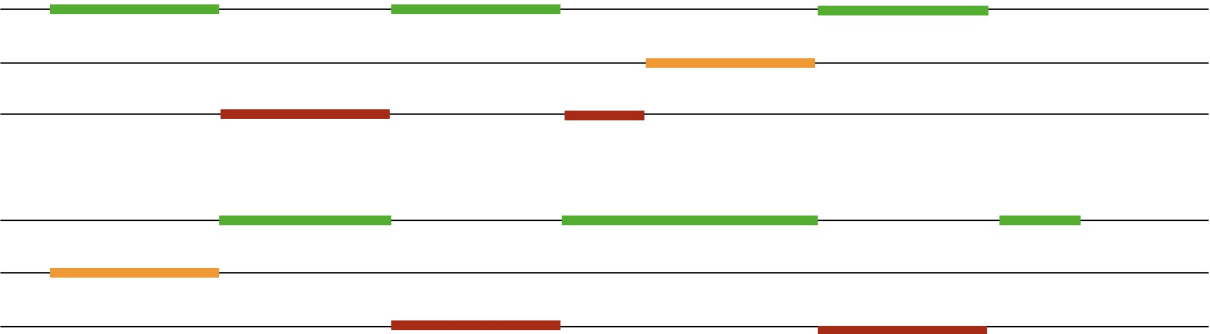




universität
uulm



Inhaltsverzeichnis

1 Motivation

2 First Come, First Served Scheduling

3 Shortest Job First Scheduling

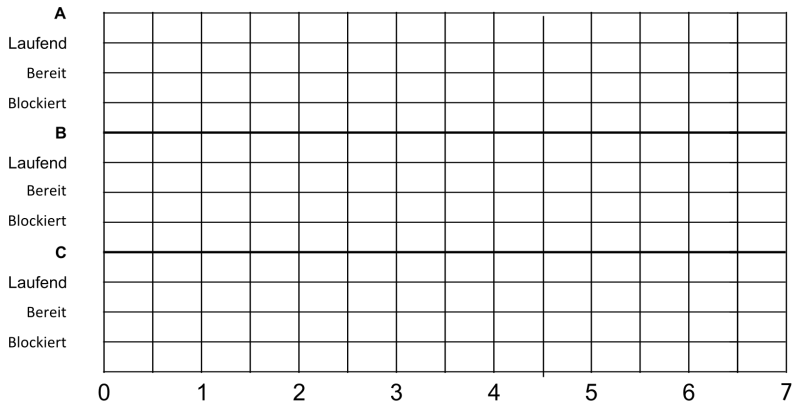
Motivation: Scheduling

- Der Scheduler verwaltet momentan aktive Prozesse
- Eingriff über Unterbrechungen möglich
- Verdrängende (präemptive) sowie nicht-verdrängende (nicht-präemptive) Strategien sind möglich
- Auswahlkriterien für Scheduling-Strategien:
 - CPU Auslastung (möglichst 100%)
 - Durchsatz (Möglichst viele Prozesse pro Zeiteinheit)
 - Verweilzeit (Möglichst kurz)
 - Wartezeit (Möglichst kurz)
 - Antwortzeit (Möglichst kurz)

First Come, First Served Scheduling

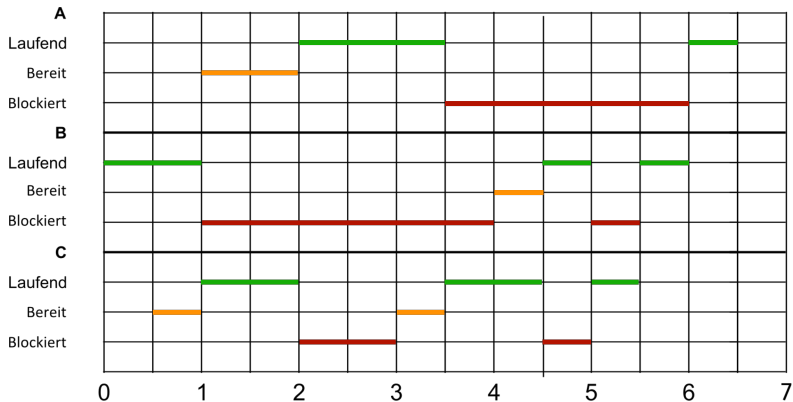
- Prozesse die bereit sind zu laufen werden in eine Warteschlange eingereiht
- Bei Blockierung eines laufenden Prozesses wird der vorderste Prozess in der Warteschlange entnommen
- Diese Strategie ist nicht präemptiv, das heißt der aktuell laufende Prozess wird nicht verdrängt wenn neue Prozesse in die Warteschlange eingereiht werden
- Die Wartezeit ist hierbei nicht optimal, da Prozesse mit langer Laufzeit nachfolgende Prozesse aufhalten

- Prozess A: Start bei $t_A=1s$, läuft 1,5s, blockiert 2,5s, läuft 0,5s
- Prozess B: Start bei $t_B=0s$, läuft 1s, blockiert 3s, läuft 0,5s, blockiert 0,5s, läuft 0,5s
- Prozess C: Start bei $t_C=0,5s$, läuft 1s, blockiert 1s, läuft 1s, blockiert 0,5s, läuft 0,5s



First Come, First Served - Beispiel

- Prozess A: Start bei $t_A=1s$, läuft 1,5s, blockiert 2,5s, läuft 0,5s
- Prozess B: Start bei $t_B=0s$, läuft 1s, blockiert 3s, läuft 0,5s, blockiert 0,5s, läuft 0,5s
- Prozess C: Start bei $t_C=0,5s$, läuft 1s, blockiert 1s, läuft 1s, blockiert 0,5s, läuft 0,5s

