



Moderne Betriebssysteme

5., aktualisierte Auflage

Andrew S. Tanenbaum
Herbert Bos

 Pearson

 EXTRAS
ONLINE

Moderne Betriebssysteme

Moderne Betriebssysteme

Inhaltsverzeichnis

Moderne Betriebssysteme

Impressum

Inhaltsübersicht

Inhaltsverzeichnis

Vorwort

Kapitel 1 - Einführung

1.1 Was ist ein Betriebssystem?

1.1.1 Das Betriebssystem als eine erweiterte Maschine

1.1.2 Das Betriebssystem als Ressourcenverwalter

1.2 Geschichte der Betriebssysteme

1.2.1 Die erste Generation (1945-1955) auf Basis von Elektronenröhren

1.2.2 Die zweite Generation (1955-1965) Transistoren und Stapelverarbeitungssysteme

1.2.3 Die dritte Generation (1965-1980) integrierte Schaltkreise und Multiprogrammierung

1.2.4 Die vierte Generation (1980 bis heute) der PC

1.2.5 Die fünfte Generation (1990 bis heute) mobile Computer

1.3 Überblick über die Computerhardware

1.3.1 Prozessoren

1.3.2 Arbeitsspeicher

1.3.3 Nicht flüchtiger Speicher

1.3.4 Ein-/Ausgabegeräte

1.3.5 Bussysteme

1.3.6 Hochfahren des Computers

1.4 Die Betriebssystemfamilie

Inhaltsverzeichnis

- 1.4.1 Betriebssysteme für Großrechner
- 1.4.2 Betriebssysteme für Server
- 1.4.3 Betriebssysteme für PCs
- 1.4.4 Betriebssysteme für Handheld-Computer und Smartphones
- 1.4.5 Betriebssysteme für eingebettete Systeme und dem Internet der Dinge
- 1.4.6 Echtzeitbetriebssysteme
- 1.4.7 Betriebssysteme für Smartcards

1.5 Betriebssystemkonzepte

- 1.5.1 Prozesse
- 1.5.2 Adressräume
- 1.5.3 Dateien
- 1.5.4 Ein- und Ausgabe
- 1.5.5 Datenschutz und Datensicherheit
- 1.5.6 Die Shell
- 1.5.7 Die Ontogenese rekapituliert die Phylogenese

1.6 Systemaufrufe

- 1.6.1 Systemaufrufe zur Prozessverwaltung
- 1.6.2 Systemaufrufe zur Dateiverwaltung
- 1.6.3 Systemaufrufe zur Verzeichnisverwaltung
- 1.6.4 Sonstige Systemaufrufe
- 1.6.5 Die Win32-Programmierschnittstelle (API) unter Windows

1.7 Betriebssystemstrukturen

- 1.7.1 Monolithische Systeme
- 1.7.2 Geschichtete Systeme
- 1.7.3 Mikrokern
- 1.7.4 Das Client-Server-Modell
- 1.7.5 Virtuelle Maschinen
- 1.7.6 Exokernel und Unikernel

