HW #1

- 1.16

a. CPU will sets up DMA register, and have pointers to the source and destination of transfer, and also a counter of number of bytes to transferred. Then DMA starts to process, put address to bus to perform transfer. CPU could do other works.

b. When transfer is completed, the DMA controller interrupts the CPU

c. If both CPU and DMA wants to access memories at the same time, it would cause some problems. CPU should avoid access the memory when DMA seizes the memory bus. If the CPU still access data in primary and secondary caches, the CPU and the DMA may update the same memory locations.

- 2.15

Share memory & message passing
"Share memory" has different process, it can communciate only by reading or writing, thus it is fast. Some synchronizations need to avoid 2 or more processes to access same memory in the same time.
"Message passing" requires less synchronizations than shared memory because there is no overlap between the physical memory of the process. But more overhead required to setup message passing between processes.

- 2.19

(a) Microkernel is earsier to design and maintain because it implemented using a small kernel and a separate privileged systems.

(b) User level programs make most service requests to system processes by IPC. The processes fulfill the service requests by making system calls to the microkernel or requests to other processes.

(c) It has abundant communciation between user and kernel mode, OS may be slower than general.

- 3.12

Context switching between kernel-level threads usually saving the value of CPU registers from the thread being switched out and restoring the CPU registers when new thread being scheduled.

- 3.18

Communications which are simple work well in ordinary pipes. For instance, assume we have a process to counts characteries in a file. Ordinary pipes can be used where the producer writes the file to the pipe and consumer read the file and count the number of characters.

Communications which are complicate work well in named pipes. For instance, if there are a lot of files want to write content into log. When process wish to write a message to log, they write it to the named pipe. A server reads the messages from the named pipe and write back into log.