



Name:

Vorname:

Matr.Nr.:

---

# Aufgabe 1)

Punkte: .....

Maximale Punkte:  $1+1+2+2+1+1+1+1=10$

- a) Zu jedem Zeitpunkt kann nur ein einziges Programm laufen. Wie ist der passende Fachbegriff für diese Betriebsart?  
*Einzelprogrammbetrieb (Singletasking).*
- b) Was versteht man unter halben Multi-User-Betriebssystemen?  
*Verschiedene Benutzer können nur nacheinander am System arbeiten, aber die Daten und Prozesse der Benutzer sind voreinander geschützt.*
- c) Nennen Sie einen Vorteil und einen Nachteil von monolithischen Kernen.
- Vorteile:
    - *Weniger Kontextwechsel als Mikrokern  $\implies$  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*
- d) 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*
- e) Beschreiben Sie, was ein Administrator mit dem Kommando `whoami` machen kann.  
*Den eigenen Benutzernamen in der Shell ausgeben.*
- f) Beschreiben Sie, was ein Administrator mit dem Kommando `chmod` machen kann.  
*Die Dateirechte von Dateien und Verzeichnissen ändern.*
- g) Beschreiben Sie, was ein Administrator mit dem Kommando `head` machen kann.  
*Zeilen vom Anfang einer Datei in der Shell ausgeben.*
- h) Beschreiben Sie, was ein Administrator mit dem Kommando `touch` machen kann.  
*Leere Dateien erzeugen und die Modifikationszeit von Dateien verändern.*

Name:

Vorname:

Matr.Nr.:

---

## Aufgabe 2)

Punkte: .....

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

a) Nennen Sie zwei rotierende magnetische digitale Datenspeicher.

*z.B. Festplatte, Trommelspeicher, Diskette.*

b) Nennen Sie zwei nichtrotierende magnetische digitale Datenspeicher.

*z.B. Kernspeicher, Magnetband, Magnetstreifen, Magnetkarte, Compact Cassette (Datasette), Magnetblasenspeicher.*

c) Nennen Sie drei Vorteile von Datenspeicher ohne bewegliche Teile gegenüber Datenspeichern mit beweglichen Teilen.

*z.B. Weniger Energieverbrauch, weniger Abnutzung, weniger Abwärme, unempfindlichkeit gegen Stöße, keine Laufgeräusche.*

d) Was ist wahlfreier Zugriff?

*Es heißt, dass das Medium nicht - wie z.B. bei Bandlaufwerken - von Beginn an sequentiell durchsucht werden muss, um eine bestimmte Stelle (Datei) zu finden.*

e) Nennen Sie einen nicht-persistenten Datenspeicher.

*Hauptspeicher (DRAM).*

Name:

Vorname:

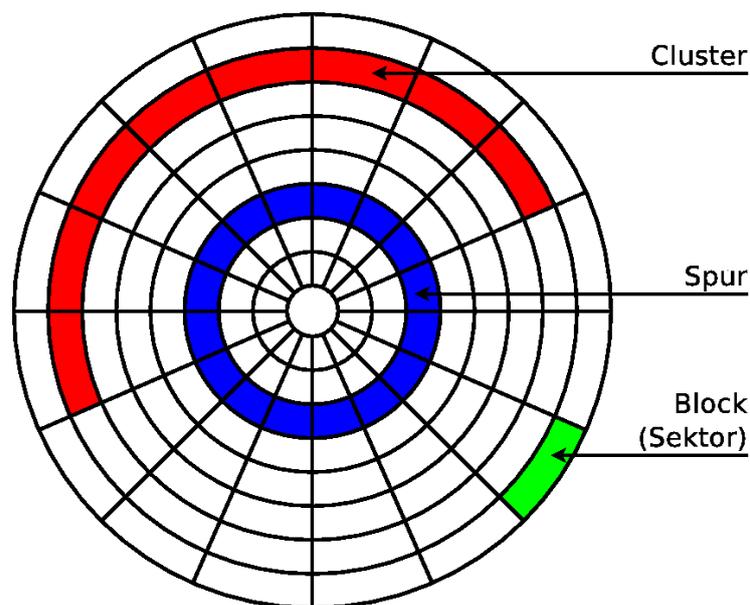
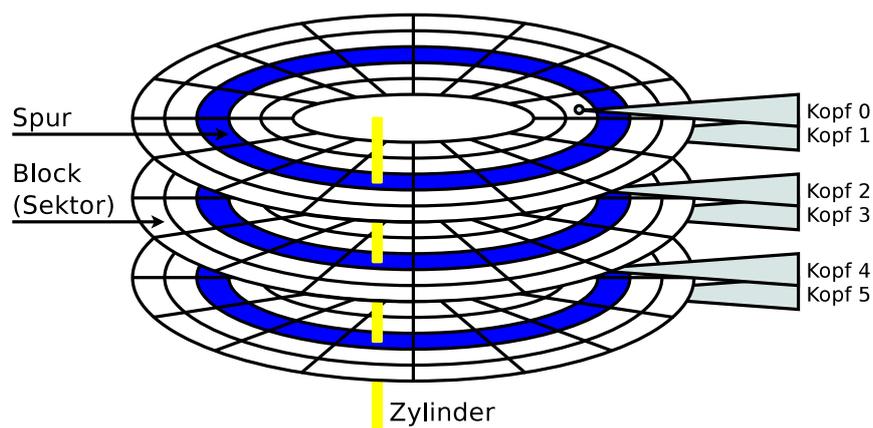
Matr.Nr.:

# Aufgabe 3)

Punkte: .....

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

- Zeichnen Sie den Aufbau einer Festplatte schematisch. Machen Sie anhand Ihrer Zeichnung(en) deutlich, was folgende Begriffe bedeuten:
  - a) Sektor (= Block)
  - b) Spur
  - c) Zylinder
  - d) Cluster



Name:

Vorname:

Matr.Nr.:

---

## Aufgabe 4)

Punkte: .....

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

a) Warum ist es falsch, SSDs als Solid State Disks zu bezeichnen?

*SSDs enthalten keine beweglichen Teile und darum auch keine „Disks“.*

b) Beschreiben Sie den Unterschied zwischen NAND-Speicher der Kategorien Single-Level Cell (SLC), Multi-Level Cell (MLC) und Triple-Level Cell (TLC).

- *TLC-Speicherzellen speichern 3 Bits*
- *MLC-Speicherzellen speichern 2 Bits*
- *SLC-Speicherzellen speichern 1 Bit*

c) Welche Aufgabe haben Wear Leveling-Algorithmen?

*Sie verteilen Schreibzugriffe gleichmäßig über die Flash-Speicherzellen.*

d) Bei welchen Konzepten der Speicherpartitionierung entsteht interne Fragmentierung?

- Statische Partitionierung
- Dynamische Partitionierung
- Buddy-Algorithmus

e) Bei welchen Konzepten der Speicherpartitionierung entsteht externe Fragmentierung?

- Statische Partitionierung
- Dynamische Partitionierung
- Buddy-Algorithmus

Name:

Vorname:

Matr.Nr.:

---

## Aufgabe 5)

Punkte: .....

Maximale Punkte:  $0,5+0,5+2+2=5$

- a) Mit welchem Kommando können Sie die Priorität eines existierenden Prozesses ändern?

*renice*

- b) Mit welchem Kommando können Sie einen „Link“ erstellen?

*ln*

- c) Können Hard Links über Dateisystemgrenzen kopiert werden?

*(Begründen Sie ihre Antwort!)*

*Nein, weil Hard Links nur Verzeichniseinträge sind, die auf einen Inode verweisen.*

- d) Können Symbolische Links über Dateisystemgrenzen kopiert werden?

*(Begründen Sie ihre Antwort!)*

*Ja, weil symbolische Links (Text-)Dateien mit einer Pfadangabe sind.*

Name:

Vorname:

Matr.Nr.:

---

## Aufgabe 6)

Punkte: .....

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

a) Wie arbeitet der Real Mode?

*Jeder Prozess kann direkt auf den gesamten adressierbaren Speicher zugreifen.*

b) Wie arbeitet der Protected Mode?

