

Errata zur 2. Auflage von **Operating Systems / Betriebssysteme – Bilingual Edition / Zweisprachige Ausgabe**.

Erschienen 2023 bei Springer Vieweg. ISBN: 978-3-658-42229-5

Seite 23, 4. und 5. Zeile von Abschnitt 3.4

Ersetze „Mehrprogrammbetrieb“ durch „Mehrbenutzerbetrieb“.

Seite 26, 3. und 4. Zeile von unten

Ersetze „Mehrprogrammbetrieb“ durch „Mehrbenutzerbetrieb“.

Seite 35, Abbildung 3.10

Ersetze jeweils „Inter Prozess Communication“ durch „Interprocess Communication“.

Seite 35, Bildunterschrift von Abbildung 3.10

Ersetze „or“ durch „of“.

Seite 55, rechte Spalte, Abschnitt 4.4.5, 1. Zeile

„Festplatten sind pro Bit. . .“

Im Buch fehlt das Wort „sind“.

Seite 92, rechte Spalte, 6. Zeile von unten

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

Seite 100, rechte Spalte, 6. Zeile von unten

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

Im Buch fehlt das Wort „ist“.

Seite 108, rechte Spalte, Abschnitt 5.3.5, 2. Zeile

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

Seite 133, linke Spalte, 6. Zeile von unten

Ersetze „Figure 6.9“ durch „Figure 6.11“.

Seite 133, rechte Spalte, 6. Zeile von unten

Ersetze „Abbildung 6.9“ durch „Abbildung 6.11“.

Seite 131, rechte Spalte, 15. und 16. 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 154, linke Spalte, Abschnitt 8.1

Ersetze:

- Information about parent or child processes

Durch:

- Parent Process ID (PPID)

Seite 154, rechte Spalte, Abschnitt 8.1

Ersetze:

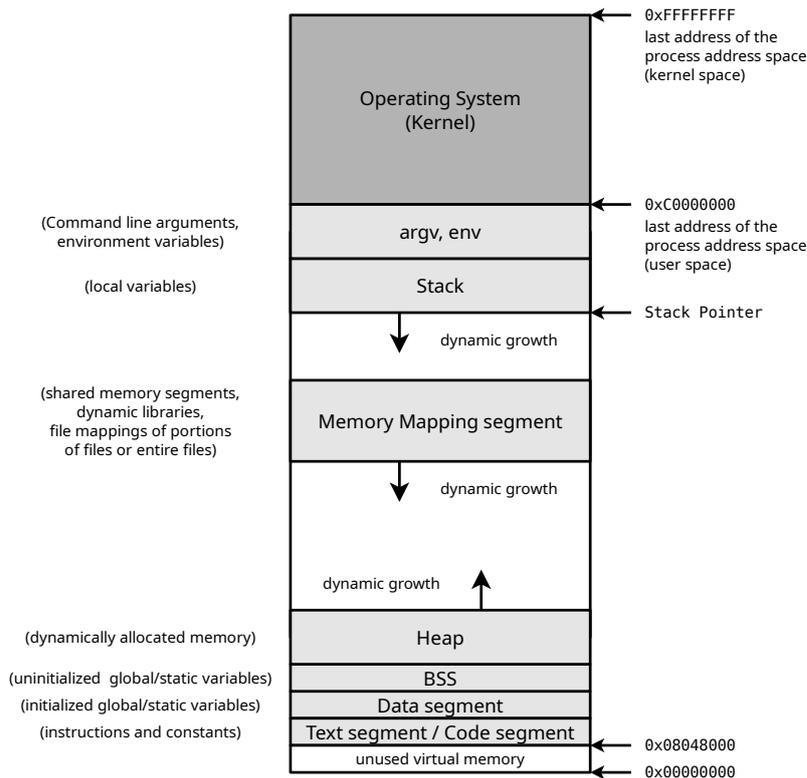
- Information über Eltern- oder Kindprozesse

Durch:

- Elternprozessnummer (PPID)

Seite 162-164, Abschnitt 8.3 (inkl. Abbildung 8.12)

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 3 des Buches.

Seite 173, rechte Spalte, vorletzte Zeile

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

Seite 234, Listing 9.6, Zeile 11 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 9.6 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 9.6 keine Rolle. Die Bedeutung der führenden Null bei der Oktalnotation mit vier Ziffern kann also hier ignoriert werden.

Die Zugriffsrechte der resultierende Pipe sind auf Seite 234 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 und 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` sind die Zugriffsrechte der benannte Pipe aus Listing 9.6 `rw-r--r--`. Die Berechnung ist wie folgt:

Definierte Zugriffsrechte in <code>mkfifo</code> in Listing 9.6:	<code>rw-rw-rw-</code>	(666)
Abzug durch <code>umask</code> auf einem System mit Debian-Linux:	<code>----w--w-</code>	(022)
Ergebnis (Zugriffsrechte der benannten Pipe):	<code>rw-r--r--</code>	(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 290, Glossar, linke Spalte, Eintrag von Data Segment 3. Zeile

Streiche „... and constants“.

Seite 290, Glossar, rechte Spalte, Eintrag von Datensegment, 2. und 3. Zeile

Streiche „... und Konstanten“.