1.8 Die Welt aus der Sicht von C

- 1.8.1 Die Programmiersprache C

Inhaltsverzeichnis

1.8.2 Header-Dateien

1.8.3 Große Programmierprojekte

1.8.4 Das Laufzeitmodell

1.9 Forschung im Bereich der Betriebssysteme

1.10 Überblick über das Buch

1.11 Metrische Einheiten

Kapitel 2 - Prozesse und Threads

2.1 Prozesse

2.1.1 Das Prozessmodell

2.1.2 Prozesserzeugung

2.1.3 Prozessbeendigung

2.1.4 Prozesshierarchien

2.1.5 Prozesszustände

2.1.6 Implementierung von Prozessen

2.1.7 Modellierung der Multiprogrammierung

2.2 Threads

2.2.1 Der Gebrauch von Threads

2.2.2 Das klassische Thread-Modell

2.2.3 POSIX-Threads

2.2.4 Implementierung von Threads im Benutzeradressraum

2.2.5 Implementierung von Threads im Kernel

2.2.6 Hybride Implementierungen

2.2.7 Einfachthread-Code in Multithread-Code umwandeln

2.3 Ereignisgesteuerte Server

2.4 Synchronisation und Interprozesskommunikation

2.4.1 Race Conditions

2.4.2 Kritische Regionen

2.4.3 Wechselseitiger Ausschluss mit aktivem Warten

2.4.4 Sleep und Wakeup

2.4.5 Semaphor

Inhaltsverzeichnis

2.4.6 Mutex

2.4.7 Monitor

2.4.8 Nachrichtenaustausch

2.4.9 Barrieren

2.4.10 Prioritätsumkehr

2.4.11 Sperren vermeiden: das Read-Copy-Update-Schema

2.5 Scheduling

2.5.1 Einführung in das Scheduling

2.5.2 Scheduling in Stapelverarbeitungssystemen

2.5.3 Scheduling in interaktiven Systemen

2.5.4 Scheduling in Echtzeitsystemen

2.5.5 Strategie versus Mechanismus

2.5.6 Thread-Scheduling

2.6 Forschung zu Prozessen und Threads

Kapitel 3 - Speicherverwaltung

3.1 Systeme ohne Speicherabstraktion

3.1.1 Mehrere Programme ohne Speicherabstraktion ausführen

3.2 Speicherabstraktion: Adressräume

3.2.1 Das Konzept des Adressraums

3.2.2 Swapping

3.2.3 Verwaltung von freiem Speicher

3.3 Virtueller Speicher

3.3.1 Paging

3.3.2 Seitentabellen

3.3.3 Beschleunigung des Paging

3.3.4 Seitentabellen für große Speicherbereiche

3.4 Seitenersetzungsalgorithmen

3.4.1 Der optimale Algorithmus zur Seitenersetzung

3.4.2 Der Not-Recently-Used-Algorithmus (NRU)

3.4.3 Der First-In-First-Out-Algorithmus (FIFO)

Inhaltsverzeichnis

- 3.4.4 Der Second-Chance-Algorithmus
- 3.4.5 Der Clock-Algorithmus
- 3.4.6 Der Least-Recently-Used-Algorithmus (LRU)
- 3.4.7 Simulation von LRU durch Software
- 3.4.8 Der Working-Set-Algorithmus
- 3.4.9 Der WSClock-Algorithmus
- 3.4.10 Zusammenfassung der Seiteneretzungsstrategien

3.5 Entwurfskriterien für Paging-Systeme

- 3.5.1 Lokale versus globale Zuteilungsstrategien
- 3.5.2 Lastkontrolle
- 3.5.3 Bereinigungsstrategien
- 3.5.4 Seitengröße
- 3.5.5 Trennung von Befehls- und Datenräumen
- 3.5.6 Gemeinsame Seiten
- 3.5.7 Gemeinsame Bibliotheken
- 3.5.8 Memory-Mapped-Dateien

3.6 Implementierungsaspekte

- 3.6.1 Aufgaben des Betriebssystems beim Paging
- 3.6.2 Behandlung von Seitenfehlern
- 3.6.3 Sicherung von unterbrochenen Befehlen
- 3.6.4 Sperren von Seiten im Speicher
- 3.6.5 Hintergrundspeicher
- 3.6.6 Trennung von Strategie und Mechanismus

3.7 Segmentierung

- 3.7.1 Implementierung von Segmentierung
- 3.7.2 Segmentierung mit Paging: MULTICS
- 3.7.3 Segmentierung mit Paging: x86 von Intel

3.8 Forschung zur Speicherverwaltung

Kapitel 4 - Dateisysteme

4.1 Dateien

Inhaltsverzeichnis

- 4.1.1 Benennung von Dateien
- 4.1.2 Dateistruktur
- 4.1.3 Dateitypen
- 4.1.4 Dateizugriff
- 4.1.5 Dateiattribute
- 4.1.6 Dateioperationen
- 4.1.7 Beispielprogramm mit Aufrufen zum Dateisystem

4.2 Verzeichnisse

- 4.2.1 Verzeichnissysteme mit einer Ebene
- 4.2.2 Hierarchische Verzeichnissysteme
- 4.2.3 Pfadnamen
- 4.2.4 Operationen auf Verzeichnissen

4.3 Implementierung von Dateisystemen

- 4.3.1 Layout eines Dateisystems
- 4.3.2 Implementierung von Dateien
- 4.3.3 Implementierung von Verzeichnissen
- 4.3.4 Gemeinsam benutzte Dateien
- 4.3.5 Log-basierte Dateisysteme
- 4.3.6 Journaling-Dateisysteme
- 4.3.7 Flash-basierte Dateisysteme
- 4.3.8 Virtuelle Dateisysteme

