# Q2_CS341-OperatingSystemQuiz-2

Total points 87/100

Email address *

maheeth2000@gmail.com

Roll Number *

1801CS31

Name *

Maheeth Reddy

✓ A thread is usually defined as a 'light weight process' because an operating system (OS) maintains smaller data structures for a thread than for a process. In relation to this, which of the followings is correct

- (●) On per-thread basis, the OS does not maintain virtual memory state ✓
- ( ) On per-thread basis, the OS maintains only CPU register state
- ( ) The OS does not maintain a separate stack for each thread
- ( ) On per thread basis, the OS maintains only scheduling and accounting information

✗ Kernel functions, pick the odd one out 0/1

- ○ Management of processes
- ○ First level interrupt handling
- ○ DMA
- ○ Provide a mechanism of interaction between processes
- ◉ Timer functions ✗

✓ A process is represented by the PCB (Process Control Block) or PD (Process Descriptor), which does not contain 1/1

- ○ Program Counter
- ○ Processor Status Word
- ○ Stack Pointer
- ○ Registers
- ○ Memory addressing registers
- ◉ Exception address ✓
- ○ Other: .....

✓ A trap is a software-generated interrupt 1/1

- (●) YES ✓
- ( ) NO

✓ An user is able to develop a new command interpreter using the system-call interface provided by the operating system. 1/1

- (●) True ✓
- ( ) False

✓ A Microkernal is more secure as more operations are done in user mode than in kernel mode 1/1

- (●) True ✓
- ( ) False

✓ In Microkernal, user programs and system services interact in a microkernel architecture by using interprocess communication mechanisms such as messaging. 1/1

- (●) True ✓
- ( ) False

✓ The modular kernel approach requires subsystems to interact with each other through carefully constructed interfaces that are typically narrow 1/1

- (●) True ✓
- ( ) False

✓ The threads of a multithreaded process share heap memory and global variables 2/2

- (●) True ✓
- ( ) False

✗ A multithreaded system comprising of multiple user-levelthreads cannot make use of the different processors in a multiprocessorsystem simultaneously 0/2

- ( ) True
- (●) False ✗

✓ Linux does not distinguish between processes and threads. 2/2

- (●) True ✓
- ( ) False

✓ I/O-bound programs have the property of performing only a small amount of computation before performing IO. 2/2

- (●) True ✓
- ( ) False

✓ The layered kernel approach is similar modular. However,the layered kernel imposes a strict ordering of subsystems such that subsystems at the lower layers are not allowed to invoke operations corresponding to the upper-layer subsystems 1/1

- (●) True ✓
- ( ) False

✓ FCFS—discriminates against short jobs since any short jobs arriving after long jobs will have a longer waiting time. 1/1

- (●) True ✓
- ( ) False

✓ RR—treats all jobs equally (giving them equal bursts of CPU time)so short jobs will be able to leave the system faster since they will finish first. 1/1

- (●) True ✓
- ( ) False

✓ Multilevel feedback queues—work similar to the RR algorithm—they discriminate favorably toward short jobs. 1/1

- (●) True ✓
- ( ) False

✓ The traditional UNIX scheduler enforces an inverse relationship between priority numbers and priorities: The higher the number, the lower the priority. 1/1

- (●) True ✓
- ( ) False

✓ How does the Hardware trigger an interrupt? 2/2

- ( ) Executing a special program called interrupt program
- (●) Sending signals to CPU through a system bus ✓
- ( ) Executing a special program called system program
- ( ) Executing a special operation called system call

✓ Consider a system running ten I/O-bound tasks and one CPU-bound task. Assume that the I/O-bound tasks issue an I/O operation once for every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context switching overhead is 0.1millisecond and that all processes are long-running tasks. What is the CPU utilization for a round-robin scheduler when the time quantum is 1 millisecond is 2/2

- ○ 90%
- ◉ 91% ✓
- ○ 75%
- ○ 100%

✓ How does the software trigger an interrupt? 2/2

- ○ Sending signals to CPU through bus
- ○ Executing a special program called system program
- ◉ Executing a special operation called system call ✓
- ○ Executing a special program called interrupt trigger program

✓ Which of the following scheduling algorithms could result in starvation? 2/2

- ○ First-come, first-served
- ○ Round robin
- ◉ Priority ✓
- ○ All of the above

✓ A process is selected from the _____ queue by the _____ scheduler, to be executed. 2/2

