

Errata zur 1. Auflage von **Betriebssysteme kompakt**.
Erschienen 2017 bei Springer Vieweg. ISBN: 978-3-662-53142-6

Seite 6, Tabelle 2.2

Das niedwertigste Bit ist x_0 und nicht x_1 und das höchstwertigste Bit ist im konkreten Beispiel x_7 und nicht x_8 .

k	<u>Quotient</u>	<u>Rest</u>
	$k \text{ DIV } 2$	$k \text{ MODULO } 2$
164	82	$0 = x_0$
82	41	$0 = x_1$
41	20	$1 = x_2$
20	10	$0 = x_3$
10	5	$0 = x_4$
5	2	$1 = x_5$
2	1	$0 = x_6$
1	0	$1 = x_7$

Seite 8, Tabelle 2.4

Aus mathematischer und didaktischer Sicht ist es sinnvoller „Bytes“ und nicht „Bedeutung“ als Überschrift der dritten Spalte zu verwenden.

Name	Symbol	Bytes
Kilobyte	kB	$2^{10} = 1.024$
Megabyte	MB	$2^{20} = 1.048.576$
Gigabyte	GB	$2^{30} = 1.073.741.824$
Terabyte	TB	$2^{40} = 1.099.511.627.776$
Petabyte	PB	$2^{50} = 1.125.899.906.842.624$
Exabyte	EB	$2^{60} = 1.152.921.504.606.846.976$
Zettabyte	ZB	$2^{70} = 1.180.591.620.717.411.303.424$
Yottabyte	YB	$2^{80} = 1.208.925.819.614.629.174.706.176$

Seite 15, 3. Zeile von Abschnitt 3.1

Ersetze „Eine ein“ durch „Ein“.

Seite 16, 2. Zeile von Abschnitt 3.2

Ersetze „installiert und“ durch „installiert, das“.

Seite 19, Tabelle 3.1

In der 5. Generationen der Computersysteme ist es sinnvoll noch die Multicore-Prozessoren aufzunehmen.

Generation	Zeitraum	Technologischer Fortschritt
0	bis 1940	(Elektro-)mechanische Rechenmaschinen
1	1940–1955	Elektronenröhren, Relais, Steckfelder
2	1955–1965	Transistoren, Stapelverarbeitung
3	1965–1980	Integrierte Schaltungen, Dialogbetrieb
4	1980–2000	Hochintegrierte Schaltungen, PCs/Workstations
5	2000 bis ?	Multicore-Prozessoren, Verteilte Systeme, Mobile Systeme

Seite 23, 3. Zeile von Abschnitt 3.4

Ersetze „Mehrprogrammbetrieb“ durch „Mehrbenutzerbetrieb“.

Seite 25, Letzte Zeile von Abschnitt 3.4.1

Ersetze „(siehe Abschn. 8.6)“ durch „(siehe Abschn. 3.4.2)“.

Seite 26, 14. Zeile von Abschnitt 3.4.2

Ersetze „Mehrprogrammbetrieb“ durch „Mehrbenutzerbetrieb“.

Seite 36, 8. und 9. Zeile von Abschnitt 3.9

Ersetze „unabhängig“ durch „abhängig“.

Seite 41, Abschnitt 4.1.3, 3. Zeile

Ersetze „Bytes“ durch „Bits“.

Seite 44, Tabellenüberschrift von Tabelle 4.2

Ersetze „Generationen von Computersystemen“ durch „Ausgewählte Bussysteme“.

Seite 47, 3. Zeile

Ersetze „bedeutete“ durch „bedeutet“.

Seite 55, Abschnitt 4.4.4, 1. Zeile

„Festplatten sind pro Bit...“

Im Buch fehlt das Wort „sind“.

