Process Management

Process : A program in execution (active)

* Text section (Program Code)
* A Stack (storing the function order
* Global Variables
* Heap
* Program Counter
* Contents of registers

- Process = Task = Job

  ↳ Process State Diagram

New :

being created

Running :

Using the CPU , executing instructions.

Blocked:

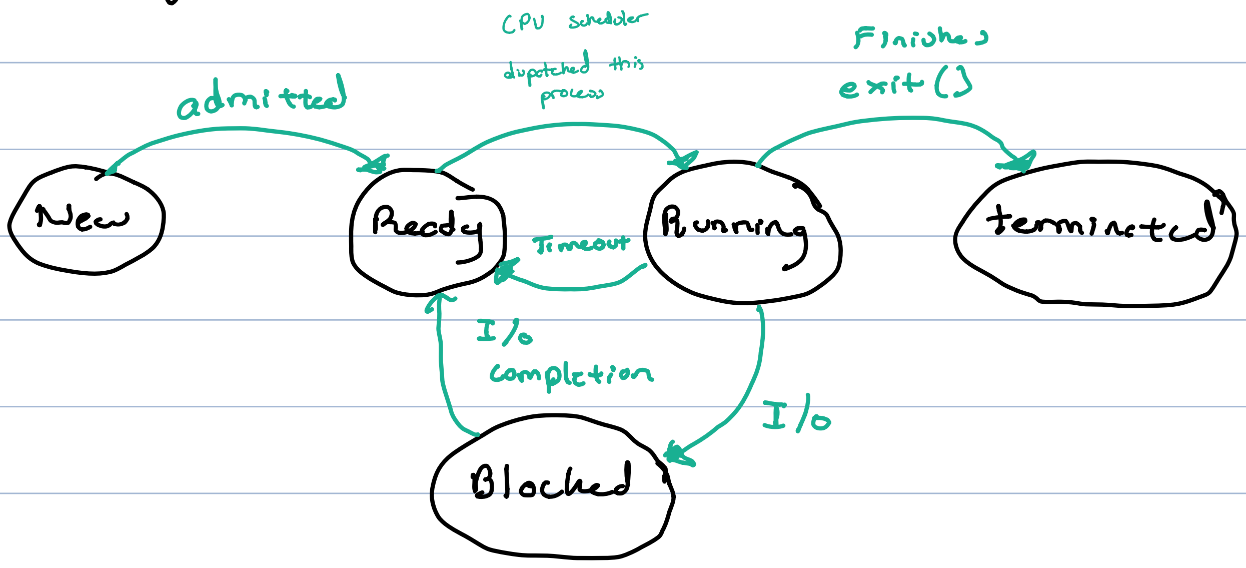
Waiting for Idle (waiting for an interupt)

Terminated:

Process is exiting

Ready :

Waiting for the CPU



PCB (Process control block) - Task control block

How the system views processes

↳ Each process is represented by PCB inside the OS.

Data Structure

* Process ID
* Priority
* Memory location
* Accounting Information (CPU Time, memory size)
* Process State
* Parent Process
* List open files / devices   is → child process
* Registers
* Program Counter
* Threading Information

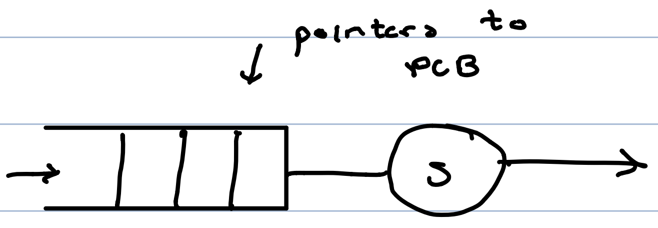
CPU Scheduling

Goal: is to maximize CPU Utilization

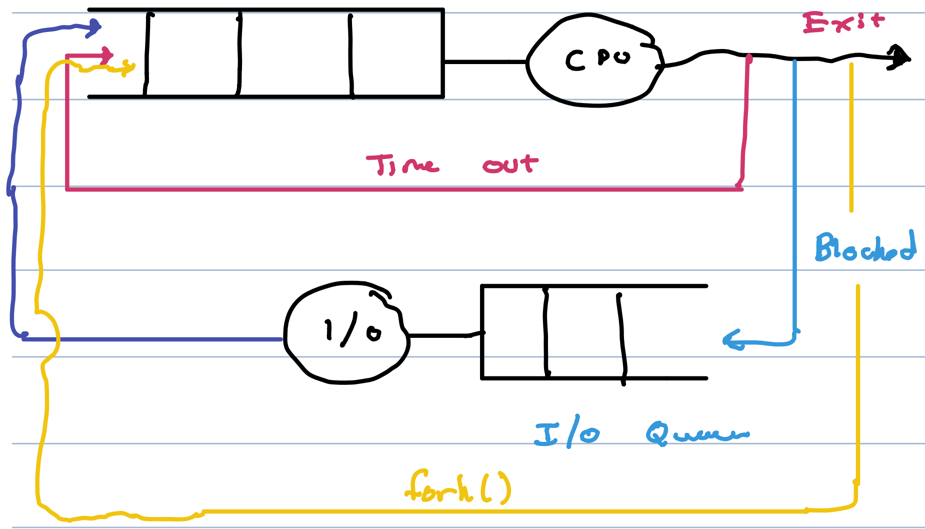
Scheduler: Selects which process gets the CPU

