BUBT | BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY

Committed to Academic Excellence

Assignment-1

Course Code: CSE 309

Course Title: Operating Systems

Submitted to:

Name: Suman Saha

Assistant Professor

Dept. of CSE

at Bangladesh University of Business and Technology.

Submitted by:

Name: Syeda Nowshin Ibnat

ID: 17183103020

Intake: 39

Section: 01

Program: B.Sc. in CSE

Date of Submission: 21/01/21

Answer to the question (1)

Timesharing:

Time sharing is the sharing of computing resources among several users at the same time. In time sharing systems, several terminals are attached to a single dedicated server having its own CPU.

Multiprogramming:

Multiprogramming is the fast switching of CPU between several programs. A program is generally made up of several tasks. A task usually ends with some request to move data which would require some I/O operations to be executed.

The differences between timesharing and multiprogramming systems:

No.	Timesharing	Multiprogramming
1.	Processors time is shared with multiple users that's why it is called as time sharing operating system.	In this, the process can be executed by a single processor.
2.	Time sharing OS has fixed time slice.	Multi-programming OS has no fixed time slice.
3.	Time sharing system maximizes response time.	Multiprogramming system maximizes response time.
4.	System model of time sharing system is multiple programs and multiple users.	System model of multiprogramming system is multiple programs.
5.	Example: Windows NT.	Example: Mac OS.

Answer to the question (2)

Empirical evidence shows that memory access exhibits the principle of locality of reference, where if one location is read then the probability of accessing nearby locations next is very high, particularly the following memory locations. So, by caching an entire cache line, the probability of a cache hit next is increased. Also, modern hardware can do a block transfer of 32 or 64 bytes into a cache line much faster than reading the same data as individual words.

Answer to the question (3)

By placing the operating system in a memory partition that could not be modified by either the user job or the operating system itself, the difficulties are listed below:

i. The critical data such as passwords and access control information that are required by or generated by the operating system would have to be passed through or stored in unprotected memory slots and would be accessible to unauthorized users.
ii. The operating system could never be updated or patched, since it is not modifiable or accessible by the user or the operating system itself.

Answer to the question (4)

We can use two types of clustering:

i. Asymmetric clustering: Here one node will do all work, while the other will be in standby and will activate in the case that the active node fails.

Benefit: The benefit is that the system will clearly not halt if the active system fails.

Disadvantage: The disadvantage is that it is not really efficient, we have two nodes, but we actively only use one of them.

ii. Symmetric clustering: To be more precise, we will observe so called parallel clustering. In this case multiple nodes (here 2 of them) can read the data from the disk.

Benefit: The benefit is that a system with parallel clustering is far more efficient than a system with asymmetric clustering.

Disadvantage: The disadvantage is that we need the parallel clustering. special software to implement RESULT HINT, asymmetric and symmetric (parallel) clustering.

Answer to the question (5)

Context switching between kernel threads typically requires saving the value of the CPU registers from the thread being switched out and restoring the CPU registers of the new thread being scheduled.

Actions taken by a kernel to context-switch between processes are -

1.The OS must save the PC and user stack pointer of the currently executing process, in response to a clock interrupt and transfers control to the kernel clock interrupt handler

2.Saving the rest of the registers, as well as other machine state, such as the state of the floating point registers, in the process PCB is done by the clock interrupt handler.

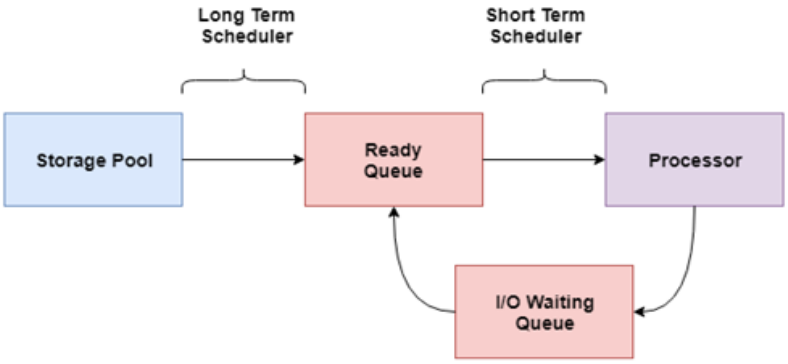
3.The scheduler to determine the next process to execute is invoked the OS.

4.Then the state of the next process from its PCB is retrieved by OS and restores the registers. The restore operation takes the processor back to the state in which the previous process was previously interrupted, executing in user code with user-mode privileges.

Many architecture-specific operations, including flushing data and instruction caches also must be performed by Context switches

Answer to the question (6)

- Short-term (CPU scheduler): selects a process from those that are in memory and ready to execute, and allocates the CPU to it.
- Medium-term (memory manager): selects processes from the ready or blocked queue and removes them from memory, then reinstates them later to continue running.
- Long-term (job scheduler): determines which jobs are brought into the system for processing.



Representation of Short Term and Long Term Scheduler using a Queuing Diagram

The differences among short-term, medium-term, and long- term scheduling:

No.	Short-term	Medium-term	Long- term
1.	It is CPU scheduler	It is process swapping scheduler.	It is job Scheduler.
2.	Speed is fastest than other two schedulers and invoked frequently.	speed is between both short and long term scheduler.	speed is lesser than short term scheduler.
3.	It provide lesser control over degree of multiprogramming.	It reduces the degree of multiprogramming.	It controls degree of multiprogramming.
4.	Transition of process state from ready to executing state.	No process state transition.	Transition of process state from new to ready state.
5.	Supplies a mix of jobs, such as I/O bound and CPU bound.	Selects a new process to allocate CPU frequently.	Process are swapped in and out for balanced process mix.