Seite 56, Unterabschnitt zu „Adressierung der Daten auf Festplatten“

Ersetze bei der Auflistung zu PATA den Eintrag

- 8 Bits für die Sektoren/Spur (maximal 255, da ab 1 gezählt wird)

durch das besser formulierte

- 8 Bits für die Sektoren/Spur (maximal 255, da Sektornummer 0 nicht verwendet wird)

Seite 56, Unterabschnitt zu „Adressierung der Daten auf Festplatten“

Ersetze bei der Auflistung zum BIOS den Eintrag

- 8 Bits für die Köpfe (maximal 256)

durch

- 8 Bits für die Köpfe (maximal 255, da Kopfnummer 0 nicht verwendet wird)

Seite 56, Unterabschnitt zu „Adressierung der Daten auf Festplatten“

Ersetze bei der Auflistung zum BIOS den Eintrag

- 6 Bits für die Sektoren/Spur (maximal 63, da ab 1 gezählt wird)

durch das besser formulierte

- 6 Bits für die Sektoren/Spur (maximal 63, da Sektornummer 0 nicht verwendet wird)

Seite 57, Unterabschnitt zu „Zugriffszeit bei Festplatten“

Relevant ist hier nicht die „Zugriffsverzögerung durch Umdrehung“ sondern die „Durchschnittliche Zugriffsverzögerung durch Umdrehung“. Diese entspricht der halben Zugriffsverzögerung durch Umdrehung. Sobald der Kopf die richtige Spur erreicht hat, muss im Durchschnitt eine halbe Umdrehung der Scheibe abgewartet werden, bis sich der richtige Sektor unter dem Kopf befindet.

Seite 58, Formel am Ende von Abschnitt 4.4.4

Ersetze die Formel durch:

Durchschnittliche Zugriffsverzögerung durch Umdrehung [ms] =

$$\frac{1000 \frac{\text{[ms]}}{\text{[sec}}} \times 60 \frac{\text{[sec]}}{\text{[min}}} \times 0,5}{\frac{\text{Umdrehungen}}{\text{[min]}}} = \frac{30.000 \frac{\text{[ms]}}{\text{[min]}}}{\frac{\text{Umdrehungen}}{\text{[min]}}}$$

Seite 78, Abbildung 5.2

In der letzten Spalte unterhalb von „Prozess A wird beendet und G gestartet“ muss der erste Prozess mit 18 MB Speicherbelegung G heißen und nicht A.

Seite 86, 1. und 2. Zeile von Abschnitt „Organisation und Adressierung des Speichers im Real Mode“

Ersetze

„Im Real Mode wird der verfügbare Speicher in maximal 65.536 Segmente unterteilt. Jedes Segment ist 16 Bytes ($= 2^{16} = 65.536$ Bits) groß.“

durch

„Im Real Mode wird der verfügbare Speicher in gleich große Segmente unterteilt. Jedes Segment ist 64 kB groß.“

Seite 90, Abbildung 5.9

Ersetze „SDD/HDD“ durch „SSD/HDD“.

Seite 92, 7. Zeile von oben

Ersetze „... der Grad...“ durch „... den Grad...“.

Seite 94, 12. Zeile von oben

Ersetze „zweistufiges Paging“ durch „mehrstufiges Paging“.

Üblicherweise arbeiteten die Betriebssysteme auf 32-Bit-Architekturen mit zweistufigem Paging, aber der Pentium Pro Prozessor war auch eine 32-Bit-Architektur und bei diesem arbeiteten die Betriebssysteme mit dreistufigem Paging.

Seite 101, 5. Zeile von oben

„... Speicherschutz mehr bietet ist wegen des...“.

Im Buch fehlt das Wort „ist“.

Seite 102, 10. Zeile

Ersetze „0xc0000000“ durch „0xC0000000“.

Seite 105, Abbildung 5.20

Die Positionierung der Seiteninhalte in Abbildung 5.20 ist verwirrend, weil so der Eindruck entsteht, dass die Seiten beim Miss-Event um jeweils eine Seite nach oben

rutschen. Das ist aber nicht der Fall. Besser ist folgende Darstellung, bei der klar ersichtlich ist, dass die Seiten an Ort und Stelle ersetzt werden.

