

# Klausur Betriebssysteme und Sicherheit, 01.03.2023

— Bearbeitungszeit: 90 Minuten — Prüfer: Prof. Dr. Schirmeier, Dr. Köpsell —

1	2	3	4	5	6	$\Sigma$
14	14	16	15	15	16	90

Alle Aussagen sind so ausführlich wie nötig, aber so knapp wie möglich zu begründen.

## Aufgabe 1 – Prozesse und Unix

14 Punkte

Ein Client und ein Serverprogramm kommunizieren über eine Pipe. Die Kommunikation über die Pipe läuft über den Filedeskriptor `pipe`. Die Programme führen folgenden Code aus.

HINWEIS:

- Die Funktion `wait` blockiert den aufrufenden Prozess so lange, bis sich einer seiner Kindprozesse beendet. Existiert kein solcher, so kehrt die Funktion sofort zurück.
- `puts(buf)` gibt den Inhalt des Buffers `buf` auf der Konsole aus.

Server

```

1 char buf[2];
2
3 int pid = fork();
4 if (pid == 0) {
5     read(pipe, buf, 1);
6     puts(buf);
7     exit();
8 }
9
10 puts("Receiving...");
11 wait();
12 puts("Shutting down...");
13 exit();

```

Client

```

1 write(pipe, "a", 1);
2 exit();

```

- a) Erklären Sie, weshalb mehrere mögliche Ausgaben auf der Konsole des Serverprozesses erscheinen können, und geben Sie zwei dieser Ausgaben an. **3 P**

- b) Welche Zeilen müsste man vertauschen, damit es nur noch eine mögliche Ausgabe gibt? **1 P**

- c) Was passiert mit dem Serverprozess und dessen Kindprozess, wenn der Clientprozess nicht ausgeführt wird und niemand in die Pipe schreibt? **2 P**

---

d) Welche alternative Kommunikationsmethode (als Ersatz für die Pipe) kann zwischen Client und Server verwendet werden? **1 P**

e) Welcher Mechanismus könnte anstelle von `fork` verwendet werden, wenn der Server nebenläufig mehrere Verbindungen aufbauen möchte? **1 P**

Das Verzeichnis, in welchem das Serverprogramm gespeichert ist, sieht wie folgt aus:

Rechte			Besitzer	Gruppe	Datei
Besitzer	Gruppe	Andere			
<code>rwX</code>	<code>r--</code>	<code>r--</code>	<code>gisela</code>	<code>user</code>	<code>Server</code>
<code>rw-</code>	<code>rw-</code>	<code>r--</code>	<code>system</code>	<code>system</code>	<code>server.log</code>

Im System existieren diese Nutzer mit folgender Gruppenzuordnung:

Nutzer	Gruppen
<code>gisela</code>	<code>user</code>
<code>system</code>	<code>system</code>

f) Der Serverprozess wird von der Benutzerin `gisela` gestartet und soll nun in die Datei `server.log` schreiben. Dies ist jedoch in der aktuellen Konfiguration nicht möglich. Begründen Sie kurz, weshalb. Geben Sie eine Änderung der Zugriffsrechte an, um das Lesen und Schreiben in die Datei `server.log` zu ermöglichen. **2 P**

g) Der Admin möchte, dass die Datei `server.log` nur für den Nutzer `system` les- und schreibbar ist. Mit welchem Unix-Mechanismus ist es möglich, dass `gisela` das Programm `Server` startet und der Prozess anschließend dennoch auf die Datei `server.log` zugreifen kann? Wie sehen die Zugriffsrechte für die Datei `Server` aus? **4 P**

---

## Aufgabe 2 – Sicherheit

14 Punkte

- a) Nennen Sie die drei wesentliche Schutzziele in IT-Systemen und geben Sie jeweils eine kurze Beschreibung an! **3 P**
- b) Nennen Sie für zwei der in a) aufgeführten Schutzziele den Typ Kryptosystem, mit dem sie umgesetzt werden können! **2 P**
- c) Welche Typen von Kryptosystemen kommen in Sicherheitsprotokollen üblicherweise zum Einsatz, und warum? **2 P**
- d) Nennen und erläutern Sie drei Datenschutzprinzipien! **3 P**
- e) Nennen Sie ein Kryptosystem, das informationstheoretische Sicherheit bietet! Erläutern Sie kurz, wie mit diesem System ver- und entschlüsselt wird! Nennen Sie zwei Anforderungen bzgl. der Schlüssel, die dieses System erfüllen muss, um informationstheoretische Sicherheit zu erlangen! **2 P**
- f) Welche vier Schritte umfasst der Sicherheitsprozess beim Informationssicherheitsmanagement? **2 P**

