


BASH

Jetzt mit
eLearning
*#besser
lernen*



Grundlagen und Praxis der Bash- und C-Programmierung in Unix/Linux

Ein Arbeits- und Übungsbuch

**Roumiana Antonova
Vassilena Slaveva
Tihomira Slavova**

Jetzt registrieren & **besser lernen**

Mit Pearson MyLab zu mehr Lernerfolg

Die interaktive eLearning-Plattform Pearson MyLab erweitert unsere Lehrbücher um die digitale Welt. Selbst komplexe Inhalte werden so anschaulicher und leichter verständlich. Über die Theorie hinaus können Sie das Erlernte praktisch anwenden und unmittelbar erleben.

Lernen wo und wann immer Sie wollen

mit Ihrem persönlichen Lehrbuch als
kommentierbaren eText.

Prüfungen effizient vorbereiten

mit vielzähligen Übungsaufgaben inklusive
Lösungshinweisen und sofortigem Feedback.

Komplexe Inhalte leichter verstehen

dank interaktiver Zusätze wie z.B. Videos,
interaktive Grafiken o.ä.

Sie sind Dozent*in

und möchten Zugang zu exklusiven Dozent*innenmaterialien bzw. MyLab
in Ihrem Kurs einsetzen? Wenden Sie sich bitte an Ihren Dozentenberater und
fordern Sie Ihren persönlichen Zugang an.

<https://www.pearson.de/studium/dozierende/>





Zugangscode

Falls Sie beim Kauf Ihres eBooks keinen Zugangscode erhalten haben, kontaktieren Sie uns bitte über die folgende Seite und halten Sie Ihre Rechnung/Bestellbestätigung bereit:
<https://www.pearson.de/ebook-zugangscode>



Grundlagen und Praxis der Bash- und C-Programmierung in Unix/Linux

Ein Arbeits- und Übungsbuch

Roumiana Antonova
Vassilena Slaveva
Tihomira Slavova

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.dnb.de> abrufbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten ist nicht zulässig. Fast alle Produktbezeichnungen und weitere Stichworte und sonstige Angaben, die in diesem Buch verwendet werden, sind als eingetragene Marken geschützt.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. GERMAN language edition published by PEARSON DEUTSCHLAND GMBH, Copyright © 2022.

Der Umwelt zuliebe verzichten wir auf Einschweißfolie.

10 9 8 7 6 5 4 3 2 1

26 25 24 23 22

ISBN 978-3-86894-425-9 (Buch)
ISBN 978-3-86326-322-5 (E-Book)
MyLab-Zugangscode einlösbar bis 30.09.2026

© 2022 by Pearson Deutschland GmbH
St.-Martin-Straße 82, D-81541 München
Alle Rechte vorbehalten
www.pearson.de
A part of Pearson plc worldwide

Programmleitung: Birger Peil, bpeil@pearson.de
Übersetzung: Petra Alm
Sprachkorrektorat: Katharina Pieper
Coverabbildung: © wrightstudio, 123RF
Satz: Micaela Krieger-Hauwede
Druck und Verarbeitung: Drukkerij Wilco, Amersfoort

Printed in the Netherlands

Inhaltsverzeichnis

Ziel dieses Buchs	9
Danksagung	11
1 Einführung	13
1.1 Was ist Unix?	15
1.2 Und was ist Linux?	15
1.3 Manual Pages – die Dokumentation	16
2 Unix/Linux-Dateisystem	19
2.1 Partitionen und Dateisysteme	21
2.2 Filesystem Hierarchy Standard (FHS)	21
3 Versteckte Dateien und Systemdateien, UID und GID	25
3.1 Versteckte Dateien	27
3.2 Systemdateien	27
3.3 UID und GID	27
4 Weit verbreitete Befehle	31
4.1 Basisbefehle	33
4.2 Befehle für die Arbeit mit Benutzern	35
4.3 Bedingte Ausführung	35
4.4 Gruppieren von Befehlen	36
4.5 Befehlssubstitution	36
5 Basisbefehle für das Arbeiten mit Dateien und Verzeichnissen	41
5.1 Basisbefehle für das Arbeiten mit Verzeichnissen	43
5.2 Befehle zum Arbeiten mit Dateien	44
5.3 Befehle zum Anzeigen des Dateiinhalts	45
5.4 Dateien und Verzeichnisse vergleichen	48
5.5 Dateieigenschaften	49
6 Harte und weiche Links	55
7 Metazeichen in Dateinamen; Anführungs- und Escapezeichen; Variablen	65
7.1 Skripte	67
7.2 Metazeichen in Dateinamen	67
7.3 Anführungs- und Escapezeichen	68
7.4 Variablen	68
7.5 Mit Variablen arbeiten	69

