

1

1.a

Prozesse	P_1	P_2	P_3	P_4	P_5	P_6
Ankunftszeit	0	4000	5000	39000	42000	43000
Rechenzeit	15000	20000	5000	50000	25000	10000

Variablen:

t ist die Zeit, P der momentan ausgewählte Prozess, W die Menge der bereiten Prozesse und ggf. deren verbleibender Ausführungszeit und P_{complete} die Menge der abgeschlossenen Prozesse.

Unterbrechendes Scheduling mit FCFS

t	P	W	P_{complete}
0	P_1	$\{P_2, P_3\}$	\emptyset
15000	P_2	$\{P_3\}$	$\{P_1\}$
35000	P_3	$\{P_4\}$	$\{P_1, P_2\}$
40000	P_4	$\{P_5, P_6\}$	$\{P_1, P_2, P_3\}$
90000	P_5	$\{P_6\}$	$\{P_1, P_2, P_3, P_4\}$
115000	P_6	\emptyset	$\{P_1, P_2, P_3, P_4, P_5\}$
125000		\emptyset	$\{P_1, P_2, P_3, P_4, P_5, P_6\}$

Resultat:

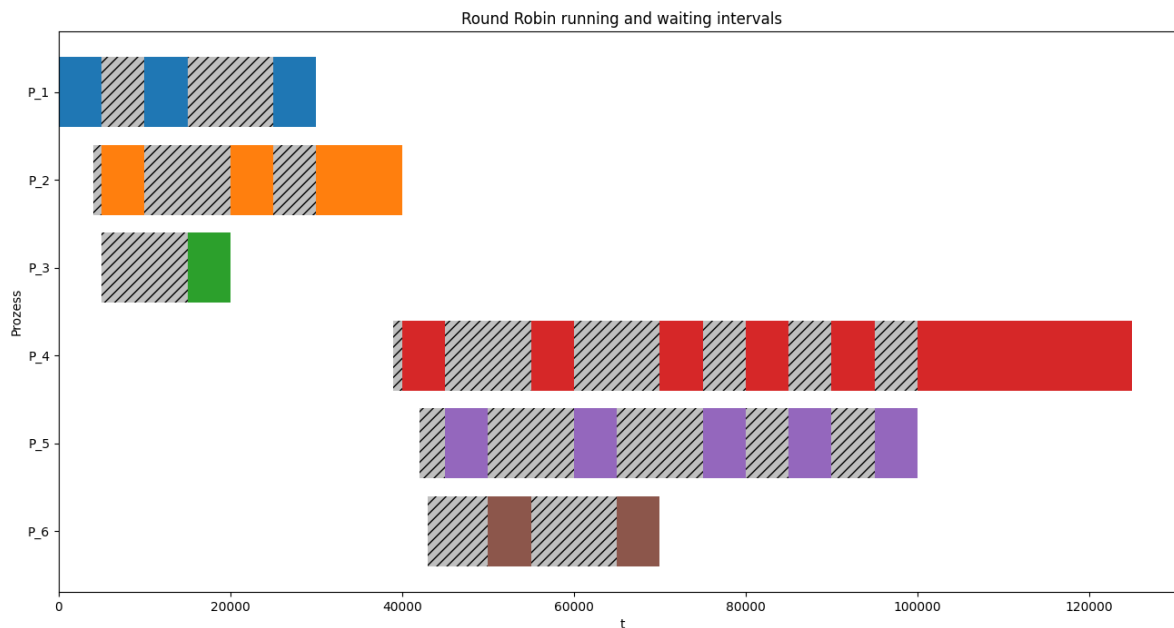
Processing math: 100%

id	t_{arrival}	t_{start}	t_{end}	w
P_1	0	0	15000	0
P_2	4000	15000	35000	11000
P_3	5000	35000	40000	30000
P_4	39000	40000	90000	1000
P_5	42000	90000	115000	48000
P_6	43000	115000	125000	72000

$$\bar{w} = 27000$$

Nicht-unterbrechendes Scheduling mit Round-Robin

Die Länge eines Zeitschlitzes beträgt in diesem Beispiel 5000 Zeiteinheiten.



t	P	W	P_{complete}
0	P_1	\emptyset	\emptyset
5000	P_2	$\{(P_1, 10000)\}$	\emptyset