## Aufgabe 3 – Speicher

16 Punkte

In einem 32-Bit-System wird eine 2-stufige Seitentabelle zur Verwaltung der virtuellen Adressräume verwendet. Die verwendeten Seiten des Systems sind 4096 Byte groß. Die Seitentabellen der ersten Stufe enthalten 256 Einträge. Die Seitentabellen der zweiten Stufe enthalten 4096 Einträge.

HINWEIS:

- Bei den folgenden Aufgaben genügt die Angabe von Zweierpotenzen.
- Es ist immer auch eine geeignete Einheit anzugeben.

a) Beantworten Sie folgende Fragen:

- Wie groß ist der virtuelle Adressraum?
- Wie viele Bit der virtuellen Adresse bilden das Offset?
- Wie viele Bit der virtuellen Adresse bilden den Index für die erste Stufe der Seitentabelle?

3 P

b) Gegeben sind die folgenden Seitentabellen. Die Seitentabelle der ersten Stufe liegt bei 0x1 000.

Addr.: 0x1 000	Addr.: 0x2 000	Addr.: 0x3 000	Addr.: 0x4 000												
0x0 <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0x2</td><td style="padding: 2px;">p</td><td style="padding: 2px;">u</td></tr></table>	0x2	p	u	0x0 <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0x0AF</td><td style="padding: 2px;">p</td><td style="padding: 2px;">u</td></tr></table>	0x0AF	p	u	0x0 <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0x0AF</td><td style="padding: 2px;">p</td><td style="padding: 2px;">u</td></tr></table>	0x0AF	p	u	0x0 <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0x0AF</td><td style="padding: 2px;">p</td><td style="padding: 2px;">u</td></tr></table>	0x0AF	p	u
0x2	p	u													
0x0AF	p	u													
0x0AF	p	u													
0x0AF	p	u													
0x1 <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0x1</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr></table>	0x1			...	...	0x1 <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0xF00</td><td style="padding: 2px;">p</td><td style="padding: 2px;"></td></tr></table>	0xF00	p							
0x1															
0xF00	p														
...	0xF00 <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0x815</td><td style="padding: 2px;">p</td><td style="padding: 2px;">u</td></tr></table>	0x815	p	u	0x51 <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0xF00</td><td style="padding: 2px;">pw</td><td style="padding: 2px;">u</td></tr></table>	0xF00	pw	u	0x2 <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0xBA5</td><td style="padding: 2px;">pw</td><td style="padding: 2px;"></td></tr></table>	0xBA5	pw				
0x815	p	u													
0xF00	pw	u													
0xBA5	pw														
0xD <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0x3</td><td style="padding: 2px;">pw</td><td style="padding: 2px;">u</td></tr></table>	0x3	pw	u	0xF01 <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0xABC</td><td style="padding: 2px;">pw</td><td style="padding: 2px;"></td></tr></table>	0xABC	pw		...	...						
0x3	pw	u													
0xABC	pw														
...	...	0xEAD <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0xBAD</td><td style="padding: 2px;">p</td><td style="padding: 2px;">u</td></tr></table>	0xBAD	p	u	0x12 <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0x123</td><td style="padding: 2px;">pw</td><td style="padding: 2px;"></td></tr></table>	0x123	pw							
0xBAD	p	u													
0x123	pw														
0xFE <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0x4</td><td style="padding: 2px;">pw</td><td style="padding: 2px;"></td></tr></table>	0x4	pw		0xFFF <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0xA55</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr></table>	0xA55			...	...						
0x4	pw														
0xA55															
0xFF <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0x2</td><td style="padding: 2px;">pw</td><td style="padding: 2px;"></td></tr></table>	0x2	pw			0xFFF <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0x0FA</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr></table>	0x0FA			0xFFF <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px;">0x004</td><td style="padding: 2px;">pw</td><td style="padding: 2px;"></td></tr></table>	0x004	pw				
0x2	pw														
0x0FA															
0x004	pw														

p	w	u
Present Bit	Write Bit	User Bit

Ein Nutzerprogramm führt im folgenden Zugriffe aus. Bestimmen Sie für diese Zugriffe entweder die physische Adresse oder geben Sie an, bei welcher Stelle der Adressübersetzung ein Seitenfehler auftritt. **5 P**