8	Standardeingabe, Standardausgabe und Standardfehler; Hintergrundprozesse; Exit-Status	77
8.1	Eine Zeile aus der Standardeingabe in Variable(n) speichern	79
8.2	Umleiten	79
8.3	Pipe (Pipeline)	80
8.4	Hintergrundprozesse	81
8.5	Exit-Status	82
9	Bedingte Konstrukte und Schleifenstrukturen	87
9.1	Der <code>test</code> -Befehl	89
9.2	Bedingtes <code>if</code> -Konstrukt	91
9.3	Bedingtes <code>case</code> -Konstrukt	91
9.4	<code>while</code> -Schleifenstruktur	92
9.5	<code>until</code> -Schleifenkonstrukt	92
9.6	<code>for</code> -Schleifenstruktur	92
9.7	Eine Schleife mit <code>break</code> vorzeitig verlassen	94
9.8	<code>continue</code> in einer Schleife	94
10	Arithmetische Ausdrücke und Stringmanipulation	103
10.1	Der Befehl <code>expr</code>	105
10.2	Der Befehl <code>let</code>	106
10.3	Doppelklammern und arithmetische Expansion	107
10.4	Manipulation von Strings	107
11	Dateisicherheit	115
11.1	Der Befehl <code>chmod</code>	118
11.2	Die Befehle <code>chown</code> und <code>chgrp</code>	119
11.3	Der Befehl <code>umask</code>	119
12	Mit Prozessen arbeiten	125
12.1	Der Befehl <code>ps</code>	127
12.2	Der Befehl <code>top</code>	128
12.3	Der Befehl <code>kill</code>	128
13	Funktionen und Arrays	133
13.1	Funktionen	135
13.2	Arrays	136
14	C-Programmierung: Systemfunktionen zum Arbeiten mit Dateien	145
14.1	Datei öffnen	148
14.2	Datei schließen	149
14.3	Lesen aus und Schreiben in eine Datei	149
14.4	Arbeiten mit Fehlern	151

15 C-Programmierung: Systemfunktionen zum Ersetzen und Beenden von Prozessen	157
15.1 Die <i>exec</i> -Funktionen	159
15.2 Die Funktion <i>exit</i>	160
16 C-Programmierung: Systemfunktionen zum Verwalten von Prozessen	165
16.1 Die Funktion <i>fork</i>	167
16.2 Die Funktion <i>wait</i>	168
16.3 Abrufen von Prozess-IDs	168
A Antworten	173
B Texteditoren: vi, Vim und Nano	205
B.1 Der Editor <i>vi</i>	206
B.2 Der Editor <i>Vim</i>	210
B.3 Der Editor <i>nano</i>	210
C Fernverbindung zu Unix-Systemen über SSH	213
C.1 Verwendung von Linux und Mac	214
C.2 Verwendung von Windows	215
Literatur	221

Heute schon ? im Lab gewesen ?

Mehr Lernerfolg durch effiziente Vorbereitung

Neu Erlerntes in Ruhe reflektieren und praktisch anwenden – das ist das A und O für nachhaltigen Lernerfolg. Wirklich effizient lernen Sie vor allem in einer entspannten Umgebung, in der Sie das Tempo vorgeben. An der Uni ist das ein Ausnahmefall.

Pearson MyLab ermöglicht Ihnen, was eine Vorlesung nicht leisten kann: Trainieren Sie Erlerntes in Ihrem eigenen Tempo. Dabei steht Pearson MyLab Ihnen wie ein Coach zur Seite, so lange bis der Stoff hundertprozentig sitzt.



Erlerntes praktisch anwenden und vertiefen ■

Vielzählige, teils auch interaktive Übungsaufgaben, perfekt abgestimmt auf den Lernstoff, ermöglichen die praktische Anwendung der theoretischen Kenntnisse.