Processing math: 100%

t	P	W	P_{complete}
10000	P_1	$\{(P_3, 5000), (P_2, 15000)\}$	\emptyset
15000	P_3	$\{(P_2, 15000), (P_1, 5000)\}$	\emptyset
20000	P_2	$\{(P_1, 5000)\}$	$\{P_3\}$
25000	P_1	$\{(P_2, 10000)\}$	$\{P_3\}$
30000	P_2	\emptyset	$\{P_3, P_1\}$
35000	P_2	\emptyset	$\{P_3, P_1\}$
40000	P_4	\emptyset	$\{P_3, P_1, P_2\}$
45000	P_5	$\{(P_6, 10000), (P_4, 45000)\}$	$\{P_3, P_1, P_2\}$
50000	P_6	$\{(P_4, 45000), (P_5, 20000)\}$	$\{P_3, P_1, P_2\}$
55000	P_4	$\{(P_5, 20000), (P_6, 5000)\}$	$\{P_3, P_1, P_2\}$
60000	P_5	$\{(P_6, 5000), (P_4, 40000)\}$	$\{P_3, P_1, P_2\}$
65000	P_6	$\{(P_4, 40000), (P_5, 15000)\}$	$\{P_3, P_1, P_2\}$
70000	P_4	$\{(P_5, 15000)\}$	$\{P_3, P_1, P_2, P_6\}$
75000	P_5	$\{(P_4, 35000)\}$	$\{P_3, P_1, P_2, P_6\}$
80000	P_4	$\{(P_5, 10000)\}$	$\{P_3, P_1, P_2, P_6\}$
85000	P_5	$\{(P_4, 30000)\}$	$\{P_3, P_1, P_2, P_6\}$
90000	P_4	$\{(P_5, 5000)\}$	$\{P_3, P_1, P_2, P_6\}$
95000	P_5	$\{(P_4, 25000)\}$	$\{P_3, P_1, P_2, P_6\}$

Processing math: 100%

t	P	W	P_{complete}
100000	P_4	\emptyset	$\{P_3, P_1, P_2, P_6, P_5\}$
125000		\emptyset	$\{P_3, P_1, P_2, P_6, P_5, P_4\}$

Resultat:

id	t_{arrival}	t_{start}	t_{end}	w	A
P_3	5000	15000	20000	10000	$\{(15000, 20000)\}$
P_1	0	25000	30000	15000	$\{(0, 5000), (10000, 15000), (25000, 30000)\}$
P_2	4000	35000	40000	16000	$\{(5000, 10000), (20000, 25000), (30000, 35000), (35000, 40000)\}$
P_6	43000	65000	70000	17000	$\{(50000, 55000), (65000, 70000)\}$
P_5	42000	95000	100000	33000	$\{(45000, 50000), (60000, 65000), (75000, 80000), (85000, 90000)\}$
P_4	39000	100000	125000	36000	$\{(40000, 45000), (55000, 60000), (70000, 75000), (80000, 85000)\}$

$$\bar{w} = 21166.666$$

1.b

Das allgemeine Kriterium für die Auswahl des nächsten zu aktivierenden Prozesses sowohl im nicht-unterbrechenden als auch im unterbrechenden Scheduling ist die kürzeste verbleibende Ausführungszeit. Dieses Kriterium lässt sich durch das Schedulingverfahren SJN für nicht-unterbrechendes Scheduling und SRT für unterbrechendes Scheduling erfüllen.

1. Nicht-unterbrechendes Scheduling: SJN

- Wenn ein Prozess beendet wird und der Prozessor einen neuen zu aktivierenden Prozess auswählt, wählt er den Prozess mit der kürzesten Gesamtausführungszeit aus der Menge der bereiten Prozesse
- Wenn es mehrere Prozesse mit der gleichen Ausführungszeit gibt, kann der Scheduler den Prozess auswählen, der zuerst eingetroffen ist (First Come First Serve - FCFS), oder andere Mechanismen zur Aufhebung von Gleichständen verwenden.
- Indem der Scheduler den Prozess mit der kürzesten verbleibenden Ausführungszeit als nächstes auswählt, wird sichergestellt, dass kürzere Prozesse schnell beendet werden,

Processing math: 100%

wodurch sich die Wartezeit für alle Prozesse in der Warteschlange verkürzt.

2. Unterbrechendes Scheduling: SRT

- In diesem Fall sollte der Scheduler die verbleibende Ausführungszeit der Prozesse berücksichtigen, nicht nur ihre Gesamtausführungszeit. Wenn ein neuer Prozess eintrifft oder der aktuelle Prozess unterbrochen wird, wählt der Planer den Prozess mit der kürzesten verbleibenden Ausführungszeit aus der Menge der bereiten Prozesse aus.
- Wenn es mehrere Prozesse mit der gleichen verbleibenden Ausführungszeit gibt, kann der Scheduler dieselben Mechanismen zur Aufhebung des Gleichstands wie oben erwähnt verwenden.
- Indem der Scheduler den Prozess mit der kürzesten verbleibenden Zeit zuerst auswählt, wird sichergestellt, dass Prozesse mit weniger verbleibender Zeit schnell beendet werden, wodurch die durchschnittliche Wartezeit für alle Prozesse in der Warteschlange verringert wird.

Durch die konsequente Auswahl des Prozesses mit der kürzesten verbleibenden Ausführungszeit ermöglicht der Scheduler, dass kürzere Prozesse schneller abgeschlossen werden können, wodurch sich die Gesamtwartezeit für alle Prozesse in der Warteschlange verringert. Dieses Kriterium trägt dazu bei, die durchschnittliche Wartezeit für das System zu minimieren und gleichzeitig die konkurrierenden Anforderungen der verschiedenen Prozesse auszugleichen.