

Beispiellösung - Übungsblatt 6

Erstellt von Daniel Bub

Aufgabe 1.

Unter Linux beträgt die Frequenz für den Scheduler 1.000 Hz. Wie hoch ist somit die minimale Zeitscheibe für die Zuordnung von Rechenzeit (in ms)?

Lösung 1.

$$t_{min} = \frac{1}{1000Hz} = 0.001s = 1ms$$

Aufgabe 2.

Erläutern Sie das Optimierungsproblem eines Schedulers, ohne auf einen spezifischen Algorithmus einzugehen.

Lösung 2.

Die unterschiedlichen Schedulingstrategien haben das Ziel, Prozesse (und auch Threads) nach bestimmten Kriterien (Fairness, Policy Enforcement, Balance, ...) für den Start auszuwählen.

Das Optimierungsproblem eines Schedulers liegt in der Tatsache, dass dieser selbst Teil des Kernels ist. Um den Prozess auszuwählen, der als nächstes gestartet werden soll, muss der Scheduler Berechnungen, basierend auf dessen Strategie, durchführen. Je komplexer der Scheduling-Algorithmus, desto höher ist der Rechenaufwand für den Scheduler. Dies hat zur Folge, dass weniger Rechenzeit für die Prozesse übrig bleibt.

Aufgabe 3.

Zeichnen Sie ein Diagramm für den zeitlichen Ablauf der folgenden Prozesse im Falle des SJF-Algorithmus. Tragen Sie in das Diagramm auf einer Zeitachse Wartezeit und Ausführungszeit jedes Prozesses ein.

Die Wertepaare für die zu verwendenden Prozesse haben die folgende Form:

Prozessnummer (Zeit der Verfügbarkeit, notwendige Rechenzeit)

p1 (0, 1)
p2 (0, 1)
p3 (0, 1)
p4 (3, 1)
p5 (3, 2)
p6 (3, 3)
p7 (7, 3)
p8 (7, 2)
p9 (7, 1)
p10 (13, 1)
p11 (13, 2)

Lösung 3.

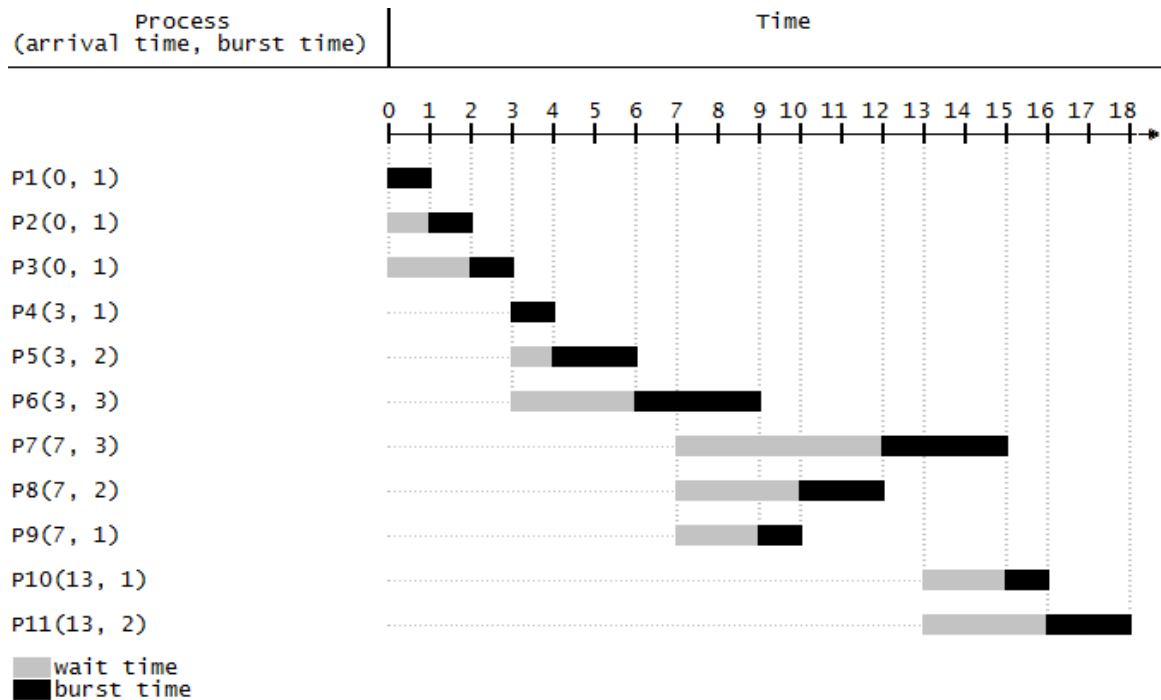


Abbildung 1: Beispiellösung des SJF-Algorithmus, angewandt auf die gegebenen Prozesse

Aufgabe 4.

Wie hoch ist die durchschnittliche Wartezeit der Prozesse aus Aufgabe 3?

Lösung 4.

$$\frac{0+1+2+0+1+3+5+3+2+2+3}{11} = \frac{22}{11} = 2 \text{ Zeiteinheiten}$$

Aufgabe 5.

Wie hoch wäre die durchschnittliche Wartezeit der Prozesse aus Aufgabe 3 im Falle des FCFS-Algorithmus?

Lösung 5.

$$\frac{0+1+2+0+1+3+2+5+7+2+3}{11} = \frac{26}{11} = 2,36 \text{ Zeiteinheiten}$$

Aufgabe 6.

Zeigen Sie, dass ein Echtzeitsystem mit den folgenden Bedingungen als 'scheduable' gilt.

Ereignis	e_1	e_2	e_3
Periode (ms)	100	200	500
notwendige Rechenzeit (in ms)	50	30	100

Lösung 6.

Ein Echtzeitsystem gilt als 'scheduable', wenn die Bedingung $\sum_{i=1}^N \frac{C_i}{C_p} \leq 1$ gilt. Einsetzen der gegebenen Daten ergibt

$$\sum_{i=1}^N \frac{C_i}{C_p} = \frac{50}{100} + \frac{30}{200} + \frac{100}{500} = 0.85 \leq 1$$

\Rightarrow Das Echtzeitsystem ist scheduable.

Aufgabe 7.

Gehen Sie von den Werten aus Ausgabe 6 aus. Ein neues Ereignis mit einer Periode von 1 Sekunde wird in das System einbezogen. Wie hoch darf der Bedarf an Rechenzeit dieses Ereignisses maximal sein, damit das System noch als ‘scheduable’ gilt?

Lösung 7.

Hier gilt es nun einfach die Gleichung $\sum_{i=1}^N \frac{C_i}{C_p} = 1$ zu lösen:

$$\begin{aligned}\frac{50}{100} + \frac{30}{200} + \frac{100}{500} + \frac{x}{1000} &= 1 \\ \frac{17}{20} + \frac{x}{1000} &= 1 \\ \frac{x}{1000} &= \frac{3}{20} \\ x &= 150\end{aligned}$$

Das Ereignis darf somit eine Rechenzeit von höchstens 150ms beanspruchen.