Lösungswege einfach nachvollziehen ■

Schritt für Schritt-Hinweise zu jeder Aufgabe verschaffen Klarheit und ein tiefgehendes Verständnis für den Lernstoff.

Den Leistungsfortschritt im Blick ■

Mit sofortigem Feedback haben Sie nicht nur die Möglichkeit zur Kontrolle und Einschätzung der eigenen Leistung sondern auch Erfolgserlebnisse, die zusätzlich motivieren und anspornen, noch besser zu werden.

Ziel dieses Buchs

Dieses Buch soll den Leser in die Grundlagen von Theorie und Praxis eines Unix/Linux-Betriebssystems einführen. Es konzentriert sich dabei auf den ersten und dritten Abschnitt der Manpages und möchte angehenden Programmierern von Unix/Linux-Anwendungen alle dafür notwendigen Kenntnisse vermitteln.

Dieses Buch enthält zum größten Teil Übungen mit verschiedenen Systembefehlen. Diese Übungen machen den Leser mit der allgemeinen Funktionsweise des Unix/Linux-Systems und mit seinem Terminal vertraut. Die letzten Kapitel sind den Funktionen der C-Bibliothek und dem Arbeiten mit Dateien und Prozessen gewidmet.

Tutor | Bash- und C-Programmierung in Unix/Linux – Learn a little...do a little

Ebenfalls enthalten, wie im Klassiker Giancoli Physik, ist die Umsetzung des bei Studierenden bewährten didaktischen Konzepts der digitalen Arbeitsblätter *Learn a little...do a little*. Hier werden mit Ihnen im zum Buch gehörigen **Tutor | Bash- und C-Programmierung in Unix/Linux** interaktive Skript-Aufgaben mit Lösungshinweisen durchgearbeitet. Sie können diese Aufgaben solange üben und mit neuen Anordnungen wiederholen, bis es wirklich sitzt. Knapp **200 Aufgaben** unterschiedlicher Schwierigkeitsgrade helfen Ihnen dabei, sich am Verständnis des Bash-Scripting zu erfreuen.

Zugangscode
MyLab über
Verlagssupport

Das *Gradebook* in **Tutor | Bash- und C-Programmierung in Unix/Linux** gibt Ihnen eine Übersicht der Aufgaben, die sie absolviert haben, sodass Sie stets das Gelernte im Überblick haben. Eine detaillierte Anleitung finden Sie auf der Startseite zum **Tutor | Bash- und C-Programmierung in Unix/Linux**.

Tutor | Bash- und C-Programmierung in Unix/Linux – Learning to go

In **Tutor | Bash- und C-Programmierung in Unix/Linux** kann Mensch sich mit jedem Browser einloggen, der JavaScript aktiviert hat. Wir empfehlen ein Laptop oder Tablet. Zusätzlich können Sie das Lehrbuch als E-Book zum Lesen überall mitnehmen.

Nutzen Sie diese einzigartige Möglichkeit zur Wiederholung des Stoffs und zur (Klausur-) Vorbereitung. Lernen Sie damit im Team, an der Uni oder im Rahmen des Selbststudiums. Fordern Sie bei Fragen die Unterstützung der Tutoren ein, sie helfen Ihnen gern. Lassen Sie sich nicht durch kleine Rückschläge entmutigen, sondern bleiben Sie dran. Dafür haben wir den **Tutor | Bash- und C-Programmierung in Unix/Linux** entwickelt, der Ihnen in kleinen *Learn a little...do a little*-Schritten zur Seite steht.

Bei technischen Fragen zum **Tutor | Bash- und C-Programmierung in Unix/Linux** können Sie sich jederzeit an unseren Support wenden.

Wir wünschen Ihnen viel Erfolg.

Ihr Pearson Team

Jetzt & registrieren

Mit Pearson MyLab zu mehr Lernerfolg

Die interaktive eLearning-Plattform Pearson MyLab erweitert unsere Lehrbücher um die digitale Welt. Selbst komplexe Inhalte werden so anschaulicher und leichter verständlich. Über die Theorie hinaus können Sie das Erlernte praktisch anwenden und unmittelbar erleben.



