

(1.9)

- a) To initiate a DMA transfer, the CPU will set up the DMA registers first, which contains a pointer byte for transferring, a pointer to the destination transfer & finally a counter of the number of transfers. While the CPU is available to finish another work.
- b) Once the transfers are done, the DMA controller will interrupt the CPU
- c) The CPU & DMA controllers are bus masters. A problem is formed when both the CPU & DMA masters wants to access the memory at the same time. The CPU will momentarily prevented from accessing main memory when the DMA controller seizes the memory bus accordingly. However, if the CPU is still allowed to access data in its primary & secondary caches, a coherency issue may be created if both CPU & DMA controller update the same memory location.

(2.7)

shared memory & message passing

Strength: shared memory is quick compared to the message passing model if the techniques are on a similar machine.

Weakness: various techniques are required to confirm which they aren't composing to the similar direction concurrently.

(2.10)

Advantage:

- ↳ It is easier to extend the operating system. All new services are added to userspace & consequently do not require modification of the kernel.
- ↳ provides more security & reliability, since the components included can disrupt the functionality of the system otherwise.
- ↳ microkernel architecture is isolated & small so it can function better.
- ↳ Its more accessible since it can be added to the system application without disturbing the kernel.

How user programs & system services interact? :

- ↳ By using interprocess communication mechanisms such as messaging which are conveyed by operating system

Disadvantage :

- ↳ It is expensive compared to ^{→ normal} monolithic system.
- ↳ A context switch or a function call is needed when the drivers are implemented as procedures or processes.
- ↳ the performance can be indifferent & may lead to some problems.

(2.1)

long term	Medium-term	Short-term
<ul style="list-style-type: none"> ↳ Job scheduler ↳ the speed is slower than short term ↳ controls the degree of multiprogramming. ↳ almost zero or minimal in time sharing system ↳ selects processes from the pool & loads them into memory for execution 	<ul style="list-style-type: none"> ↳ swapping scheduler ↳ the speed is in between long & short term ↳ reduces the degree of multiprogramming. ↳ a part of the time sharing system ↳ can reintroduce the process into memory & execution can be continued. 	<ul style="list-style-type: none"> ↳ CPU scheduler ↳ the fastest among the three. ↳ provides lesser control over the degree of multi programming. ↳ minimal time sharing system ↳ selects those processes that are ready to execute.

(3.11)

- a) benefit: allows a rendezvous between sender & receiver
 Disadvantage: rendezvous might not be required & the message could be delivered asynchronously.
 As a result, message passing systems often provide both forms of synchronization.
- b) Automatic buffering provides a queue with indefinite length, which ensures the sender will never have to block while waiting to copy a message. There are no specification on how automatic buffering will be provided; one scheme might reserve enough large memory where most of the memory is wasted. Explicit buffering specifies how large the buffer is. In this situation, the sender might be blocked while waiting for available space in the queue. However, it is less likely that memory will be wasted with explicit buffering.
- c) Send by copy doesn't allow the receiver to alter the state of parameter; send by reference does allow it. A benefit of send by reference is that it allows the programmer to write a distributed version of a centralized application. Java RMI provides both; but, passing a parameter by reference requires declaring the parameter as a remote object as well.
- d) The implication of this are mostly related to buffering issues; with fixed-size message, a buffer with a specific size can hold a known number of messages, a buffer with a specific size can't hold a known number of messages. The number of variable sized messages that can be held by such a buffer is unknown.