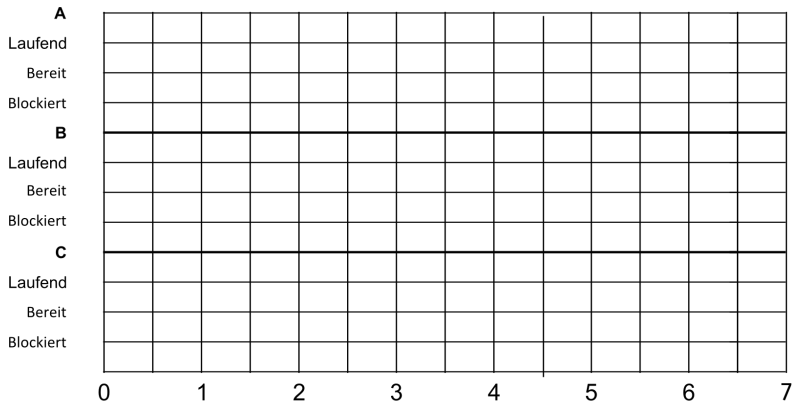


Shortest Job First Scheduling

- Bereite Prozesse werden basierend auf ihrer benötigten Laufzeit bis zur nächsten Blockierung sortiert
- Erfordert Kenntnis über die Laufzeit der Prozesse
- Präemptive und nicht-präemptive Varianten möglich

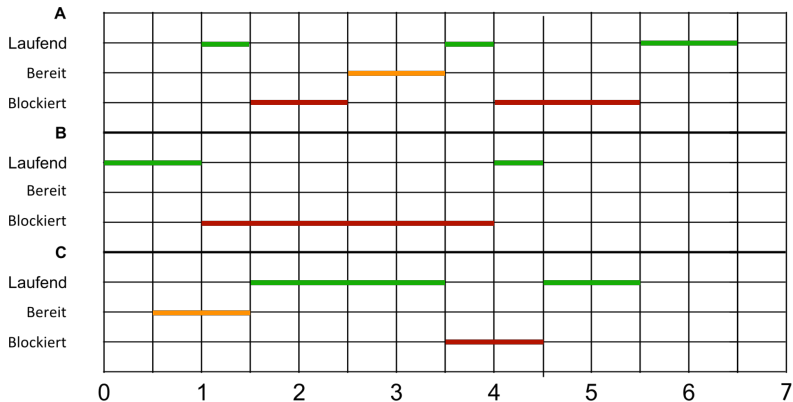
Shortest Job First - Beispiel

- Prozess A: Start bei $t_A=1s$, läuft 0,5s, blockiert 1s, läuft 0,5s, blockiert 1,5s, läuft 1s
- Prozess B: Start bei $t_B=0s$, läuft 1s, blockiert 3s, läuft 0,5s
- Prozess C: Start bei $t_C=0,5s$, läuft 2s, blockiert 1s, läuft 1s



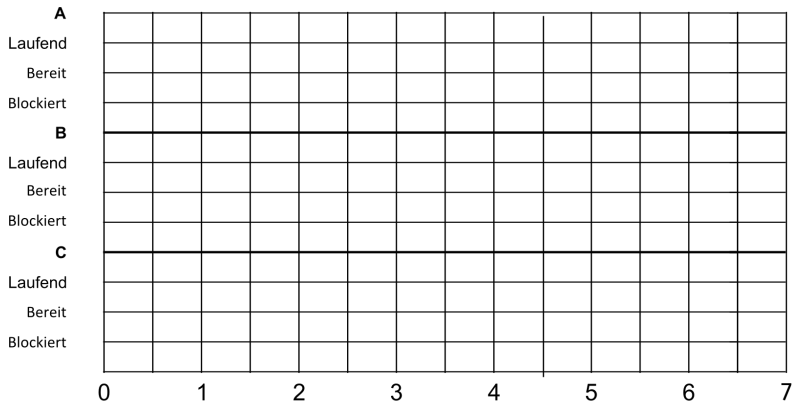
Shortest Job First - Beispiel

- Prozess A: Start bei $t_A=1s$, läuft 0,5s, blockiert 1s, läuft 0,5s, blockiert 1,5s, läuft 1s
- Prozess B: Start bei $t_B=0s$, läuft 1s, blockiert 3s, läuft 0,5s
- Prozess C: Start bei $t_C=0,5s$, läuft 2s, blockiert 1s, läuft 1s



Präemptives Shortest Job First - Beispiel

- Prozess A: Start bei $t_A=1s$, läuft 0,5s, blockiert 1s, läuft 0,5s, blockiert 1,5s, läuft 1s
- Prozess B: Start bei $t_B=0s$, läuft 1s, blockiert 3s, läuft 0,5s
- Prozess C: Start bei $t_C=0,5s$, läuft 2s, blockiert 1s, läuft 1s



Präemptives Shortest Job First - Beispiel

- Prozess A: Start bei $t_A=1s$, läuft 0,5s, blockiert 1s, läuft 0,5s, blockiert 1,5s, läuft 1s
- Prozess B: Start bei $t_B=0s$, läuft 1s, blockiert 3s, läuft 0,5s
- Prozess C: Start bei $t_C=0,5s$, läuft 2s, blockiert 1s, läuft 1s

