Time: 30 Minutes CS-2005 Operating Systems Marks: 20

Quiz-1

1. What is the purpose of multiprogramming? Is it good idea to run some process all time to achieve multiprogramming? Briefly explain.

   Solution

   Multiprogramming aims to maximize CPU utilization by allowing multiple programs to execute concurrently. It enables the CPU to switch between different processes, ensuring that the processor is continuously busy and minimizing idle time.

   Running processes all the time isn't needed for multiprogramming. The operating system dynamically schedules processes based on factors like priority and resource availability. While some processes may run continuously will result in starvation of other processes for resources.

2. Define cache coherency and describe how I/O request (read/write) is fulfilled from user to till I/O device by giving sequence to the following process.

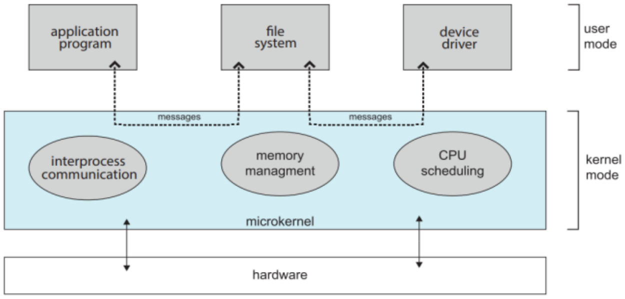
   System Call Handling, User-space Request, I/O Scheduling, Kernel Space Handling, Data Transfer, User-space Notification, Device Driver Interaction, Interrupt Handling.

   Solution:

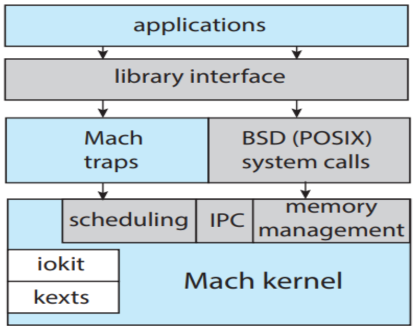
   Cache coherency ensures that multiple caches storing copies of the same data remain synchronized to maintain data consistency.

   1. User-space Request: The I/O request originates from a user application through a system call, such as read() or write().
   2. Kernel Space Handling: Upon receiving the system call interrupt, the CPU switches to kernel mode to execute the corresponding kernel code.
   3. System Call Handling: The operating system kernel identifies and processes the system call, which involves validating parameters, allocating resources, and initiating the I/O operation.
   4. I/O Scheduling: If necessary, the operating system schedules the I/O request along with other pending requests, considering factors like priority and fairness.
   5. Device Driver Interaction: The operating system interacts with the appropriate device driver to translate the request into hardware-specific commands understandable by the I/O device.
   6. Data Transfer: The device driver initiates data transfer between the CPU's memory and the I/O device. This can involve DMA (Direct Memory Access) for efficient data movement without CPU intervention.
   7. Interrupt Handling: During data transfer, the I/O device generates interruptions to signal completion or request further action. The CPU handles these interruptions by executing interrupt service routines (ISRs) to update the system state or notify waiting processes.
   8. User-space Notification: Once the I/O operation is completed, the user application may be notified of the result through mechanisms like callbacks, signals, or by returning from the system call with relevant information.

3. Identify the architecture in the following diagram and state the differences accordingly to structures.



(a)



(b)

Solution:

The diagram (a) is typical microkernel structures, and this structure the operating system by removing all nonessential components from the kernel and implementing them as userlevel programs that reside in separate address spaces. The result is a smaller kernel. There is little consensus regarding which services should remain in the kernel and which should be implemented in user space.

In diagram (b) is Darwin Mach kernel, which uses a hybrid structure. Darwin is a layered system that consists primarily of the Mach microkernel and the BSD UNIX kernel.