Mid-term Assignment

Course Name: Operating System

Submitted By:

Abdul Razzaq (12938)

BS (SE) Section: A

Submitted To:

Sir Daud

Dated: 21th April 2020

Mid Semester Assignment

Spring 2020

Subject: Operating System Concepts

Section A

Question No: 1 (M - 1)

The hardware mechanism that enables a device to notify CPU is called an ----------

► Interrupt
► Signal
► Trap
► Process

Question No: 2 (M - 1)

The section of the process control block comprises of page and segment tables

► Memory related information
► Accounting information
► Register information

► Scheduling information

Question No: 3 (M - 1)

The ---------- system call suspends the calling process.

► fork
► wait
► exec
► exit

Question No: 4 (M - 1)

In ---------addressing, the recipient is not required to name the sender.

► Symmetric
► Asymmetric
► Both symmetric and asymmetric
► None of the given options

Question No: 5 (M - 1)

-------- command gives a snapshot of the current processes.

► ps

► top
► who
► ls

Question No: 6 (M - 1)
----------command to resume the execution of a suspended job in the foreground
► fg

► bg
► jobs
► kill

Question No: 7 (M - 1)
You can use the ---------- command to display the status of suspended and background processes
► fg
► bg
► jobs
► kill

Question No: 8 (M - 1)
You can terminate a foreground process by pressing ----------
► <Ctrl-A>
► <Ctrl-C>
► <Ctrl-Z>
► None of the given options

Question No: 9 (M - 1)
A time sharing system is
► Multi-tasking
► Interactive
► Multi user
► All of these

Question No: 10 (M - 1)
The main characteristic of a Real time system is
► Efficiency
► Large Virtual Memory
► Large secondary storage device
► Usability

Question No: 11 (M - 1)

Shared libraries and kernel modules are stored in _____ directory

► /bin
► /dev
► /boot
► /lib

Question No: 12 (M - 1)
_____ scheduler selects the process from the job pool and put them in main memory.

► Long term
► Short term
► Medium term
► Swapper

Question No: 13 (M - 1)
In indirect inter process communication, a sender ___ mention the name of the recipient.

► do
► do not

Question No: 14 (M - 1)
A _____ is an integer variable that, apart from initialization is accessible only through two standard atomic operations: wait and signal.

► Semaphore
► Monitor
► Critical region
► Critical section

Question No: 15 (M - 1)
A semaphore that cause Busy-Waiting is termed as _____.

► Spinlock
► Monitor
► Critical region
► Critical section

Question No: 16 (M - 1)
The execution of critical sections must NOT be mutually exclusive

► True
► False

Question No: 17 (M - 1)
The performance of Round Robin algorithm does NOT depends heavily on the size of the time quantum.

► True
► False

Question No: 18 (M - 1)
The following requirement for solving critical section problem is known as _____.
"There exists a bound on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted."

► Progress
► Bounded Waiting
► Mutual Exclusion
► Critical Region

Question No: 19 (M - 1)
The critical section problem can be solved by the following except

► Software based solution
► Firmware based solution
► Operating system based solution
► Hardware based solution

Question No: 20 (M - 1)
_____ is also called Swapper.

► Swap space
► Medium term scheduler
► Short term scheduler
► Long term scheduler

Section B

Question No: 21 (M - 2)

Write the formula/ procedure for calculating the waiting time in preemptive Shortest Job First scheduling.

Question No: 22 (M - 3)

If a process exits and there are still threads of that process running, will they continue to run?

Question No: 23 (M - 5)

Considering the Resource sharing feature of thread, what do you think is 'resource sharing' an advantage of a thread or disadvantage of a thread. Explain yours answer briefly.

Solution to MCQ's:

1) Interrupt
2) Register information
3) Wait
4) A symmetric
5) Ps
6) Fg
7) Jobs
8) <Ctrl-C>
9) All of these
10) Usability
11) /lib
12) Long term
13) Do not
14) Semaphore
15) Spinlock
16) False
17) True
18) Bounded waiting
19) Firmware
20) Medium term scheduler

Section – B

Q#21 Answer:

In preemptive Shortest Job First scheduling, as the jobs arrive, they are put into ready queue for execution. The processes with having shortest burst time will be executed first and when the process is preempted, which means it's done, it is deleted from execution.

Process	Burst Time	Arrival Time
P1	9	0
P2	4	1
P3	5	2
P4	7	3
P5	3	4

P5	P2	P3	P4	P1	
0	3	7	12	19	28

Waiting time = total waiting time – no. of millisecond – arrival time.

P1= 20 – 1 – 0 = 19ms.

P2= 4 – 3 – 1 = 0ms.

P3= 8 – 0 – 2 = 6ms.

P4= 13 – 0 – 3 = 10ms.

P5= 5 – 0 – 4 = 1ms.

Total waiting time = 36ms**.**

Average waiting time will be:

36/5= 7.2ms

- P1 arrives at first, its arrival time is zero. At zero millisecond we have only one process available in ready queue which is p1. So p1 will get the CPU first.
- After one millisecond a new process comes into the ready queue which is p2 with burst time of 4 milliseconds. The current process running is p1 and is completed in one millisecond when p2 arrives. Now we have 2 processes in ready queue which are based on shortest CPU burst time.
- Now p3 arrives with the burst time of 5 milliseconds and gets into ready queue, p2 is executing with burst time of 3 milliseconds.
- Now there are three processes into the ready queue, again the burst time will be compared and the process with shortest burst time will be selected. Which is p2.
- In similar standards, whenever a new process arrives, their burst time will be compared and will be executed.

Preemptive SJF is also called shortest remaining time first because at any given point of time, the job with the shortest remaining time will be executed first.

Formula: Void Process Proc [], int n, int wt [])

Q#22) Answer:

No, when a process is ended, everything of that process ends with it. we include a process structure I this, the memory space and threads are included etc.

Q#23) Answer:

Yes, resource sharing is an advantage of thread because threads share common code data and other resources and the processes by default which enables multiple tasks to be performed simultaneously in one same single address space.