

# Grundlagen der Informatik III

Wintersemester 2010/2011

Wolfgang Heenes, Patrik Schmittat



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## 11. Aufgabenblatt

31.01.2011

**Hinweis:** Der Schnelltest und die Aufgaben sollen in den Übungsgruppen bearbeitet werden. Die Hausaufgaben sind in der Kalenderwoche 6 (07.02. bis 11.02.) bei den Tutoren in **physikalischer Form** (handschriftlich oder gedruckt) abzugeben. Bei allen Abgaben ist der Name des Tutors und die Übungsgruppe deutlich anzugeben. Bei Teamabgaben wird nur eine Lösung eingereicht, die alle Namen der Teammitglieder enthält.

Schicken Sie Ihre Lösungen von Programmieraufgaben zusätzlich zur schriftlichen Abgabe per E-Mail an Ihren Tutor. Kommentieren Sie Ihren Quellcode.

### Aufgabe 1: Schnelltest

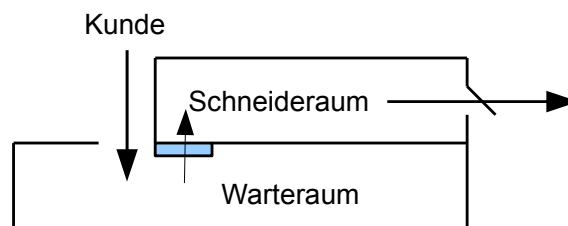
Fragen	Antworten
1. Welche der folgenden Aussagen sind korrekt ?	<input type="checkbox"/> Im Benutzermodus sind privilegierte (Prozessor)-Befehle verfügbar. <input type="checkbox"/> Asynchrone Interrupts sind im Allgemeinen nicht reproduzierbar. <input type="checkbox"/> Das Betriebssystem behandelt Interrupts selbständig. <input type="checkbox"/> Multithreading sorgt für parallele Ausführung mehrerer Prozesse. <input type="checkbox"/> Threads teilen sich einen gemeinsamen Adressraum.
2. Zur Unterbrechungsbehandlung muss	<input type="checkbox"/> das Prozessorstatuswort gesichert werden <input type="checkbox"/> das IC-Register neu geladen werden. <input type="checkbox"/> der Cache vollständig geleert werden.
3. Welche Aussagen zu Semaphoren sind richtig?	<input type="checkbox"/> Mit Semaphoren läßt sich wechselseitiger Ausschluss verhindern. <input type="checkbox"/> Mit Semaphoren läßt sich wechselseitiger Ausschluss realisieren. <input type="checkbox"/> Semaphoren können Verklemmungen nicht verhindern. <input type="checkbox"/> Das Verändern von Semaphoren muss atomar ablaufen
4. Bei welchen der folgenden Schedulingverfahren können Prozesse verhungern?	<input type="checkbox"/> FIFO <input type="checkbox"/> Shortest Job First <input type="checkbox"/> Round Robin <input type="checkbox"/> Statische Prioritätenvergabe <input type="checkbox"/> Dynamische Prioritätenvergabe

## Aufgabe 2: Betriebssysteme

- Was ist im Zusammenhang von Prozesssynchronisation mit dem Begriff „atomar“ gemeint? Warum müssen die Semaphore Operationen „P“ und „V“ atomar sein?
- Was sind die Anforderungen an eine Realisierung des wechselseitigen Ausschlusses?

## Aufgabe 3: Semaphore: Sleeping Barber Problem

Der Laden eines Friseurs besteht aus einem Warteraum, der beliebig viele Kunden aufnehmen kann und einem Haarschneiderraum. Beide Räume besitzen eine gemeinsame, jeweils nur einen Eingang schließende Schiebetür.



- Die Kunden betreten nacheinander den Warteraum.
- Hat der Friseur einen Haarschnitt beendet, so verläßt der Kunde den Schneiderraum und der Friseur öffnet die Tür zum Warteraum und schaut nach, ob ein weiterer Kunde wartet.
  - Ist dies der Fall, so bittet er diesen in den Schneiderraum;
  - andernfalls setzt er sich auf seinen Stuhl im Schneiderraum und schläft.
- Tritt ein Kunde in den Warteraum und findet den Friseur schlafend,
  - so weckt er diesen und läßt sich dann die Haare schneiden;
  - andernfalls wartet der Kunde.

Schreiben Sie Algorithmen für die Verhaltensweise des Friseurs und der Kunden unter Verwendung von Semaphoren. Als Syntax bietet sich die Systemprogrammiersprache aus der Vorlesung an.