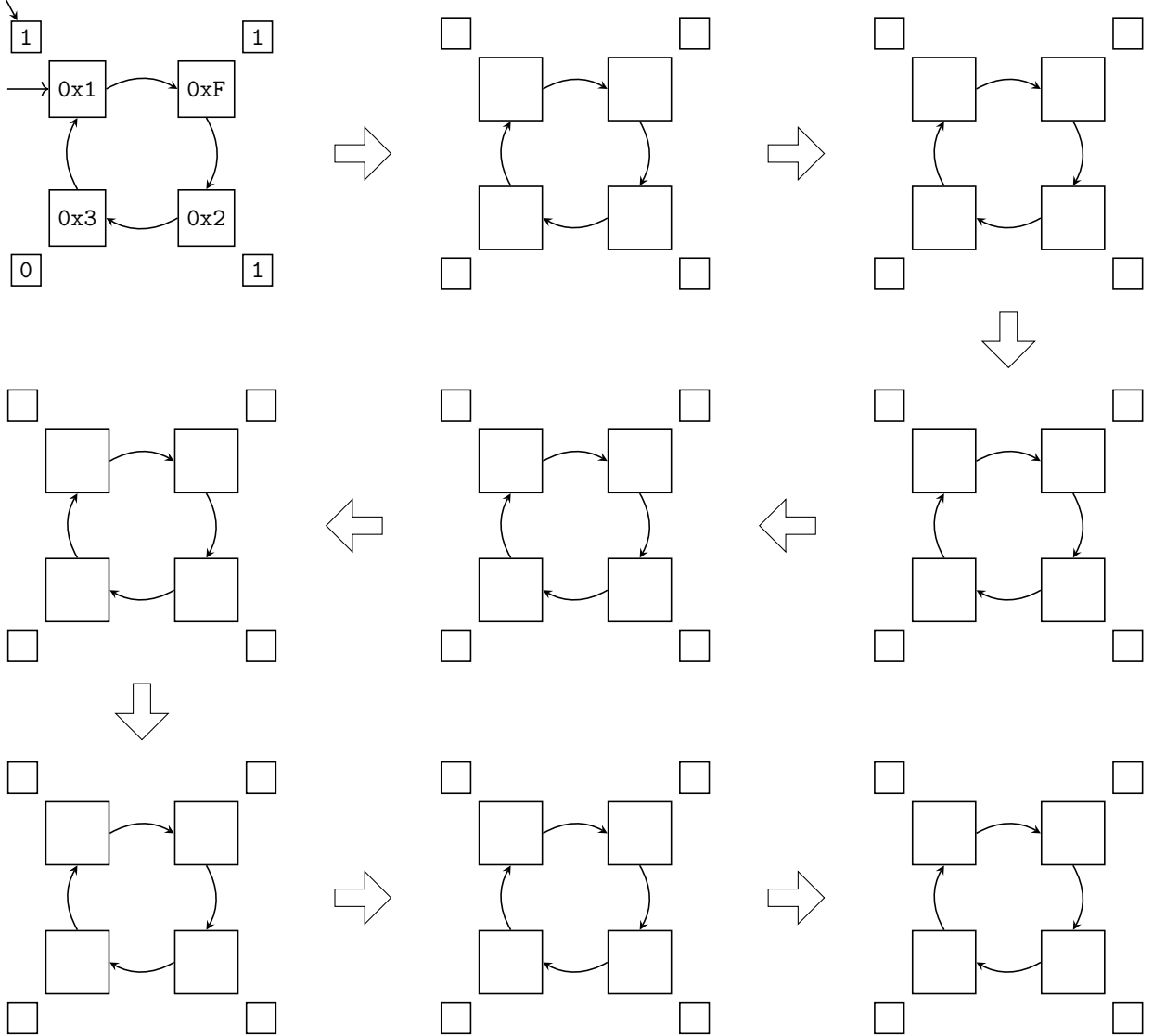
- Lesender Zugriff auf 0x1ABAFCC
- Lesender Zugriff auf 0xF00BA5
- Schreibender Zugriff auf 0xDEADA55
- Schreibender Zugriff auf 0xD051599
- Schreibender Zugriff auf 0xFE012345

c) Für die Seitenverdrängung nutzt das verwendete Betriebssystem den Clock-Algorithmus. Gegeben sei folgende Zugriffsreihenfolge auf Seiten im System:

0x1, 0x2, 0x3, 0x4, 0x2, 0xA, 0xB, 0x1

Notieren Sie alle Änderungen und Ersetzungen gemäß dem Clock-Algorithmus, die sich aus dieser Zugriffsreihenfolge ergeben. Starten Sie dabei vom ersten ausgefüllten Zustand. **8 P**

Referenzbit



## Aufgabe 4 – Synchronisation

15 Punkte

Ein Betriebssystem stellt die Abstraktion von Semaphoren bereit, die von Programmen mittels des Typs `sem_t` genutzt werden kann. Ein `sem_t s` unterstützt die zwei Operationen `s.signal()` und `s.wait()`, deren Verhalten der Vorlesung entsprechend ist. Der initiale Wert des Zählers wird bei der Initialisierung des Objektes übergeben.

- a) Die folgenden drei Funktionen werden von drei Threads ausgeführt und sind gleichzeitig bereit. Sie sollen mit Objekten des Typs `sem_t` so synchronisiert werden, dass sie gemeinsam „Leave now, and never come back!“ ausgeben.

Vervollständigen Sie den Code entsprechend. Initialisieren Sie die Objekte vom Typ `sem_t`. Verwenden Sie genau *drei* Objekte vom Typ `sem_t` und so wenige Aufrufe an `signal` und `wait` wie möglich. **7 P**

```
/* Initialisierung der sem_t-Variablen */
```

**T<sub>1</sub>**

```
void f1() {  
    _____  
    printf("now, ");  
    _____  
    printf("come ");  
    _____  
}
```

**T<sub>2</sub>**

```
void f2() {  
    _____  
    printf("Leave ");  
    _____  
    printf("back!");  
    _____  
}
```

**T<sub>3</sub>**

```
void f3() {  
    _____  
    printf("and ");  
    _____  
    printf("never ");  
    _____  
}
```

- b) Für das Anlernen eines Sprachmodells sollen Videos aus dem Internet verwendet werden. Dazu müssen wiederholt Videos jeweils mittels der Funktion `download_video` heruntergeladen und lokal gespeichert werden. Nach der Verarbeitung durch die Funktion `process_video` wird das Video automatisch gelöscht. Zur Erhöhung der Effizienz passiert das jeweils parallel und in separaten Threads. Es können beliebig viele Threads parallel `download_video` und `process_video` ausführen. Aufgrund begrenzter Ressourcen können jedoch maximal drei Videos gleichzeitig heruntergeladen und lokal gespeichert werden. Geben Sie an, wie viele Objekte vom Typ `sem_t` nötig sind, um dieses Problem zu lösen ohne aktives Warten einzusetzen:

Geben Sie nachfolgend Initialisierungen und Code für eine Lösung mit der von Ihnen genannten Anzahl an Semaphoren an. **6 P**

```
/* Initialisierung */
```

```
void do_download() {
```

```
    download_video();
```

```
}
```

```
void do_processing() {
```

```
    process_video();
```

```
}
```

- c) Es wird festgestellt, dass `process_video` exklusiven Zugriff auf eine Variable benötigt. Beschreiben Sie die Änderungen, die dadurch nötig sind. Nennen Sie auch die Initialisierung etwaiger neuer Variablen. **2 P**

## Aufgabe 5 – Ein-/Ausgabe und Dateisysteme

15 Punkte

Auf einem UNIX-artigen System ergibt die Listung der verfügbaren Blockgeräte folgende (stark vereinfachte) Ausgabe:

NAME	MAJ:MIN	SIZE	RO	MOUNTPOINTS
nvme0n1	259:1	147G	0	/
nvme0n2	259:2	314G	0	/home
nvme0n3	259:3	16G	0	[SWAP]

a) Was bezeichnen die Major- und Minornummern in Spalte zwei der Ausgabe?

2 P

Gegeben ist ein Dateisystem im unten angegebenen Zustand. Das Dateisystem nutzt eine Blockgröße von 4 KiB. Die dargestellten Blöcke enthalten entweder Inodes (IB), Superblöcke (SB), Verzeichnisblöcke (VB) oder eine Allokations-Bitmap für Blöcke (AB) bzw. für Inodes (IAB). Weitere Blöcke (z.B. für Dateinhalte) existieren, sind aber nicht dargestellt.