Two processes

P1	P2
1 millisec CPU	0.5 milli sec
	0.5 I/o sec
10 sec I/o	0.5 CPU
	0.5 I/o

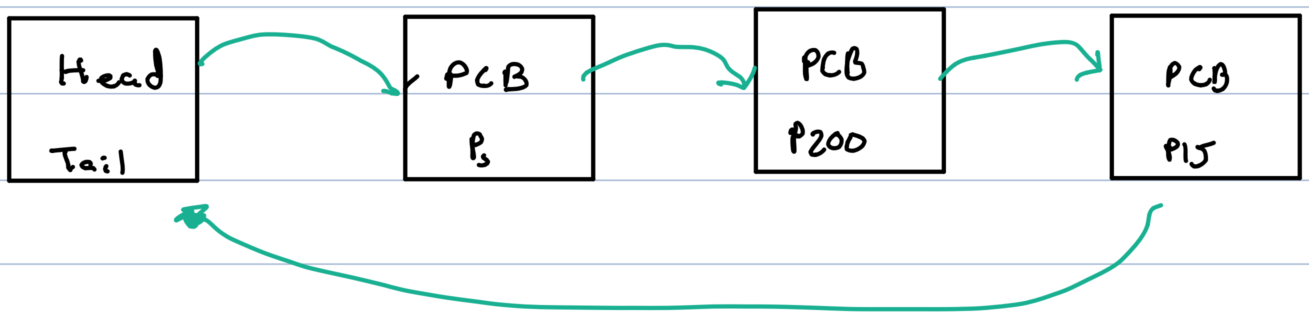


Ready Queue



• If process is being blocked

Ready Queue



Long term scheduler

Selects processes memory (admission)

Short - term Scheduling

Select a process to execute on the CPU

Scheduler also uses CPU to make scheduling decisions

es//

It takes 10 ms to pick a process to run for 100 msec

What % of CPU is wasted making scheduling decisions?

$$\frac{10\text{ msec}}{110\text{ msec}} \cdot 100 = 9\%$$

CPU - bound : Process spend most of it's time executing
I/o - bound : Process spend most of it's time doing I/o's

- Want to have the right mix of CPU vs I/O bound processes.

Medium - term Scheduler

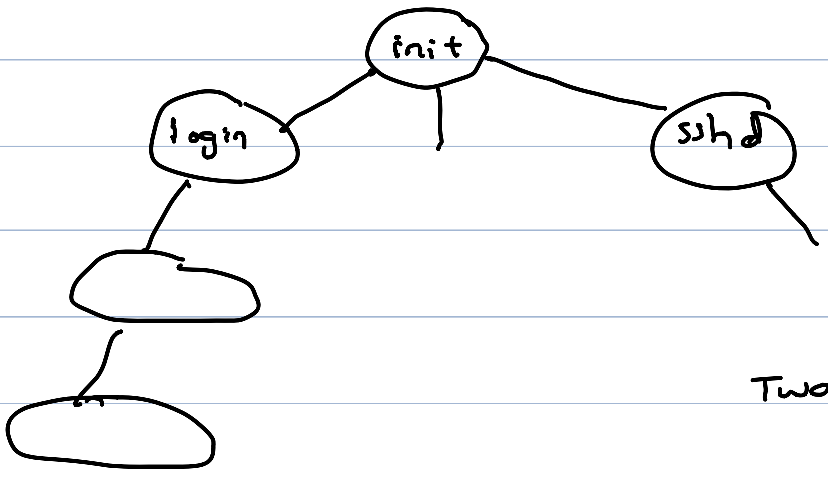
Swap processes in/out of memory to/from disk

Context switching

- Done by short - term scheduler
  ↳ Switching the CPU to another process (involves saving the state and restoring another)

Process Creation

A process creates another process (child)



Two comands at a time using &

- When a process is created
  1. Parent continues to run
  2. Parent waits until the child function finishes
     → wait (NULL)

Process termination

→ exit()

why a parent terminates a child process?

1. Exceed it's usage
2. No longer needed
3. Parent exiting