Quiz 02 - Scheduling

Total points 9/11

The respondent's email address (**rpn241@nyu.edu**) was recorded on submission of this form.

Section score 9/11

✓ In a non-preemptive strategy, when is an election triggered by the scheduler? (Choose all of the correct answers) 1/1

- [x] When a task switches from the running state to the waiting state ✓
- [x] When a task switches from the waiting state to the ready state and the CPU is idle ✓
- [ ] When a task switches from the waiting state to the ready state and the CPU is busy
- [x] When a task switches from the running state to the ready state ✓
- [x] When a task switches from the running state to the terminated state ✓
- [ ] Never: elections are triggered by the dispatcher

✓ Preemption means that an election can be triggered: 1/1

- ○ When a task switches from the waiting state to the ready state and the CPU is idle
- ○ When a task switches from the waiting state to the ready state and the CPU is busy
- ○ When a new task gets inserted in the system
- ◉ All of the above ✓
- ○ None of the above

✓ When designing and choosing scheduling policies, which of the following metrics should be maximized? (Choose all of the correct answers) 1/1

- ☐ average waiting time
- ☐ average turnaround time
- ☑ CPU utilization ✓
- ☐ average response time
- ☑ throughput ✓

✓ The time interval between the submission date of a process and its completion date is called: 1/1

- ◉ turnaround time ✓
- ○ response time
- ○ waiting time
- ○ throughput

✓ Which evaluation metric uses the difference between turnaround time and computation time? 1/1

- ○ average response time
- ◉ average waiting time ✓
- ○ throughput

✓ Which of the following process scheduling algorithms may lead to starvation? 1/1

- ☐ First Come, First Served
- ☐ Round Robin
- ☑ Shortest Job First ✓
- ☑ Multilevel Feedback Queue ✓
- ☑ Priority Scheduling ✓
- ☐ None of the above

✓ In order to ensure fairness when scheduling tasks in a time-sharing strategy, why do we not simply minimize the size of the quantum? 1/1

- ○ It penalizes larger tasks because it can induce famine
- ◉ It penalizes every task because it increases the scheduling overhead ✓
- ○ It penalizes smaller tasks that could have terminated faster with a larger quantum size

✓ What is the effect of a very large quantum size when scheduling tasks in a time-sharing strategy? 1/1

- ○ It increases the throughput by decreasing the overhead
- ○ It penalizes larger tasks because it mimics the Shortest Job First strategy
- ◉ It penalizes smaller tasks because it mimics the First Come First Served strategy ✓

✗ When does it make sense to choose Round Robin instead of Shortest Job First? (Choose all of the correct answers) 0/1

- ☑ To prevent famine ✓
- ☑ To decrease the average response time ✗
- ☐ To decrease the average waiting time
- ☑ To introduce fairness in the presence of interactive tasks ✓

Correct answer

- ☑ To prevent famine
- ☑ To introduce fairness in the presence of interactive tasks

✗ Which statement about the Preemptive SJF scheduling policy is false? 0/1

- ○ May cause famines
- ○ Upon insertion of a new process, the running process gets evicted if its expected CPU time is longer than that of the new process
- ○ Requires the scheduler to predict the CPU time of each task
- ◉ Guarantees that the average waiting time is minimal ✗
- ○ All of the above
- ○ None of the above

Correct answer

- ◉ None of the above

✓ What is graceful ageing? 1/1

- ○ The priority of processes in the ready queue increases periodically.
- ○ Processes move to a lower priority queue after finishing their quantum. Upon reaching the lowest priority queue, they are flushed out using the FCFS scheduling policy.
- ◉ Processes move to a lower priority queue after finishing their quantum. After reaching the lowest priority queue, they may move back to the highest priority queue. ✓
- ○ Processes get the CPU for a limited duration (quantum), then go back to the end of the ready queue.

Google Forms