

Name:

Vorname:

Matr.Nr.:

Aufgabe 1)

Punkte:

Maximale Punkte: 1+1+1+1+2=6

- a) Für welche Aufgaben eignet sich Stapelbetrieb besonders gut?

Stapelbetrieb eignet sich gut zur Ausführung von Routineaufgaben.

- b) Was ist der Zweck des Speicherschutzes?

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

- c) Was ist der 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.

- d) Es gibt zwei Arten von Echtzeitbetriebssystemen. Geben Sie deren Namen an.

Harte Echtzeitsysteme und weiche Echtzeitsysteme.

- e) 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 mindestens einen Vorteil und einen Nachteil.

Von Vorteil wäre, dass 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.

Name:

Vorname:

Matr.Nr.:

Aufgabe 2)

Punkte:

Maximale Punkte: 2+3=5

- a) Erklären Sie die beiden Dateieinträge „.“ und „..“ in der Ausgabe von `ls`?

```
$ mkdir new_directory
$ cd new_directory
$ ls -l --all --size --human-readable
insgesamt 8,0K
4,0K drwxr-xr-x  2 bnc users 4,0K Jul 12 11:03 .
4,0K drwxr-xr-x 119 bnc users 4,0K Jul 12 11:03 ..
```

Der Punkt „.“ ist ein Hard-Link, der auf den Ordner `new_directory` zeigt und „..“ ist ein Hard-Link, der auf den Ordner eine Ebene höher in der Verzeichnisstruktur zeigt.

- b) Erklären Sie die Dateirechte der Datei `convert_script.py`.

```
$ ls -l --all --size --human-readable
insgesamt 16K
4,0K drwxr-xr-x  2 bnc users 4,0K Jul 12 09:14 .
4,0K drwxr-xr-x 119 bnc users 4,0K Jul 12 09:13 ..
8,0K -rwxr-xr--  1 bnc users 7,0K Jul 12 09:22 convert_script.py
```

(Hinweis: Beschreiben Sie, welche Aktionen die verschiedenen Benutzer/Benutzergruppen mit der Datei durchführen dürfen.)

Der Benutzer `bnc` darf auf die Datei lesend und schreibend zugreifen und er darf sie ausführen.

Alle Benutzer, die Mitglied in der Gruppe `users` sind, dürfen auf die Datei lesend zugreifen und diese ausführen.

Alle übrigen Benutzer dürfen auf die Datei lesend zugreifen.

Name:

Vorname:

Matr.Nr.:

Aufgabe 3)

Punkte:

Maximale Punkte: $1+0.5+0.5+1+1+1+1=6$

- a) Nennen Sie die beiden grundsätzlichen Cache-Schreibstrategien.

Write-Through und Write-Back.

- b) Bei welcher Cache-Schreibstrategie aus Teilaufgabe a) kann es zu Inkonsistenzen kommen?

Write-Back.

- c) Bei welcher Cache-Schreibstrategie aus Teilaufgabe a) ist die System-Geschwindigkeit geringer?

Write-Through.

- d) Bei welcher Cache-Schreibstrategie aus Teilaufgabe a) kommen sogenannte „Dirty Bits“ zum Einsatz?

Write-Back.

- e) Was ist die Aufgabe der „Dirty Bits“?

Für jede Seite im Cache wird ein Dirty Bit im Cache gespeichert, das angibt, ob die Seite geändert wurde.

- f) Welche Faktoren beeinflussen die Zugriffszeit einer Festplatte?

- Suchzeit („Average Seek Time“)
- Zugriffsverzögerung durch Umdrehung („Average Rotational Latency Time“)

- g) Beschreiben Sie die Faktoren aus Teilaufgabe f).

- Suchzeit („Average Seek Time“): Die Zeit, die der Schwungarm braucht, um eine Spur zu erreichen.
- Zugriffsverzögerung durch Umdrehung („Average Rotational Latency Time“): Verzögerung der Drehgeschwindigkeit bis der Schreib-/Lesekopf den gewünschten Block erreicht. Hängt ausschließlich von der Drehgeschwindigkeit der Scheiben ab.

Name:

Vorname:

Matr.Nr.:

Aufgabe 4)

Punkte:

Maximale Punkte: $1+1+1+1+1+1=6$

- a) Nehmen Sie Stellung zu der Aussage: „Ein RAID-Verbund kann das regelmäßige Backup wichtiger Daten ersetzen“.