4.4 Dateisystemverwaltung und -optimierung

- 4.4.1 Plattenspeicherverwaltung
- 4.4.2 Sicherung von Dateisystemen
- 4.4.3 Konsistenz eines Dateisystems
- 4.4.4 Performanz eines Dateisystems
- 4.4.5 Defragmentierung von Plattenspeicher
- 4.4.6 Komprimierung und Deduplizierung
- 4.4.7 Sicheres Löschen von Dateien und Verschlüsseln von Festplatten

4.5 Beispiele von Dateisystemen

Inhaltsverzeichnis

4.5.1 Das MS-DOS-Dateisystem

4.5.2 Das UNIX-V7-Dateisystem

4.6 Forschung zu Dateisystemen

Kapitel 5 - Eingabe und Ausgabe

5.1 Grundlagen der Ein-/Ausgabehardware

5.1.1 Ein-/Ausgabegeräte

5.1.2 Controller

5.1.3 Memory-Mapped-Ein-/Ausgabe

5.1.4 Direct Memory Access

5.1.5 Interrupts

5.2 Grundlagen der Ein-/Ausgabesoftware

5.2.1 Ziele von Ein-/Ausgabesoftware

5.2.2 Programmierte Ein-/Ausgabe

5.2.3 Interruptgesteuerte Ein-/Ausgabe

5.2.4 Ein-/Ausgabe mit DMA

5.3 Schichten der Ein-/Ausgabesoftware

5.3.1 Unterbrechungsroutinen

5.3.2 Gerätetreiber

5.3.3 Geräteunabhängige Ein-/Ausgabesoftware

5.3.4 Ein-/Ausgabesoftware im Benutzeradressraum

5.4 Massenspeicher: Festplatten und SSDs

5.4.1 Magnetplatten

5.4.2 SSDs

5.4.3 RAID

5.5 Uhren

5.5.1 Hardwareuhren

5.5.2 Softwareuhren

5.5.3 Soft-Timer

5.6 Benutzerschnittstellen: Tastatur, Maus, Bildschirm

5.6.1 Eingabesoftware

Inhaltsverzeichnis

5.6.2 Ausgabesoftware

5.7 Thin Clients

5.8 Energieverwaltung

5.8.1 Hardwareaspekte

5.8.2 Betriebssystemaspekte

5.8.3 Energieverwaltung und Anwendungsprogramme

5.9 Forschung im Bereich Ein-/Ausgabe

Kapitel 6 - Deadlocks

6.1 Ressourcen

6.1.1 Unterbrechbare und nicht unterbrechbare Ressourcen

6.1.2 Ressourcenanforderung

6.1.3 Das Philosophenproblem

6.2 Einführung in Deadlocks

6.2.1 Voraussetzungen für Ressourcen-Deadlocks

6.2.2 Modellierung von Deadlocks

6.3 Der Vogel-Strauß-Algorithmus

6.4 Erkennen und Beheben von Deadlocks

6.4.1 Deadlock-Erkennung bei einer Ressource je Typ

6.4.2 Deadlock-Erkennung bei mehreren Ressourcen je Typ

6.4.3 Beheben von Deadlocks

6.5 Vermeidung von Deadlocks (Avoidance)

6.5.1 Ressourcenspuren

6.5.2 Sichere und unsichere Zustände

6.5.3 Der Bankier-Algorithmus für eine einzelne Ressource

6.5.4 Der Bankier-Algorithmus für mehrere Ressourcen

6.6 Verhinderung von Deadlocks (Prevention)

6.6.1 Unterlaufen der Bedingung des wechselseitigen Ausschlusses

6.6.2 Unterlaufen der Hold-and-Wait-Bedingung

6.6.3 Unterlaufen der Bedingung der Ununterbrechbarkeit

6.6.4 Unterlaufen der zyklischen Wartebedingung

Inhaltsverzeichnis

6.7 Weitere Themen zu Deadlocks

- 6.7.1 Zwei-Phasen-Sperren
- 6.7.2 Kommunikationsdeadlocks
- 6.7.3 Livelock
- 6.7.4 Verhungern

6.8 Forschung zu Deadlocks

Kapitel 7 - Virtualisierung und die Cloud

7.1 Geschichte der Virtualisierung

7.2 Anforderungen für die Virtualisierung

7.3 Typ-1- und Typ-2-Hypervisoren

7.4 Techniken für die effiziente Virtualisierung

- 7.4.1 Das Nichtvirtualisierbare virtualisieren
- 7.4.2 Kosten der Virtualisierung