Lernen wo und wann immer Sie wollen ■

mit Ihrem persönlichen Lehrbuch
als kommentierbaren eText.

Prüfungen effizient vorbereiten ■

mit vielzähligen Übungsaufgaben inklusive
Lösungshinweisen und sofortigem Feedback.

Komplexe Inhalte leichter verstehen ■

dank interaktiver Zusätze wie z.B. Videos,
interaktive Grafiken o.ä.

Danksagung

Wir möchten uns bei den folgenden Personen bedanken, die dieses Buch möglich gemacht haben:

- Christina Zasheva und Sabina Bocheva, Assistenzprofessorinnen der Universität »St. Kliment Ohridski« in Sofia, für ihre Anregungen zu den Hauptthemen und zum Aufbau des Buches.
- Atanas Dobrev, Barzana Panajotova, Iliana Petrova, Jordan Babukov, Mincho Gaydarov, Svetozar Ilkov und Yavor Mihaylov, die mit uns als wissenschaftliche Assistenten an der Universität »St. Kliment Ohridski« in Sofia gearbeitet haben und deren Unterrichtserfahrungen mit Studenten zahlreiche Ideen zu vielen Übungen der einzelnen Kapitel geliefert haben.

Roumiana Hristova Antinova, Vassilena Iordanova Slaveva, Tihomira Encheva Slavova

besser lernen mit Pearson MyLab

92%

der MyLab-Nutzer
verstehen den
Lernstoff besser,

91%

fühlen sich besser
auf Prüfungen
vorbereitet,

90%

finden Pearson
MyLabs leicht zu
bedienen.

mehr als
11 Mio.
Studenten weltweit
lernen besser mit
Pearson MyLab

Wecken Sie das Potenzial, das in Ihnen steckt!

Registrieren Sie sich jetzt kostenfrei und
nutzen Sie alle Vorteile Ihres Pearson MyLabs!

Ihren Zugangscode sowie alle Infos zur
Registrierung finden Sie vorne in
diesem Buch.



Einführung

1.1	Was ist Unix?	15
1.2	Und was ist Linux?	15
1.3	Manual Pages – die Dokumentation	16

» Dieses Kapitel stellt die Betriebssysteme Unix und Linux vor und gibt einen kurzen Überblick über die Geschichte von Unix und dessen Weiterentwicklung zu Linux. Es beschreibt auch die allgemeine Struktur der Dokumentation von Unix-Betriebssystemen und hebt hervor, auf welche Teile davon sich dieses Buch konzentriert. «

1.1. Was ist Unix?

Menschen nutzen in ihrem Alltag häufig Unix-basierte Betriebssysteme (kurz OS für *Operating System*), ohne es zu merken. Zu den Geräten mit solchen Betriebssystemen gehören Telefone, Router, Spielekonsolen, Smart-TV-Betriebssysteme usw.