RAID ist kein Ersatz für Datensicherung. Fehlerhafte Dateioperationen oder Virenbefall finden auf allen Laufwerken statt.

- b) Warum ist es sinnvoll, Paritätsinformationen nicht auf einem Laufwerk zu speichern, sondern auf allen Laufwerken zu verteilen?

Jede Schreiboperation auf das RAID führt zu Schreiboperationen auf das Paritätslaufwerk \implies Flaschenhals.

- c) Welche Nettokapazität hat ein RAID-0-Verbund?

Die Nettokapazität ist n , wobei n die Anzahl der Laufwerke ist.

- d) Welche Nettokapazität hat ein RAID-1-Verbund?

Die Nettokapazität entspricht der Kapazität des kleinsten Laufwerks.

- e) Welche Nettokapazität hat ein RAID-5-Verbund?

Die Nettokapazität ist $n - 1$, wobei n die Anzahl der Laufwerke ist.

- f) Wie funktioniert die Berechnung der Paritätsinformationen bei RAID-5?

$P(16-19) = \text{Block 16 XOR Block 17 XOR Block 18 XOR Block 19.}$

Name:

Vorname:

Matr.Nr.:

Aufgabe 5)

Punkte:

Maximale Punkte: $3+1+1+1+1+0.5+0.5+1=9$

- a) Was ist virtueller Speicher?

Jeder Prozess besitzt einen eigenen Adressraum. Der Adressraum ist eine Abstraktion des physischen Speichers. Es handelt sich dabei um virtuellen Speicher. Er besteht aus logischen Speicheradressen, die von der Adresse 0 aufwärts durchnummeriert sind und er ist unabhängig von der verwendeten Speichertechnologie und den gegebenen Ausbaumöglichkeiten.

- b) Erklären Sie, warum mit virtuellem Speicher der Hauptspeicher besser ausgenutzt wird.

Die Prozesse müssen nicht am Stück im Hauptspeicher liegen. Externe Fragmentierung entsteht, spielt aber keine Rolle.

- c) Was ist Mapping?

Abbilden des virtuellen Speichers auf den realen Speicher.

- d) Was ist Swapping?

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

- e) Nennen Sie die beiden Konzepte von virtuellem Speicher.

Segmentierung und Paging.

- f) Bei welchem Konzept aus Teilaufgabe e) entsteht interne Fragmentierung?

Paging (nur in der letzten Seite eines Prozesses).

- g) Bei welchem Konzept aus Teilaufgabe e) entsteht externe Fragmentierung?

Segmentierung.

- h) Wie entsteht eine Page Fault Ausnahme (Exception)?

Ein Programm versucht auf eine Seite zuzugreifen, die nicht im physischen Hauptspeicher ist.

Name:

Vorname:

Matr.Nr.:

Aufgabe 6)

Punkte:

Maximale Punkte: $1+1+2+1+1+1=7$

a) Was ist ein absoluter Pfadname?

Ein kompletter Pfad von der Wurzel bis zum Ziel (Datei oder Verzeichnis).

b) Was ist ein relativer Pfadname?

Ein Pfad, der nicht mit der Wurzel beginnt.

c) Was ist die Dateizuordnungstabelle bzw. File Allocation Table (FAT) und welche Informationen enthält sie?

Für jeden Cluster des Dateisystems existiert in der FAT ein Eintrag mit folgenden Informationen über den Cluster:

- *Cluster ist frei oder das Medium an dieser Stelle beschädigt.*
- *Cluster ist von einer Datei belegt und enthält die Adresse des nächsten Clusters, der zu dieser Datei gehört bzw. ist der letzte Cluster der Datei.*

d) Was ist die Aufgabe des Journals bei Journaling-Dateisystemen?

Im Journal werden die Schreibzugriffe gesammelt, bevor sie durchgeführt werden.

e) Nennen Sie einen Vorteil von Journaling-Dateisystemen gegenüber Dateisystemen ohne Journal.

Nach einem Absturz müssen nur diejenigen Dateien (Cluster) und Metadaten überprüft werden, die im Journal stehen.

f) Welche Art der Datenverarbeitung wird durch Defragmentieren maximal beschleunigt?

Eine zusammenhängende Anordnung beschleunigt das fortlaufende Vorwärtslesen der Daten maximal, da keine Warte- und Suchzeiten mehr vorkommen können.