Anfragen: **1 2 3 4 1 2 5 1 2 3 4 5**

1. Seite:	1	1	1	4	4	4	5	5	5	3	3	3
2. Seite:		2	2	2	1	1	1	1	1	1	4	4
3. Seite:			3	3	3	2	2	2	2	2	2	5

Queue:

1	2	3	4	1	2	5	1	2	3	4	5
1	2	3	4	1	2	5	1	2	3	4	5
1	2	3	4	1	2	5	1	2	3	4	5

Seite 106, Vorletzte Zeile von Abschnitt 5.3.3

Ersetze „LRU“ durch „LFU“.

Seite 107, rechte Spalte, Abschnitt 5.3.5, 2. Zeile

Ersetze „... bei die Auswahl...“ durch „... bei der Auswahl...“.

Seite 121, Abbildung 6.7

Ersetze „Dateizuordnungstabellen“ durch „Dateizuordnungstabelle“.

Seite 123, Tabellenüberschrift von Tabelle 6.3

Streiche „Dateigröße und“.

Seite 123, Kopfzeile der zweiten Spalte Tabelle 6.3

Streiche „Minimale“.

Seite 124, Tabellenüberschrift von Tabelle 6.4

Ersetze

„Maximale Dateigröße und Dateisystemgröße von FAT32 bei unterschiedlich großen Clustern“.

durch

„Standardmäßige Clustergröße von FAT32 bei unterschiedlich großen Partitionen“.

Seite 124, Kopfzeile der zweiten Spalte Tabelle 6.4

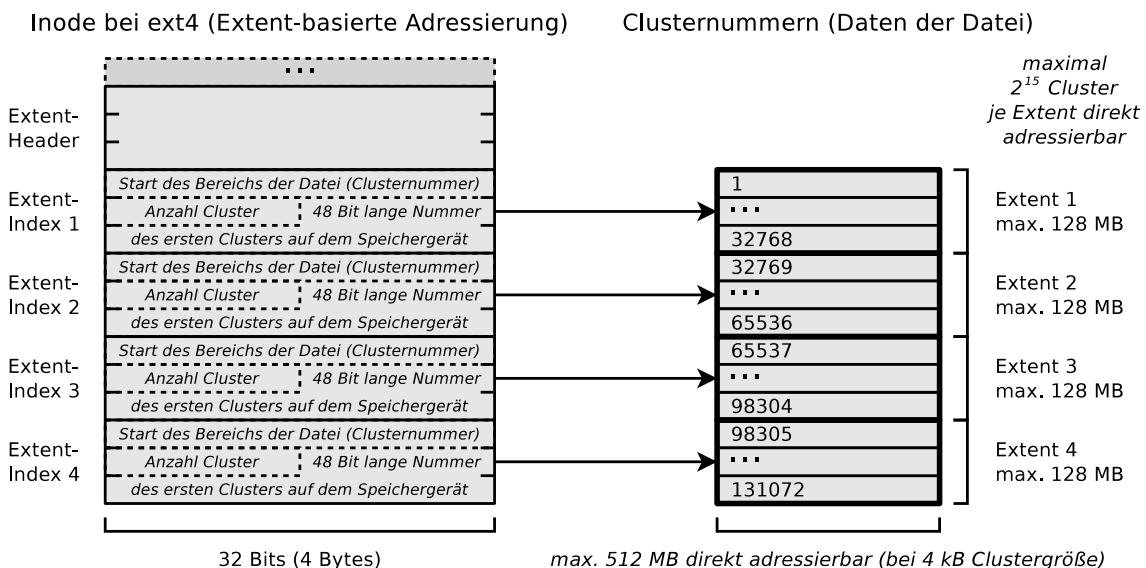
Streiche „Minimale“.

Seite 126, 10. und 11. Zeile des 3. Abschnitts

Streiche im Satz „Allerdings ist auch bei diesem Konzept nur die Konsistenz der Metadaten ~~ist~~ garantiert.“ das hier durchgestrichene Wort.