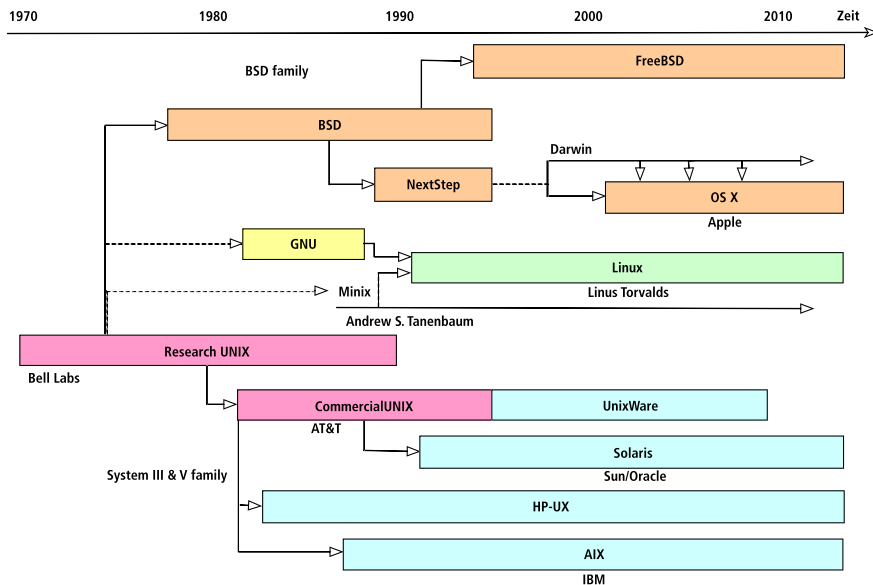


Abbildung 1.1 Linux – ein geschichtlicher Abriss

Einfach ausgedrückt ist Unix ein Multitasking- und Multiuser-Computerbetriebssystem, das in den späten 1960er Jahren im Forschungszentrum Bell Labs entwickelt wurde. Unix bietet eine Reihe von kleinen, modularen Hilfsprogrammen, die zur Lösung komplexerer Aufgaben kombiniert werden können. Während seiner Entwicklung verzweigte sich Unix grob in zwei Betriebssystemfamilien – BSD und System V (► Abbildung 1.1) [11, 7].

Im Allgemeinen interagieren Benutzer mit einem Unix-System über ein Terminal – eine textbasierte Schnittstelle zum System, auch bekannt als Kommando- oder Befehlszeile. Der Benutzer kann Befehle eintippen und erhält dann im Terminal eine Rückmeldung. Die Befehle werden von der Shell interpretiert, die gleichzeitig Kommandozeileninterpreter und Sprache ist. Die beliebteste Shell ist Bash.

1.2. Und was ist Linux?

Linux ist ein unixoides Computerbetriebssystem. Genau wie die Systeme der BSD-Familie ist Linux quelloffen [8]. Die Leistungsfähigkeit von Linux basiert auf dem Linux-Kernel, entwickelt von Linus Torvalds [8]. Der Kernel ist das zentrale Computerprogramm, auf

das das Betriebssystem des Computers aufsetzt. Er übersetzt Anfragen von Software in Anweisungen, die die CPU versteht, verwaltet den Speicher und kommuniziert mit Peripheriegeräten wie Druckern, Lautsprechern usw.

1.3. Manual Pages – die Dokumentation

Nahezu alles, was man über das Unix-Betriebssystem erfahren kann, ist in seiner Dokumentation, den sogenannten »Manual Pages« oder kurz »Manpages« [14] beschrieben. In der Regel bestehen diese Manpages aus acht Abschnitten (die bei den meisten Unix-Systemen gleich sind) und mehreren mit Buchstaben versehenen Anhängen (die sich je nach verwendetem Unix-System unterscheiden).

Die wichtigsten Informationen finden Sie in den ersten drei Abschnitten der Manpages:

- Der erste Abschnitt enthält detaillierte Informationen über die allgemeinen Befehle des Systems, d. h. über Programme und Anwendungen, die für den durchschnittlichen Benutzer nützlich sind. Sie finden in diesem Abschnitt eine Vielzahl von Benutzerbefehlen und Tools.
- Abschnitt zwei erklärt die für das Betriebssystem verfügbaren Systemaufrufe, mit denen Computerprogramme Dienste vom Kernel anfordern. Diese Systemaufrufe werden nicht direkt aufgerufen, sondern über eine C-Bibliothek, die als API verwendet wird.
- Abschnitt drei enthält Informationen über die Bibliotheken (Unterprogramme) des Systems, einschließlich der oben erwähnten C-Bibliothek. Zu fast jedem Systemaufruf gibt es eine C-Wrapper-Funktion für den Aufruf in C-Programmen.

Zusammenfassung

- ✓ Unix ist ein Multitasking- und Multiuser-Betriebssystem, das in den späten 1960er Jahren im Bell-Labs-Forschungszentrum entwickelt wurde.
- ✓ Im Allgemeinen interagiert der Benutzer mit einem Unix-System über sein Terminal – eine textbasierte Schnittstelle zum System, auch bekannt als Kommando- bzw. Befehlszeile.
- ✓ Die Befehle werden von der Shell interpretiert, die Kommandozeileninterpreter und Sprache in einem ist.
- ✓ Linux ist ein unixoides Betriebssystem.
- ✓ Der Kernel bildet den Systemkern, auf den das Linux-Betriebssystem aufbaut.
- ✓ Nahezu alles über das Unix-Betriebssystem kann man in seiner Dokumentation – den »Manual Pages«, auch »Manpages« genannt – nachlesen.
- ✓ Der erste Abschnitt der Manpages enthält detaillierte Informationen zu den allgemeinen Befehlen des Systems.
- ✓ Der zweite Teil der Manpages erklärt die für das Betriebssystem zur Verfügung stehenden Systemaufrufe, mit denen Computerprogramme Dienste vom Kernel anfordern.
- ✓ Der dritte Teil der Manpages liefert Informationen über die Bibliotheken (Unterprogramme) des Systems.

