NTNU
Norwegian University of
Science and Technology

Department of Computer Science – IDI TDT4186 Operating Systems

https://folk.ntnu.no/michaeng/tdt4186_22/
michael.engel@ntnu.no

Theoretical exercises
Spring 2022

Theoretical Exercises 5

Scheduling

Please submit solutions on Blackboard by Monday, 7.3.2022 12:00h

5.1 Scheduling

Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

Process	Burst time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

- Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling.
- What is the turnaround time of each process for each of the scheduling algorithms in part a?
- What is the waiting time of each process for each of the scheduling algorithms in part a?
- Which of the schedules in part a results in the minimal average waiting time (over all processes)?

5.2 Starvation

Which of the following scheduling algorithms could result in starvation? Explain why the starvation can occur.

- First-come, first-served
- Shortest job first
- Round robin
- Priority

5.3 Round-Robin

Consider a variant of the RR scheduling algorithm where the entries in the ready queue are pointers to the PCBs.

- What would be the effect of putting two pointers to the same process in the ready queue?
- What would be the major advantages and disadvantages of this scheme?
- How would you modify the basic RR algorithm to achieve the same effect without the duplicate pointers?

5.4 Scheduling behavior

Explain the differences in the degree to which the following scheduling algorithms discriminate in favor of short processes:

- FCFS
- RR
- Multilevel feedback queues

5.5 Corner cases

Consider a preemptive priority scheduling algorithm based on dynamically changing priorities. Larger priority numbers imply higher priority. When a process is waiting for the CPU (in the ready queue, but not running), its priority changes at a rate α when it is running, its priority changes at a rate β. All processes are given a priority of 0 when they enter the ready queue. The parameters α and β can be set to give many different scheduling algorithms.

- What is the algorithm that results from $\beta > \alpha > 0$?
- What is the algorithm that results from $\alpha < \beta < 0$?