Block	0	1	2	3	4	5	6
Typ	SB	AB	IAB	IB	IB	VB	VB
Inhalt	free: 90 root: I <sub>0</sub>	0–10	0–3	I <sub>0</sub> → 5/32 I <sub>1</sub> → 6/64 I <sub>2</sub> → 7–9/9000	I <sub>3</sub> → 10/3000	users → I <sub>1</sub>	stefan → I <sub>2</sub> ede → I <sub>3</sub>

Inode-Verweis für  
Wurzelverzeichnis

Blöcke 0 bis  
10 belegt

Inodes 0 bis  
3 belegt

Blockzeiger für  
Inodes, Größe (in Byte)

Abbildung von Dateinamen  
auf zugehörige Inodes

Einträge in den Inodeblöcken haben die Form  
 <Inode-Nummer> → <Liste der Datenblöcke>/<Datei- bzw. Verzeichnisgröße>

b) Welche Blöcke des oben gezeigten Dateisystems müssen modifiziert bzw. neu geschrieben werden, wenn eine Datei `/users/julia` mit einer Größe von 1 KiB in das Dateisystem eingefügt werden soll?

7 P

c) Was ist auf Geräteebene die Ursache dafür, dass es beim unvollständigen Ausführen einer Dateisystemmodifikation zu Inkonsistenzen kommen kann?

1 P

d) Nennen Sie je ein Beispiel für eine reparable (unkritische) und eine nicht reparable (kritische) Inkonsistenz, die bei der in Teilaufgabe b) beschriebenen Modifikation auftreten können.

2 P

e) Nennen Sie zwei Mechanismen, um das Entstehen von Inkonsistenzen in Dateisystemen zu verhindern.

2 P

f) Wir nehmen an, dass das Dateisystem aus Aufgabe b) unter `/home` eingehängt wurde. Ist es in diesem System möglich von der Datei `/home/users/stefan` einen Hardlink namens `/vo65` anzulegen? Begründen Sie ihre Antwort.

1 P



# Aufgabe 6 – Scheduling

16 Punkte

a) Ein Einprozessor-Betriebssystem verwaltet drei Prozesse  $P_1$ ,  $P_2$  und  $P_3$ . Die Prozesse treffen in dieser Reihenfolge im System ein und sind alle zum Zeitpunkt  $t = 0$  rechenbereit. Die Prozesse wiederholen sich unendlich lange. Nach jedem CPU-Stoß führen die Prozesse einen E/A-Stoß durch. Die CPU-Stöße (in ms) und E/A-Stöße (in ms) der Prozesse sind in der folgenden Tabelle angegeben:

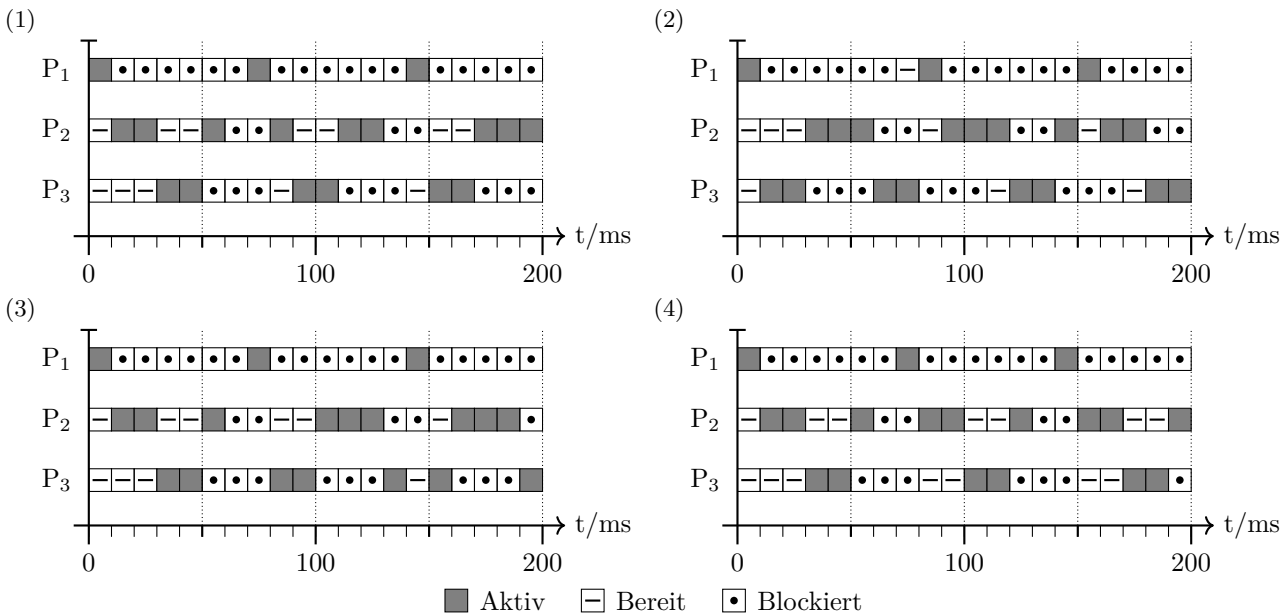
Prozesse	$P_1$	$P_2$	$P_3$
CPU-Stöße	10	30	20
E/A-Stöße	60	20	30

