Errata zur 2. Auflage von **Betriebssysteme kompakt**.
Erschienen 2020 bei Springer Vieweg. ISBN: 978-3-662-61410-5

Seite 23, 3. und 4. Zeile von Abschnitt 3.4

Ersetze „Mehrprogrammbetrieb“ durch „Mehrbenutzerbetrieb“.

Seite 27, 3. Zeile unter Tabelle 3.2

Ersetze „Mehrprogrammbetrieb“ durch „Mehrbenutzerbetrieb“.

Seite 56, Abschnitt 4.4.4, 1. Zeile

„Festplatten sind pro Bit. . . “

Im Buch fehlt das Wort „sind“.

Seite 58, 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 59, Formel am Ende von Abschnitt 4.4.4

Ersetze die Formel durch:

$$\text{Durchschnittliche Zugriffsverzögerung durch Umdrehung [ms]} =$$

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

Seite 80, 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 88, 2. und 3. Zeile von Abschnitt „Organisation und Adressierung des Speichers im Real Mode“

Ersetze

„Im Real Mode wird der verfügbare Speicher in gleich große Segmente unterteilt. Die Speicheradressen sind 16 Bits lang. Jedes Segment ist dementsprechend 64 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 94, 3. Zeile von unten

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

Seite 102, 4. Zeile von unten

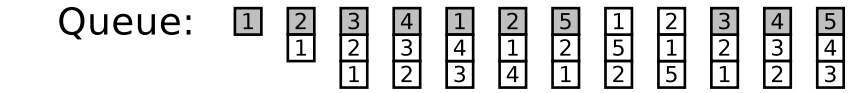
„. . . Speicherschutz mehr bietet ist wegen des. . . “.

Im Buch fehlt das Wort „ist“.

Seite 108, Abbildung 5.20

Die Positionierung der Seiteninhalte in Abbildung 5.20 ist verwirrend, weil so der Eindruck entsteht, das 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, das 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



Seite 110, rechte Spalte, Abschnitt 5.3.5, 2. Zeile

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

Seite 127, Tabellenüberschrift von Tabelle 6.3

Streiche „Dateigröße und“.

Seite 127, Kopfzeile der zweiten Spalte Tabelle 6.3

Streiche „Minimale“.

Seite 128, 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 128, Kopfzeile der zweiten Spalte Tabelle 6.4

Streiche „Minimale“.

Seite 131, 10. bis 12. Zeile des 2. Abschnitts

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

Seite 133, letzte Zeile

Ersetze „Abb. 6.8“ durch „Abb. 6.10“.

Seite 137, 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 137, Kopfzeile der zweiten Spalte Tabelle 6.5

Streiche „Minimale“.

Seite 150, Abschnitt 8.1

Ersetze:

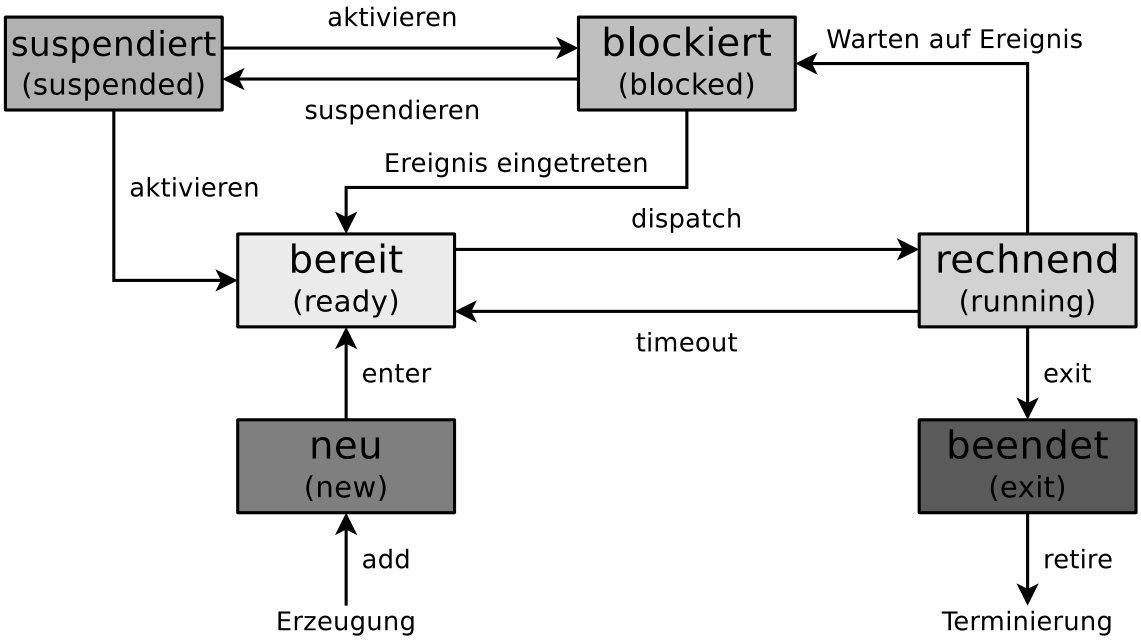
- Information über Eltern- oder Kindprozesse