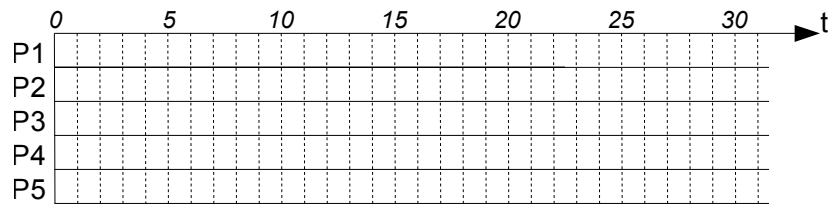
## Aufgabe 4: Scheduling

Gegeben seien fünf Prozesse mit jeweiligen Start- und CPU-Zeiten, sowie deren Priorität (Prozesse mit kleinerem Prioritätswert kommen vor Prozessen mit größeren Werten):

Prozess	Startzeit	CPU-Zeit	Statische Priorität
P1	0	8	6
P2	2	4	7
P3	3	2	3
P4	3	7	4
P5	6	3	2

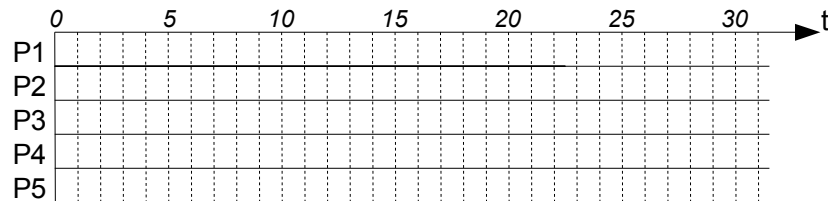
Betrachten Sie nur die Prozesszustände bereit (B) und laufend (L).

Zur Darstellung des Scheduling sind Gantt-Diagramm sehr gut geeignet (vgl. Vorlesung 24). Ein Gantt-Diagramm hat folgendes Aussehen:

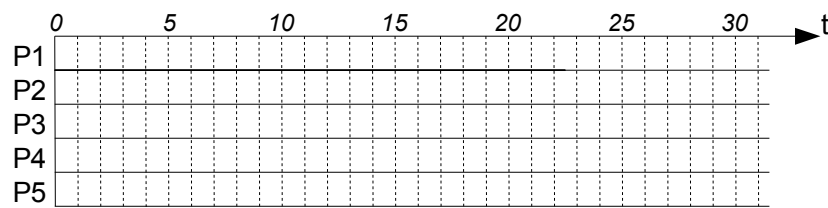


Zeichnen Sie für jedes der folgenden Schedulingverfahren ein Gantt-Diagramm der Prozesse.

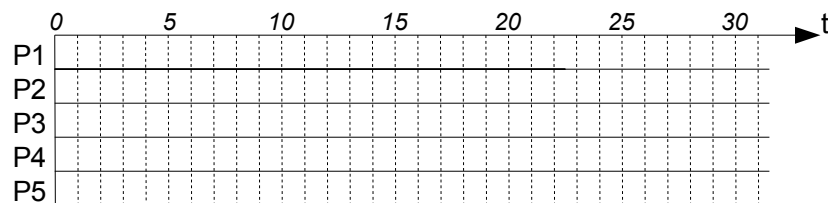
a) Statische Priorität (non-preemptive):



b) Shortest Job First (non-preemptive):



c) Shortest remaining time next (preemptive):

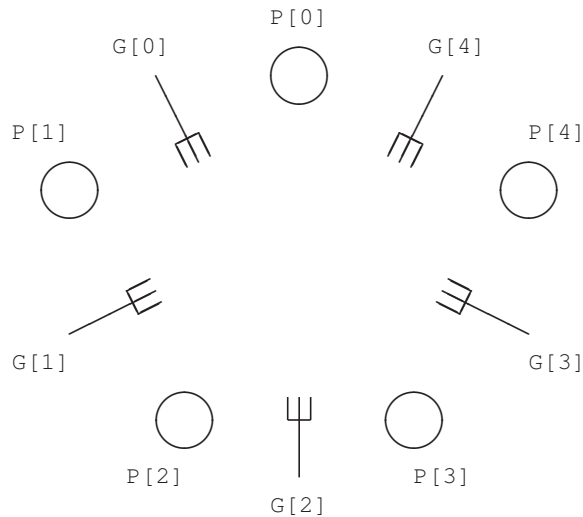


d) Berechnen Sie die mittlere Wartezeit für jedes Schedulingverfahren.