Heute schon ? im Lab gewesen ?

Mehr Lernerfolg – immer und überall

Nie mehr Bücher schleppen oder nach verloren gegangenen Merktzetteln suchen! Mit Ihrem persönlichen Lehrbuch als kommentierbarem eText gibt Person MyLab Ihnen die Freiheit zu lernen, wann und wo immer Sie wollen. Zu Hause am PC ebenso wie unterwegs mit dem Tablet – und das ganz ohne lästiges Schleppen.

Individualisierbar ■

Hinterlegen Sie wichtige Notizen oder Markierungen direkt im eText. So schaffen Sie ihr persönliches Lehrbuch, mit dem Ihnen nichts mehr entgeht.



Ihr Lehrbuch als erweitertes PDF ■

Ihr eText steht Ihnen im Look & Feel Ihres Lehrbuchs zur Verfügung. So behalten Sie die Orientierung, gerade wenn Sie on- und offline arbeiten.

Unterwegs einfach dabei ■

Lesen Sie den eText in Pearson MyLab am PC oder auf dem Tablet. Mit der MyLab App haben sie Ihr Buch auch einfach unterwegs dabei.

Unix/Linux-Dateisystem

2.1	Partitionen und Dateisysteme	21
2.2	Filesystem Hierarchy Standard (FHS)	21

» Dieses Kapitel gibt einen Überblick über das Unix/Linux-Dateisystem, einschließlich der Definition der verschiedenen im System verwendeten Dateitypen. Außerdem geht es auf das Thema Partitionen ein, stellt die Standardhierarchie des Linux-Dateisystems vor und beschreibt den Inhalt einiger der gängigsten Verzeichnisse. «

Ein Dateisystem ist eine Methode zum Speichern und Organisieren beliebiger Datensammlungen in einer benutzerfreundlichen Struktur.

Eine einfache Beschreibung des Unix-Systems, die auch auf Linux anwendbar ist, lautet:

»Auf einem Unix-System ist alles eine Datei; und wenn es keine Datei ist, ist es ein Prozess.«¹ [5]

Die verschiedenen Dateitypen sind [5]:

- **Regulär:** Textdatei, Binärdatei, Bilddatei etc.
- **Verzeichnis:** Binärdatei, die eine Liste von anderen Dateien enthält
- **Gerät (Spezial):** Datei, die ein Gerät repräsentiert
- **Link:** Datei, die ein Verweis auf eine andere Datei ist
- **(Domain-)Socket:** Datei, die für die Kommunikation zwischen Prozessen verwendet wird, ähnlich wie TCP/IP-Sockets
- **Benannte Pipe:** Datei, die es Prozessen ermöglicht, ohne die Semantik von Netzwerk-Sockets zu kommunizieren

2.1. Partitionen und Dateisysteme

Eine Partition ist ein logischer Teil der Festplatte, während ein Dateisystem eine Methode zum Speichern/Suchen von Dateien auf einer Festplatte ist (normalerweise in einer Partition). Zur Veranschaulichung könnte man sich Dateisysteme als Stammbäume vorstellen, die die Nachkommen einer Familie und deren Beziehungen abbilden, während Partitionen jeweils verschiedene Familien repräsentieren (die jeweils ihren eigenen Stammbaum haben).

2.2. Filesystem Hierarchy Standard (FHS)

Das Linux-Dateisystem wird üblicherweise als Baumstruktur ausgedrückt (► Abbildung 2.1).

