helfen, indem eine Index- (oder Tag-)Datei mit im Quellcode gefundenen Namen erzeugt wird.

Tipp
Konfigurieren Sie Ihren Lieblingseditor (z.B. `emacs` oder `vim`), so dass er asynchrone Lint-Engine-Plugins nutzt und Sie so beim Schreiben von Code unterstützt. Diese Plugins werden sehr mächtig, indem Sie die Vorteile des Language-Server-Protokolls nutzen. Da sie sich sehr schnell ändern, könnte es eine gute Option sein, die Software von Upstream zu verwenden statt des entsprechenden Debian-Pakets.

Paket	Popcon	Größe	Beschreibung
vim-ale	I:0.72	2833	Asynchrone Lint-Engine für Vim 8 und NeoVim
vim-syntastic	I:1.8	1379	Syntaxprüfungen für Vim
elpa-flycheck	V:0.1, I:1.5	1130	moderne on-the-fly Syntaxprüfung für Emacs
elpa-relint	I:0.05	150	Fehlersuche mittels Regexp für Emacs-Lisp-Dateien
cppcheck-gui	V:0.09, I:0.93	7973	Werkzeug für die statische C/C++-Code-Analyse (GUI)
shellcheck	V:4, I:17	22860	Lint-Werkzeug für Shell-Skripte
pyflakes3	V:2, I:14	20	passive Prüfung von Python3-Programmen
pylint	V:4, I:19	2109	Statisches Prüfprogramm für Python-Code
perl	V:672, I:991	840	Interpreter mit internem statischen Code-Prüfmechanismus: `B::Lint`(3perl)
rubocop	V:0.08, I:1.00	4708	statischer Code-Analyser für Ruby
clang-tidy	V:2, I:11	20	clang-basiertes C++-Linter-Werkzeug
splint	V:0.05, I:0.75	2325	Werkzeug zur statischen Überprüfung von C-Programmen auf Programmfehler
flawfinder	V:0.05, I:0.41	187	Werkzeug zur Durchsuchung von Quellcode auf Sicherheitsschwächen
black	V:3, I:15	11130	kompromissloser Python-Code-Formatierer
perltidy	V:0.4, I:2.9	3086	Einrücker und Neuformatierer für Perl-Skripte
indent	V:0.3, I:4.7	438	Code-Formatierer für C
astyle	V:0.2, I:2.4	770	Quelltextformatierer für C, C++, Objective-C, C# und Java
bcpp	V:0.02, I:0.21	114	C(++)-Code schön machen
xmlindent	V:0.06, I:0.70	52	Neuformatierung von XML-Datenströmen
global	V:0.1, I:1.4	1923	Werkzeuge zum Suchen und Browsen in Quelltext
exuberant-ctags	V:1, I:12	341	Erzeugen von Indexdateien aus Quelltextdefinitionen
universal-ctags	V:1, I:12	4238	Erzeugen von Indexdateien aus Quelltextdefinitionen

Tabelle 12.12: Liste von Werkzeugen für die statische Code-Analyse

12.5 Fehlersuche (Debugging)

Debugging ist ein wichtiger Teil der Programmieraktivitäten. Das Wissen darüber, wie man in Programmen einen Fehler sucht, macht Sie zu einem guten Debian-Nutzer, der aussagekräftige Fehlerberichte erstellen kann.

Paket	Popcon	Größe	Dokumentation
gdb	V:94, I:163	13472	"`info gdb`" aus dem `gdb-doc`-Paket
ddd	V:0.2, I:4.8	4519	"`info ddd`" aus dem `ddd-doc`-Paket

Tabelle 12.13: Liste von Debugging-Paketen

12.5.1 Grundlegende Ausführung von gdb

Das primäre Programm zur Fehlersuche (Debugger) im Debian-System ist gdb(1), welches Ihnen erlaubt, ein Programm zu inspizieren, während es läuft.

