**Summary**

IPC:

1. Shared memory – memory shared by two or more processes
2. Exchange messages - message passing between processes
3. Synchronization – processes wait until the other process completes it usage of a resource.

Message passing creating, sending and receiving the messages between processes.

OS creates and maintains a channel and it also provides interface to:

1. send/write messages to a port.
2. receive/read messages from a port.

Kernel is responsible for establishing communication, perform each IPC operation,

send() and recv().

Forms of message passing:

1. Pipes – Processes communicate from 2 ends and carry byte stream between processes.

Eg: Sending output from one process as another process’s input

1. Message queues - OS management includes priorities, scheduling of message delivery
2. Sockets:
3. send(), recv()
4. socket() - create kernel level socket buffer, associate necessary kernel level processing. If they are different machines, it channels between processes and network devices. If the same machine, it bypasses the full protocol stack.

4. Shared memory - read and write to shared memory region.

1. Physical pages mapped into virtual address space.
2. Virtual address of process 1 and virtual address of process 2 map to the same physical address.
3. Virtual address regions that correspond to shared buffer do not have the same virtual address space.
4. Physical memory doesn’t need to be contiguous.

Copy - CPU cycles to copy data to/from port.

Map - CPU cycles to map memory into address space. CPU to copy data to channel.

When a process requests that a shared memory segment is created, the OS allocation requires physical memory and assigns unique key.

1. Create - OS assigns unique key.
2. Attach - map virtual address to physical addresses.
3. Detach - invalidate address mappings
4. Destroy - only remove when explicitly deleted.

Shared memory and Synchronization methods:

1. Mechanisms supported by process threading library.
2. OS supported IPC for synchronization.

Other IPC synchronization methods are:

1. Message queues - implement mutual exclusion via send/recv.
2. Semaphores - binary semaphores which takes 0,1 values only.
3. If value is 0 then stop/blocked.
4. If value is 1 then decrement and go/proceed.