

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 ist die Bash [1]. Das Apple Mac OS X Terminal reicht für die meisten praktischen Übungen aus. Die Windows Eingabeaufforderung und Windows PowerShell sind für die Übungsaufgaben nicht ausreichend.

Optimal ist, wenn Sie auf Ihrem System zur Vorbereitung das Betriebssystem Linux installieren. Eine Installation in einer virtuellen Maschine ist ausreichend. Einfach zu bedienende Distributionen sind z.B. Debian [2], Ubuntu [3] und Mint [4]. Eine freie Virtualisierungslösung ist VirtualBox [5]

Alternativ können Sie mit einem Live-Systeme auf CD, DVD oder USB-Laufwerk arbeiten. Hierbei ist keine lokale Installation nötig. Eine leistungsfähige Live-Distributionen ist KNOPPIX [6].

[1] <http://tiswww.case.edu/php/chet/bash/bashtop.html>

[2] <http://www.debian.org>

[3] <http://www.ubuntu.com>

[4] <http://www.linuxmint.com>

[5] <http://www.virtualbox.org>

[6] <http://www.knopper.net/knoppix>

Aufgabe 1 (Stapel- bzw. Batchbetrieb)

1. Was ist das Ziel des Stapelbetriebs?

Maximierung der Prozessorausnutzung.

2. Warum führt Stapelbetrieb zu einer Beschleunigung, wenn mehrere Aufgaben abgearbeitet werden?

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

3. Welche Voraussetzung muss bei Stapelbetrieb erfüllt sein, bevor mit der Abarbeitung einer Aufgabe begonnen werden kann

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

4. Für welche Aufgaben eignet sich Stapelbetrieb besonders gut?

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. Was ist Spooling?

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

Aufgabe 2 (Dialogbetrieb – Time-Sharing)

1. Was ist das Ziel des Dialogbetriebs (*Time-Sharing*)?

Faire Verteilung der Rechenzeit zwischen den Benutzern.

2. Wie geschieht beim Dialogbetrieb (*Time-Sharing*) die Verteilung der Rechenzeit?

*Mit Zeitscheiben (*Time Slices*).*

3. Was ist das Ziel der quasi-parallelen Programm- bzw. Prozessausführung?

Mehrprogrammbetrieb oder Multitasking.

4. Was ist das Ziel der quasi-parallelen Programm- bzw. Prozessausführung?

Minimierung der Antwortzeit.

5. Was ist Scheduling?

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

6. Was ist Swapping?

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

7. Wie funktioniert Speicherschutz?

Der Arbeitsspeicher wird aufgeteilt und laufende Programme voneinander getrennt.

8. Was ist der Zweck des Speicherschutzes?

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

Aufgabe 3 (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
```