Wir installieren `gdb` und zugehörige Programme wie folgt:

```
# apt-get install gdb gdb-doc build-essential devscripts
```

Gute Einführungen zu `gdb` finden Sie unter:

- "`info gdb`"
- "Debugging with GDB" in `/usr/share/doc/gdb-doc/html/gdb/index.html`
- "Einführungen im Netz"

Hier ein einfaches Beispiel zur Verwendung von gdb(1) bei einem Programm namens "`program`", kompiliert mit der Option "`-g`", um Debugging-Informationen auszugeben.

```
$ gdb program
(gdb) b 1                # set break point at line 1
(gdb) run args           # run program with args
(gdb) next               # next line
...
(gdb) step               # step forward
...
(gdb) p parm             # print parm
...
(gdb) p parm=12          # set value to 12
...
(gdb) quit
```

Tipp
Viele gdb(1)-Befehle können abgekürzt werden. Vervollständigungen funktionieren wie in der Shell mit der Tabulator-Taste.

12.5.2 Fehlersuche (Debugging) in einem Debian-Paket

Da installierte Binärdateien auf einem Debian-System normalerweise nicht unnötig aufgebläht sein sollten, werden Debugging-Symbole in normalen Paketen meistens entfernt. Um bei solchen Paketen Debugging mit gdb(1) durchführen zu können, müssen `*-dbgsym`-Pakete installiert werden (z.B. `coreutils-dbgsym` für das `coreutils`-Paket).

Die Quellpakete erzeugen diese `*-dbgsym`-Debug-Pakete automatisch mit den normalen Binärpaketen und sie werden in dem separaten Archiv debian-debug abgelegt. Bitte lesen Sie die entsprechenden Artikel im Debian Wiki, wenn Sie weitere Informationen benötigen.

Falls ein Paket, bei dem eine Fehlersuche durchgeführt werden soll, kein `*-dbgsym`-Paket anbietet, müssen Sie es händisch neu bauen und dann installieren, wie hier:

```
$ mkdir /path/new ; cd /path/new
$ sudo apt-get update
$ sudo apt-get dist-upgrade
$ sudo apt-get install fakeroot devscripts build-essential
$ apt-get source package_name
$ cd package_name*
$ sudo apt-get build-dep ./
```

Beheben Sie die Fehler, falls erforderlich.

Erhöhen Sie die Paketversion auf eine Versionsnummer, die nicht mit offiziellen Debian-Versionen kollidiert (Sie können z.B. ein "`+debug1`" anhängen, wenn Sie eine existierende Paketversion neu kompilieren, oder Sie hängen "`~pre1`" an, wenn Sie eine noch nicht veröffentlichte Paketversion selbst kompilieren). Verwenden Sie dazu:

```
$ dch -i
```

Kompilieren und installieren Sie ein Paket mit Debugging-Symbolen wie folgt:

```
$ export DEB_BUILD_OPTIONS="nostrip noopt"
$ debuild
$ cd ..
$ sudo debi package_name*.changes
```

Sie müssen die Build-Skripte des Pakets überprüfen und sicherstellen, dass "`CFLAGS=-g -Wall`" zum Kompilieren der Binärdateien verwendet wird.

12.5.3 Gewinnen von Backtrace-Informationen

Wenn Sie einen Programmabsturz erlitten haben, ist es eine gute Idee, einen Fehlerbericht einzureichen und an diesen zusätzliche Backtrace-Informationen (Daten zur Rückverfolgung von Vorgängen in Programmen zwecks Fehleranalyse) anzuhängen, die Sie mittels Kopieren-und-Einfügen erhalten.

Solche Backtrace-Informationen lassen sich mittels gdb(1) über einen der folgenden Ansätze gewinnen:

- der Ansatz "Absturz in GDB":
  - ein Programm über GDB starten
  - einen Absturz des Programms herbeiführen
  - "`bt`" am GDB-Prompt eingeben
- der Ansatz "Zunächst den Absturz herbeiführen":
  - Aktualisieren der Datei "**/etc/security/limits.conf**", so dass sie folgendes enthält:

    ```
    * soft core unlimited
    ```

  - Eingabe von "`ulimit -c unlimited`" in der Shell
  - Aufruf des Programms aus dieser Shell
  - einen Absturz des Programms herbeiführen, um eine Dump-Datei zu erzeugen
  - Laden der Dump-Datei in GDB mit "`gdb gdb ./program_binary core`"
  - "`bt`" am GDB-Prompt eingeben

Im Falle einer Endlosschleife oder bei eingefrorener Tastatur können Sie einen Absturz des Programms mit `Strg-\` oder `Strg-C` herbeiführen, oder indem Sie “`kill -ABRT PID`” ausführen (Näheres dazu in Abschnitt 9.4.12).