*Jeder Prozess darf nur auf seinen eigenen virtuellen Speicher zugreifen.*

c) Wie entsteht eine Page Fault Ausnahme (Exception)?

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

d) Wie entsteht eine General Protection Fault Ausnahme (Exception)?

*Ein Prozess versucht auf eine virtuelle Speicheradresse zuzugreifen, auf die er nicht zugreifen darf.*

e) Welche Auswirkung hat eine General Protection Fault Ausnahme (Exception)?

*Crash des Betriebssystems.*

f) Was enthält der Kernelspace?

*Den Betriebssystemkern (Kernel) und Kernelerweiterungen (Treiber).*

g) Was enthält der Userspace?

*Den aktuell ausgeführten Prozess, der um den Erweiterungsspeicher („Swap“, Windows: „Page-File“) vergrößert wird.*

Name:

Vorname:

Matr.Nr.:

---

## Aufgabe 7)

Punkte: .....

Maximale Punkte: 6

Kreuzen Sie bei jeder Aussage zu Dateisystemen an, ob die Aussage wahr oder falsch ist.

Aussage	wahr	falsch
Inodes speichern alle Verwaltungsdaten (Metadaten) der Dateien.		X
Dateisysteme adressieren Cluster und nicht Blöcke des Datenträgers oder Laufwerks.	X	
Je kleiner die Cluster, desto größer ist der Verwaltungsaufwand für große Dateien.	X	
Je größer die Cluster, desto geringer ist der Kapazitätsverlust durch interne Fragmentierung.		X
Absolute Pfadnamen beschreiben den kompletten Pfad von der Wurzel bis zur Datei.	X	
Ein Vorteil der Blockgruppen bei ext2 ist, dass die Inodes physisch nahe bei den Clustern liegen, die sie adressieren.	X	
Eine Dateizuordnungstabelle (FAT) erfasst die belegten und freien Cluster im Dateisystem.	X	
Ein Journal im Dateisystem reduziert die Anzahl der Schreibzugriffe.		X
Journaling-Dateisysteme grenzen die bei der Konsistenzprüfung zu überprüfenden Daten ein.	X	
Bei Dateisystemen mit Journal sind Datenverluste garantiert ausgeschlossen.		X
Vollständiges Journaling führt alle Schreiboperation doppelt aus.	X	
Extents verursachen weniger Verwaltungsaufwand als Blockadressierung.	X	

Name:

Vorname:

Matr.Nr.:

---

# Aufgabe 8)

Punkte: .....

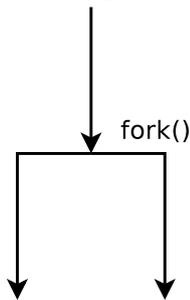
Maximale Punkte: 1+3+1+1+4=10

- a) Was passiert, wenn ein neuer Prozess erstellt werden soll, es aber im Betriebssystem keine freien Prozessidentifikation (PIDs) mehr gibt?

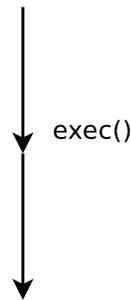
*Dann kann kein neuer Prozess erstellt werden.*

- b) Die drei Abbildungen zeigen alle existierenden Möglichkeiten, einen neuen Prozess zu erzeugen. Schreiben Sie zu jeder Abbildung, welche(r) Systemaufruf(e) nötig sind, um die gezeigte Prozesserschöpfung zu realisieren.

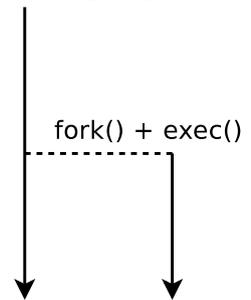
Prozessvergabeung



Prozessverkettung



Prozesserschöpfung



- c) Was unterscheidet einen Kindprozess vom Elternprozess kurz nach der Erzeugung?

*Die PID und die Speicherbereiche.*

- d) Was passiert, wenn ein Elternprozess vor dem Kindprozess beendet wird?