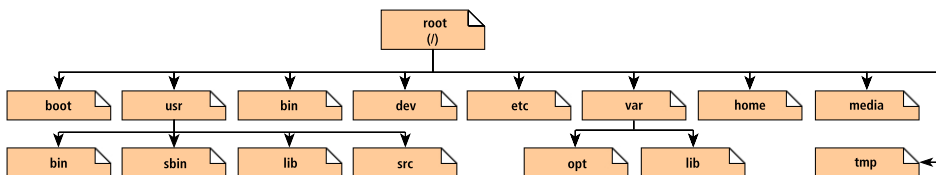


Abbildung 2.1 Linux-Dateisystem

¹ Anm. d. Übers.: Der Begriff *everything is a file* wird auch im Deutschen als Beschreibung von Unix-Systemen verwendet.

Linux-Systeme speichern ihre wichtigen Dateien in einem Standardlayout, dem sogenannten Filesystem Hierarchy Standard (FHS). Dieser Standard stellt sicher, dass Benutzer zwischen Distributionen wechseln können, ohne neu lernen zu müssen, wie das System organisiert ist. Eine Distribution ist eine Variante eines Linux-Systems.

Linux verwendet den Schrägstrich `/` als Trennzeichen in seinen Pfadangaben (im Gegensatz zu Windows, das den Backslash `\` verwendet). Außerdem kennt Linux keine Laufwerksbuchstaben. Neue Laufwerke werden als Verzeichnisse in dem Dateisystem angelegt, oft unter `/media` (eine CD-ROM mit der Bezeichnung *FEDORA* wäre also z. B. unter `/media/FEDORA` und eine Datei `README.txt` auf dieser CD-ROM unter `/media/FEDORA/README.txt` zu finden).

Einige der am häufigsten verwendeten Verzeichnisse sind [5]:

`/` – Das Root-Verzeichnis. Dieses Verzeichnis bildet die Wurzel des Dateisystems und enthält alles andere.

`/boot` – Wichtiges Verzeichnis, das die Startdateien und den Kernel enthält.

`/bin` – Verzeichnis der ausführbaren Binärdateien. In diesem Verzeichnis befinden sich die ausführbaren Dateien der Befehle, die wir normalerweise zum Interagieren mit der Shell benötigen. Auf dieses Verzeichnis können alle Systembenutzer zugreifen, um Dateien auszuführen. Im Dateisystem gibt es weitere `bin`-Unterverzeichnisse, die ausführbare Binärdateien für unterschiedliche Zwecke enthalten.

`/usr` – Verzeichnis, das Binärdateien, Bibliotheken, Dokumentation und Quellcode für Secondlevel-Programme enthält. Dieses Verzeichnis hat mehrere Unterverzeichnisse mit nützlichen Dateien. Zum Beispiel enthält `/usr/bin` Binärdateien für Benutzerprogramme, `/usr/sbin` enthält Binärdateien für Systemadministratoren, `/usr/lib` enthält Bibliotheken für `/usr/bin` und `/usr/sbin`; `/usr/src` enthält Benutzerprogramme, die aus Quellcode installiert werden.

`/dev` – Verzeichnis für Gerätedateien. Das sind alle Geräte, die an das System angeschlossen sind: USB, Terminals, Drucker usw.

`/etc` – Verzeichnis für Konfigurationsdateien, die von allen Programmen benötigt werden. Eine Konfigurationsdatei dient der Steuerung eines Programmablaufs und sollte nicht ausführbar sein.

`/var` – Verzeichnis für Systemprotokolldateien, Cache-Verzeichnisse, Mail- und Drucker-Spool-Verzeichnisse sowie andere Dateien, in die das System regelmäßig Daten schreibt.

`/home` – Verzeichnis, das die Homeverzeichnisse aller Benutzer enthält, um darin ihre persönlichen Dateien zu speichern.

`/tmp` – Verzeichnis, in das verschiedene Programme temporäre Dateien speichern. Es wird normalerweise beim Neustart des Systems gelöscht.

Bei der Installation einer Linux-Distribution ist es sinnvoll, bestimmte Verzeichnisse in eigenen, separaten Partitionen abzulegen, um die wichtigsten Daten zu schützen [5] und den zunehmenden Umfang bestimmter Verzeichnisse (z. B. `/usr`, `/var`, `/home`, `/tmp`) zu begrenzen. Andernfalls können die Benutzerdaten den gesamten Plattenspeicher belegen, was dazu führt, dass sich das System irgendwann aufhängt.