

Lösung von Übungsblatt 1

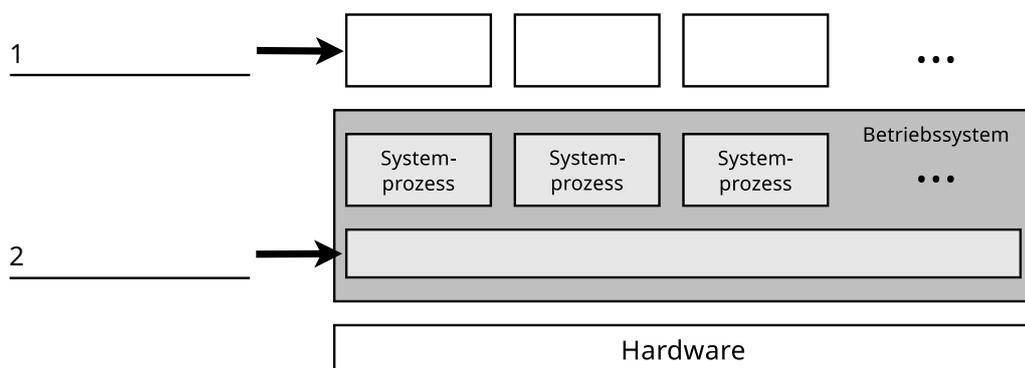
Die Übungsblätter der Vorlesung Betriebssysteme enthalten theoretische und praktische Übungen. Für die Bearbeitung der praktischen Übungen benötigen Sie eine UNIX-Shell. Weit verbreitet sind u.a. die `bash` (Bourne-Again SHell), `fish` (Friendly Interactive Shell), `ksh` (KornShell), `csch` (C Shell), `tcsh` und `zsh` (Z Shell). Das Apple Mac OS X Terminal reicht für die meisten praktischen Übungen aus. Mac OS X von Version 10.3 bis 10.14 verwendet standardmäßig die `bash` und neuere Versionen verwenden die `zsh`. Die Windows Eingabeaufforderung und Windows PowerShell sind für die Übungsaufgaben nicht ausreichend! Die Installation der `bash` oder einer vergleichbaren Shell unter Windows 10 und Windows 11 ist aber problemlos möglich.

Optimal ist, wenn Sie auf Ihrem System zur Vorbereitung des Betriebssystems Linux installieren. Eine Installation in einem Container oder in einer virtuellen Maschine ist ausreichend. Einfach zu bedienende Linux-Distributionen sind z.B. Debian, Ubuntu, Manjaro, Mint, Pop!_OS, Elementary OS oder Fedora.

Auf einem System mit Windows-Betriebssystem ist in vielen Fällen das Windows-Subsystem for Linux (WSL) die beste Wahl zur Installation eines Linux-Gastsystems. Wenn das nicht funktioniert, ist die freie Virtualisierungslösung VirtualBox eine gute Wahl. VirtualBox auch für Systeme mit Mac OS X verfügbar. Eine freie alternative Virtualisierungslösung für Geräte mit Mac OS X ist UTM.

Aufgabe 1 (Aufbau eines Betriebssystems)

1. Die Abbildung zeigt den grundsätzlichen Aufbau eines Betriebssystems. Schreiben Sie auf die beiden Linien die Namen der mit Pfeilen gekennzeichneten Komponenten.



Aufgabe 2 (Stapel- bzw. Batchbetrieb)

1. Beschreiben Sie das Ziel des Stapelbetriebs.

Maximierung der Prozessorausnutzung.

2. Beschreiben Sie warum Stapelbetrieb zu einer Beschleunigung führt, wenn mehrere Aufgaben abgearbeitet werden.

Wegen der Automatisierung. Es geht keine Zeit für den Benutzerwechsel verloren.

3. Geben Sie die Voraussetzung an, die bei Stapelbetrieb erfüllt sein muss, bevor mit der Abarbeitung einer Aufgabe begonnen werden kann.

Jedes Programm muss (mit allen Eingabedaten!) vollständig vorliegen, bevor die Abarbeitung beginnen kann.

4. Geben Sie an, für welche Aufgaben sich Stapelbetrieb besonders gut eignet.

Stapelbetrieb eignet sich gut zur Ausführung von Routineaufgaben.

