# Operating systems

# Midterm examination

# 2009/2010

Student Name:

NIA:

**1.** (3pt) Given the following set of processes:

| Process | Burst Time (ms) | Priority |
|---|---|---|
| *P*1 | 10 | 3 |
| *P*2 | 1 | 1 |
| *P*3 | 2 | 3 |
| *P*4 | 1 | 4 |
| *P*5 | 5 | 2 |

The processes are assumed to have arrived in the order *P*1, *P*2, *P*3, *P*4, *P*5, all at time 0.

a. Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, a non-preemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FCFS | | | | | | | | | | | | | | | | | | | |
| RR | | | | | | | | | | | | | | | | | | | |
| SJF | | | | | | | | | | | | | | | | | | | |
| Priority | | | | | | | | | | | | | | | | | | | |

b. What is the turnaround time of each process for each of the scheduling algorithms in part a?

| | FCFS | RR | SJF | Priority |
|---|---|---|---|---|
| *P*1 | | | | |
| *P*2 | | | | |
| *P*3 | | | | |
| *P*4 | | | | |
| *P*5 | | | | |

c. What is the waiting time of each process for each of the scheduling algorithms in part a?

| | FCFS | RR | SJF | Priority |
|---|---|---|---|---|
| *P*1 | | | | |
| *P*2 | | | | |
| *P*3 | | | | |
| *P*4 | | | | |
| *P*5 | | | | |

d. Which of the schedules in part a results in the minimal average waiting time (over all processes)?

2. (1pt) Which of the following scheduling algorithms could result in starvation? Why?

a. First-come, first-served
b. Shortest job first
c. Round robin
d. Priority

3. (1pt) Draw a state transition diagram of process execution states. Can a process transition from waiting for an I/O operation to the terminated state? Why or why not?

4. (0.5pt) What is the essential cause of the difference in cost between a context switch for

kernel-level threads and a switch that occurs between user-level threads?

5. (0.5 pt) Which of the following components of program state are shared across threads in a multithreaded process?

a. Register values
b. Heap memory
c. Global variables
d. Stack memory

6. (0.5pt) The threading model supported by the typical Linux kernel is:

(a) one-to-one
(b) one-to-many
(c) many-to-one
(d) many-to-many
(e) two-level
(f) all of the above

7. (0.5pt) Explain the difference between “user mode” and “kernel mode” execution.

8. (3pt) Given the following code:

```
main() {
        printf(“Hello ”);
        fork();
        printf(“world”);
        fork();
        printf(“!”);
}
```

(a) Draw the process tree.
(b) What will the program print? Why?
(c) Is the order of printing the same for all executions? Justify your answer.
(d) Modify the program in such a way that it **creates exactly 3 processes** (including father) the father prints “Hello”, a child prints “world” and a child of a child prints “!” **exactly in this order**.