Prozess	Prio.	SJF	SRTN
P1			
P2			
P3			
P4			
P5			
Mittlere Wartezeit			

## Hausaufgabe 1: Semaphore: 5-Philosophenproblem (6 Punkte)

Fünf Philosophen sitzen gemeinsam an einem runden Tisch, an dem sie unabhängig voneinander von Zeit zu Zeit Spaghetti essen. Jeder Philosoph hat rechts neben seinem Teller nur eine Gabel, benötigt aber zum Essen zwei Gabeln, also auch die seines linken Nachbarn. Aus diesem Grund können nicht alle Philosophen gleichzeitig essen, sondern sie müssen sich bezüglich ihrer kritischen Abschnitte **Essen** synchronisieren



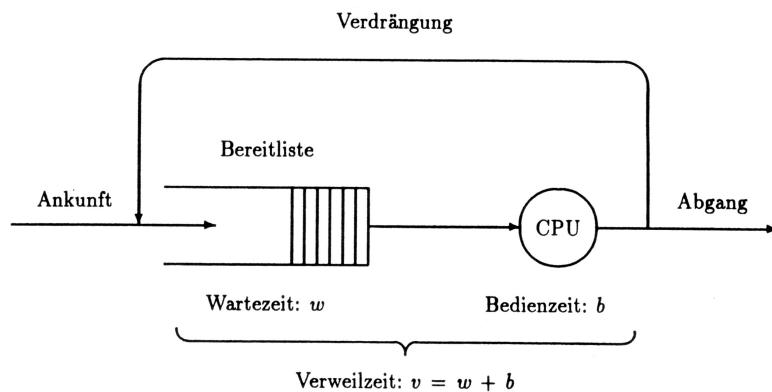
- a) Formulieren Sie die Synchronisationsbedingung mittels eines Ausschlußgraphen.
- b) Geben Sie eine Programmlösung unter Verwendung der P- und V-Operationen an, bei der es zu keinerlei Verklemmungen kommen kann. Als Syntax für die Programmlösung bietet sich die Systemprogrammiersprache aus der Vorlesung an.

Hinweise:

- Die Philosophen sollten als Prozesse modelliert werden.
- Jeder Philosoph erlebt zyklisch drei Zustände
  - \* denkend,
  - \* hungrig und
  - \* essend
- Die Gabeln stellen die Betriebsmittel dar.

## Hausaufgabe 2: Scheduling (4 Punkte)

Die folgende Abbildung beschreibt ein einfaches Modell für die Prozeßbearbeitung.



Fünf Prozesse treffen zu gegebenen Zeitpunkten in der Bereitliste ein. Es ist bekannt, wieviel Bedienzeit (=Rechenzeit) sie benötigen. Jeder Prozess hat eine Priorität (0 stellt die höchste Priorität dar).

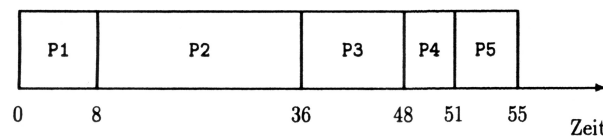
Die folgende Tabelle gibt die Ankunftszeitpunkte, Bedienzeiten und Prioritäten der einzelnen Prozesse wieder.

Prozeß	Ankunftszeit	Bedienzeit	Priorität
1	0	8	4
2	3	28	1
3	7	12	0
4	9	3	2
5	15	4	3

Die Prozesse sollen nun unter verschiedenen Dispatcher-Strategien laufen.

- Ohne Verdrängung, d. h. nach der Zuteilung der CPU arbeitet ein Prozess, bis er von selbst die Prozessor abgibt.
  - a) First-In-First-Out (FIFO)
  - b) Highest-Priority-First (HPF)
  - c) Shortest-Processing-Time-First (SPTF)  
Dies entspricht einer HPF-Strategie, bei der die Priorität nach der Dauer der Bedienzeit vergeben werden.
- Mit Verdrängung
  - d) Round-Robin (RR) mit dem Zeitscheibenwert 5
  - e) HPF mit Verdrängung bei der Ankunft eines Prozesses (mit höherer Priorität)
  - f) Shortest-Remaining-Time-First  
Dies entspricht einer HPF-Strategie, bei der nach der Ankunft eines neuen Prozesses derjenige ausgewählt wird, der die kürzeste Restbedienzeit besitzt.

Das Gantt-Diagramm für die Dispatcher-Strategie a) ist in der folgenden Abbildung angegeben.



- a) Zeichnen Sie die Gantt-Diagramme für die Dispatcher-Strategien b) bis f).