Prof. Dr. Carsten Vogt

Exercises „Operating Systems and Distributed Systems 2“

2019/20

Technology
Arts Sciences
TH Köln

Process Management and CPU Scheduling – Solutions

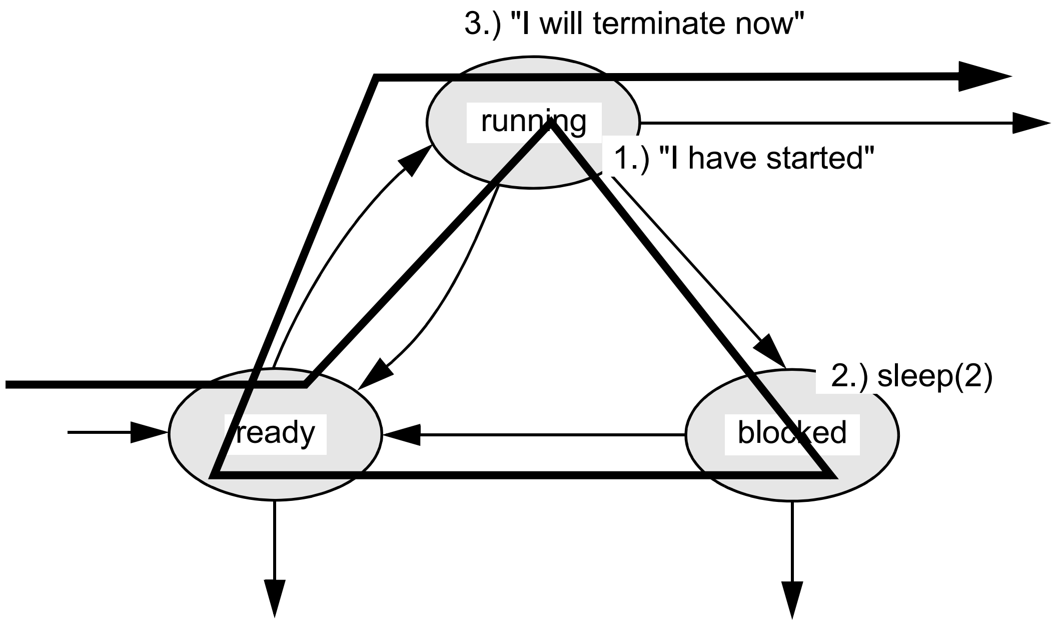
Exercise 1: Process States

- Consider the following program:

```
main() {  printf("I have started.\n");
          sleep(2);
          printf("I will terminate now.\n"); }
```

- A process executing this programm will run through a sequence of states. Illustrate, with the state transition diagram, this state sequence.

Solution:



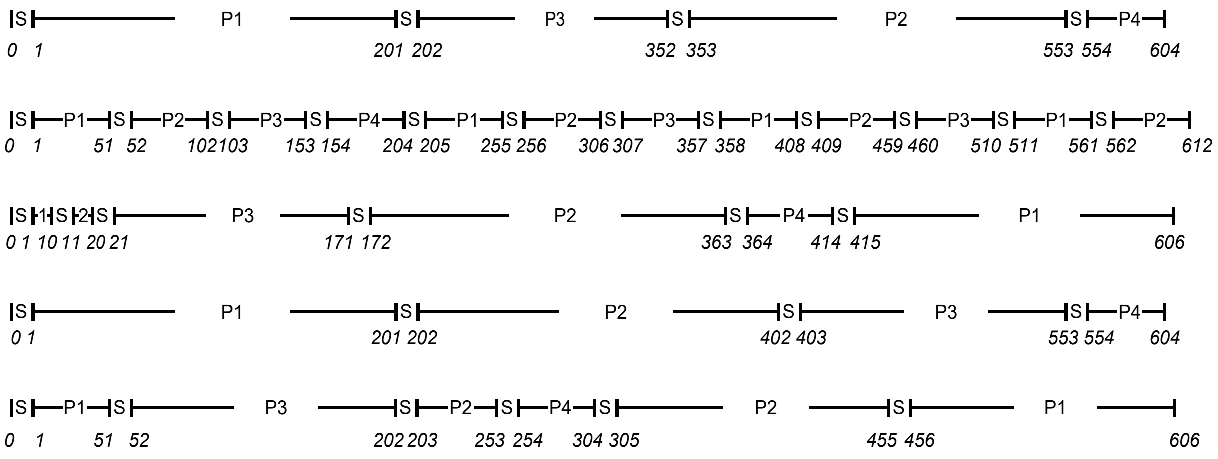
- Please find a video explanation at http://www.nt.th-koeln.de/vogt/bs/blaetter.html.