- ○ blocked, short term
- ◉ ready, short term ✓
- ○ wait, long term
- ○ ready, long term

✓ The switching of the CPU from one process or thread to another is called _____ 2/2

- ○ process switch
- ○ task switch
- ○ context switch
- ◉ all of the mentioned ✓

✓ Process are classified into different groups in _____ 3/3

- shortest job scheduling algorithm
- round robin scheduling algorithm
- (●) multilevel queue scheduling algorithm ✓
- priority scheduling algorithm

✓ What is 'Aging'? 2/2

- keeping track of cache contents
- (●) increasing the priority of jobs to ensure termination in a finite time ✓
- keeping track of what pages are currently residing in memory
- keeping track of how many times a given page is referenced

✓ Consider the following set of processes, the length of the CPU burst time given in milliseconds. The burst time for processes P1 , P2, P3, and P4 are 6, 8, 7, and 3. Assuming the above process being scheduled with the SJF scheduling algorithm. 3/3

- (●) The waiting time for process P1 is 3ms ✓
- The waiting time for process P1 is 0ms
- The waiting time for process P1 is 16ms
- The waiting time for process P1 is 9ms

✓ Which of the following statements are true? i) Shortest remaining time first scheduling may cause starvation ii) Preemptive scheduling may cause starvation iii) Round robin is better than FCFS in terms of response time 2/2

- ○ ii and iii only
- ◉ i, ii and iii ✓
- ○ i only
- ○ I and ii only

✓ Which operation is performed by an interrupt handler? 2/2

- ○ Saving the current state of the system
- ○ Loading the interrupt handling code and executing it
- ○ Once done handling, bringing back the system to the original state it was before the interrupt occurred
- ◉ All of the mentioned ✓

✓ 3/3

```
Q. A process executes the code
fork();
fork();
fork();

The total number of child processes created is
A: 3
B: 4
C: 7
D: 8
```

- ○ 3
- ○ 4
- ◉ 7 ✓
- ○ 8

✓ Consider the following statements about user level threads and kernel level threads. Which one of the following statement is FALSE? 1/1

- ○ Context switch time is longer for kernel level threads than for user level threads.
- ○ User level threads do not need any hardware support.
- ○ Related kernel level threads can be scheduled on different processors in a multiprocessor system.
- ◉ Blocking one kernel level thread blocks all related threads ✓

✓ An operating system implements a policy that requires a process to release all resources before making a request for another resource. Select the TRUE statement from the following:

- ○ Both starvation and deadlock can occur
- ◉ Starvation can occur but deadlock cannot occur * ✓
- ○ Starvation cannot occur but deadlock can occur
- ○ Neither starvation nor deadlock can occur

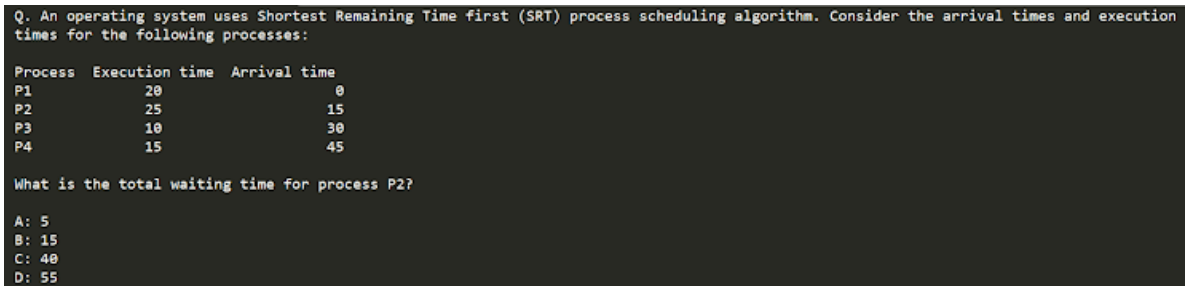
✓ Which of the following process scheduling algorithm may lead to starvation 2/2

- ○ FIFO
- ○ Round Robin
- ◉ Shortest Job Next ✓
- ○ None of the above

✓ If the quantum time of round robin algorithm is very large, then it is equivalent to: 2/2

- ◉ First in first out ✓
- ○ Shortest Job Next
- ○ Lottery scheduling
- ○ None of the above

