**Bilal Saleem SP 18-Mcs-019**

**Assignment #2**

**Subject : Operating System**

**Submitted to Sir Irshadullah**

Inter Process Communication

A process can be of two type:

* Independent process.
* Co-operating process.

 The communication between these processes can be seen as a method of co-operation between them. Processes can communicate with each other in following two ways:

1. Shared Memory
2. Message passing

**IPC Mechanism Working in Windows XP OS:**

The following IPC mechanisms are supported by Windows:

* [Clipboard](https://docs.microsoft.com/en-us/windows/desktop/ipc/interprocess-communications#using-the-clipboard-for-ipc)
* [Data Copy](https://docs.microsoft.com/en-us/windows/desktop/ipc/interprocess-communications#using-data-copy-for-ipc)
* [DDE](https://docs.microsoft.com/en-us/windows/desktop/ipc/interprocess-communications#using-dde-for-ipc)
* [File Mapping](https://docs.microsoft.com/en-us/windows/desktop/ipc/interprocess-communications#using-a-file-mapping-for-ipc)
* [Pipes](https://docs.microsoft.com/en-us/windows/desktop/ipc/interprocess-communications#using-pipes-for-ipc)
* [RPC](https://docs.microsoft.com/en-us/windows/desktop/ipc/interprocess-communications#using-rpc-for-ipc)
* [Windows Sockets](https://docs.microsoft.com/en-us/windows/desktop/ipc/interprocess-communications#using-windows-sockets-for-ipc)

Examples of IPC Systems - Windows XP OS provides support for multiple operating environments (i.e. system call APIs) for different types of processes (e.g. Windows Processes, MS-DOS processes, POSIX processes, etc.) using an easily extended subsystem architecture. Message-passing centric, via local procedure call (LPC) facility, so works only between processes on the same system

**IPC Mechanism Working in MAC OS:**

A micro-kernel architecture used within Mac OS where communication is message based Even system calls are messages Each task gets two mailboxes at creation, Kernel and Notify, to communicate with the kernel. Only three system calls needed for message transfer msg send(), msg receive(), msg rpc() The OS may allocate mailboxes in shared memory to reduce inefficient double copying of writes and reads between processes. Mailboxes needed for communication, created via port allocate().

**IPC Mechanism Working in UNIX OS:**

Process first creates shared memory segment: segment id = (IPC PRIVATE, size, S IRUSR | S IWUSR);

Process wanting access to that shared memory must attach to it: shared memory = (char \*) (segment id, NULL, 0);

Now the process could write to the shared memory: (shared memory, "Writing to shared memory");

When done, a process can detach the shared memory from its address space

(shared memory);