



**Systeme I - WS 07/08**  
**Musterlösung - Übungsblatt 3 - Theorie**

**Aufgabe 1:**

**Prozess (P) Laufzeit**

- 1 9
- 2 6
- 3 3
- 4 5
- 5 X

Die mittlere Durchlaufzeit ist optimal bei einem Shortest- Job First Scheduler.

Fällunterscheidung:

- $3 > X \Rightarrow P5, P3, P4, P2, P1$
- $3 = X \Rightarrow P5, P3, P4, P2, P1$  oder  $P3, P5, P4, P2, P1$
- $3 < X < 5 \Rightarrow P3, P5, P4, P2, P1$
- $X = 5 \Rightarrow P3, P5, P4, P2, P1$  oder  $P3, P4, P5, P2, P1$
- $5 < X < 6 \Rightarrow P3, P4, P5, P2, P1$
- $X = 6 \Rightarrow P3, P4, P5, P2, P1$  oder  $P3, P4, P2, P2, P1$
- $6 < X < 9 \Rightarrow P3, P4, P2, P5, P1$
- $X = 9 \Rightarrow P3, P4, P2, P1, P5$  oder  $P3, P4, P2, P5, P1$
- $X > 9 \Rightarrow P3, P4, P2, P1, P5$

**Aufgabe 2:**

FCFS: Kann nicht verhungern, da Ausführung strikt nach Eingangsreihenfolge und nicht präemptiv.

Round-Robin: Ähnlich wie FCFS, aber präemptiv. Der am längsten wartende Prozess wird aktiviert. Somit kann kein Prozess verhungern, da jeder nach endlicher Zeit aktiviert wird.

HRRN: Kann nicht verhungern, da die Wartezeit beachtet wird. Je länger ein Prozess wartet, um so größer sind seine Chancen zu laufen.

Feedback: Können prinzipiell verhungern! Mit längerer Laufzeit sinkt seine Priorität. Wenn zu jedem Zeitpunkt Prozesse mit höherer Priorität verfügbar sind, verhungern die Prozesse mit niedriger Priorität.

**Aufgabe 3:**

Prozess	Zedpunkt	Laufzeit
A	t=0	8
B	t=0	4
C	t=2	4
D	t=6	4

Mit Scheduling Shortest- Job First:

t=0 B, A da Laufzeit von B kleiner als Laufzeit von A.  
 t=12 C,D

Mittlere Durchlaufzeit:

Prozess	Durchlaufzeit
B	4 (fertig zu t=4, lauffertig seit t=0)
A	12 (fertig zu t=12, lauffertig seit t=0)
C	14 (fertig zu t=16, lauffertig seit t=2)
D	14 (fertig zu t=20, lauffertig seit t=6)

$$\frac{4+12+14+14}{4} = 11$$

Shortest Remaining Time (nicht präemptiv!):

- t=0 scheidung: B, A
- t=2 C wird lauffertig
- t=4 B beendet, suche erneut nach SRT  $\Rightarrow$  C wird ausgewählt.
- t=6 D wird lauffertig.
- t=8 C beendet, suche erneut nach SRT  $\Rightarrow$  D wird ausgewählt.
- t=12 D beendet, A einzig verbleibender Prozess.

Prozess	Durchlaufzeit
B	4 (fertig zu t=4, lauffertig seit t=0)
C	6 (fertig zu t=8, lauffertig seit t=2)
D	6 (fertig zu t=12, lauffertig seit t=6)
A	20 (fertig zu t=20, lauffertig seit t=0)

Mittlere Durchlaufzeit:

$$\frac{4+6+6+20}{4} = 9$$

#### Aufgabe 4:

**Angenommen:** P0 im kritischen Abschnitt und P1 betritt den kritischen Abschnitt.

- $t_0$ : P1 betritt den kritischen Abschnitt.
- $t_1 < t_0$ : Zeitpunkt zu dem P1 zum letzten Mal `solange(flag[0] = true)` durchlaufen hat. Zu diesem Zeitpunkt befindet sich P0 nach Annahme im kritischen Abschnitt.
- $t_2 < t_1$ : P0 zum letzten Mal `solange(flag[1] = true)` durchlaufen hat.
- $t_3 < t_2$ : P0 `flag[0] = true` gesetzt hat.

D.h. es muss einen Zeitpunkt  $t'$  mit  $t_3 < t' < t_1$  gegeben haben zu dem `flag[0]=false` gesetzt wurde.

Dies kann aber nur geschehen sein:

1. bei der Initialisierung
2. indem P0 den kritischen Abschnitt verlassen hat.

Da, nach Annahme, aber noch P0 im kritischen Abschnitt ist, gibt es keine Möglichkeit in der `flag[0]=false` gesetzt werden könnte.

Widerspruch: P0 und P1 können nicht zusammen im kritischen Abschnitt sein.  
⇒ wechselseitigen Ausschluss ist garantiert!