✗ 0/3




- ○ 5
- ○ 15
- ◉ 40 ✗
- ○ 55

✓ The maximum number of processes that can be in Ready state for a computer system with n CPUs is 2/2

- ○ n
- ○ n^2
- ○ 2n
- ◉ Independent of n ✓

✓ Assume every process requires 3 seconds of service time in a system with single processor. If new processes are arriving at the rate of 10 processes per minute, then estimate the fraction of time CPU is busy in system? 2/2

- ○ 20%
- ○ 30%
- ◉ 50% ✓
- ○ 60%

✓ Which of the following is not an optimization criterion in the design of a CPU scheduling algorithm? 2/2

- ◉ Minimum CPU utilization ✓
- ○ Maximum throughput
- ○ Minimum turnaround time
- ○ Minimum waiting time

✗ Given the following pieces of code. 1 will be printed x times and hello y times. The values of x and y are 0/3

```
Prog.(a)

main() {
int i=0; printf(" %d\n", i+1);
printf(" Hello\n");
fork();
fork();
fork(); }
```

- ○ 1,1
- ◉ 2,1 ✗
- ○ 1,2
- ○ 2,2

✓ Given the following pieces of code. 1 will be printed x times and hello y times. The values of x and y are 2/2

```
Prog.(b)

main() {
fork();
fork();
fork();
int i=0; printf(" %d\n", i+1);
printf(" Hello\n"); }
```

- 3,3
- 4,4
- 7,8
- 8,8 ✓

✗ Given the following pieces of code. 1 will be printed x times and hello y times. The values of x and y are 0/4

```
Prog.(c)

main() {
int i=0; printf(" %d\n", i+1);
fork();
fork();
printf(" Hello\n");
fork(); }
```

- 2,2
- 2,4
- 1,4
- 4,4 ✗

✓ Which of the following statements is not true for the Multi-Level Feedback Queue processor scheduling algorithm? 2/2

- Queues have different priorities
- Each queue may have different scheduling algorithm
- Processes are permanently assigned to a queue ✓
- This algorithm can be configured to match a specific system under design

✓ examples in which multithreading does not provide better performance than a single-threaded solution 2/2

- ○ Any kind of sequential program is not a good candidate to be threaded.
- ○ "shell" program such as the C-shell
- ○ program that calculates an individual tax return
- ◉ All of the above ✓

Comparison of FIFO, RR and SRTF Algorithms. Given the following mix of job, job lengths, and arrival times, assume a time slice of 10 and compute the completion for each job and average response time for the FIFO, RR, and SRTF algorithms

| Job | Length (secs) | Arrival time |
|---|---|---|
| | | |
| 0 | 85 | 0 |
| 1 | 30 | 10 |
| 2 | 35 | 10 |
| 3 | 20 | 80 |
| 4 | 50 | 85 |

| | 111 | 120 | 75 | 80 | Score | |
|---|---|---|---|---|---|---|
| FIFO-Completion Time (Average) | ☑ | ☐ | ☐ | ☐ | 1/1 | ✓ |
| RR Completion Time (Average) | ☐ | ☑ | ☐ | ☐ | 1/1 | ✓ |
| SRTF Completion Time (Average) | ☐ | ☐ | ☐ | ☑ | 1/1 | ✓ |

Comparison of FIFO, RR and SRTF Algorithms. Given the following mix of job, job lengths, and arrival times, assume a time slice of 10 and compute the completion for each job and average response time for the FIFO, RR, and SRTF algorithms

| Job | Length (secs) | Arrival time |
|---|---|---|
| | | |
| 0 | 85 | 0 |
| 1 | 30 | 10 |
| 2 | 35 | 10 |
| 3 | 20 | 80 |
| 4 | 50 | 85 |

| | 145 | 220 | 80 | 215 | 125 | Score | |
|---|---|---|---|---|---|---|---|
| RR - Completion Time P0 | ☐ | ☑ | ☐ | ☐ | ☐ | 1/1 | ✓ |
| RR - Completion Time P1 | ☐ | ☐ | ☑ | ☐ | ☐ | 1/1 | ✓ |
| RR - Completion Time P2 | ☐ | ☐ | ☐ | ☐ | ☑ | 1/1 | ✓ |
| RR - Completion Time P3 | ☑ | ☐ | ☐ | ☐ | ☐ | 1/1 | ✓ |
| RR - Completion Time P4 | ☐ | ☐ | ☐ | ☑ | ☐ | 1/1 | ✓ |