Tipp
Oft stellt man fest, dass in den Backtrace-Informationen eine oder mehrere der ersten Zeilen ”`malloc()`” oder ”`g_malloc()`” enthalten. Wenn dies passiert, besteht eine hohe Wahrscheinlichkeit, dass Ihr Backtrace nicht sehr nützlich sein wird. Der einfachste Weg nützliche Informationen zu bekommen ist, die Umgebungsvariable ”`$MALLOC_CHECK_`” auf einen Wert von 2 zu setzen (Näheres in malloc(3)). Sie können dies wie folgt erledigen, während `gdb` läuft:

```
$ MALLOC_CHECK_=2 gdb hello
```

12.5.4 Erweiterte gdb-Befehle

Befehl	Beschreibung
`(gdb) thread apply all bt`	Backtrace für alle Threads eines Multi-Thread-Programms auslesen
`(gdb) bt full`	Parameter auslesen, die auf dem Aufruf-Stack der Funktionsaufrufe aufgelaufen sind
`(gdb) thread apply all bt full`	Backtrace und Parameter als Kombination der vorangegangenen Optionen auslesen
`(gdb) thread apply all bt full 10`	Backtrace und Parameter für die ersten 10 Aufrufe auslesen, um irrelevante Ausgaben abzuschneiden
`(gdb) set logging on`	Protokoll der Ausgabe von `gdb` in eine Datei schreiben (Standard ist ”`gdb.txt`”)

Tabelle 12.14: Liste erweiterter gdb-Befehle

12.5.5 Überprüfen der Abhängigkeiten von Bibliotheken

Verwenden Sie ldd(1) wie hier, um zu ermitteln, von welchen Bibliotheken ein Programm abhängt:

```
$ ldd /usr/bin/ls
        librt.so.1 => /lib/librt.so.1 (0x4001e000)
        libc.so.6 => /lib/libc.so.6 (0x40030000)
        libpthread.so.0 => /lib/libpthread.so.0 (0x40153000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Damit ls(1) in einer ”chroot”-Umgebung funktioniert, müssen die obigen Bibliotheken in der ”chroot”-Umgebung vorhanden sein.

Lesen Sie dazu Abschnitt 9.4.6.

12.5.6 Werkzeuge zur dynamischen Aufrufverfolgung

Es gibt verschiedene Werkzeuge zur dynamischen Aufrufverfolgung in Debian. Schauen Sie in Abschnitt 9.4.

12.5.7 Fehleranalyse bei X-Fehlern

Wenn ein GNOME-Programm `preview1` einen X-Fehler empfangen hat, sollten Sie eine Meldung wie diese sehen:

```
The program 'preview1' received an X Window System error.
```

Sollte dies der Fall sein, können Sie versuchen, das Programm mit der Option "`--sync`" zu starten und einen Haltepunkt für die Funktion "`gdk_x_error`" zu setzen, um einen Backtrace zu bekommen.

12.5.8 Werkzeuge zur Erkennung von Speicherlecks

Es gibt verschiedene Werkzeuge zur Erkennung von Speicherlecks in Debian:

Paket	Popcon	Größe	Beschreibung
`libc6-dev`	V:259, I:577	12863	mtrace(1): malloc-Debugging-Funktionalität in glibc
`valgrind`	V:4, I:31	89318	Speicher-Debugging- und Analyse-Programm
`electric-fence`	V:0.2, I:2.2	69	malloc(3)-Debugging-Programm
`libdmalloc5`	I:0.54	373	Bibliothek zur Fehlersuche bei Speicherzuweisungen
`duma`	I:0.06	295	Bibliothek zur Detektierung von Pufferüber- oder -unterläufen in C- und C++-Programmen

Tabelle 12.15: Liste von Werkzeugen zur Erkennung von Speicherlecks

12.5.9 Disassemblieren von Binärdateien

Sie können Binär-Code wie folgt mit objdump(1) disassemblieren:

```
$  objdump -m i386 -b binary -D /usr/lib/grub/x86_64-pc/stage1
```

Anmerkung
gdb(1) kann verwendet werden, um Code interaktiv zu disassemblieren.

12.6 Bauwerkzeuge

Paket	Popcon	Größe	Dokumentation
`make`	V:155, I:558	1762	"`info make`" aus dem `make-doc`-Paket
`autoconf`	V:28, I:189	2244	"`info autoconf`" aus dem `autoconf-doc`-Paket
`automake`	V:27, I:188	1959	"`info automake`" aus dem `automake`-Paket
`libtool`	V:23, I:172	1276	"`info libtool`" aus dem `libtool-doc`-Paket
`cmake`	V:23, I:124	63999	cmake(1) - plattformübergreifendes, quelloffenes Make-System
`ninja-build`	V:9, I:55	456	ninja(1) - kleines Bausystem nah am Geiste von Make
`meson`	V:6, I:29	4402	meson(1) - hochproduktives Bausystem, aufbauend auf `ninja`
`xutils-dev`	V:0.6, I:6.6	1495	imake(1), xmkmf(1) usw.

Tabelle 12.16: Liste von Paketen mit Bauwerkzeugen

12.6.1 Make

Make ist ein Werkzeug, um Gruppen von Programmen zu betreuen. Bei Ausführung des Befehls make(1) liest `make` die Regeldatei "`Makefile`" und aktualisiert ein Ziel (Target), falls sich Dateien, von denen das Makefile abhängt, seit der letzten Modifizierung des Targets verändert haben oder falls das Target nicht existiert. Die Ausführungen dieser Aktualisierungen können zeitgleich erfolgen.

Die Syntax der Regeldatei ist folgende:

```
target: [ prerequisites ... ]
 [TAB]  command1
 [TAB]  -command2 # ignore errors
 [TAB]  @command3 # suppress echoing
```

Hierbei ist "`[TAB]`" ein TAB-Code. Jede Zeile wird nach Ersetzung der Variablen durch die Shell interpretiert. Verwenden Sie "\" am Ende einer Zeile, um den Befehl in der nächsten Zeile fortzusetzen. Zur Angabe von Umgebungsvariablen müssen Sie statt "$" hier "$$" schreiben.

Implizite Regeln für das Target und Voraussetzungen können z.B. wie folgt angegeben werden:

```
%.o: %.c header.h
```

Hier enthält das Target das Zeichen "%" (exakt eines davon). Das "%" passt auf jeden nicht leeren Teil-String in den eigentlichen Dateinamen des Targets. Auch die Voraussetzungen nutzen auf ähnliche Art ein "%", um den Bezug zum Namen des Targets anzuzeigen.

automatische Variable	Wert
$@	Ziel
$<	erste Voraussetzung
$?	alle neueren Voraussetzungen
$^	alle Voraussetzungen
$*	"%" trifft auf den Stamm im Target-Muster zu

Tabelle 12.17: Liste von automatischen make-Variablen

Variablenexpandierung	Beschreibung
`foo1 := bar`	einmalige Expandierung
`foo2 = bar`	rekursive Expandierung
`foo3 += bar`	anhängen

Tabelle 12.18: Liste von make-Variablenexpandierungen

Führen Sie "`make -p -f/dev/null`" aus, um alle internen automatischen Regeln zu sehen.

12.6.2 Autotools

Autotools ist eine Sammlung von Programmierwerkzeugen, entwickelt, um dabei zu helfen, Quellcode-Pakete auf viele Unix-artige Systeme portierbar zu machen.

- Autoconf ist ein Werkzeug, das ein "`configure`"-Shell-Skript aus einer "`configure.ac`"-Datei erzeugt.
  - "`configure`" wird dann später verwendet, um aus der Vorlage "`Makefile.in`" die "`Makefile`"-Datei zu generieren.
- Automake erstellt aus einer "`Makefile.am`" eine "`Makefile.in`".
- Libtool ist ein Shell-Skript für die Problematik der Portabilität beim Kompilieren dynamischer Bibliotheken aus Quellcode.

12.6.2.1 Kompilieren und Installieren eines Programms



Warnung
Überschreiben Sie keine Systemdateien, wenn Sie Ihre selbst kompilierten Programme installieren.

Debian verändert keine Dateien unter "`/usr/local/`" oder "`/opt`". Wenn Sie also ein Programm aus den Quellen kompilieren, sollten Sie es in "`/usr/local/`" installieren, damit es nicht mit Debian kollidiert.

