

Lösung von Übungsblatt 1

Die Übungsblätter 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], Manjaro [4] und Mint [5]. Eine freie Virtualisierungslösung ist VirtualBox [6]. Unter Windows können Sie einfach das Windows Subsystem für Linux [7] verwenden.

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 [8].

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

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

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

[4] <https://manjaro.org>

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

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

[7] <https://learn.microsoft.com/en-us/windows/wsl/>

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

Aufgabe 1 (Stapel- bzw. Batchbetrieb)

1. Nennen Sie ein 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 2 (Dialogbetrieb – Time-Sharing)

1. Nennen Sie ein 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. Nennen Sie den Fachbegriff der quasi-parallelen Programm- bzw. Prozessausführung.

Mehrprogrammbetrieb oder Multitasking.

4. Nennen Sie ein 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.

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

Aufgabe 3 (Betriebssysteme – Klassifikationen)

1. Zu jedem Zeitpunkt kann nur ein einziges Programm laufen. Nennen Sie den passenden Fachbegriff für diese Betriebsart.

Einzelprogrammbetrieb (Singletasking).

2. Beschreiben Sie was ein halbes Multi-User-Betriebssystemen ausmacht.

Verschiedene Benutzer können nur nacheinander am System arbeiten, aber die Daten und Prozesse der Benutzer sind voreinander geschützt.

3. Beschreiben Sie den Unterschied zwischen 8 Bit-, 16 Bit-, 32 Bit- und 64 Bit-Betriebssystemen.

Die Bit-Zahl gibt die Länge der Speicheradressen an, mit denen das Betriebssystem intern arbeitet.

4. Beschreiben Sie den Aufbau eines monolithischen Kernels.

Monolithische Kerne enthalten...

- *Funktionen für Speicherverwaltung*
- *Funktionen für Prozessverwaltung*
- *Funktionen für Prozesskommunikation*
- *Gerätetreiber*
- *Dateisysteme-Treiber*

Außerhalb des Kerns befinden sich die Benutzerprozesse.

5. Beschreiben Sie den Aufbau eines minimalen Kerns (Mikrokernels).

Im Kern befinden sich üblicherweise nur:

- *Notwendigste Funktionen zur Speicher- und Prozessverwaltung*
- *Funktionen zur Synchronisation und Interprozesskommunikation*

Gerätetreiber, Dateisysteme und Dienste (Server), befinden sich außerhalb des Kerns und laufen wie die Anwendungsprogramme im Benutzermodus

6. Beschreiben Sie den Aufbau eines hybriden Kernels.

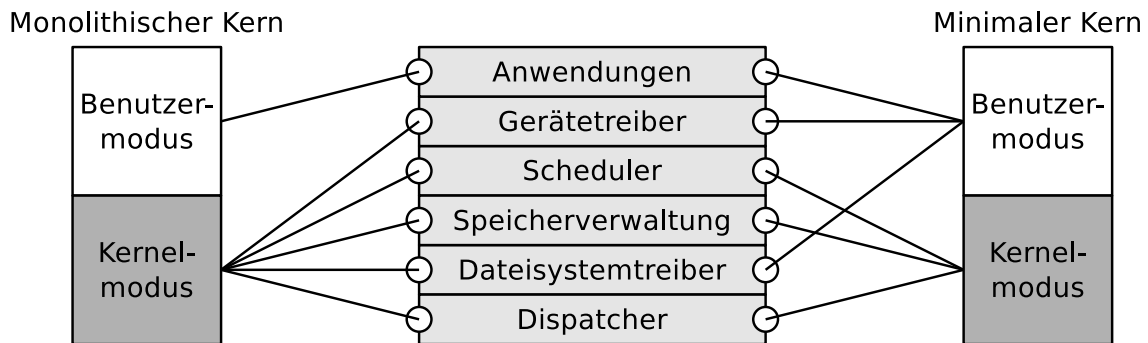
Hybride Kerne sind ein Kompromiss zwischen monolithischen Kernen und minimalen Kernen. Die enthalten aus Geschwindigkeitsgründen Komponenten,

die bei minimalen Kernen außerhalb des Kerns liegen. Es ist nicht festgelegt, welche Komponenten bei Systemen mit hybriden Kernen zusätzlich in den Kernel einkompiliert sind.

