Interprocess Communication Series

1. What is interprocess Communication (IPC) ✓
2. Independent vs Cooperating process ✓ — Process / Threads
3. Advantage of interprocess communication ✓ — PCB
4. Models of interprocess communication ✓
5. Shared memory vs Message Passing ✓
6. Methods for implementing message passing ✓
7. Examples of IPC Systems in modern OS

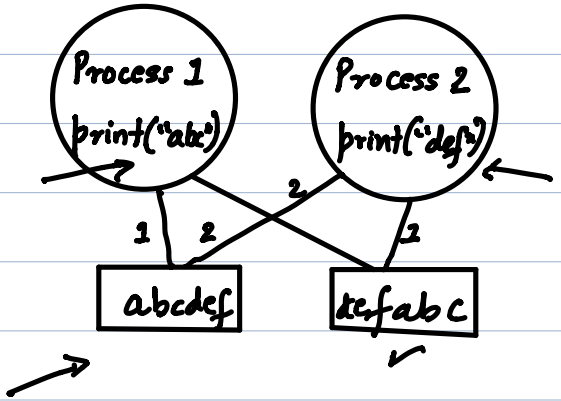
* Will talk in details about shared memory in next series - Process synchronization

Interprocess Communication

* A very important feature of an operating system
* Allows multiple process to communicate with each other - exchange data and information.

Independent vs Cooperating process

* Depending upon where a process shares data with others or not, we can classify processes into two.

Independent process	Cooperating process
* Doesn't share any data with any processes	shares data with other processes
* Cannot affect or be affected by other processes on system	can affect or be affected by other processes on system
* Result of the program is deterministic	Result of the program is not deterministic
* Behaviour is easily reproducible	Behaviour is not easily reproducible
* print("abc") print("def") Output: abcdef	

* Cooperating process requires an interprocess communication mechanism to allow them to exchange data.

Advantages of interprocess communication

Information sharing: Several programs may be interested in same piece of information.

Computation speedup:

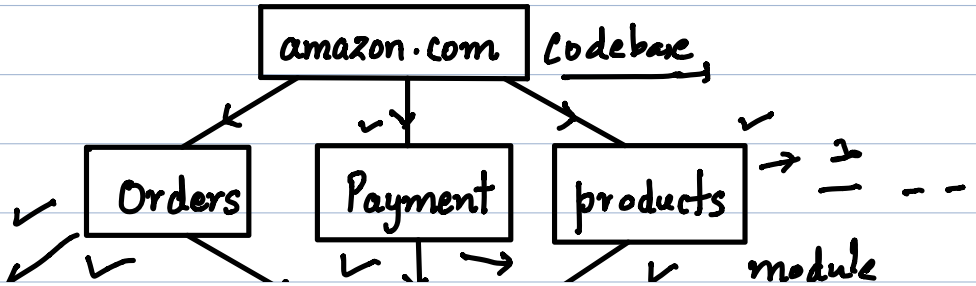
* Break up a program into multiple parts
* Run each part on a different processor
* Communicate between different parts using IPC mechanism.

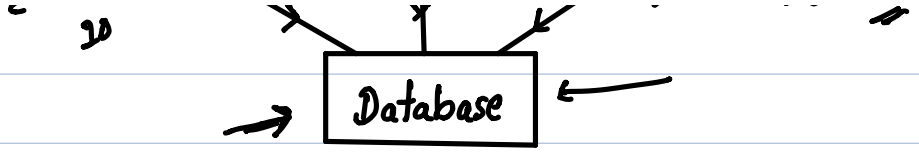
Modularity:

* Breaking up a program into multiple logical parts
* Makes understanding program easier
* Easy to maintain and debug code

Convenience:

* A single user can work on many tasks at same time
* He/she can edit, listen to music, compile etc at same time.





Two models of IPC :

i) shared memory
ii) Message passing