```
$ cd src
$ ./configure --prefix=/usr/local
$ make # this compiles program
$ sudo make install # this installs the files in the system
```

12.6.2.2 Deinstallation eines Programms

Wenn Sie noch den Original-Quellcode haben, dieser autoconf(1)/automake(1) nutzt und Sie noch wissen, wie Sie es konfiguriert haben, verfahren Sie wie folgt, um das Programm zu deinstallieren:

```
$ ./configure all-of-the-options-you-gave-it
$ sudo make uninstall
```

Wenn Sie sich absolut sicher sind, dass der Installationsprozess Dateien nur unter "`/usr/local/`" abgelegt hat und und es nichts Wichtiges mehr dort gibt, können Sie alternativ auch alles löschen, wie hier:

```
# find /usr/local -type f -print0 | xargs -0 rm -f
```

Falls Sie nicht sicher sind, wo die Dateien installiert sind, sollten Sie einen Blick auf checkinstall(8) aus dem `checkinstall`-Paket werfen, das einen leeren Pfad für die Deinstallation liefert. Es unterstützt jetzt auch die Erzeugung eines Debian-Pakets mit der Option "`-D`".

12.6.3 Meson

Das Software-Bausystem hat sich weiterentwickelt:

- Autotools als Aufsatz für Make war seit den 1990'ern der De-Facto-Standard für portable Bauinfrastruktur. Es ist sehr langsam.
- CMake - erstmals veröffentlicht im Jahr 2000 - verbesserte die Geschwindigkeit erheblich, baute allerdings ursprünglich auf dem grundsätzlich langsamen Make auf. (Jetzt kann Ninja als Backend dienen.)
- Ninja - erstmals veröffentlicht in 2012 - trat an, um Make zu ersetzen, um so die Baugeschwindigkeit weiter zu erhöhen, und ist auch darauf angewiesen, dass seine Eingabedateien von einem höherwertigen Bausystem erzeugt werden.
- Meson - erstmals veröffentlicht 2013 - ist das neue populäre und schnelle, hochwertige Bausystem; es verwendet Ninja als Backend.

Sie finden Dokumente hierzu unter "The Meson Build system" and "The Ninja build system".

12.7 Web

Einfache interaktive dynamische Webseiten können wie folgt erstellt werden:

- Abfragen werden mittels HTML-Formularen dem Browser-Nutzer präsentiert.
- Das Ausfüllen und Anklicken von Formulareinträgen sendet einen URL-String mit kodierten Parametern vom Browser zum Webserver:
  - "`https://www.foo.dom/cgi-bin/program.pl?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`"
  - "`https://www.foo.dom/cgi-bin/program.py?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`"
  - "`https://www.foo.dom/program.php?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`"
- "%nn" in einer URL wird durch ein Zeichen mit hexadezimalem nn-Wert ersetzt.
- Die Umgebungsvariable wird gesetzt als: "`ABFRAGE_STRING="VAR1=WERT1 VAR2=WERT2 VAR3=WERT3"`".
- Ein CGI-Programm (irgendeines von "`program.*`") auf dem Webserver führt sich selbst mit der Umgebungsvariable "`$ABFRAGE_STRING`" aus.
- Die Standardausgabe (`stdout`) eines CGI-Programms wird zum Webbrowser gesandt und dort als interaktive dynamische Webseite angezeigt.

Aus Sicherheitsgründen wird empfohlen, keinen eigenen zusammengebastelten Code zum Parsen von CGI-Parametern zu verwenden. Es gibt dafür etablierte Module in Perl und Python. PHP unterstützt diese Funktionalitäten. Wenn eine Speicherung der Daten auf dem Client nötig ist, werden HTTP-Cookies verwendet. Ist eine Verarbeitung der Daten auf dem Client erforderlich, wird häufig Javascript genutzt.