*init adoptiert den Kind-Prozess. Die PPID des Kind-Prozesses hat dann den Wert 1.*

- e) Ein Elternprozess (PID = 102) mit den in der folgenden Tabelle beschriebenen Eigenschaften erzeugt mit Hilfe des Systemaufrufs `fork()` einen Kindprozess (PID = 103). Tragen Sie die vier fehlenden Werte in die Tabelle ein.

	Elternprozess	Kindprozess
UID	100	100
PID	102	103
PPID	101	102
Rückgabewert von <code>fork()</code>	103	0

Name:

Vorname:

Matr.Nr.:

---

## Aufgabe 9)

Punkte: .....

Maximale Punkte:  $1+2+3+1+1+1=9$

- a) Welche Daten enthält das Textsegment?

*Den ausführbaren Programmcode (Maschinencode).*

- b) Welche Daten enthält der Heap?

*Konstanten und Variablen die außerhalb von Funktionen deklariert sind.*

- c) Welche Daten enthält der Stack?

*Kommandozeilenargumente des Programmaufrufs, Umgebungsvariablen, Aufrufparameter und Rücksprungadressen der Funktionen, lokale Variablen der Funktionen.*

- d) Was sind Interrupts?

*Es sind externe Unterbrechungen. Sie werden durch Ereignisse außerhalb des zu unterbrechenden Prozesses ausgelöst (z.B. ein Ein-/Ausgabe-Gerät meldet ein E/A-Ereignis).*

- e) Was ist der Unterbrechungsvektor (Interrupt Vector)?

*Eine Liste mit den Adressen aller Unterbrechungsrouinen im Betriebssystem.*

- f) Was sind Exceptions?

*Es sind interne Unterbrechungen oder Ausnahmen. Im Gegensatz zu Interrupts werden Sie vom zu unterbrechenden Prozess selbst ausgelöst.*

Name:

Vorname:

Matr.Nr.:

---

## Aufgabe 10)

Punkte: .....

Maximale Punkte:  $2+3+2+1+1+1=10$

- a) Wie funktioniert statisches Multilevel-Scheduling?

*Die bereit-Liste wird in mehrere Teillisten unterteilt. Für jede Teilliste wird eine andere Scheduling-Strategie verwendet. Die Teillisten haben unterschiedliche Prioritäten oder Zeitmultiplexe.*

- b) Wie funktioniert Multilevel-Feedback-Scheduling?

*Es arbeitet mit mehreren Warteschlangen. Jede Warteschlange hat eine andere Priorität oder Zeitmultiplex. Jeder neue Prozess kommt in die oberste Warteschlange und hat damit die höchste Priorität. Innerhalb jeder Warteschlange wird Round Robin eingesetzt. Gibt ein Prozess die CPU freiwillig wieder ab, wird er wieder in die selbe Warteschlange eingereiht. Hat ein Prozess seine volle Zeitscheibe genutzt, kommt er in die nächst tiefere Warteschlange mit einer niedrigeren Priorität.*

- c) Nennen Sie vier Schedulingverfahren, die „fair“ sind.

*z.B. First Come First Served, Round Robin mit Zeitquantum, Highest Response Ratio Next, Earliest Deadline First, Fair-Share.*

- d) Was ist ein kritischer Abschnitt?

*Mehrere Prozesse greifen lesend und schreibend auf gemeinsame Daten zu.*

- e) Was ist eine Race Condition?

*Eine unbeabsichtigten Wettlaufsituation zweier Prozesse, die auf die gleiche Speicherstelle schreibend zugreifen wollen.*

- f) Wie werden Race Conditions vermieden?

*Durch das Konzept der Semaphore.*

Name:

Vorname:

Matr.Nr.:

---

# Aufgabe 11)

Punkte: .....

Maximale Punkte:  $1+1+0,5+0,5+1+1+3+1=9$

- a) Was ist bei Interprozesskommunikation über gemeinsame Speichersegmente (Shared Memory) zu beachten?

