User Id: mame23

# CSE 303: Quiz #4

Due November 28th, 2022, at 11:59 PM

The quiz has THREE questions. Please submit your answer on Course Site as a pdf whose name is exactly your user Id and the "pdf" extension (e.g., abc123.pdf) before the deadline.

**Question 1:** We studied four scheduling algorithms in lectures: FIFO, SJF, RR, MFQ.

(a) Which one of them is closer to the way you schedule your homework assignments? One valid answer is "neither of them" -- in this case explain what the scheduling algorithm is you follow.
(b) Use the terms we discussed in Chapter 7 (response time, fairness, throughput, starvation, …) to argue why or why not you should stick with the way you schedule your homework assignments.

Give a detailed answer but keep your entire response to a single side of an 8.5x11 page.

(a) The scheduling algorithm that most closely resembles the *overall* process of how I complete my homework is the SJF, where I prioritize or complete the assignments that will take the least amount of time or effort to complete first, then I prioritize/complete the assignments that will take the most amount of time or effort. However, if the homework is a 2-week project or even a semester project, it would look more like a RR, where I complete some parts of the homework, then move to other homework.
(b) In general, response time is highly valued as homework assignments have required deadlines, so it is an advantage to complete the processes in a short amount of time. This process favors SJF. However, to prevent starvation and implement an algorithm that is fair, completing pieces of different homework assignments is an advantage versus completing the shortest one first; The most common case for me is to do SJF only on the first assignment, then RR the other, more difficult assignments.

User Id: mame23

**Question 2:** Given the following mix of tasks, task lengths, and arrival times, compute the completion and response time for each task, along with the average response time for the FIFO, RR, and SJF algorithms. Assume a time slice of 20 milliseconds and that all times are in milliseconds.

| Task | Length (Burst Time) | Arrival Time |
|---|---|---|
| 0 | 90 | 0 |
| 1 | 30 | 15 |
| 2 | 40 | 15 |
| 3 | 20 | 80 |
| 4 | 50 | 85 |

| Process (SJF) | Arrival Time | Length (Burst Time) | Completion Time | Turn Around Time | Waiting Time |
|---|---|---|---|---|---|
| p3 | 80 | 20 | 100 | 20 | 0 |
| p1 | 15 | 30 | 130 | 115 | 85 |
| p2 | 15 | 40 | 170 | 155 | 115 |
| p4 | 85 | 50 | 220 | 135 | 85 |
| p0 | 0 | 90 | 310 | 310 | 220 |
| | | | | avg=135 | avg=85 |
| Gantt Chart | | | | | |
| idle time | p3 | p1 | p2 | p4 | p0 |
| 0 | 100 | 130 | 170 | 220 | 310 |

| Process (RR) | Arrival Time | Length (Burst Time) | Completion Time | Turn Around Time | Waiting Time |
|---|---|---|---|---|---|
| p0 | 0 | 90 | 260 | 260 | 170 |
| p1 | 15 | 30 | 140 | 125 | 95 |
| p2 | 15 | 40 | 160 | 145 | 105 |
| p3 | 80 | 20 | 80 | 0 | -20 |
| p4 | 85 | 50 | 220 | 135 | 85 |
| | | | | avg=135 | avg=95 |
| Gantt Chart | | | | | |
| idle time | p0 | p1 | p2 | p3 | p4 |
| 0 | 20 | 40 | 60 | 80 | 100 |
| p0 | p1 | p2 | p4 | p0 | p4 |
| 120 | 140 | 160 | 180 | 200 | 220 |
| p0 | p0 | | | | |
| 240 | 260 | | | | |

User Id: mame23

| Process (FIFO) | Arrival Time | Length (Burst Time) | Completion Time | Turn Around Time | Waiting Time |
|---|---|---|---|---|---|
| p0 | 0 | 90 | 90 | 90 | 0 |
| p1 | 15 | 30 | 120 | 105 | 75 |
| p2 | 15 | 40 | 160 | 145 | 105 |
| p3 | 80 | 20 | 180 | 100 | 80 |
| p4 | 85 | 50 | 230 | 145 | 95 |
| | | | | avg=105 | avg=80 |
| Gantt Chart | | | | | |
| idle time | p0 | p1 | p2 | p3 | p4 |
| 0 | 90 | 120 | 160 | 180 | 230 |

User Id: mame23

**Question 3:** Assume we have a 32-bit virtual address that is split into 20 bits for virtual page number and 12 bits for page offset.

A. What is the maximum number of virtual pages in this system? What is the size of each page?
B. Is it possible to have a different number of physical frames than the number of virtual pages? Why or why not?
C. Is it possible to have a different physical frame size than the virtual page size? Why or why not?
D. Considering a single-level page table is used to translate those virtual pages into physical page frames, and assuming that each page table entry is 4 bytes, how many pages are needed to load this entire page table into memory?

(a) Maximum number of virtual pages = 2^20
Page size = (2^32) / (2^20) = 4096
(b) The virtual address appears linear, but the physical pages in memory can be split any which way. This creates only internal fragmentation within the page size. This is evident in the fact that a virtual page can have multiple mappings to different frames
(c) No. The page size must be equal to the frame size to maximize the internal fragmentation of the main memory. A page is a region of the virtual address space, and the page frame is a region of physical memory; A page must have the same size as that piece of physical memory.
(d) 4096/4 = 1024 pages