Für weitere Informationen wird auf das Common Gateway Interface, die Apache Software Foundation und JavaScript verwiesen.

Die Suche nach "CGI tutorial" auf Google durch Eingabe der kodierten URL https://www.google.com/search?hl=en&ie=UTF-8&q=CGI+tutorial direkt in der Adresszeile des Browsers ist eine gute Möglichkeit, das CGI-Skript auf dem Google-Server in Aktion zu beobachten.

12.8 Die Quellcode-Übersetzung

Es gibt verschiedene Programme zur Übersetzung von Quellcode:

Paket	Popcon	Größe	Schlüsselwort	Beschreibung
`f2c`	V:0.1, I:1.9	443	`FORTRAN → C`	Quellcode von FORTRAN 77 nach C/C++ konvertieren: f2c(1)
`intel2gas`	V:0.01, I:0.17	178	`NASM → GAS`	Konvertierer von NASM (Intel-Format) nach GNU Assembler (GAS)

Tabelle 12.19: Liste von Programmen zur Übersetzung von Quellcode

12.9 Erstellen von Debian-Paketen

Wenn Sie ein Debian-Paket erstellen möchten, lesen Sie folgendes:

- Kapitel 2, um die Grundlagen des Paketsystems zu verstehen;

- Abschnitt 2.7.13, um den grundlegenden Portierungsprozess zu verstehen;
- Abschnitt 9.11.4, um die grundlegenden chroot-Techniken zu verstehen;
- debuild(1) und sbuild(1)
- Abschnitt 12.5.2 für Informationen zum Rekompilieren von Quellcode zwecks Fehlersuche (Debugging);
- Handbuch für Debian-Paketbetreuer (aus dem `debmake-doc`-Paket).
- Debian-Entwicklerreferenz (aus dem `developers-reference`-Paket);
- Debian Policy-Handbuch (aus dem `debian-policy`-Paket);

Es gibt auch Pakete wie `debmake`, `dh-make`, `dh-make-perl` usw., die beim Paketieren helfen.

Anhang A

Anhang

Hier einige Hintergrundinformationen zu diesem Dokument.

A.1 Das Debian-Labyrinth

Das Linux-System ist eine sehr leistungsfähige Computer-Plattform für einen mit dem Netzwerk verbundenen Rechner. Allerdings ist es nicht einfach, zu erlernen, wie man all dessen Fähigkeiten nutzen kann. Die Einrichtung der LPR-Drucker-Warteschlange mit einem nicht PostScript-fähigen Drucker war ein gutes Beispiel für die möglichen Stolpersteine. (Es gibt mittlerweile keine solchen Probleme mehr, da neue Installationen jetzt das CUPS-System nutzen).

Es gibt eine vollständige Quelle für detaillierte Informationen, genannt der ”QUELLCODE”. Dieser ist sehr exakt, aber auch sehr schwer zu verstehen. Es gibt auch andere Quellen wie HOWTOs oder mini-HOWTOs. Diese sind leichter verständlich, neigen aber dazu, zu viele Details zu liefern und den großen Überblick zu verlieren. Ich habe manchmal ein Problem, den richtigen Abschnitt in einem langen HOWTO zu finden, wenn ich nur ein paar Befehle, die ich aufrufen muss, suche.

Ich hoffe, diese ”Debian-Referenz (Version 2.153)” (2026-11-29 05:38:00 UTC) bietet eine gute Starthilfe für Leute im Debian-Labyrinth.

A.2 Copyright-Vergangenheit

Die Debian Reference wurde initiiert von mir, Osamu Aoki <osamu at debian dot org> als persönliches Memo zur Systemadministration. Viele Inhalte entstammen dem Wissen, das ich von der debian-user-Mailingliste und anderen Debian-Ressourcen bezogen habe.

Einem Vorschlag von Josip Rodin folgend, der sehr aktiv im Debian Documentation Project (DDP) war, wurde die ”Debian Reference (version 1, 2001-2007)” als Teil der DDP-Dokumente erstellt.

Nach sechs Jahren stellte ich fest, dass die originale ”Debian Reference (version 1)” veraltet war und begann, viele Inhalte neu zu schreiben. Die neue ”Debian Reference (version 2)” wurde 2008 veröffentlicht.

