

GSS Abgabe 04

Carolyn Konietzny, Paul Bienkowski, Julian Tobergte, Oliver Sengpiel

27. Mai 2015

1.1 a)

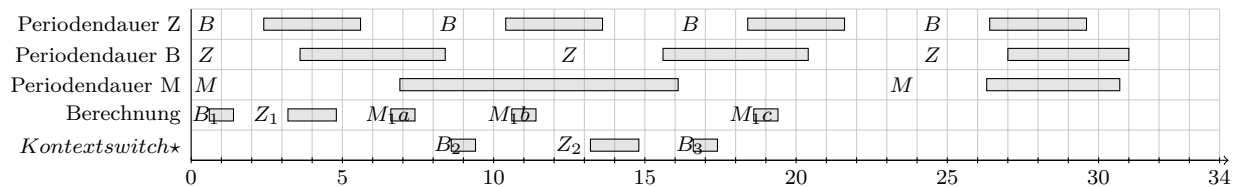
b)

2 a)

b)

c)

3 a) Zeitskala: $t/5$



* : Während der Bearbeitung von M1 werden Prozessen mit höherer Priorität Rechenzeit zugewiesen, welche "Zwischengeschoben" werden, dh. die Mutexlocks von dem vorherig laufenden Prozess bleiben erhalten, in diesem Fall M1. Dies bedeutet aber immernoch eine Sequentielle Ausführung.

Problematisch wird das ganze wenn ein mittelhoch Priorisierter Prozess (Z) Rechenzeit erhält, der dann von einem hochpriorisierten Prozess (B) wiederum abgelöst werden soll. Dabei kann nämlich dann der Mutexlock auf einen Wert, auf den M und B zugreifen nicht von M aufgelöst werden, weil M zu diesem Zeitpunkt ja garnicht aktiv ist. Daher würden erst alle anderen Prozess durchlaufen, die höhere Priorität als M haben, bevor B endlich zur Berechnung kommen darf. Dies hat dann bei zeitkritischen Laufplänen die Folge, dass B zu spät rechnen kann.

Der Graph mit dem Blockgraph Package sieht bei uns unlesbar aus, daher nochmal der Source-code:

```
\begin{blockgraph}{34}{5}{0.4} % 170 / 5 = 34 cols

\bglabelxx{0}
\bglabelxx{5}
\bglabelxx{10}
\bglabelxx{15}
\bglabelxx{20}
\bglabelxx{25}
\bglabelxx{30}
\bglabelxx{34}
```

```

\bglabely{4}{Periodendauer Z}
\bglabely{3}{Periodendauer B}
\bglabely{2}{Periodendauer M}
\bglabely{1}{Berechnung}
\bglabely{0}{\Kontextswitch \star$}

\bgblock[1]{0}{2}{\B_1$}
\bgblock[1]{2}{6}{\Z_1$}
\bgblock[1]{6}{8}{\M_1 a$}
\bgblock[0]{8}{10}{\B_2$}
\bgblock[1]{10}{12}{\M_1 b$}
\bgblock[0]{12}{16}{\Z_2$}
\bgblock[0]{16}{18}{\B_3$}
\bgblock[1]{18}{20}{\M_1 c$}

\bgblock[2]{0}{23}{\M$}
\bgblock[2]{23}{34}{\M$} \% 115 / 5 = 23
\bgblock[3]{0}{12}{\Z$}
\bgblock[3]{12}{24}{\Z$}
\bgblock[3]{24}{34}{\Z$} \% 60 / 5 = 12
\bgblock[4]{0}{8}{\B$}
\bgblock[4]{8}{16}{\B$}
\bgblock[4]{16}{24}{\B$}
\bgblock[4]{24}{32}{\B$} \% 40 / 5 = 8

```

```

\end{blockgraph}

```