Exercise 2: Comparison of Scheduling Strategies

- Consider these scheduling strategies: FCFS (First Come First Served), FP_np (Fixed priorities, non-preemptive), FP_p (Fixed priorities, preemptive), RR_50 (Round Robin, time slice = 50 ms), MQ_50 (Multilevel Queueing, round robin time slice = 50 ms)
- Given are four processes with arrival times (= instant of the arrival of the process, in ms after time 0), execution durations (= CPU time required, in ms) and priorities:

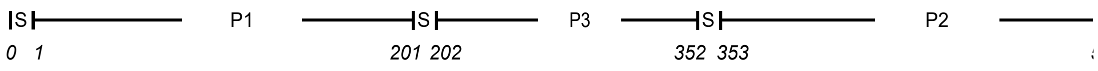
process no.	arrival time	execution duration	priority
P1	0	200	3 (low)
P2	10	200	2 (medium)
P3	20	150	1 (high)
P4	30	50	2 (medium)

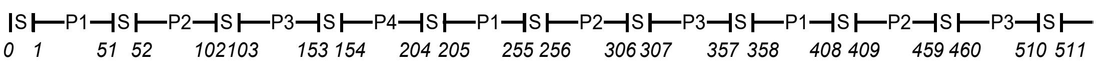
- Scheduling the processes on the CPU leads to the following timelines. These timelines show when the CPU serves which process ('S' = CPU switch between processes):

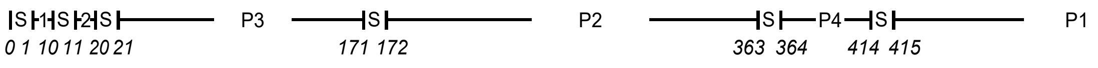


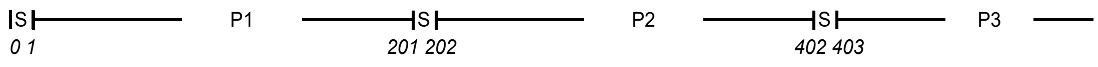
- Which timeline belongs to which strategy?

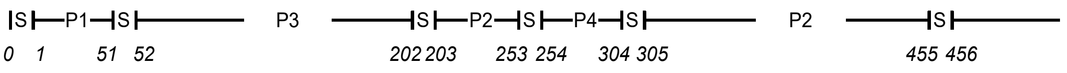
Solution:

FP_np:


RR 50:


FP_p:


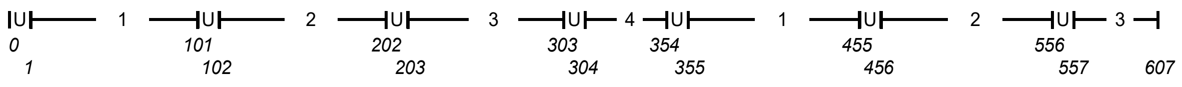
FCFS:


MQ 50:


- Please find a video explanation at http://www.nt.th-koeln.de/vogt/bs/blaetter.html.

- Draw a similar timeline for RR_100 (i.e. Round Robin with a timeslice of 100 ms). What improves (= becomes better) with a longer timeslice? What is the drawback of the longer timeslice?

Solution:



  - Longer timeslice better: Less management overhead = fewer switch operations
  - Longer timeslice worse: Processes with short execution times terminate later

Exercise 3: Traditional UNIX CPU Scheduling

- Given are the following constants:
  - $B_u = 60$, norm = 0.1, decay = 2, nice = 0, priority at process start time = 60
  - CPU time used by the process:
    480 ms during the 1st second, 600 ms during the 2nd second, 0 ms afterwards.
- Calculate the values of "cpu_usage" and "prio" after each second:

second ...	0 (= start)	1	2	3	4	5
cpu_usage						
prio						

Solution:

second ...	0 (= start)	1	2	3	4	5
cpu_usage	0	480	840	420	210	105
prio	60	108	144	102	81	70

calculation:
variable *cpu_usage* at time 1
= variable *cpu_usage* at time 0 / 2 + CPU time used during the 1st second
= 0 / 2 + 480 = 480
variable *cpu_usage* at time 2
= variable *cpu_usage* at time 1 / 2 + CPU time used during the 2nd second
= 480 / 2 + 600 = 840
etc.

- Please find a video explanation at http://www.nt.th-koeln.de/vogt/bs/blaetter.html.

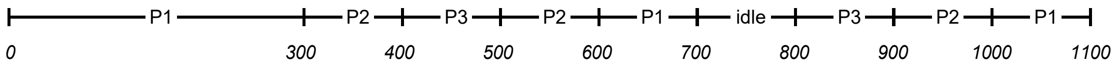
Exercise 4: CFS Scheduling

- Consider a system that activates its scheduler every 100 ms. At these points in time, the scheduler assigns the CPU to one of the active processes (i.e. the processes that are ready to run) according to the CFS (= Completely Fair Scheduling) strategy.
- Given are three processes:
  - P1: Starts at instant 0 and then requests 400 ms CPU time.
  - P2: Starts at instant 300 ms (= 300 ms after P1) and then requests 200 ms CPU time.
  - P3: Starts at instant 400 ms and then requests 100 ms CPU time.
  - At instant 800 ms, all processes request another 100 ms CPU time each.
- Draw a timeline that shows when the CPU will execute which process.
  - Assume that a context switch requires no time.
- While drawing the timeline, fill in the following table. This table shall show the "runtimes" of the processes, i.e. the total CPU times they have received up to the given instants:

ms	100	200	300	400	500	600	700	800	900	1000	1100
P1											
P2											
P3											

Solution:

- timeline:



- table:

ms	100	200	300	400	500	600	700	800	900	1000	1100
P1	100	200	300	300	300	300	400	400	400	400	500
P2	0	0	0	100	100	200	200	200	200	300	300
P3	0	0	0	0	100	100	100	100	200	200	200

- Please find a video explanation at http://www.nt.th-koeln.de/vogt/bs/blaetter.html.