

1.16: Direct memory access is used for high-speed I/O devices in order to avoid increasing the CPU's execution load.

(a) How does the CPU interface with the device to coordinate the transfer? (b) How does the CPU know when the memory operations are complete? (c) The CPU is allowed to execute other programs while the DMA controller is transferring data. Does this process interfere with the execution of the user programs? If so, describe what forms of interference are caused.

Answer: (a) All devices have their hardware controller. The OS has device drivers that communicates with the controllers. The device drivers have registers, counters and buffer to store arguments and results. To not let CPU sit idle while the I/O is being performed, asynchronous I/O is used. The CPU can perform other tasks while the I/O is being performed. The device driver signals the CPU when the I/O is complete. (b) The device controller sends an interrupt to the CPU so it know that the I/O is complete. (c) No. However, when the device controller sends an interrupt to the CPU, a user process could be suspended.

2.15: What are the two models of interprocess communication? What are the strengths and weakness of the two approaches?

Answer: The two models of interprocess communication are shared memory and message passing. In shared memory, the processes communicate with each other by reading and writing data to a shared area in memory. In message passing, the processes communicate with each other by sending and receiving messages.

	Shared Memory	Message Passing
Speed	Faster	Slower
Overhead	no	yes
synchronization issues	yes	no

2.19: What is the main advantage of the microkernel approach to system design? How do user programs and system services interact in a microkernel architecture? What are the disadvantages of using the microkernel approach?

Answer: There are two main advantages. First, the microkernel approach is easier to extend and port the OS to new architecture. Second, it is more reliable and more secure because less code is running in kernel mode. the user programs and system service interact through message passing. One of the detriments is that message passing introduces overhead.

3.12: Describe the actions taken by a kernel to context-switch between processes.

Answer: The CPU save the old state of the process in the PCB, and then loads the new state of the process from the PCB. The time taken to switch between processes is pure overhead.

3.18: Give an example of a situation in which ordinary pipes are more suitable than named pipes and an example of a situation in which named pipes are more suitable than ordinary pipes.

Answer: Ordinary pipes allow communication in a standard producer-consumer scenario. It is suitable to use ordinary pipes when the processes are related and are running on the same machine. Named pipes, on the other hand, allow communication between unrelated processes. It is suitable to use named pipes when the processes are unrelated or are running on different machines.