

# GSS-Übungsblatt 2

Alexander Timmermann, Jannis Krämer

## 1 Scheduling-Algorithmen

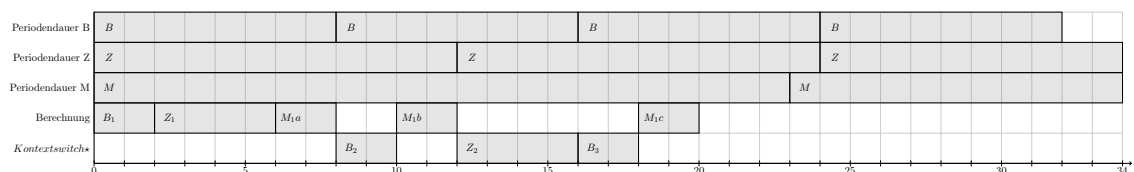
a)

## 2 Echtzeit- und Multiprozessor-Scheduling

a)

## 3 Prioritätsinversion

a) Es ergibt sich folgende Abbildung:



Während der Bearbeitung von M1 wird B aufgrund seiner höheren Priorität eingeschoben. M1 gibt dabei seine Mutexlocks weiter, obwohl M1 noch nicht alle Daten schreiben konnte. Im Zuge des Bus Management, das B ausführt, benötigt B nämlich auch Zugriff auf die Daten von M1. Nachdem B durchgelaufen ist läuft M1 somit weiter. Bevor M1 seinen Task jedoch beenden kann wird er abermals unterbrochen, diesmal jedoch von Z. Z läuft mit mittlerer Priorität, löst M1 somit ab. Z benötigt jedoch keinen Zugriff auf von M1 geschriebene Daten, erhält also auch nicht die Mutexlocks von M1. Nachdem Z nun fertig ist wird er allerdings von B abgelöst, nicht von M, da B die höchste Priorität besitzt. B1 hat nun keinen Zugriff auf Ms Daten, da M nicht aktiv ist und somit B auch keine Mutexlocks übertragen kann. B kann deshalb nicht zuende rechnen und muss warten bis M seine Daten fertig geschrieben hat. Da jedoch zuerst alle Prozesse mit höherer Priorität als M ausgeführt werden, kann die Ausführung B unter Umständen sehr lange

verhindert sein und ein zeitkritisches System somit zum Absturz bringen. Der Computer der Pathfinder-Mission beispielweise führte automatisch einen Neustart, welcher mit Datenverlust verbunden war, aus wenn der Bus Management Task (B) zu lange nicht ausgeführt wurde.