Name:

Vorname:

Matr.Nr.:

Aufgabe 7)

Punkte:

Maximale Punkte: $1,5+0,5+1+1+1+1+1+1=8$

- a) Welche drei Arten Prozesskontextinformationen speichert das Betriebssystem?

Benutzerkontext, Hardwarekontext und Systemkontext.

- b) Welche Prozesskontextinformationen sind nicht im Prozesskontrollblock gespeichert?

Der Benutzerkontext, also die Daten im zugewiesenen Adressraum (virtuellen Speicher).

- c) Warum sind nicht alle Prozesskontextinformationen im Prozesskontrollblock gespeichert?

Weil der virtuelle Speicher jedes Prozesses je nach verwendeter Architektur mehrere GB oder mehr groß sein kann. Der Benutzerkontext ist damit einfach zu groß, um ihn doppelt zu speichern.

- d) Was ist die Aufgabe des Dispatchers?

Aufgabe des Dispatchers ist die Umsetzung der Zustandsübergänge der Prozesse.

- e) Was ist die Aufgabe des Schedulers?

Er legt die Ausführungsreihenfolge der Prozesse mit einem Scheduling-Algorithmus fest.

- f) Was macht der Systemaufruf `fork()`?

Ruft ein Prozess `fork()` auf, wird eine identische Kopie als neuer Prozess gestartet.

- g) Was macht der Systemaufruf `exec()`?

Der Systemaufruf `exec()` ersetzt einen Prozess durch einen anderen.

- h) Was ist ein `cron`-Job?

Ein `cron`-Job legt die zeitgesteuerte Ausführung eines Kommandoaus oder Prozesses fest.

Name:

Vorname:

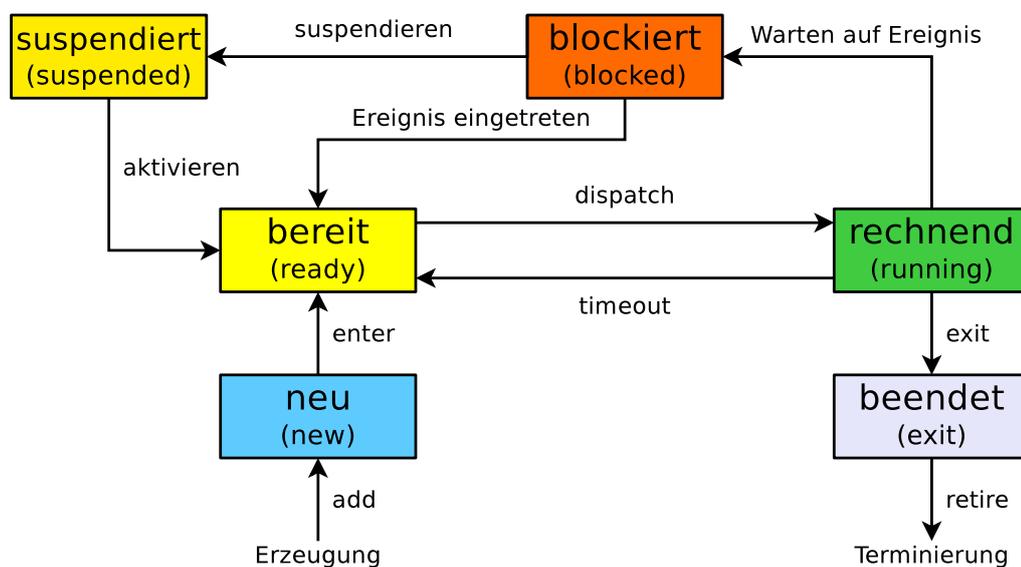
Matr.Nr.:

Aufgabe 8)

Punkte:

Maximale Punkte: 6+1=7

- a) Tragen Sie die Namen der Zustände in die Abbildung des 6-Zustands-Prozessmodells ein.



- b) Was ist ein Zombie-Prozess?

Ein Zombie-Prozess ist fertig abgearbeitet (via Systemaufruf `exit`), aber sein Eintrag in der Prozesstabelle existiert so lange, bis der Elternprozess den Rückgabewert (via Systemaufruf `wait`) abgefragt hat. Seine PID kann noch nicht an einen neuen Prozess vergeben werden.