*Die Prozesse müssen die Zugriffe selbst koordinieren und sicherstellen, dass ihre Speicherzugriffe sich gegenseitig ausschließen. Der Senter-Prozess darf nichts aus dem gemeinsamen Speicher lesen, bevor der Sender-Prozess fertig geschrieben hat. Ist die Koordinierung der Zugriffe nicht sorgfältig  $\implies$  Inkonsistenzen.*

- b) Kreuzen Sie an, welche Auswirkungen ein Neustart (Reboot) des Betriebssystems auf die bestehenden gemeinsamen Speichersegmente (Shared Memory) hat.

Die gemeinsamen Speichersegmente werden beim Neustart erneut angelegt und die Inhalte werden wieder hergestellt.

Die gemeinsamen Speichersegmente und deren Inhalte sind verloren.

Die gemeinsamen Speichersegmente werden beim Neustart erneut angelegt, bleiben aber leer. Nur die Inhalte sind also verloren.

Nur die gemeinsamen Speichersegmente sind verloren. Die Inhalte speichert das Betriebssystem in temporären Dateien im Ordner `\tmp`.

- c) Nach welchem Prinzip arbeiten Nachrichtenwarteschlangen (Message Queues)?

Round Robin

LIFO

SJF

FIFO

LJF

- d) Wie viele Prozesse können über eine Pipe miteinander kommunizieren?

*Pipes können immer nur zwischen 2 Prozessen tätig sein.*

- e) Was passiert, wenn ein Prozess in eine volle Pipe schreiben will?

*Der in die Pipe schreibende Prozess wird blockiert.*

- f) Was passiert, wenn ein Prozess aus einer leeren Pipe lesen will?

*Der aus der Pipe lesende Prozess wird blockiert.*

- g) Welche drei Formen der Interprozesskommunikation funktionieren bidirektional?

*Gemeinsame Speichersegmente, Sockets, benannte Pipes.*

- h) Bei welcher Form der Interprozesskommunikation garantiert das Betriebssystem die Synchronisierung?

*Anonyme und benannte Pipes, Sockets, Nachrichtenwarteschlangen*

*(Hinweis: Eine korrekte Antwort war bei dieser Teilaufgabe ausreichend, um den Punkt zu bekommen.)*

Name:

Vorname:

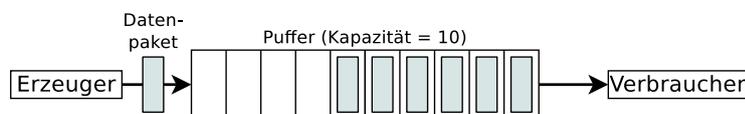
Matr.Nr.:

# Aufgabe 12)

Punkte: .....

Maximale Punkte: 7

- Ein Erzeuger schreibt Daten in den Puffer und der Verbraucher entfernt diese.
- Gegenseitiger Ausschluss ist nötig, um Inkonsistenzen zu vermeiden.
- Ist der Puffer voll, muss der Erzeuger blockieren.
- Ist der Puffer leer, muss der Verbraucher blockieren.



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

```
typedef int semaphore;           // Semaphore sind von Typ Integer
semaphore voll = 0;              // zählt die belegten Plätze im Puffer
semaphore leer = 10;            // zählt die freien Plätze im Puffer
semaphore mutex = 1;           // steuert Zugriff auf kritische Bereiche

void erzeuger (void) {
    int daten;

    while (TRUE) {              // Endlosschleife
        erzeugeDatenpaket(daten); // erzeuge Datenpaket
        P(leer);                 // Zähler "leere Plätze" erniedrigen
        P(mutex);               // in kritischen Bereich eintreten
        einfüegenDatenpaket(daten); // Datenpaket in Puffer schreiben
        V(mutex);               // kritischen Bereich verlassen
        V(voll);                 // Zähler für volle Plätze erhöhen
    }
}

void verbraucher (void) {
    int daten;

    while (TRUE) {              // Endlosschleife
        P(voll);                 // Zähler "volle Plätze" erniedrigen
        P(mutex);               // in kritischen Bereich eintreten
        entferneDatenpaket(daten); // Datenpaket aus dem Puffer holen
        V(mutex);               // kritischen Bereich verlassen
        V(leer);                 // Zähler für leere Plätze erhöhen
        verbraucheDatenpaket(daten); // Datenpaket nutzen
    }
}
```