5. Stapelbetrieb ist immer. . .

interaktiv interaktionslos

6. Nennen Sie eine Anwendung des Stapelbetriebs, die heute noch populär ist.

Batch-Dateien, Shell-Skripte, usw.

7. Beschreiben Sie was Spooling ist.

Spooling ist die Entlastung des Hauptprozessors durch zusätzliche Hardware für Ein-/Ausgabeoperationen.

Aufgabe 3 (Dialogbetrieb – Time-Sharing)

1. Beschreiben Sie das Ziel des Dialogbetriebs (*Time-Sharing*).

Faire Verteilung der Rechenzeit zwischen den Benutzern.

2. Beschreiben Sie wie beim Dialogbetrieb (*Time-Sharing*) die Verteilung der Rechenzeit geschieht.

Mit Zeitscheiben (Time Slices).

3. Geben Sie den Fachbegriff der quasi-parallele Programm- bzw. Prozessausführung an.

Mehrprogrammbetrieb oder Multitasking.

4. Beschreiben Sie das Ziel der quasi-parallelen Programm- bzw. Prozessausführung.

Minimierung der Antwortzeit.

5. Beschreiben Sie was Scheduling ist.

Automatische Erstellung eines Ablaufplanes (schedule), der Benutzern bzw. Prozessen zeitlich begrenzte Ressourcen zuteilt.

6. Beschreiben Sie was Swapping ist.

Prozess des Ein- und Auslagerns von Daten in den/vom Arbeitsspeicher vom/in den Hintergrundspeicher (Festplatten/SSDs).

7. Beschreiben Sie wie Speicherschutz funktioniert.

Der Arbeitsspeicher wird aufgeteilt und laufende Programme voneinander getrennt.

8. Beschreiben Sie den Zweck des Speicherschutzes es.

Ein Programmierfehler oder Absturz eines einzelnen Programms kann nicht die Stabilität anderer Programme und des Gesamtsystems beeinträchtigen.

Aufgabe 4 (Dateien und Verzeichnisse)

1. Erzeugen Sie in Ihrem Benutzerverzeichnis (Home-Verzeichnis) ein Verzeichnis BTS.

```
$ mkdir ~/BTS
```

2. Wechseln Sie in das Verzeichnis BTS und erzeugen Sie darin eine leere Datei mit dem Dateinamen `Datei1.txt`.

- Verwenden Sie zum Anlegen der Datei keinen Editor, sondern einen Kommandozeilenbefehl.

```
$ cd ~/BTS && touch Datei1.txt
```

3. Überprüfen Sie die Dateigröße der Datei `Datei1.txt`.

```
$ ls -lh Datei1.txt
```

4. Ändern Sie die Modifikationszeit der Datei `Datei1.txt` auf Ihr Geburtsdatum.

```
touch -t XXXXYYZZAABB Datei1.txt
```

XXXX enthält das Jahr.
YY enthält den Monat.
ZZ enthält den Tag des Monats.
AA enthält die Stunde.
BB enthält die Minute.

5. Erzeugen Sie auf der Shell eine neue Datei `Datei2.txt` und fügen Sie in die neue Datei einen beliebigen mehrzeiligen Text als Inhalt ein.

- Verwenden Sie für das Einfügen des Textes keinen Editor, sondern einen Kommandozeilenbefehl.

```
$ echo -e "Zeile1\nZeile2" > Datei2.txt
```

6. Geben Sie die erste Zeile der Datei `Datei2.txt` aus.

```
$ head -n 1 Datei2.txt
```

7. Hängen den Inhalt von `Datei2.txt` an `Datei1.txt` an.

- Verwenden keinen Editor, sondern einen Kommandozeilenbefehl.

```
$ cat Datei2.txt >> Datei1.txt
```

8. Erzeugen Sie in Ihrem Home-Verzeichnis ein Verzeichnis mit dem Verzeichnisnamen `BTS_neues_Semester`.

```
$ mkdir ~/BTS_neues_Semester
```

9. Kopieren Sie die Dateien `Datei1.txt` und `Datei2.txt` aus dem Verzeichnis `BTS` in das Verzeichnis `BTS_neues_Semester`.

```
$ cp ~/BTS/* ~/BTS_neues_Semester
```

10. Löschen Sie das Verzeichnis `BTS`.

```
$ rm -rf ~/BTS
```