Name:

Vorname:

Matr.Nr.:

Aufgabe 9)

Punkte:

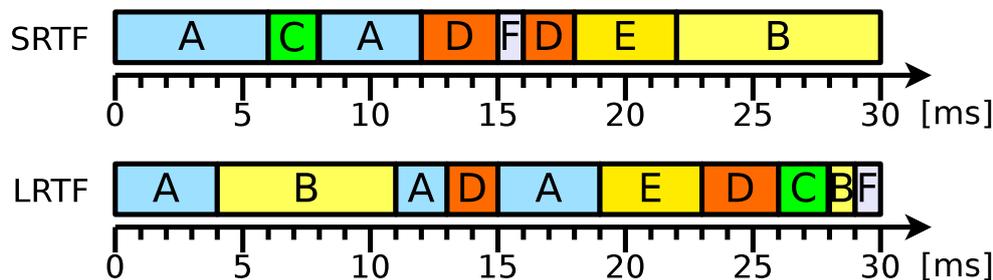
Maximale Punkte: 5+5+2+2+2+2=18

- a) Auf einem Einprozessorrechner sollen sechs Prozesse mit unterschiedlichen Ankunftszeiten verarbeitet werden.

Prozess	CPU-Laufzeit [ms]	Ankunftszeit [ms]
A	10	0
B	8	4
C	2	6
D	5	11
E	4	13
F	1	15

Skizzieren Sie die Ausführungsreihenfolge der Prozesse mit einem Gantt-Diagramm (Zeitleiste) für...

- Shortest Remaining Time First und
- Longest Remaining Time First.



- b) Berechnen Sie die mittleren Laufzeiten der Prozesse.

Shortest Remaining Time First $\frac{12+26+2+7+9+1}{6} = 9,5 \text{ ms}$

Longest Remaining Time First $\frac{19+25+22+15+10+15}{6} = 17, \bar{6} \text{ ms}$

- c) Berechnen Sie die mittleren Wartezeiten der Prozesse.

Shortest Remaining Time First $\frac{2+18+0+2+5+0}{6} = 4,5 \text{ ms}$

Longest Remaining Time First $\frac{9+17+20+10+6+14}{6} = 12, \bar{6} \text{ ms}$

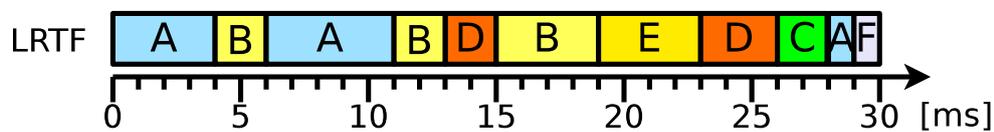
Name:

Vorname:

Matr.Nr.:

Aufgabe 9 – Fortsetzung)

Diese Lösung für LRTF wurde auch als korrekt bewertet. Dabei wird gilt im Zweifelsfall, also wenn mehrere Prozesse die gleiche Restlaufzeit bei einem Ereignis haben, immer FIFO. Auf einem Einprozessorrechner macht diese Lösung auch mehr Sinn, weil der laufende Prozess in jedem Fall unterbrochen werden muss. Immerhin braucht der Scheduler für die Überprüfung ja auch die CPU.



Mittlere Laufzeiten der Prozesse.

$$\text{Longest Remaining Time First } \frac{29+15+22+15+10+15}{6} = 17,\bar{6} \text{ ms}$$

Mittlere Wartezeiten der Prozesse.

$$\text{Longest Remaining Time First } \frac{19+7+20+10+6+14}{6} = 12,\bar{6} \text{ ms}$$

Name:

Vorname:

Matr.Nr.:

Aufgabe 10)

Punkte:

Maximale Punkte: 1+1+1+2=5

- a) Welchen Vorteil haben Signalisieren und Warten gegenüber aktivem Warten (Warteschleife)?

Bei aktivem Warten wird Rechenzeit der CPU verschwendet, weil diese immer wieder vom wartenden Prozess belegt wird. Bei Signalisieren und Warten wird die CPU entlastet, weil der wartende Prozess blockiert und zu einem späteren Zeitpunkt deblockiert wird.

- b) Was ist eine Barriere?

Eine Barriere synchronisiert die beteiligten Prozesse an einer Stelle.

- c) Welche beiden Probleme können durch Blockieren entstehen?