7.5 Der Hypervisor: ein idealer Mikrokern?

7.6 Speichervirtualisierung

7.7 Ein-/Ausgabevirtualisierung

7.8 Virtuelle Maschinen bei Mehrkernprozessoren

7.9 Clouds

- 7.9.1 Clouds-as-a-Service
- 7.9.2 Migration von virtuellen Maschinen
- 7.9.3 Checkpointing

7.10 Virtualisierung auf Betriebssystemebene

7.11 Fallstudie: VMware

- 7.11.1 Die Anfänge von VMware
- 7.11.2 VMware Workstation
- 7.11.3 Aufgaben bei der Virtualisierungseinführung im x86
- 7.11.4 VMware Workstation: Überblick über die Lösungen
- 7.11.5 Die Weiterentwicklung von VMware Workstation
- 7.11.6 ESX Server: Typ-1-Hypervisor von VMware

Inhaltsverzeichnis

7.12 Forschung zu Virtualisierung und der Cloud

Kapitel 8j - Multiprozessorsysteme

8.1 Multiprozessoren

- 8.1.1 Hardware von Multiprozessoren
- 8.1.2 Betriebssystemarten für Multiprozessoren
- 8.1.3 Synchronisation in Multiprozessorsystemen
- 8.1.4 Multiprozessor-Scheduling

8.2 Multicomputer

- 8.2.1 Hardware von Multicomputern
- 8.2.2 Low-Level-Kommunikationssoftware
- 8.2.3 Kommunikationssoftware auf Benutzerebene
- 8.2.4 Entfernter Prozeduraufruf (RPC)
- 8.2.5 Distributed Shared Memory
- 8.2.6 Multicomputer-Scheduling
- 8.2.7 Lastausgleich

8.3 Verteilte Systeme

- 8.3.1 Netzwerkhardware
- 8.3.2 Netzwerkdienste und -protokolle
- 8.3.3 Dokumentenbasierte Middleware
- 8.3.4 Dateisystembasierte Middleware
- 8.3.5 Objektbasierte Middleware
- 8.3.6 Koordinationsbasierte Middleware

8.4 Forschung zu Multiprozessorsystemen

Kapitel 9 - IT-Sicherheit

9.1 Die Grundlagen der Betriebssystemsicherheit

- 9.1.1 Die CIA-Triade
- 9.1.2 Sicherheitsprinzipien
- 9.1.3 Sicherheit der Betriebssystemstruktur
- 9.1.4 Trusted Computing Base
- 9.1.5 Angreifer