Comparison of FIFO, RR and SRTF Algorithms. Given the following mix of job, job lengths, and arrival times, assume a time slice of 10 and compute the completion for each job and average response time for the FIFO, RR, and SRTF algorithms

| Job | Length (secs) | Arrival time |
|---|---|---|
| | | |
| 0 | 85 | 0 |
| 1 | 30 | 10 |
| 2 | 35 | 10 |
| 3 | 20 | 80 |
| 4 | 50 | 85 |

| | 150 | 100 | 75 | 40 | 220 | Score | |
|---|---|---|---|---|---|---|---|
| SRTF - Completion Time P0 | ☐ | ☐ | ☐ | ☐ | ☑ | 1/1 | ✓ |
| SRTF - Completion Time P1 | ☐ | ☐ | ☐ | ☑ | ☐ | 1/1 | ✓ |
| SRTF - Completion Time P2 | ☐ | ☐ | ☑ | ☐ | ☐ | 1/1 | ✓ |
| SRTF - Completion Time P3 | ☐ | ☑ | ☐ | ☐ | ☐ | 1/1 | ✓ |
| SRTF - Completion Time P4 | ☑ | ☐ | ☐ | ☐ | ☐ | 1/1 | ✓ |

Comparison of FIFO, RR and SRTF Algorithms. Given the following mix of job, job lengths, and arrival times, assume a time slice of 10 and compute the completion for each job and average response time for the FIFO, RR, and SRTF algorithms

| Job | Length (secs) | Arrival time |
|---|---|---|
| | | |
| 0 | 85 | 0 |
| 1 | 30 | 10 |
| 2 | 35 | 10 |
| 3 | 20 | 80 |
| 4 | 50 | 85 |

| | 70 | 220 | 115 | 130 | 65 | Score | |
|---|---|---|---|---|---|---|---|
| RR - Response Time P0 | ☐ | ☑ | ☐ | ☐ | ☐ | 1/1 | ✓ |
| RR - Response Time P1 | ☑ | ☐ | ☐ | ☐ | ☐ | 1/1 | ✓ |
| RR - Response Time P2 | ☐ | ☐ | ☑ | ☐ | ☐ | 1/1 | ✓ |
| RR - Response Time P3 | ☐ | ☐ | ☐ | ☐ | ☑ | 1/1 | ✓ |
| RR - Response Time P4 | ☐ | ☐ | ☐ | ☑ | ☐ | 1/1 | ✓ |

Comparison of FIFO, RR and SRTF Algorithms. Given the following mix of job, job lengths, and arrival times, assume a time slice of 10 and compute the completion for each job and average response time for the FIFO, RR, and SRTF algorithms

| Job | Length (secs) | Arrival time |
|---|---|---|
| | | |
| 0 | 85 | 0 |
| 1 | 30 | 10 |
| 2 | 35 | 10 |
| 3 | 20 | 80 |
| 4 | 50 | 85 |

| | 30 | 220 | sixty five | 20 | 65 | Score | |
|---|---|---|---|---|---|---|---|
| SRTF-Response Time P0 | ☐ | ☑ | ☐ | ☐ | ☐ | 1/1 | ✓ |
| SRTF - Response Time P1 | ☑ | ☐ | ☐ | ☐ | ☐ | 1/1 | ✓ |
| SRTF-Response Time P2 | ☐ | ☐ | ☑ | ☐ | ☐ | 1/1 | ✓ |
| SRTF - Response Time P3 | ☐ | ☐ | ☐ | ☑ | ☐ | 1/1 | ✓ |
| SRTF - Response Time P4 | ☐ | ☐ | ☐ | ☐ | ☑ | 1/1 | ✓ |

Mach

| | selects from jobs in memory those jobs that are ready to execute and allocates the CPU to them | A swapping scheme is implemented to remove partially run programs from memory and reinstate them later to continue where they left off. | determines which jobs are brought into memory for processing. | None | Score | |
|---|---|---|---|---|---|---|
| Short-term Scheduler | ☑ | ☐ | ☐ | ☐ | 1/1 | ✓ |
| Medium-term | ☐ | ☑ | ☐ | ☐ | 1/1 | ✓ |
| Long-term (job scheduler) | ☐ | ☐ | ☑ | ☐ | 1/1 | ✓ |