Exam: MID SEMESTER ASSIGNMENT

SPRING 2020

Paper: Operating System   Teacher Name: Sir. Muhammad Dawood

Semester: 3rd

Name: AMIR ABBAS   **ID: 15499**

Section A

Question No: 1 (M - 1)

The hardware mechanism that enables a device to notify CPU is called an **Interrupt**.

► **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 **wait** *system call suspends the calling process.*

► fork
► **wait** ☑
► exec
► exit

Question No: 4 (M - 1)

*In **Asymmetric** 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)

***ps** command gives a snapshot of the current processes.*

►**ps** ☑

► top

► who

► ls

Question No: 6 (M - 1)

***fg** 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 **Jobs** 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-C>**

► <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 **/lib** directory*

► /bin

► /dev

► /boot

► **/lib** ☑

Question No: 12 (M - 1)

***Long term** 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 **<u>Semaphore</u>** *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 **<u>Spinlock</u>**.

► **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
Bounded Waiting.
"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)
Medium term scheduler *is also called Swapper.*

► Swap space

► **Medium term scheduler** ☑
► Short term scheduler
► Long term scheduler

Section B   (Subjective Part)

Question No: 21 (M - 2)
Write the formula/ procedure for calculating the waiting time in preemptive Shortest Job First scheduling.

Ans: 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: 22 (M - 3)
If a process exits and there are still threads of that process running, will they continue to run?

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

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.

Ans: **The advantages and Disadvantages of Threads:**

Four main advantages of thread are:

1. **Responsiveness:** Multithreading and 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.
2. **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.
3. **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
4. **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.

Multithreading on multi-CPU machines increases concurrency.

Some of the main disadvantages of thread are:

1. **Resource sharing:** whereas 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)
2. **Difficult programming model:** it is difficult to write, debug, and maintain multithreaded programs for an average user. This is particularly true when it comes to writing code for synchronized access to shared resources.