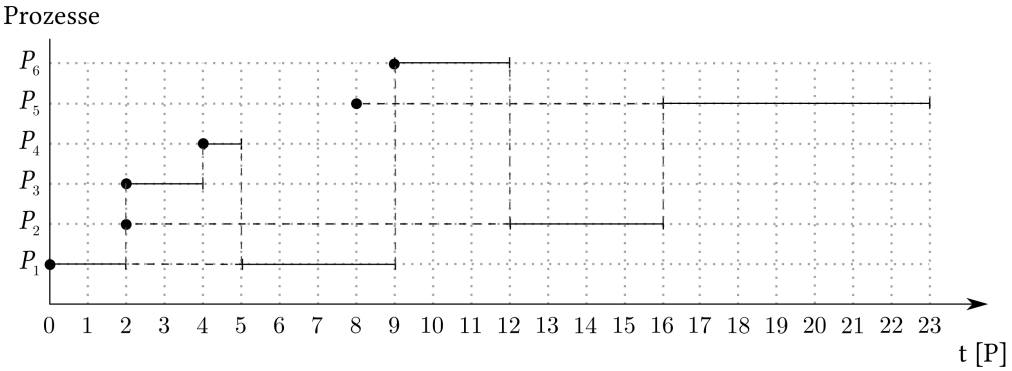
Betriebsysteme (WS19/20)
Übungsblatt 5

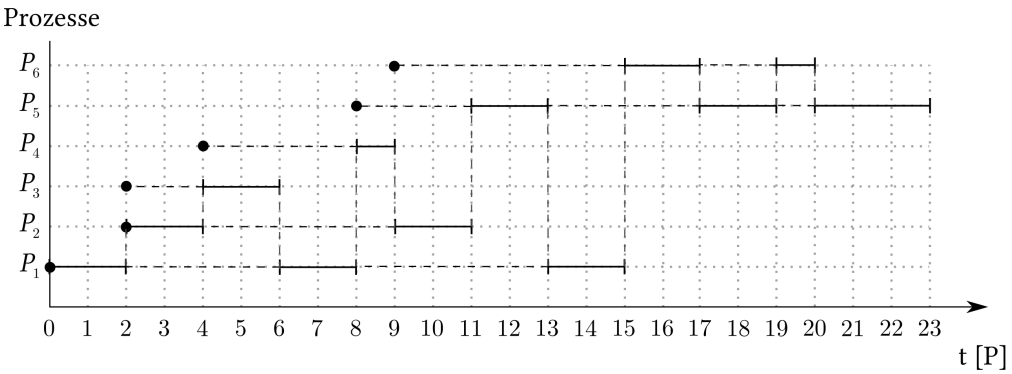
Yudong Sun
12141043

24. November 2019

Aufgabe H23 (a) **SRPT**



(b) **RR**



(c) **SRPT**

Proz.	Ankunftz.	Bedienz.	Beendigungsz.	Verweildauer	Nom. Vwd.	Wartezeit
P_1	0	6	9	9	$9/6 = 1,5$	3
P_2	2	4	16	14	$14/4 = 3,5$	10
P_3	2	2	4	2	$2/2 = 1$	0
P_4	4	1	5	1	$1/1 = 1$	0
P_5	8	7	23	15	$15/7 = 2,14$	8
P_6	9	3	12	3	$2/3 = 0,66$	0

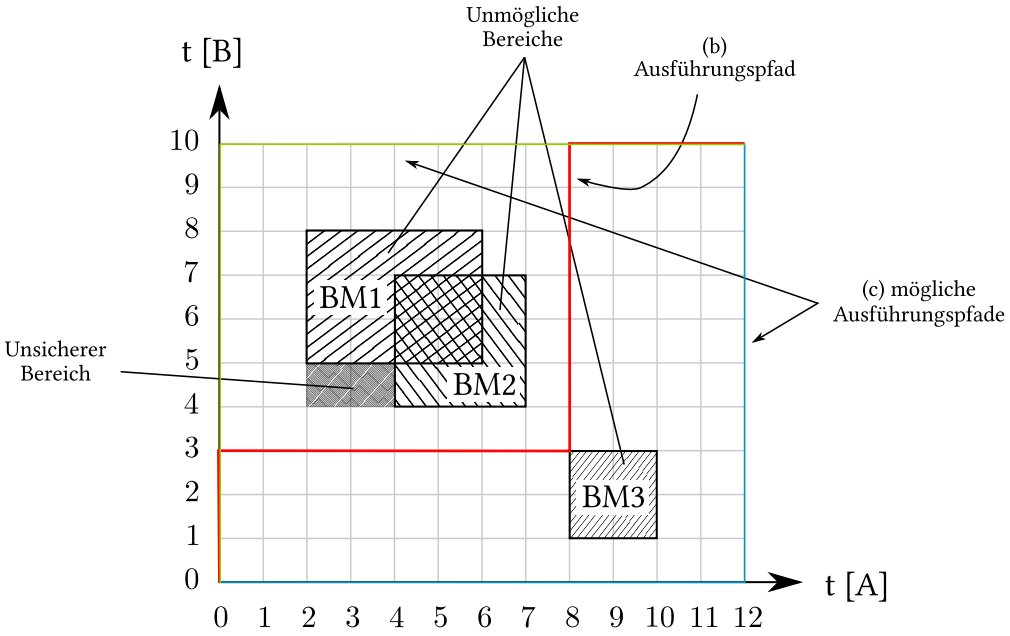
Mittlere Verweildauer $= (9 + 14 + 2 + 1 + 15 + 3)/6 = 7,33$
Mittlere normalisierte Verweildauer $= \left(9/6 + 14/4 + 2/2 + 1/1 + 15/7 + 2/3\right)/6 = 1,63$
Mittlere Wartezeit $= (3 + 10 + 0 + 0 + 8 + 0)/6 = 3,50$