Seite 129, Abbildung 6.10

In Abbildung 6.10 haben sich nicht nur ein Layout-Fehler in der Darstellung des Inodes, sondern auch mehrere inhaltliche Fehler eingeschlichen. Aus didaktischer Sicht ist es besser im Inode im 48 Bits langen Datenfeld nicht von Blöcken, sondern von Clustern zu sprechen. Die Clusternummern in den Extents sind auch nicht korrekt. Mit 2^{15} Clusternummern kann jeder Extent maximal 32768 Clusternummern umfassen und nicht 131072. Die korrekte Darstellung ist wie folgt:

**Seite 129, 3. Zeile**

Ersetze „Abb. 6.8“ durch „Abb. 6.10“.

Seite 131, Tabellenüberschrift von Tabelle 6.5 Ersetze

„Maximale Dateigröße und Dateisystemgröße von NTFS bei unterschiedlich großen Clustern“.

durch

„Standardmäßige Clustergröße von NTFS bei unterschiedlich großen Partitionen“.

Seite 131, Kopfzeile der zweiten Spalte Tabelle 6.5

Streiche „Minimale“.

Seite 132, 17. Zeile von oben

Ersetze „Auch all“ durch „Auch alle“.

Seite 144, Abschnitt 8.1

Ersetze:

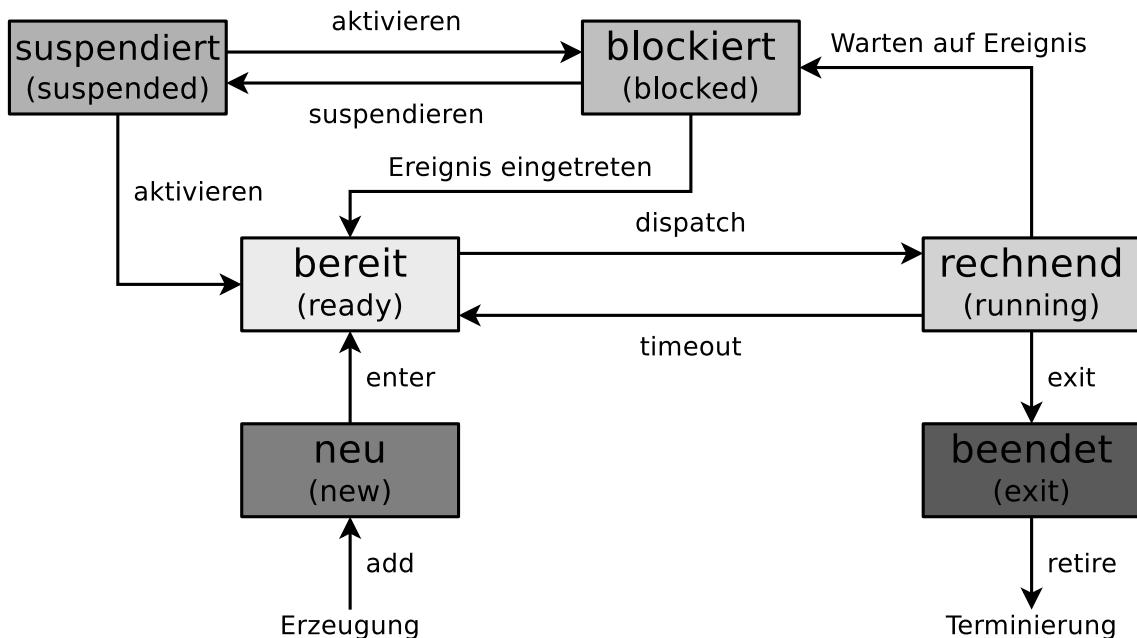
- Information über Eltern- oder Kindprozesse

Durch:

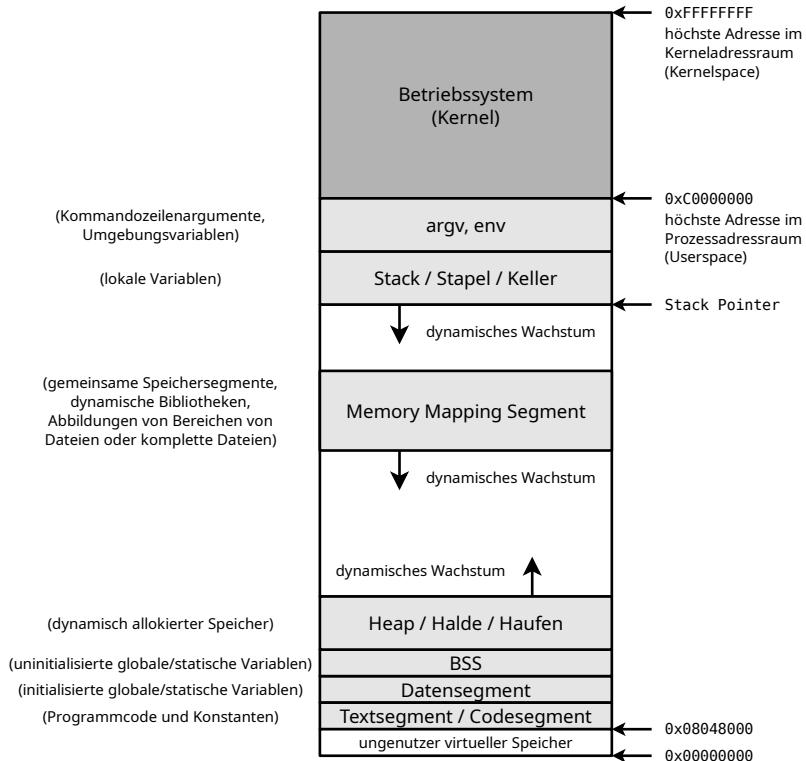
- Elternprozessnummer (PPID)

Seite 149, Abbildung 8.8

Im 6-Zustands-Prozessmodell in Abbildung 8.8 fehlt ein Prozessübergang aktivieren von Prozesszustand suspendiert zu Prozesszustand blockiert.

**Seite 151-152, Abschnitt 8.3 (inkl. Abbildung 8.11)**

Die Struktur eines Linux-Prozesses auf einem 32-Bit-Systemen im Speicher (wie in Abbildung 8.12) gezeigt ist im Buch nicht korrekt dargestellt.



- Das Textsegment enthält den ausführbaren Programmcode (Maschinencode) und ausschließlich lesbare Daten wie Konstanten.
- Das Datensegment enthält initialisierte Variablen, die entweder global sind oder lokal und zugleich statisch.
- Der Bereich BSS enthält diejenigen globalen Variablen und lokalen statischen Variablen, die beim Start des Prozesses nicht initialisiert werden.
- Der Heap wächst dynamisch. Hier kann ein Prozess dynamisch zur Laufzeit Speicher allokalieren (mit `malloc`). Der Heap kann im Gegensatz zum Textsegment, Datensegment und BSS während der Laufzeit eines Programms wachsen.
- Kommandozeilenargumente (argv) des Programmaufrufs und die Umgebungsvariablen (env) liegen in einem Bereich, der am Ende des Userspace beginnt.
- Der Stack ermöglicht die Realisierung geschachtelter Funktionsaufrufe und arbeitet nach dem Prinzip LIFO. Mit jedem Funktionsaufruf wird eine Datenstruktur auf den Stack gelegt, die die Aufrufparameter, die Rücksprungadresse und einen Zeiger auf die aufrufende Funktion im Stack enthält. Die Funktionen legen auch ihre lokalen Variablen auf den Stack. Beim Rücksprung aus einer Funktion wird die Datenstruktur der Funktion aus dem Stack entfernt. Der Stack kann also während der Laufzeit eines Programms wachsen.

- Im Speicherbereich Memory Mapping, der sich im Adressraum zwischen Stack und Heap befindet, werden dynamische Bibliotheken (*Shared Libraries*) geladen. Auch gemeinsame Speicherbereiche sind hier abgebildet. Zusätzlich können komplette Dateien oder Bereiche von Dateien hier mit dem Systemaufruf `mmap` abgebildet werden.