Verhungern (Starving) und Verklemmung (Deadlock).

- d) Was ist der Unterschied zwischen Signalisieren und Blockieren?

Signalisieren legt die Ausführungsreihenfolge der kritische Abschnitte der Prozesse fest.

Blockieren sichert kritische Abschnitte. Die Reihenfolge, in der die Prozesse ihre kritische Abschnitte abarbeiten, ist nicht festgelegt. Es wird nur sichergestellt, dass es keine Überlappung in der Ausführung der kritischen Abschnitte gibt.

Name:

Vorname:

Matr.Nr.:

Aufgabe 11)

Punkte:

Maximale Punkte: 13

In einer Lagerhalle werden ständig Pakete von einem Lieferanten angeliefert und von zwei Auslieferern abgeholt. Der Lieferant und die Auslieferer müssen dafür ein Tor durchfahren. Das Tor kann immer nur von einer Person durchfahren werden. Der Lieferant bringt mit jeder Lieferung 3 Pakete zum Wareneingang. An der Ausgabe holt ein Auslieferer jeweils 2 Pakete ab, der andere Auslieferer 1 Paket.

Es existiert genau ein Prozess `Lieferant`, ein Prozess `Auslieferer_X` und ein Prozess `Auslieferer_Y`.

Synchronisieren Sie die beiden Prozesse, indem Sie die nötigen Semaphoren erzeugen, diese mit Startwerten versehen und Semaphore-Operationen einfügen.

Folgende Bedingungen müssen erfüllt sein:

- Es darf immer nur ein Prozess das Tor durchfahren.
- Es darf immer nur einer der beiden Auslieferer die Warenausgabe betreten.
- Es soll möglich sein, dass der Lieferant und ein Auslieferer gleichzeitig Waren entladen bzw. aufladen.
- Die Lagerhalle kann maximal 20 Pakete aufnehmen.
- Es dürfen keine Verklemmungen auftreten.
- Zu Beginn sind keine Pakete in der Lagerhalle vorrätig und das Tor, der Wareneingang und die Warenausgabe sind frei.

Quelle: TU-München, Übungen zur Einführung in die Informatik III, WS01/02

Man benötigt folgende Semaphore:

- Boolescher Semaphor `tor` zum wechselseitigen Ausschluss des Tores mit Startwert 1. Zu Beginn ist das Tor frei.
- Boolescher Semaphor `ausgabe` zum wechselseitigen Ausschluss der Warenausgabe mit Startwert 1. Zu Beginn ist die Warenausgabe frei.
- Semaphor `frei` zum Zählen der freien Plätze in der Lagerhalle mit Startwert 20. Zu Beginn sind alle Plätze frei.
- Semaphor `belegt` zum Zählen der belegten Plätze in der Lagerhalle mit Startwert 0. Zu Beginn ist kein Platz belegt.

Name:

Vorname:

Matr.Nr.:

Aufgabe 11 – Fortsetzung)

```
sema tor      = 1
sema ausgabe = 1
sema frei     = 20
sema belegt  = 0
```

```
Lieferant      Auslieferer_X      Auslieferer_Y
{
  while (TRUE)
  {
    P(tor);
    <Tor durchfahren>;
    V(tor);

    <Wareneingang betreten>;

    P(frei);
    P(frei);
    P(frei);
    <3 Pakete entladen>;
    V(belegt);
    V(belegt);
    V(belegt);

    <Wareneingang verlassen>;

    P(tor);
    <Tor durchfahren>;
    V(tor);
  }
}

{
  while (TRUE)
  {
    P(tor);
    <Tor durchfahren>;
    V(tor);

    P(ausgabe);
    <Warenausgabe betreten>;

    P(belegt);
    P(belegt);
    <2 Paket2 aufladen>;
    V(frei);
    V(frei);

    <Warenausgabe verlassen>;
    V(ausgabe);

    P(tor);
    <Tor durchfahren>;
    V(tor);
  }
}

{
  while (TRUE)
  {
    P(tor);
    <Tor durchfahren>;
    V(tor);

    P(ausgabe);
    <Warenausgabe betreten>;

    P(belegt);
    <1 Paket aufladen>;
    V(frei);

    <Warenausgabe verlassen>;
    V(ausgabe);

    P(tor);
    <Tor durchfahren>;
    V(tor);
  }
}
```