Ich habe die ”Debian Reference (version 2)” aktualisiert, um neuen Themen (wie Systemd, Wayland, IMAP, PipeWire, Linux-Kernel 5.10) Rechnung zu tragen sowie veraltete (SysV init, CVS, Subversion, SSH-Protokoll 1, Linux-Kernel vor 2.5) zu entfernen. Referenzen zur Situation während Jessie 8 (2015-2020) oder älter wurden überwiegend gelöscht.

Diese ”Debian-Referenz (Version 2.153)” (2026-11-29 05:38:00 UTC) deckt zum Großteil die Debian-Veröffentlichungen Trixie (`=stable`) und Forky (`=testing`) ab.

Die Lehrinhalte haben ihren Ursprung und ihre Inspiration in folgenden Quellen:

- ”Linux User’s Guide” von Larry Greenfield (Dezember 1996)
  - abgelöst durch das ”Debian Tutorial”
- ”Debian Tutorial” von Havoc Pennington (11. Dezember 1998)
  - teilweise geschrieben von Oliver Elphick, Ole Tetlie, James Treacy, Craig Sawyer und Ivan E. Moore II
  - abgelöst durch ”Debian GNU/Linux: Guide to Installation and Usage”
- ”Debian GNU/Linux: Guide to Installation and Usage” von John Goerzen und Ossama Othman (1999)
  - abgelöst durch die ”Debian Reference (version 1)”

Die Paket- und Archivbeschreibungen habe ihren Ursprung und ihre Inspiration in folgenden Quellen:

- ”Debian FAQ” (Version von März 2002, zu dieser Zeit betreut durch Josip Rodin)

Andere Inhalte haben ihren Ursprung und ihre Inspiration in folgenden Quellen:

- ”Debian Reference (version 1)” von Osamu Aoki (2001-2007)
  - abgelöst durch die neuere ”Debian Referenz (Version 2)” in 2008.

Die vorherige ”Debian Reference (version 1)” wurde unter Mithilfe von vielen Beteiligten erstellt:

- ein Großteil der Beiträge zur Netzwerkkonfiguration von Thomas Hood;
- erhebliche Beiträge zu X- und VCS-bezogenen Themen von Brian Nelson;
- Hilfe bei den Build-Scripten und viele Korrekturen des Inhaltes durch Jens Seidel;
- ausgiebige Korrekturlesung durch David Sewell;
- viele Beiträge von Übersetzern und Leuten, die Inhalte beigetragen oder Fehler berichtet haben.

Viele Handbuch- und Infoseiten im Debian-System sowie Upstream-Webseiten und Wikipedia-Dokumente wurden als primäre Referenzen verwendet, um dieses Dokument zu schreiben. Bis zu einem gewissen Maß, das Osamu Aoki im Sinne von Fair Use als angemessen angesehen hat, wurden viele Teile davon, speziell Befehlsdefinitionen, in die Debian Reference übernommen (nach sorgfältiger Überarbeitung zwecks Anpassung an Stil und Zielvorgabe dieses Dokuments).

Die Beschreibung des Debuggers gdb wurde unter Verwendung von Inhalten aus dem Debian Wiki über backtrace in Konsens mit Ari Pollak, Loïc Minier und Dafydd Harries erweitert.

Die Inhalte der derzeitigen ”Debian Referenz (Version 2.153)” (2026-11-29 05:38:00 UTC) sind überwiegend meine eigene Arbeit, mit Ausnahme der oben genannten. Diese wurden auch durch diejenigen, die damals die Inhalte beigetragen haben, aktualisiert.

Der Autor Osamu Aoki dankt allen, die geholfen haben, dieses Dokument möglich zu machen.

A.3 Dokumentenformat

Der Quelltext des englischen Originaldokuments wird derzeit in DocBook-XML-Dateien geschrieben. Diese Quellen werden in HTML, reine Textdateien, PostScript und PDF konvertiert. (Einige Formate könnten für die Verteilung ausgelassen werden.)