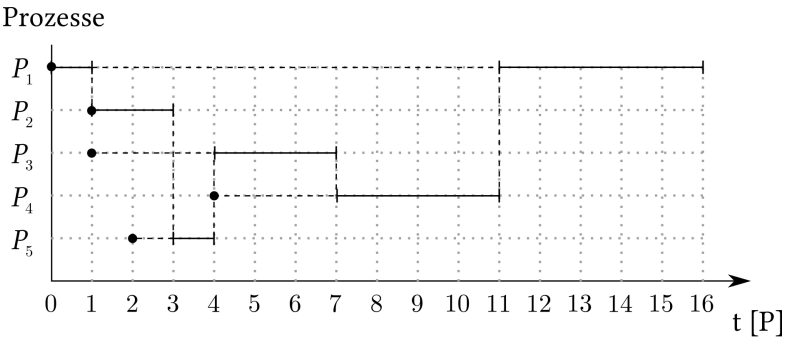
Betriebsysteme (WS19/20)
Übungsblatt 4

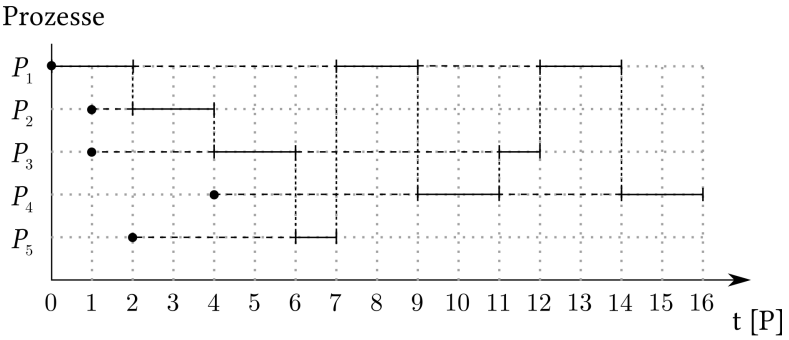
Yudong Sun
12141043

17. November 2019

Aufgabe H20 (a) **SRPT**



(b) **RR**



(c) **SRPT**

Proz.	Ankunftz.	Bedienz.	Beendigungsz.	Verweildauer	Wartezeit
P_1	0	6	16	16	10
P_2	1	2	3	2	0
P_3	1	3	7	6	3
P_4	4	4	11	7	3
P_5	2	1	4	2	1

Mittlere Verweildauer $= (16 + 2 + 6 + 7 + 2)/5 = 6,60$
Mittlere Wartezeit $= (10 + 0 + 3 + 3 + 1)/5 = 3.40$

RR

Proz.	Ankunftz.	Bedienz.	Beendigungsz.	Verweildauer	Wartezeit
P_1	0	6	14	14	8
P_2	1	2	4	3	1
P_3	1	3	12	11	8
P_4	4	4	16	12	8
P_5	2	1	7	5	4

Mittlere Verweildauer $= (14 + 3 + 11 + 12 + 5)/5 = 9,00$
Mittlere Wartezeit $= (8 + 1 + 8 + 8 + 4)/5 = 5,80$

(d) Wenn die Zeitscheibe zu lang ist, funktioniert eine RR Stratgie wie eine FCFS Strategie. Ein Nachteil davon ist, dass andere Prozessen unnötig verhungern können, wenn ein Prozess auf ein Ereignis wartet muss, aber den Prozessor nicht sofort frei gibt.

Aufgabe H21 (a) Stellen Sie vor, dass jeder Philosoph hunger hat und gleichzeitig seine linke Stäbchen nimmt. Jeder wartet auf seine rechte Stäbchen. Aber weil alle Hunger haben, legt niemand seine linke Stäbchen. Da es nur 5 Stäbchen gibt, führt es zu einem Deadlock.

(b)
- Jeder Philosoph muss auf der Bedingung warten, dass beide Stäbchen frei sein müssen, bevor er die Stäbchen nimmt und isst. Falls eines davon besetzt ist, dann nimmt er keine Stäbchen.
- Zu jedem Zeitpunkt darf nur ein Philosoph essen. Die andere Philosophen, die Hunger haben, muss in einer Warteschlange warten.

Diesen obengennante 2 Möglichkeiten lässt sich die Deadlockssituation vermeiden.

Aufgabe H22 Sehen Sie bitte `u04-h22.txt`