

NAME            FAYYAZ MUHAMMAD  
ID                14294  
SEMESTER       5<sup>TH</sup>  
SUBJECT         OPERATING SYSTEM

## QUESTION NO 1

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

❖ Interrupt

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

❖ Memory related information

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

❖ Wait

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

❖ Asymmetric

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

❖ Ps

➤ *-----command to resume the execution of a suspended job in the foreground*

❖ fg

➤ *You can use the ----- command to display the status of suspended and background processes*

❖ jobs

➤ You can terminate a foreground process by pressing  
-----

❖ <Ctrl-C>

➤ *A time sharing system is*

❖ All of these

➤ *The main characteristic of a Real time system is*

❖ Usability

➤ *Shared libraries and kernel modules are stored in \_\_\_\_\_ directory*

❖ /lib

➤ *\_\_\_\_\_ scheduler selects the process from the job pool and put them in main memory.*

❖ Long term

➤ *In indirect inter process communication, a sender \_\_\_\_\_ mention the name of the recipient.*

❖ do not

➤ *A \_\_\_\_\_ is an integer variable that, apart from initialization is accessible only through two standard atomic operations: wait and signal.*

❖ Semaphore

➤ *A semaphore that cause Busy-Waiting is termed as*

\_\_\_\_\_.

❖ Spinlock

- *The execution of critical sections must NOT be mutually exclusive*

❖ False

- *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.”*

❖ Bounded Waiting

❖

- *The performance of Round Robin algorithm does NOT depends heavily on the size of the time quantum.*

❖ True

- *The critical section problem can be solved by the following except.*

❖ Firmware based solution

➤ \_\_\_\_\_ *is also called Swapper.*

❖ Medium term scheduler

## SECTION B

### *QUESTION NO 2:*

ANSWER :

if the thread in the process is running and receives a signal(say Ctrl-C) and the default action of the signal is to terminate a process, does the running thread terminates or the parent process will also terminate. That happens to the threads if the running process terminates because of some signal.

### QUESTION NO 1:

ANSWER:

Preemptive SJF scheduling is sometimes called shortest-remaining-time-first scheduling. We illustrate the working of the SJF algorithm by using the following system state.

Process	Arrival Time	Burst Time
P1	0.0	7
P2	2.0	4
P3	4.0	1
P4	5.0	4

## QUESTION NO 3

ANSWER:

### The Advantages and Disadvantages of Threads

Advantages of threads are:

- **Responsiveness:**

Multithreading an interactive application may allow a program to continue running even if part of it is blocked or is performing a lengthy operation, thereby increasing responsiveness to the user.

- **Resource sharing:**

By default, threads share the memory and the resources of the process to which they belong. Code sharing allows an application to have several different threads of activity all within the same address space.

- **Economy:**

Allocating memory and resources for process creation is costly. Alternatively, because threads share resources of the process to which they belong, it is more economical to create and context switch threads.

- **Utilization of multiprocessor architectures:**

The benefits of multithreading of multithreading can be greatly increased in a multiprocessor environment, where each thread may be running in parallel on a different processor. A single threaded process can only run on one CPU no matter how many are available.

## Disadvantages of threads are:

- **Resource sharing:**

Whereas resource sharing is one of the major advantages of threads, it is also a disadvantage because proper synchronization is needed between threads for accessing the shared resources (e.g., data and files).

- **Difficult programming model:**

It is difficult to write, debug, and maintain multi-threaded programs for an average user. This is particularly true when it comes to writing code for synchronized access to shared resources.