Durch:

- Elternprozessnummer (PPID)

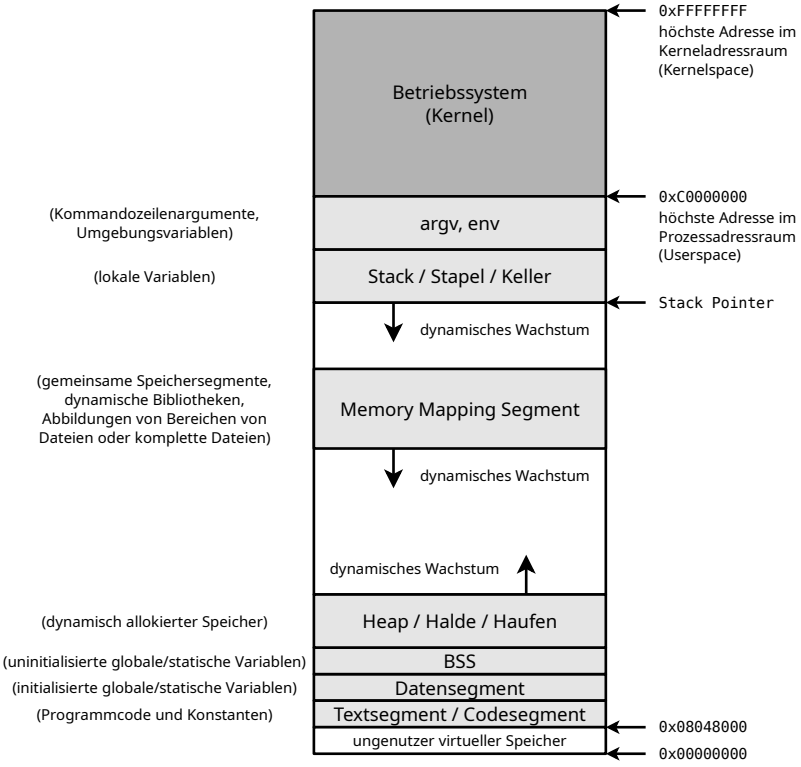
Seite 156, Abbildung 8.8

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



Seite 158-160, 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 allokieren (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 170, vorletzte Zeile

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

Seite 198, 6. Zeile von oben

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

Seite 198, 8. Zeile von oben

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

Seite 208, 2. Zeile von oben

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 212, Listing 9.2, Zeile 49 im Quellcode

Ersetze „`printf(Äus der Nachrichtenwarteschlange...`“

durch „`printf("Aus der Nachrichtenwarteschlange...`“.

Seite 216, 2. Zeile von unten

Ersetze „Konversion“ durch „Konvertierung“.

Seite 225, 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 224 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 229, 2. Zeile des vorletzten Absatzes

Ersetze

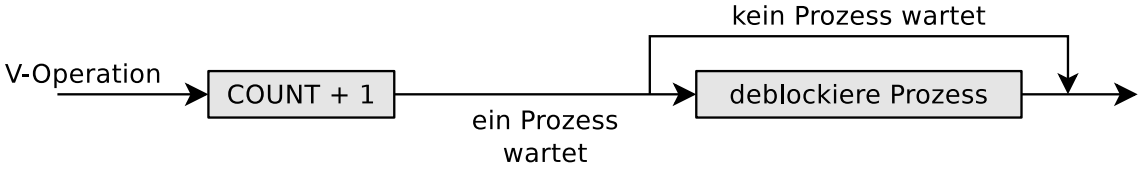
„Das Kommando `lsof` gibt in einem Linux-Betriebssystem eine Liste aller existierenden und von mindestens einem Prozess verwendeten benannten Pipes aus.“

durch

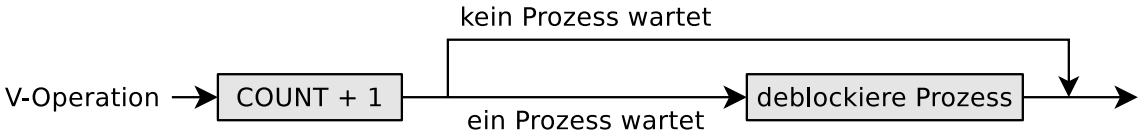
„Das Kommando `lsof` gibt in einem Linux-Betriebssystem eine Liste aller aktuell offenen Dateien, also auch die existierenden benannten Pipes aus.“

Seite 246, Abbildung 9.12

Der Pfad „kein Prozess wartet“ zweigt zu spät nach oben ab.



Bessere Darstellung:



Seite 258, vorletzte Zeile

Ersetze „Ring 1“ durch „Ring 0“.

Seite 265, Glossar, zweiter Eintrag

Ersetze „Adressbus“durch „AES“.

Seite 266, Glossar, Eintrag von Datensegment, 2. Zeile

Streiche „. . . und Konstanten“.