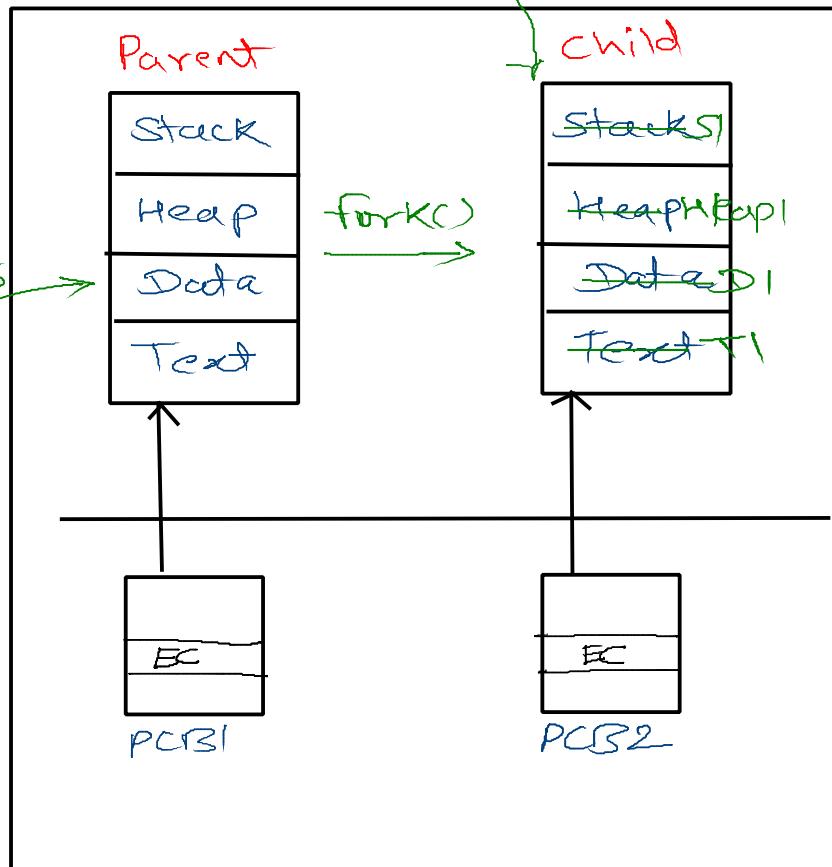
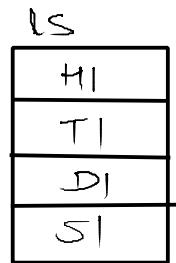


`wait()`

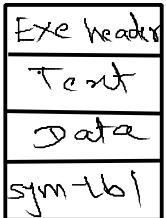
- `wait` is a blocking system call
- read exit status of child from its PCB
- clear the PCBs of child process
- return the exit status to calling process

exec("/usr/bin/ls", "ls", "-l", NULL);



cmd> ls -l
argc=2
argv[0] = "ls"
argv[1] = "-l"
argv[2] = NULL

demo\$



Data Structures in OS:

1) Job queue / process list:
- All processes of system

2) Ready queue:
- All processes which are ready for execution on CPU

3) Waiting queue:
- All processes which are waiting for I/O/Sync
- There are multiple waiting queues in system for every I/O device/sync(IPC)

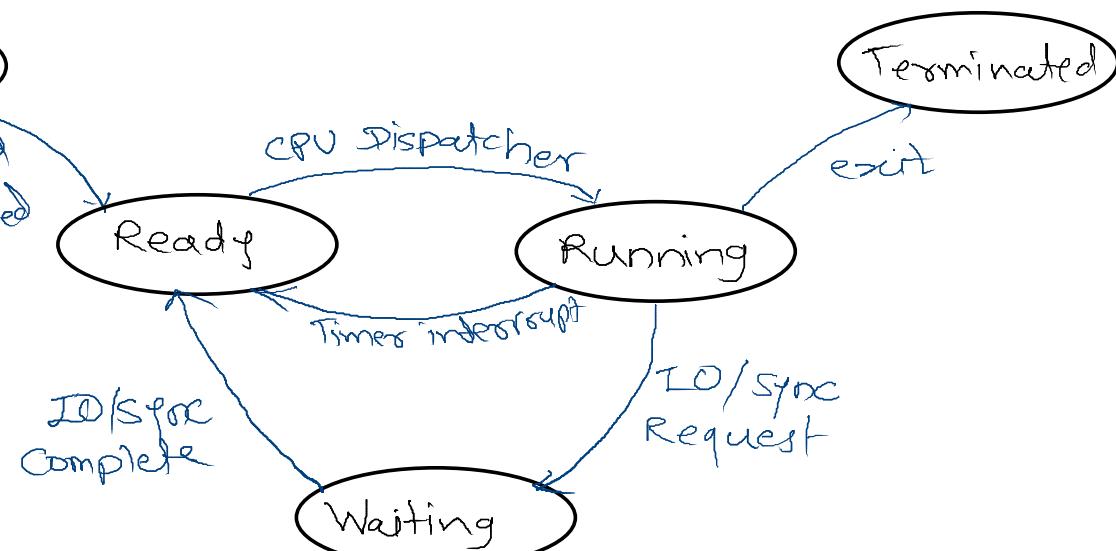
Non preemptive scheduling
① Running → Terminated
② Running → Waiting
③ Running → Ready
④ Waiting → Ready

Preemptive scheduling

Scheduling Algorithms

- ① FCFS
- ② SJF
- ③ Priority
- ④ RR
- ⑤ FS

Process Life Cycle



CPU scheduling Criterias:

① CPU utilization (ideal - Max)

- Desktop system - 70%
- Server system - 90%

② Throughput (ideal - max)

- amount of work done in unit time

③ Waiting time (ideal - min)

- time spent in ready queue

④ Response time (ideal - min)

- time from arrival of process in ready queue to first time executed by CPU (first waiting time)

⑤ Turn Around time(TAT) (ideal - min)

- total time spent in RAM

$$\text{CPU waiting} + \text{CPU burst} + \text{IO waiting} + \text{IO burst}$$