7. Nennen Sie einen Vorteil und einen Nachteil von monolithischen Kernen.
- Vorteile:
 - Weniger Kontextwechsel als Mikrokernel \implies höhere Geschwindigkeit
 - Gewachsene Stabilität
 - Nachteile:
 - Abgestürzte Komponenten können im Kernel nicht separat neu gestartet werden und das gesamte System nach sich ziehen
 - Hoher Entwicklungsaufwand für Erweiterungen am Kern, da dieser bei jedem Kompilieren komplett neu übersetzt werden muss
8. Nennen Sie einen Vorteil und einen Nachteil von minimalen Kernen (Mikrokernen).
- Vorteile:
 - Einfache Austauschbarkeit der Komponenten
 - Beste Stabilität und Sicherheit (in der Theorie!), weil weniger Funktionen im Kernelmodus laufen
 - Nachteile:
 - Langsamer wegen der größeren Zahl von Kontextwechseln
 - Entwicklung eines neuen (Mikro-)kernels ist eine komplexe Aufgabe
9. Nennen Sie einen Vorteil und einen Nachteil von hybriden Kernen.
- Vorteile:
 - Höhere Geschwindigkeit als minimale Kerne (da weniger Kontextwechsel)
 - Bessere Stabilität (in der Theorie!) als monolithische Kerne
 - Nachteile:
 - Entwicklung eines neuen (Hybrid-)kernels ist eine komplexe Aufgabe
10. Ein Kollege empfiehlt Ihnen häufig verwendete Server-Dienste wie z.B. Web-Server, Email-Server, SSH-Server und FTP-Server vom Benutzermodus in den Kernelmodus zu verlagern. Wie stehen Sie zu dieser Idee? Begründen Sie Ihre Antwort. Nennen Sie hierfür einen Vorteil und einen Nachteil.
- Von Vorteil wäre, das das Betriebssystem und die Server-Dienste insgesamt schneller arbeiten, weil im beschriebenen Szenario weniger Moduswechsel zwischen Benutzermodus und Kernelmodus nötig sind.
- Gravierender ist aber der entstehende Nachteil. Es liegt ein Sicherheitsrisiko vor. Komplexe Software wie Server-Dienste sollten nicht im Kernelmodus laufen. Softwarefehler in den Server-Diensten könnten zu Systemabstürzen oder zur vollständigen Kontrollübernahme durch Angreifer führen.

Aufgabe 4 (Kernelarchitekturen)

Die mittlere Spalte des Bildes enthält Funktionen eines Betriebssystems. Zeichnen Sie von jeder Komponente eine Linie nach links und eine Linie nach rechts, um sowohl für monolithische Kernel als auch für Mikrokern anzugeben, ob die Komponente zum Kernelmodus oder zum Benutzermodus gehört.



Aufgabe 5 (Grundlegende Linux-Kommandos)

Geben Sie ein Kommando an, mit dem Sie...

1. Handbuchseiten („Man Pages“) öffnen.

man

2. das aktuelle Verzeichnis in der Shell ausgeben.

pwd

3. ein neues Verzeichnis erzeugen.

mkdir

4. in ein Verzeichnis wechseln.

cd

5. den Inhalt eines Verzeichnisses in der Shell ausgeben.

ls

6. eine leere Datei erzeugen.

touch

7. versuchen den Inhalt einer Datei zu bestimmen.

file

8. den Inhalt von Dateien verknüpfen oder den Inhalt einer Datei ausgeben.

cat

9. Zeilen vom Ende einer Datei in der Shell ausgeben.

tail

10. Zeilen vom Anfang einer Datei in der Shell ausgeben.

head

11. Dateien oder Verzeichnisse an eine andere Stelle kopieren.

cp

12. Dateien oder Verzeichnisse an eine andere Stelle verschieben.

mv

13. Dateien oder Verzeichnisse löschen.

rm

14. ein leeres Verzeichnis löschen.

rmdir

15. eine Zeichenkette in der Shell ausgeben.

echo

16. die Dateirechte von Dateien oder Verzeichnissen ändern.

chmod

17. Das Password eines Benutzers ändern.

passwd

18. die laufende Sitzung (und damit auch die Shell) beenden und den Rückgabewert eines Shell-Skripts festlegen.

exit

19. das System neu starten.

reboot oder alternativ shutdown

20. das System ausschalten.

halt oder alternativ shutdown

21. einen neuen Benutzer erstellen.

adduser

22. einen Benutzer löschen.

deluser

23. einen Benutzer ändern.

usermod

24. die Gruppenzugehörigkeiten des Benutzers ausgeben.

groups

25. eine neue Gruppe erstellen.

groupadd

26. eine Gruppe löschen.

groupdel

27. eine Gruppe ändern.

groupmod

28. den Benutzer (\implies Besitzer) einer Datei oder eines Verzeichnisses ändern.

chown

29. die Gruppenzuordnung einer Datei oder eines Verzeichnisses ändern.

chgrp

30. einen „Link“ erstellen.

ln

31. eine Datei nach den Zeilen durchsuchen, die ein Suchmuster enthalten.

grep

32. eine Liste der laufenden Prozesse in der Shell ausgeben.

ps

33. einen im Hintergrund laufenden Prozess in den Vordergrund holen.

fg

34. einen Prozess in den Hintergrund verschieben.

bg

35. einen Prozess beenden.

kill

36. eine Gruppe von Prozessen beenden.

killall

37. die Priorität eines neuen Prozesses festlegen.

nice

38. die Priorität eines existierenden Prozesses ändern.

renice

39. eine Liste der existierenden Prozesse als Baum in der Shell ausgeben.

ps tree

Aufgabe 6 (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 `BSRN_Vorlesung`.

```
$ mkdir ~/BSRN_Vorlesung
```

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

```
$ cp ~/BTS/* ~/BSRN_Vorlesung
```

10. Löschen Sie das Verzeichnis `BTS`.

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