Inhaltsverzeichnis

9.1.6 Können wir sichere Systeme bauen?

9.2 Zugriff auf Ressourcen steuern

9.2.1 Schutzdomains

9.2.2 Zugriffskontrolllisten

9.2.3 Capabilities

9.3 Formale Modelle von sicheren Systemen

9.3.1 Multilevel-Sicherheit

9.3.2 Kryptografie

9.3.3 Trusted Platform Module (TPM)

9.4 Authentifizierung

9.4.1 Passwörter

9.4.2 Authentifizierung durch Besitz

9.4.3 Biometrische Authentifizierung

9.5 Ausnutzen von softwareseitigen Sicherheitslücken

9.5.1 Pufferüberlaufangriffe

9.5.2 Formatstring-Angriffe

9.5.3 Use-After-Free-Angriffe

9.5.4 Schwachstellen aufgrund von Typverwechslung

9.5.5 NULL-Zeiger-Dereferenzierungsangriffe

9.5.6 Angriffe durch Ganzzahlüberlauf

9.5.7 Angriffe durch Kommando-Injektion

9.5.8 Time-of-Check-to-Time-of-Use-Angriffe

9.5.9 Double-Fetch-Schwachstelle

9.6 Ausnutzen von hardwareseitigen Sicherheitslücken

9.6.1 Verdeckte Kanäle

9.6.2 Seitenkanäle

9.6.3 Angriffe auf die transiente Ausführung

9.7 Insider-Angriffe

9.7.1 Logische Bomben

9.7.2 Hintertüren

Inhaltsverzeichnis

9.7.3 Login-Spoofing

9.8 Betriebssystem-Hardening

9.8.1 Feinkörnige Randomisierung

9.8.2 Kontrollflussbeschränkungen

9.8.3 Zugangsbeschränkungen

9.8.4 Prüfung auf Code- und Datenintegrität

9.8.5 Bescheinigung unter Verwendung von TPM

9.8.6 Kapselung von nicht vertrauenswürdigem Code

9.9 Forschung zum Thema IT-Sicherheit

Kapitel 10 - Fallstudie 1: UNIX, Linux und Android

10.1 Die Geschichte von UNIX und Linux

10.1.1 UNICS

10.1.2 PDP-11-UNIX

10.1.3 Portable UNIX-Varianten

10.1.4 Berkeley-UNIX

10.1.5 Standard-UNIX

10.1.6 MINIX

10.1.7 Linux

10.2 Überblick über Linux

10.2.1 Ziele von Linux

10.2.2 Schnittstellen zu Linux

10.2.3 Die Shell

10.2.4 Hilfsprogramme unter Linux

10.2.5 Kernelstruktur

10.3 Prozesse in Linux

10.3.1 Grundlegende Konzepte

10.3.2 Systemaufrufe zur Prozessverwaltung in Linux

10.3.3 Implementierung von Prozessen und Threads in Linux

10.3.4 Scheduling in Linux

10.3.5 Synchronisation in Linux

Inhaltsverzeichnis

10.3.6 Starten von Linux

10.4 Speicherverwaltung in Linux

10.4.1 Grundlegende Konzepte

10.4.2 Systemaufrufe zur Speicherverwaltung in Linux

10.4.3 Implementierung der Speicherverwaltung in Linux

10.4.4 Paging in Linux

10.5 Ein-/Ausgabe in Linux

10.5.1 Grundlegende Konzepte

10.5.2 Netzwerkimplementierung

10.5.3 Systemaufrufe zur Ein-/Ausgabe in Linux

10.5.4 Implementierung der Ein-/Ausgabe in Linux

10.5.5 Linux-Kernelmodule

10.6 Das Linux-Dateisystem

10.6.1 Grundlegende Konzepte

10.6.2 Systemaufrufe zur Dateiverwaltung in Linux

10.6.3 Implementierung des Linux-Dateisystems

10.6.4 NFS das Netzwerkdateisystem

10.7 Sicherheit in Linux

10.7.1 Grundlegende Konzepte

10.7.2 Systemaufrufe zu Sicherheitsfunktionen in Linux

10.7.3 Implementierung von Sicherheitsfunktionen in Linux

10.8 Android

10.8.1 Android und Google

10.8.2 Geschichte von Android

10.8.3 Entwurfsziele

10.8.4 Architektur von Android

10.8.5 Linux-Erweiterungen

10.8.6 ART

10.8.7 Interprozesskommunikation mit Binder

10.8.8 Android-Anwendungen

Inhaltsverzeichnis

10.8.9 Intents

10.8.10 Prozessmodell

10.8.11 Sicherheit und Datenschutz

10.8.12 Hintergrundauführung und Social Engineering

Kapitel 11 - Fallstudie 2: Windows 11

11.1 Die Geschichte von Windows bis Windows 11

11.1.1 Die 1980er: MS-DOS

11.1.2 Die 1990er: MS-DOS-basiertes Windows

11.1.3 Die 2000er: NT-basiertes Windows

11.1.4 Windows Vista

11.1.5 Windows 8

11.1.6 Windows 10

11.1.7 Windows 11

11.2 Programmierung von Windows

11.2.1 Universelle Windows-Plattform

11.2.2 Windows-Subsysteme

11.2.3 Die native NT-Programmierschnittstelle

11.2.4 Die Win32-Programmierschnittstelle

11.2.5 Die Windows-Registrierungsdatenbank

11.3 Systemstruktur

11.3.1 Betriebssystemstruktur

11.3.2 Starten von Windows

11.3.3 Implementierung des Objekt-Managers

11.3.4 Subsysteme, DLLs und Dienste im Benutzermodus

11.4 Prozesse und Threads in Windows

11.4.1 Grundlegende Konzepte

11.4.2 API-Aufrufe zur Job-, Prozess-, Thread- und Fiberverwaltung

11.4.3 Implementierung von Prozessen und Threads

11.4.4 WoW64 und Emulation

11.5 Speicherverwaltung

Inhaltsverzeichnis

- 11.5.1 Grundlegende Konzepte
- 11.5.2 Systemaufrufe zur Speicherverwaltung
- 11.5.3 Implementierung der Speicherverwaltung
- 11.5.4 Speicherkompression
- 11.5.5 Speicherpartitionen

11.6 Caching in Windows

11.7 Ein-/Ausgabe in Windows

- 11.7.1 Grundlegende Konzepte
- 11.7.2 API-Aufrufe für die Ein-/Ausgabe
- 11.7.3 Implementierung der Ein-/Ausgabe