Gegeben ist die folgende Auswahl bezüglich CPU-Zuteilungsstrategien:

- (i) Round Robin (RR) mit einer Zeitscheibenlänge von 20 ms
- (ii) Virtual Round Robin (VRR) mit einer Zeitscheibenlänge von 20 ms
- (iii) Shortest Remaining Time First (SRTF) mit Kenntnis über die Prozesslaufzeiten
- (iv) Keine der zuvor genannten

Nachfolgend sind vier Gantt-Diagramme abgebildet, die eine mögliche Einplanung der Prozesse über 200 ms darstellen. Ordnen Sie der oben genannten Auswahl bezüglich CPU-Zuteilungsstrategien jeweils eines der Gantt-Diagramme zu.

HINWEIS: Für jede korrekte Zuordnung werden zwei Punkte vergeben. Bei einer falschen Zuordnung wird ein Punkt abgezogen, wobei die Aufgabe nicht mit negativer Punktzahl bewertet werden kann. **8 P**



---

HINWEIS: Am Ende der Seite steht ein kariertes Bereich zum Zeichnen für alle Teilaufgaben zur Verfügung. Bitte notieren Sie dabei jeweils, zu welcher Teilaufgabe die Zeichnung gehört.

---

In einem Einprozessor-Echtzeitsystem sind drei periodische Tasks  $T_1$ ,  $T_2$  und  $T_3$  einzuplanen mit folgenden Werten ( $p_i$  Periodenlänge,  $t_i$  Bearbeitungszeit, Periodenende = Zeitschranke):

$$p_1 = 4, t_1 = 1$$

$$p_2 = 3, t_2 = 1$$

$$p_3 = 6, t_3 = 2$$

Zwischen den Tasks bestehen keine Abhängigkeiten, und sie sind an beliebiger Stelle unterbrechbar. Alle Tasks sind zum Zeitpunkt  $t = 0$  rechenbereit.

- b) Ist die Taskmenge einplanbar mittels statischer Prioritäten? Falls ja, schlagen Sie einen Algorithmus vor und geben Sie die Zuordnung der Prioritäten an. Ansonsten begründen Sie Ihre Aussage.

HINWEIS:  $2^{1/2} \approx 1,41$ ;  $2^{1/3} \approx 1,26$ ;  $2^{1/4} \approx 1,19$ ;  $2^{1/5} \approx 1,15$

**5 P**

- c) Es stellt sich heraus, dass  $T_3$  zu  $T'_3$  angepasst werden muss und  $T'_3$  eine Bearbeitungszeit von  $t'_3 = 2,1$  benötigt und eine Periode von nach wie vor  $p'_3 = 6$  hat. Ist die dadurch entstehende Taskmenge  $\{T_1, T_2, T'_3\}$  mittels statischer Prioritäten einplanbar?

**1 P**

- d) Wenn kein Task aus der Taskmenge  $\{T_1, T_2, T'_3\}$  herausgenommen werden soll, welches Verfahren würde infrage kommen, um die Taskmenge einplanen zu können, wenn statische Prioritäten keine Bedingung mehr sind? Warum ist Ihr gewähltes Verfahren geeignet?

**2 P**