Diese Anpassungen betreffen auch die entsprechenden Einträge im Glossar.

Eine ausführlichere und korrekte Darstellung der Struktur eines Linux-Prozesses auf einem 32-Bit-Systemen im Speicher enthält Auflage 4 des Buches.

Seite 154, Abbildung 8.12

Ersetze „SDD/HDD“ durch „SSD/HDD“.

Seite 157, Bildunterschrift von Abbildung 8.11

Ersetze „auf einem 32 Bit-Systemen im Speicher“ durch „im Speicher eines 32 Bit-Systems“.

So wie es im Buch geschrieben ist, ist es inhaltlich korrekt, aber sprachlich unschön.

Seite 161, vorletzte Zeile

Ersetze „... von der...“ durch „... von denen...“.

Seite 161, letzte Zeile

Ersetze „exec“ durch „exec1“.

Seite 185, 2. Zeile im 3. Absatz

Ersetze „Existing Resource Vektor“ durch „Existing Resource Vector“.

Seite 185, 4. Zeile im 3. Absatz

Ersetze „Available Resource Vektor“ durch „Available Resource Vector“.

Seite 187, Vorletzte Zeile von Abschnitt 9.2.3

Im vorletzten Satz des Abschnitts fehlt die schließende Klammer nach „Klasse 4“.

Seite 194, vorletzter Absatz

Ersetze „Konversion“ durch „Konvertierung“.

Der Begriff „Konversion“ wird in vielen Bereichen (u.a. Religion, Stadtentwicklung und Konversion) verwendet, aber in der Informatik und ganz besonders im Kontext verschiedener Stellenwertsystem ist „Konvertierung“ der korrekte Fachbegriff.

Seite 194, letzte Zeile

Ersetze „Zeile“ durch „Spalte“.

Seite 195, vorletzte Zeile von Abschnitt 9.3.1

Ersetze

`$ ipcrm shm 127008859`

durch

`$ ipcrm shm 56393780`

Seite 197, 10. Zeile von unten

Ersetze „... der das Segment anlegt, auf dieses lesend...“

durch: „... der die Nachrichtenwarteschlange anlegt, auf diese lesend...“.

Seite 202, 1. Zeile des dritten Absatzes

Ersetze „... des Segments...“ durch: „... der Nachrichtenwarteschlange...“.

Seite 202, 4. Zeile von unten

Ersetze „Konversion“ durch „Konvertierung“.

Seite 203, 3. Zeile des Fließtextes

Ersetze „Zeilen“ durch „Spalten“.

Seite 204, letzte Zeile

Füge „fork“ nach „Funktion“ ein.

Seite 208, Listing 9.4, Zeile 10 im Quellcode

Eine ausführliche Erklärung zu `mkfifo` und den Zugriffsrechten wäre an der Stelle im Buch sinnvoll gewesen, da auf den ersten Blick die Zugriffsrechte der benannten Pipe nicht zum Quellcode passen.

In Listing 9.4 wird mit `mkfifo` eine benannte Pipe `testfifo` angelegt. Als Zugriffsrechte sind 0666 definiert. Die führende 0 kann hier ignoriert werden. Sie ist ein Platzhalter für das sogenannte Sticky-Bit, das Setgid Bit und das Setuid Bit. Diese erweiterten Dateirechte kommen eher selten zum Einsatz und spielen im Kontext von Listing 9.4 keine Rolle. Die Bedeutung der führenden Null bei der Oktalnotation mit vier Ziffern kann also hier ignoriert werden.

Die führende 0 kann hier ignoriert werden. Die Zugriffsrechte der resultierende Pipe sind auf Seite 207 aber in der symbolischen Notation mit **rw-r--r--** angegeben, was in Oktalnotation 644 entspricht. Auf Ubuntu-basierten Systemen wird das Ergebnis hingegen in der symbolischen Notation **rw-rw-r--** sein, was in Oktalnotation 664 entspricht. Auch ganz andere Ergebnisse sind je nach verwendetem Betriebssystem und vorgenommenen Einstellungen möglich.

