

# Scheduling 2 (Round Robin) - Handout

\*\*\*\*\* Nguyen

9. November 2020

## 1 Konzept

Es wird ein Quantum (eine Art Zeitslot) festgelegt, welcher üblicherweise zwischen 10ms und 100ms liegt. Wenn ein Prozess länger braucht als ein Quantum, wird es wieder hinten an die Warteschlange geschickt. Wenn ein Prozess während eines Quantums fertig wird, bekommt der nächste ein vollständiges Quantum.

## 2 Beispiel

Es sei ein Quantum von 100ms festgelegt.

Prozess	Ankunftszeit	Prozessdauer
P0	0ms	150ms
P1	30ms	250ms
P2	120ms	50ms
P3	130ms	170ms

Tabelle 1: Beispiel Prozesse

Zeitstrahl					Verbleibende Zeit				
Zeit	Slot 1	Slot 2	Slot 3	Slot 4	Event	P0	P1	P2	P3
0ms	P0				P0 join	150			
30ms	P0	P1			P1 join	120	250		
100ms	P1	P0			Rotate	50	250		
120ms	P1	P0	P2		P2 join		230	50	
130ms	P1	P0	P2	P3	P3 join		220		170
200ms	P0	P2	P3	P1	Rotate	50	150		
250ms	P2	P3	P1		P0 finish	0		50	
300ms	P3	P1			P2 finish			0	170
400ms	P1	P3			Rotate		150		70
500ms	P3	P1			Rotate		50		70
570ms	P1				P3 finish		50		0
620ms					P1 finish		0		

Tabelle 2: Ablauf im Round Robin Scheduling

### 3 Aufgabe

#### 3.1 Aufgabenstellung

Prozess	Ankunftszeit	Prozessdauer
P0	4ms	317ms
P1	105ms	16ms
P2	220ms	186ms

Tabelle 3: Gegebene Prozesse

1. Bestimme die Wartezeit (Wie viel länger der Prozess als die Prozessdauer braucht) für jeden Prozess mit einem Quantum von 100ms
2. Vergleiche die Wartezeiten mit der eines FCFS Algorithmus

#### 3.2 Lösung

Zeit	Zeitstrahl		Event	Verbleibende Zeit		
	Slot 1	Slot 2		P0	P1	P2
4ms	P0		P0 join	317		
104ms	P0		Rotate	217		
105ms	P0	P1	P1 join	216	16	
204ms	P1	P0	Rotate	117		
220ms	P0	P2	P1 finish & P2 join		0	186
330ms	P2	P0	Rotate	17		
430ms	P0	P2	Rotate			86
447ms	P2		P0 finish	0		
533ms			P2 finish			0

Tabelle 4: Ablauf

Prozess	Start	Ende	Wartezeit (RR)	Wartezeit (FCFS)
P0	4ms	447ms	126ms	0ms
P1	105ms	220ms	99ms	342ms
P2	220ms	533ms	127ms	0ms

Tabelle 5: Wartezeit und Vergleich mit FCFS

## 4 Fazit

- Alle Prozesse werden hier mit der gleichen Dringlichkeit bearbeitet (kann aber auch keine Prioritäten setzen)
- Es wird sicher gegangen, dass kein Prozess verhungert
- Niedrige durchschnittliche Verweilzeit für alle Prozesse
- Algorithmus ist leicht zu implementieren
- Wichtig: Das Quantum muss gut gewählt sein
  - Wenn das Quantum zu groß ist, ähnelt es einem First Come First Serve (FCFS/FIFO)
  - Wenn das Quantum zu klein ist, ähnelt es einem Shortest Job First (SJF) und der Kontextwechsel Aufwand wird größer

## Literatur

- [1] Wolfram Burgard. *Systeme I: Betriebssysteme, Kapitel 7 Scheduling*. (letzter Zugriff: 2020-11-01). 2016. URL: <http://ais.informatik.uni-freiburg.de/teaching/ws16/systems1/slides/kap07-scheduling.pdf>.
- [2] Prof. Dr. Margarita Esponda. *Scheduling*. (letzter Zugriff: 2020-11-01). 2012. URL: [http://www.inf.fu-berlin.de/lehre/WS11/OS/slides/OS\\_V7\\_Scheduling\\_Teil\\_1.pdf](http://www.inf.fu-berlin.de/lehre/WS11/OS/slides/OS_V7_Scheduling_Teil_1.pdf).
- [3] R. Bär und G. Bischofberger und E. Dehler und N. Hammer und B. Schiemann und T. Wolf. *Informatik und Informationstechnik*. EUROPA-LEHRMITTEL, 2017. URL: [http://www.inf.fu-berlin.de/lehre/WS11/OS/slides/OS\\_V7\\_Scheduling\\_Teil\\_1.pdf](http://www.inf.fu-berlin.de/lehre/WS11/OS/slides/OS_V7_Scheduling_Teil_1.pdf).