RR

Proz.	Ankunftz.	Bedienz.	Beendigungsz.	Verweildauer	Nom. Vwd.	Wartezeit
P_1	0	6	15	15	$15/6 = 2,5$	9
P_2	2	4	11	9	$9/4 = 2,25$	5
P_3	2	2	6	4	$4/2 = 2$	2
P_4	4	1	9	5	$5/1 = 1$	4
P_5	8	7	23	15	$15/7 = 2,14$	8
P_6	9	3	20	11	$11/3 = 3,67$	8

Mittlere Verweildauer $= (15 + 9 + 4 + 5 + 15 + 11)/6 = 9,83$
Mittlere normalisierte Verweildauer $= \left(15/6 + 9/4 + 4/2 + 5/1 + 15/7 + 11/3\right)/6 = 2,93$
Mittlere Wartezeit $= (9 + 5 + 2 + 4 + 8 + 8)/6 = 6,00$

Aufgabe H26 (a)



Es führt zu einem Deadlock, wenn der Ausführungspfad in den unsicheren Bereich kommt. Da der Ausführungspfad nur rechts und oben weitergehen kann, kommt der Graph unweigerlich in den unmöglichen Bereich.

(b) Sehen Sie bitte das obige Teil (a).

(c) Nein, es kann nicht bei nicht-präemptivem Scheduling zu einem Deadlock kommen. Bei nicht-präemptivem Scheduling sind Prozesse ohne Unterbrechungen hintereinander durchgeführt. Das heißt, dass kein Prozess muss auf einem anderem Prozess warten. Dabei kann auch kein Deadlock entstehen.

Sehen Sie bitte das obige Teil (a) für die mögliche Abarbeitung von Prozess A und B.

(d) Je nach dem Scheduling-Algorithm kann es zu vielen verschiendenen Abläufen führen. Man kann aber 6 prinzipiell verschiendenen Möglichkeiten bestimmen, um die Prozesse A und B erfolgreich terminieren zu lassen:

- $A(1) \Rightarrow A(2) \Rightarrow A'(1) \Rightarrow A'(2) \Rightarrow A(3) \Rightarrow A'(3) \Rightarrow B(3) \Rightarrow B'(3) \Rightarrow B(2) \Rightarrow B(1) \Rightarrow B'(2) \Rightarrow B'(1)$ (blauer Pfad)
- $B(3) \Rightarrow B'(3) \Rightarrow B(2) \Rightarrow B(1) \Rightarrow B'(2) \Rightarrow B'(1) \Rightarrow A(1) \Rightarrow A(2) \Rightarrow A'(1) \Rightarrow A'(2) \Rightarrow A(3) \Rightarrow A'(3)$ (grüner Pfad)
- $B(3) \Rightarrow B'(3) \Rightarrow A(1) \Rightarrow A(2) \Rightarrow A'(1) \Rightarrow A'(2) \Rightarrow B(2) \Rightarrow B(1) \Rightarrow B'(2) \Rightarrow B'(1) \Rightarrow A(3) \Rightarrow A'(3)$ (roter Pfad)
- $A(1) \Rightarrow B(3) \Rightarrow B'(3) \Rightarrow A(2) \Rightarrow A'(1) \Rightarrow A'(2) \Rightarrow B(2) \Rightarrow B(1) \Rightarrow B'(2) \Rightarrow B'(1) \Rightarrow A(3) \Rightarrow A'(3)$
- $B(3) \Rightarrow B'(3) \Rightarrow A(1) \Rightarrow A(2) \Rightarrow A'(1) \Rightarrow A'(2) \Rightarrow B(2) \Rightarrow B(1) \Rightarrow B'(2) \Rightarrow B'(1) \Rightarrow A(3) \Rightarrow A'(3)$
- $A(1) \Rightarrow B(3) \Rightarrow B'(3) \Rightarrow A(2) \Rightarrow A'(1) \Rightarrow A'(2) \Rightarrow B(2) \Rightarrow B(1) \Rightarrow B'(2) \Rightarrow B'(1) \Rightarrow A(3) \Rightarrow A'(3)$

wobei

$K(i)$	BMi wird von Prozess k besitzt
$K'(i)$	Prozess K gibt BMi zurück

Aufgabe H27 Sehen Sie bitte `u05-h27.txt`