11.8 Das Windows-NT-Dateisystem

- 11.8.1 Grundlegende Konzepte
- 11.8.2 Implementierung des NT-Dateisystems

11.9 Energieverwaltung in Windows

11.10 Virtualisierung in Windows

- 11.10.1 Hyper-V
- 11.10.2 Container
- 11.10.3 Virtualisierungsbasierte Sicherheit

11.11 Sicherheit in Windows

- 11.11.1 Grundlegende Konzepte
- 11.11.2 API-Aufrufe zu Sicherheitsfunktionen
- 11.11.3 Implementierung von Sicherheitsfunktionen
- 11.11.4 Schadensbegrenzende Maßnahmen in der IT-Sicherheit

Kapitel 12 - Entwurf von Betriebssystemen

12.1 Das Problem des Entwurfs

- 12.1.1 Ziele
- 12.1.2 Warum ist es schwierig, ein Betriebssystem zu entwerfen?

12.2 Schnittstellenentwurf

- 12.2.1 Leitlinien
- 12.2.2 Paradigmen

Inhaltsverzeichnis

12.2.3 Die Systemaufrufchnittstelle

12.3 Implementierung

12.3.1 Systemstruktur

12.3.2 Mechanismus versus Strategie

12.3.3 Orthogonalität

12.3.4 Namensräume

12.3.5 Zeitpunkt des Bindens

12.3.6 Statische versus dynamische Strukturen

12.3.7 Top-down- versus Bottom-up-Implementierung

12.3.8 Synchron versus asynchrone Kommunikation

12.3.9 Nützliche Techniken

12.4 Performanz

12.4.1 Warum sind Betriebssysteme langsam?

12.4.2 Was sollte verbessert werden?

12.4.3 Der Zielkonflikt zwischen Laufzeit und Speicherplatz

12.4.4 Caching

12.4.5 Hints

12.4.6 Ausnutzen der Lokalität

12.4.7 Optimieren des Normalfalls

12.5 Projektverwaltung

12.5.1 Der Mythos vom Mann-Monat

12.5.2 Teamstruktur

12.5.3 Die Bedeutung der Erfahrung

12.5.4 No Silver Bullet

Bibliografie

13.1 Empfehlungen für weiterführende Literatur

13.1.1 Einführung

13.1.2 Prozesse und Threads

13.1.3 Speicherverwaltung

13.1.4 Dateisysteme

Inhaltsverzeichnis

13.1.5 Ein- und Ausgabe

13.1.6 Deadlocks

13.1.7 Virtualisierung und die Cloud

13.1.8 Multiprozessorsysteme

13.1.9 IT-Sicherheit

13.1.10 Fallstudie 1: UNIX, Linux und Android

13.1.11 Fallstudie 2: Windows

13.1.12 Betriebssystementwurf

13.2 Alphabetische Literaturliste

Namensregister

Register

Copyright

Copyright

Daten, Texte, Design und Grafiken dieses eBooks, sowie die eventuell angebotenen eBook-Zusatzdaten sind urheberrechtlich geschützt. Dieses eBook stellen wir lediglich als **persönliche Einzelplatz-Lizenz** zur Verfügung!

Jede andere Verwendung dieses eBooks oder zugehöriger Materialien und Informationen, einschließlich

- der Reproduktion,
- der Weitergabe,
- des Weitervertriebs,
- der Platzierung im Internet, in Intranets, in Extranets,
- der Veränderung,
- des Weiterverkaufs und
- der Veröffentlichung

bedarf der **schriftlichen Genehmigung** des Verlags. Insbesondere ist die Entfernung oder Änderung des vom Verlag vergebenen Passwort- und DRM-Schutzes ausdrücklich untersagt!

Bei Fragen zu diesem Thema wenden Sie sich bitte an: **info@pearson.de**

Zusatzdaten

Möglicherweise liegt dem gedruckten Buch eine CD-ROM mit Zusatzdaten oder ein Zugangscode zu einer eLearning Plattform bei. Die Zurverfügungstellung dieser Daten auf unseren Websites ist eine freiwillige Leistung des Verlags. **Der Rechtsweg ist ausgeschlossen.** Zugangscodes können Sie darüberhinaus auf unserer Website käuflich erwerben.

Hinweis

Dieses und viele weitere eBooks können Sie rund um die Uhr und legal auf unserer Website herunterladen:

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