Zombie – ~~ts-el~~ | ps -el | grep Z

Orphan – Parent terminates, Child still executing
↳ no special marking in process table
Linux solves this by assigning init process as parent (not exactly)
↳ upstart → creates a process
↳ you'll need to check

Independent process

Cooperating process
→ Affects others, Shares data

Interprocess communication

Shared memory

- → Process can deposit information to a common mem
- → Only one process can write OR read at a time. (another process can't even read)

→ Creation of shared mem by 1 process & others attach to it

→ Only 2 syscalls

Message Passing

Message Passing Model

-> If ~~two~~ in a distributed System you can't have shared memory easily

-> Sender sends messages to recipients via a queue

-> Requires a LOT of system calls

Producer - Consumer Problems

$rate_{produced} \neq rate_{consumed}$

=> You need a store (buffer)
- -> once buffer is full, producer stops
- -> once buffer is empty, consumer waits

unbounded (no limit) <— —> bound

BUFFER_SIZE

Circular queue

Interprocess Communication

-> Hard to support variable sized messages

-> ~~Send~~

Let a process P send a message to Q
then P could mention Q in the send function. Or not.
-> Q can recieve by mentioning P

Direct   Indirect

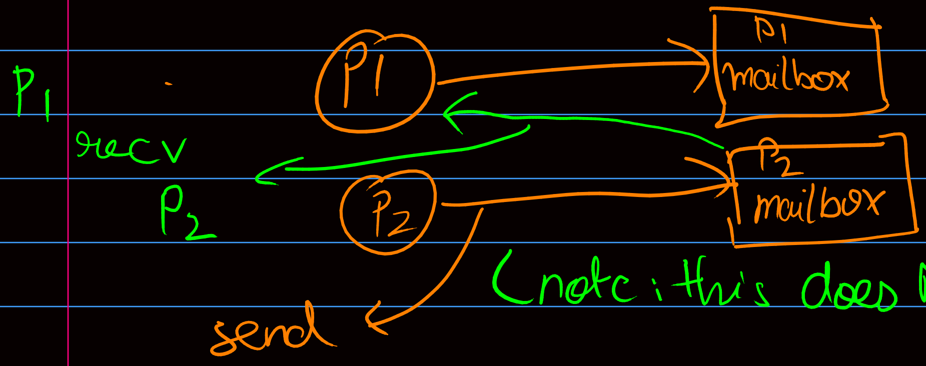
Direct link

Dedicated link (nothing else can connect in the meantime)

↓

Single link only between 2 processes

Indirect link : Create a mailbox



Processes can communicate if ~~Sh~~ they share the mailbox

(note: this does NOT work like shared memory)

NOT a dedicated link, any other process can view

mailbox a.k.a port → not a http/network port)

each process
- Create a mailbox
- Send/receive messages other processes
- destroy mailbox (Comm seizes to exist)