

Abschlussklausur

Betriebssysteme

22. Juli 2019

Name: _____

Vorname: _____

Matrikelnummer: _____

Mit meiner Unterschrift bestätige ich, dass ich die Klausur selbständig bearbeite und das ich mich gesund und prüfungsfähig fühle.
Mir ist bekannt, dass mit dem Erhalt der Aufgabenstellung die Klausur als angetreten gilt und bewertet wird.

Unterschrift: _____

- Schreiben Sie Ihre Lösungen auf die vorbereiteten Blätter. Eigenes Papier darf *nicht* verwendet werden.
- Als Hilfsmittel ist ein *selbständig vorbereitetes* und *handschriftlich einseitig beschriebenes DIN-A4-Blatt* zugelassen (keine Kopien!).
- Als Hilfsmittel ist ein *Taschenrechner* zugelassen.
- Verwenden Sie *keinen* Rotstift.
- Die Bearbeitungszeit beträgt *90 Minuten*.
- Schalten Sie Ihre Mobiltelefone aus.

Bewertung:

Question:	1	2	3	4	5	6	7	8	9	10	11	Σ	Grade
Maximum points:	8	14	8	4	10	8	4	8	10	9	7	90	—
Achieved points:													

1.0: 90.0-85.5, **1.3:** 85.0-81.0, **1.7:** 80.5-76.5, **2.0:** 76.0-72.0, **2.3:** 71.5-67.5,
2.7: 67.0-63.0, **3.0:** 62.5-58.5, **3.3:** 58.0-54.0, **3.7:** 53.5-49.5, **4.0:** 49.0-45.0, **5.0:** <45

Name:

Vorname:

Matr.Nr.:

Aufgabe 1)

Punkte:

Maximale Punkte: $1+1+2+2+2=8$

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

- b) Nennen Sie den Fachbegriff der quasi-parallelen Programm- bzw. Prozessausführung.

- c) Beschreiben Sie den Aufbau eines monolithischen Kernels.

- d) Beschreiben Sie den Aufbau eines minimalen Kerns (Mikrokernels).

- e) Beschreiben Sie den Aufbau eines hybriden Kernels.

Name:

Vorname:

Matr.Nr.:

Aufgabe 2)

Punkte:

Maximale Punkte: $1+2+2+3+6=14$

- a) Welche zwei Gruppen von Ein- und Ausgabegeräten gibt es bezüglich der kleinsten Übertragungseinheit?
- b) Vergleichen Sie die Arbeitsweise der Gruppen aus Teilaufgabe a).
- c) Nennen Sie für jede Gruppe aus Teilaufgabe a) zwei Beispiele.
- d) Nennen Sie drei Möglichkeiten, wie Prozesse Daten von Ein- und Ausgabegeräten lesen können.
- e) Nennen Sie für jede Möglichkeit aus Teilaufgabe d) jeweils einen Vorteil und einen Nachteil.

Name:

Vorname:

Matr.Nr.:

Aufgabe 3)

Punkte:

Maximale Punkte: $2+2+2+2=8$

Auf einer Festplatte befinden sich folgende Informationen:

IBM Travelstar	MODEL: DBCA-204860 E182115 T
RATED: 5V 500mA	MADE IN THAILAND BY IBM STORAGE
P/N: 21L9510 4090 MB	16NOV99
FRU: 22L0018 MLC:F41941	(7944 CYL. 16 HEADS. 63 SEC/T)

- a) Berechnen Sie die Kapazität einer Oberfläche einer Scheibe der Festplatte.

(Bei der Lösung muss der Rechenweg angegeben sein!)

Hinweis: Die Anzahl der Zylinder (CYL) ist identisch mit der Anzahl der Spuren („Tracks“) pro Scheibe. Die Größe der Sektoren (SEC) ist 512 Byte.

- b) Berechnen Sie die Größe einer Spur der Festplatte.

(Bei der Lösung muss der Rechenweg angegeben sein!)

- c) Berechnen Sie die Gesamtkapazität der Festplatte.

(Bei der Lösung muss der Rechenweg angegeben sein!)

- d) Wie viele Scheiben hat die Festplatte? *Hinweis: Jede Scheibe hat zwei Oberflächen.*

(Begründen Sie ihre Antwort!)

Name:

Vorname:

Matr.Nr.:

Aufgabe 4)

Punkte:

Maximale Punkte: 4

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

- a) Real Mode ist für Multitasking-Systeme geeignet.
☐ Wahr ☐ Falsch
- b) Beim Protected Mode läuft jeder Prozess in seiner eigenen, von anderen Prozessen abgeschotteten Kopie des physischen Adressraums.
☐ Wahr ☐ Falsch
- c) Bei statischer Partitionierung entsteht interne Fragmentierung.
☐ Wahr ☐ Falsch
- d) Bei dynamischer Partitionierung ist externe Fragmentierung unmöglich.
☐ Wahr ☐ Falsch
- e) Beim Paging haben alle Seiten die gleiche Länge.
☐ Wahr ☐ Falsch
- f) Ein Vorteil langer Seiten beim Paging ist geringe interne Fragmentierung.
☐ Wahr ☐ Falsch
- g) Ein Nachteil kurzer Seiten beim Paging ist, dass die Seitentabelle sehr groß werden kann.
☐ Wahr ☐ Falsch
- h) Die MMU übersetzt beim Paging logische Speicheradressen mit der Seitentabelle in physische Adressen.
☐ Wahr ☐ Falsch

Name:

Vorname:

Matr.Nr.:

Aufgabe 5)

Punkte:

Maximale Punkte: 10

- a) Geben Sie an, welche Informationen ein Inode speichert.
- b) Nennen Sie drei Beispiele für Metadaten im Dateisystem.
- c) Beschreiben Sie, was ein Cluster im Dateisystem ist.
- d) Beschreiben Sie, wie ein UNIX-Dateisystem (z.B. ext2/3), das keine Extents verwendet, mehr als 12 Cluster adressiert.
- e) Beschreiben Sie, wie Verzeichnisse bei Linux-Dateisystemen technisch realisiert sind.
- f) Die meisten Betriebssystemen arbeiten nach dem Prinzip...
 - ☐ Write-Back ☐ Write-Through
- g) `/home/<benutzername>/Mail/inbox/` ist ein...
 - ☐ Absoluter Pfadname ☐ Relativer Pfadname
- h) Nennen Sie die Information, die der Bootsektor eines Dateisystems speichert.
- i) Nennen Sie die Information, die der Superblock eines Dateisystems speichert.
- j) Erklären Sie warum manche Dateisysteme (z.B. ext2/3) die Cluster des Dateisystems zu Blockgruppen zusammenfassen.

Name:

Vorname:

Matr.Nr.:

Aufgabe 6)

Punkte:

Maximale Punkte: $2+1+1+3+1=8$

- a) Beschreiben Sie, was die Dateizuordnungstabelle bzw. File Allocation Table (FAT) ist und welche Informationen diese enthält.

- b) Beschreiben Sie die Aufgabe des Journals bei Journaling-Dateisystemen.

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

- d) Nennen Sie die drei Werte, die zum Speichern eines Extents nötig sind.

- e) Beschreiben Sie den Vorteil des Einsatzes von Extents gegenüber direkter Adressierung der Cluster.

Name:

Vorname:

Matr.Nr.:

Aufgabe 7)

Punkte:

Maximale Punkte: 4

- a) Beschreiben Sie, was das Defragmentieren macht.

- b) Beschreiben Sie welche Art der Datenverarbeitung durch Defragmentieren maximal beschleunigt wird.

- c) Beschreiben Sie in welchen Szenario das Defragmentieren sinnvoll ist.

- d) Ist das Defragmentieren bei SSDs sinnvoll? (Begründen Sie ihre Antwort.)

Name:

Vorname:

Matr.Nr.:

Aufgabe 8)

Punkte:

Maximale Punkte: 8

- a) Beschreiben Sie, was der Systemaufruf `fork()` macht.

- b) Beschreiben Sie, was der Systemaufruf `exec()` macht.

- c) Erklären Sie, was `init` ist.

- d) Nennen Sie den Unterschied eines Kindprozess vom Elternprozess kurz nach der Erzeugung.

- e) Beschreiben, Sie was passiert, wenn ein Elternprozess vor dem Kindprozess beendet wird.

- f) Nennen Sie den Inhalt des Textsegments.

- g) Nennen Sie den Inhalt des Heap.

- h) Nennen Sie den Inhalt des Stack.

Name:

Vorname:

Matr.Nr.:

Aufgabe 9)

Punkte:

Maximale Punkte: $6+2+2=10$

- a) Beschreiben Sie (gerne auch mithilfe einer Abbildung) wie das Multilevel-Feedback-Scheduling funktioniert.

- b) Nennen Sie vier Schedulingverfahren, die fair sind.

- c) Nennen Sie vier Schedulingverfahren, bei denen die CPU-Laufzeit (= *Rechenzeit*) der Prozesse nicht bekannt sein muss.
(Hinweis: Es sind also nur solche Schedulingverfahren gesucht, die unter realistischen Bedingungen eingesetzt werden können.)

Name:

Vorname:

Matr.Nr.:

Aufgabe 10)

Punkte:

Maximale Punkte: 2+7=9

- a) Kreuzen Sie vier Bedingungen an, die gleichzeitig erfüllt sein müssen, damit ein Deadlock entstehen kann?

☐ Rekursive Funktionsaufrufe

☐ Anforderung weiterer Betriebsmittel

☐ Wechselseitiger Ausschluss

☐ > 128 Prozesse im Zustand **blockiert**

☐ Häufige Funktionsaufrufe

☐ Iterative Programmierung

☐ Geschachtelte **for**-Schleifen

☐ Zyklische Wartebedingung

☐ Ununterbrechbarkeit

☐ Warteschlangen

- b) Kommt es zum Deadlock?

Führen Sie die Deadlock-Erkennung mit Matrizen durch.

Ressourcenvektor = (4 8 6 6 5)

Belegungsmatrix = $\begin{bmatrix} 0 & 2 & 1 & 0 & 0 \\ 2 & 3 & 1 & 0 & 4 \\ 1 & 0 & 2 & 1 & 1 \end{bmatrix}$

Anforderungsmatrix = $\begin{bmatrix} 3 & 3 & 2 & 4 & 5 \\ 0 & 3 & 1 & 4 & 0 \\ 0 & 2 & 3 & 5 & 4 \end{bmatrix}$

Name:

Vorname:

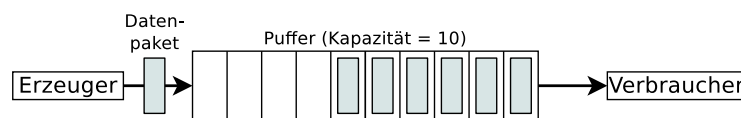
Matr.Nr.:

Aufgabe 11)

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;
```

```
void erzeuger (void) {
    int daten;
    while (TRUE) {
        erzeugeDatenpaket(daten);    // Endlosschleife
        // erzeuge Datenpaket

        einfuegenDatenpaket(daten);  // Datenpaket in Puffer schreiben

    }
}

void verbraucher (void) {
    int daten;
    while (TRUE) {
        // Endlosschleife

        entferneDatenpaket(daten);   // Datenpaket aus Puffer holen

        verbraucheDatenpaket(daten); // Datenpaket nutzen
    }
}
```