Der Grund dafür ist, dass auf dem System die mit **umask** („Dateierzeugungsmaske“) gesetzten Zugriffsrechte entfernt („maskiert“) werden. Die Standardeinstellung von **umask** hängt vom verwendeten Betriebssystem ab und kann vom Systemadministrator verändert werden. Die **umask**-Standardwerte der Linux-Distributionen Debian sind Ubuntu sind z.B. 0022 bzw. 0002.

Die aktuell eingestellte Dateierzeugungsmaske kann durch einen Aufruf des Kommandos **umask** ohne Parameter in der Kommandozeile ausgegeben werden:

```
$ umask  
0022
```

Hat **umask** den Wert 0022 (auch hier kann die führende 0 ignoriert werden) sind die Zugriffsrechte der benannte Pipe aus Listing 9.6 **rw-r--r--**. Die Berechnung ist wie folgt:

Definierte Zugriffsrechte in mkfifo in Listing 9.6:	rw-rw-rw-	(666)
Abzug durch umask auf einem System mit Debian-Linux:	----w--w-	(022)
Ergebnis (Zugriffsrechte der benannten Pipe):	rw-r--r--	(644)

Auf einem System, bei dem **umask** den Wert 0002 hat, sind die Zugriffsrechte der benannten Pipe dementsprechend **rw-rw-r--**.

Weitere Informationen zum Thema **umask** und Zugriffsrechte sind u.a. hier zu finden:

- <https://wiki.ubuntuusers.de/umask>
- https://www.debian.org/doc/manuals/debian-reference/ch01.en.html#_control_of_permissions_for_newly_created_files_umask

Seite 210, 6. Zeile von oben

Ersetze

„Das Kommando **lsof** gibt in einem Linux-Betriebssystem eine Liste aller existierenden und von mindestens einem Prozess verwendeten benannten Pipes aus.“

durch

„Das Kommando `lsof` gibt in einem Linux-Betriebssystem eine Liste aller aktuell offenen Dateien, also auch die existierenden benannten Pipes aus.“

Seite 211, 21. Zeile von oben

Ersetze „Sender und Client“ durch „Server und Client“.

Seite 213, 23. Zeile von Listing 9.5

Ersetze „Datainamen“ durch „Dateinamen“.

Seite 227, 8. Zeile von unten

Nach dem letzten Wort „angesprochen“ fehlt der Punkt (Satzzeichen).

Seite 231, 10. Zeile von oben

Ersetze „dieser Werk“ durch „dieses Werk“.

Seite 233, Fußnote 1

Ersetze „emumliert“ durch „emuliert“.

Seite 233, Fußnote 7

Ersetze „emumliert“ durch „emuliert“.

Seite 234, 3. Zeile von unten

Entferne „eine“.

Seite 236, letzte Zeile des ersten Absatzes

Ersetze „in Ring 1 abgefangen“ durch „in Ring 0 abgefangen“.

Seite 243, Bildunterschrift von Abbildung 10.7

Ersetze

„Bei Hardware-Virtualisierung läuft der Hypervisor im neuen Ring -1 und das Host-Betriebssystem wieder in Ring 1“

durch

„Bei Hardware-Virtualisierung läuft der Hypervisor im neuen Ring -1 und das Host-Betriebssystem wieder in Ring 0“

Seite 243, Glossar

Ersetze

„..., in der das Betriebssystem Teile des Hauptprozessors auslagert“

durch

„..., in den das Betriebssystem diejenigen Prozesse auslagert, die gegenwärtig keinen Zugriff auf einen Prozessor bzw. einen Prozessorkern haben“.

Seite 244, Glossar, Eintrag von Datensegment, 2. Zeile

Streiche „... und Konstanten“.

Seite 254, Literaturverzeichnis, 20. Eintrag

Ersetze „Grumm H“ durch „Gumm H“