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?

To initiate a DMA transfer, the CPU first sets up the DMA registers, which contain a pointer to the source of a transfer, a pointer to the destination of the transfer, and a counter of the number of bytes to be transferred. Then the DMA controller proceeds to place addresses on the bus to perform transfers, while the CPU is available to accomplish other work.

(b) How does the CPU know when the memory operations are complete?

Once the entire transfer is finished, the DMA controller interrupts the CPU.

(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.

Both the CPU and the DMA controller are bus masters. A problem would be created if both the CPU and the DMA controller want to access the memory at the same time. Accordingly, the CPU should be momentarily prevented from accessing main memory when the DMA controller seizes the memory bus. However, if the CPU is still allowed to access data in its primary and secondary caches, a coherency issue may be created if both the CPU and the DMA controller update the same memory locations

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

Shared Memory strength: faster than message passing weakness: need to handle syncronization since two processes may access the same memory at the same time

Message Passing: strength: no need to handle syncronization since there is no overlap between the physical memory of the processes. weakness: slower than shared memory, since it need to copy the message and required to set up before passing message.

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?

advantages: easier to extend a microkernel easier to port the OS to new architectures more reliable (less code is running in kernel mode) more secure

User programs and system services interact in a microkernel architecture by using interprocess communication mechanisms such as messaging

disadvatages: performance overhead of user space to kernel space communication

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

the system must save the state of the old process and load the saved state for the new process via a context switch.

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.

ordinary pipes:

Ordinary pipes are ideal for simple parent-child processes communication, since named pipe might be too overhead. Like unix can use command '|' to pipe data in other process.

named pipes:

Named pipes are more suitable for complex IPC scenarios, especially when communication needs to occur between unrelated processes or processes that do not have a parent-child relationship.