DIPLOMA WALLAH

(Your One Stop Hub For Diploma Resources)

OPERATING SYSTEM AND ADMINISTRATION

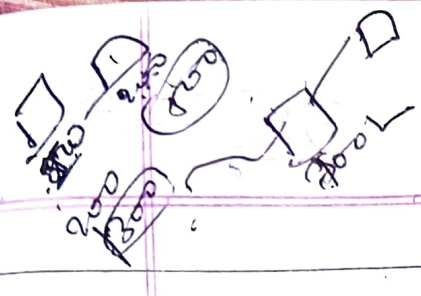
Complete Notes Based on Full Syllabus

- Diploma Engineering

4th Semester

www.diplomawallah.in

Notes prepared by **Sangam**



Unit-04

Process managements

process

- A process is basically a program in execution.
- It not just the program code (which is called a program or executable), but also the current activity of that program running on the computer.

It includes:-

- The program instruction being executed.
- The current values of program counters and registers.
- The variables and data used by the program.
- The resource (like files, memory, CPU time) it's using.

Key process

- When we run program (like opening browser), the OS creates a process for that program.
- Each process has unique Process ID (PID)
- A process can be in different states like:-
  - New: just created
  - Ready: waiting to be assigned CPU
  - Running: currently executing instructions on CPU
  - Waiting (Blocked): waiting for some event like input/output.
  - Terminated: finished execution.

* Daemon

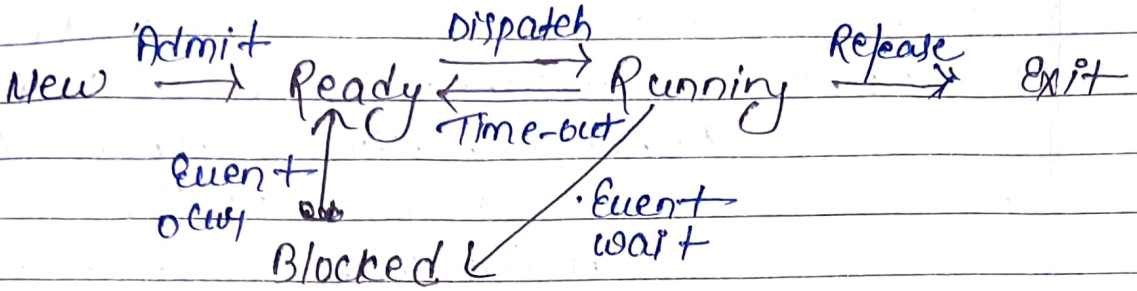
(1)

* A Daemon is a special type of background process that run without direct user interaction.
* It usually start when the system boots up and keeps running silently in the background, waiting to perform specific tasks or provide services.
* Daemons are common in unix/linux system, but similar background services exist in other OSes too.

Ex
* Sshd → manages secure remote login (SSH)
* http or apache - Runs a web server
* Cron → Schedules and runs periodic jobs.
* Syslogd → manages system logging.

* process States

- A process moves through different states during its lifecycle.
- These states describe what the process is currently doing or waiting for.
- The operating system uses these states to manage process execution efficiently.



- New → Ready : When a process is created it goes to the ready queue to wait for CPU.
- Ready → Running : When CPU Scheduler picks the process to run.
- Running → Waiting : If process needs to wait for I/O or some event.
- Waiting → Ready : When the waiting event is complete, process moves back to ready.
- Running → Terminated : When process finished or is killed.

✱ PCB (process Control Block)

- A process control block is a data structure used by the O.S. to store all the info. about a process.
- It acts like a process's identity card for the OS, keeping track of everything the OS needs to manage the process.

PCB Contains

→ PID → unique identifier for the process.
→ process state → current state (New, Ready, Running, waiting, Terminated.
→ program Counter → Address of the next instruction to execute.
→ CPU Registers – Contents of CPU registers during process switch.
→ memory management info – Information about the process memory (eg, base and limit registers, page tables).
→ Scheduling info – priority, scheduling queue pointers.
→ Accounting info – CPU usage, time limits, process execution stats.
→ I/O Status info – list of I/O devices allocated to the process.

* process Scheduling Queue
• The OS maintains queues to keep track of processes based on their current states (like Ready, waiting).
• The most common queue is the Ready Queue, which holds processes waiting to get CPU time.

Queue Types
• Ready Queue :– process waiting for CPU scheduling
• waiting (Blocked) Queue :– process waiting for some event (like I/O).

Job Queue :- All process in the System.

* operation on Queues
  - Enqueue :- Add a process to a queue. (when process become ready.)
  - Dequeue :- Remove a process from a queue (when process start running)

* Operation on process Creation
When a new process is created, the OS perform these steps:
- Assign a unique PID.
- Allocate memory and resources needed by the process.
- Initialize the PCB (process Control Block).
- Set the process State to New, then to Ready.
- Add the process to the Ready Queue.
- update Scheduling data Structure so the Scheduler can run it.

process Creation happens via System calls like fork() in unix/linux.

* ~~Operation~~

Main process (or parent process)

fork()

parent process . Child process

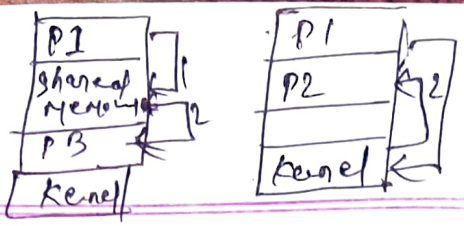
* operation on process Termination
when a process Terminated:

- The process state is changed to Terminated (or Exit).
- Release resources held by the process (memory, files, I/O devices).
- Remove the process from scheduling queues.
- Notify parent process if necessary (eg: Send exit status).
- Delete PCB and clean up data structures.

* Inter process communication (IPC)
- IPC allows processors to exchange data and synchronize with each other.
- Needed when process work together or share data.

Common IPC methods

- pipes :- unidirectional communication channel b/w related processes.
- message queues :- process send and receive messages asynchronously.
- Share memory - processes share a memory segment for fast data exchange.
- Semaphores - used to control access.
- Sockets - communicate over n/w b/w processes of different system.



* Scheduling Types

1. Long-term:- (Job Scheduling)
- Control the admission of processes into the system.
- Decides which jobs/ processes are admitted from the pool of new jobs into the ready queue.
- Runs less frequently compared to other Schedulers.
- Controls the degree of multiprogramming (how many process are in memory at once.)
- Example:- In batch systems, it decides which batch job to load next.

2. Short-Term Scheduling (CPU-scheduling)
- Decides which process in the ready queue will get the CPU next.
- Runs very frequently.

Ex- Choosing b/w two ready process to run on CPU.

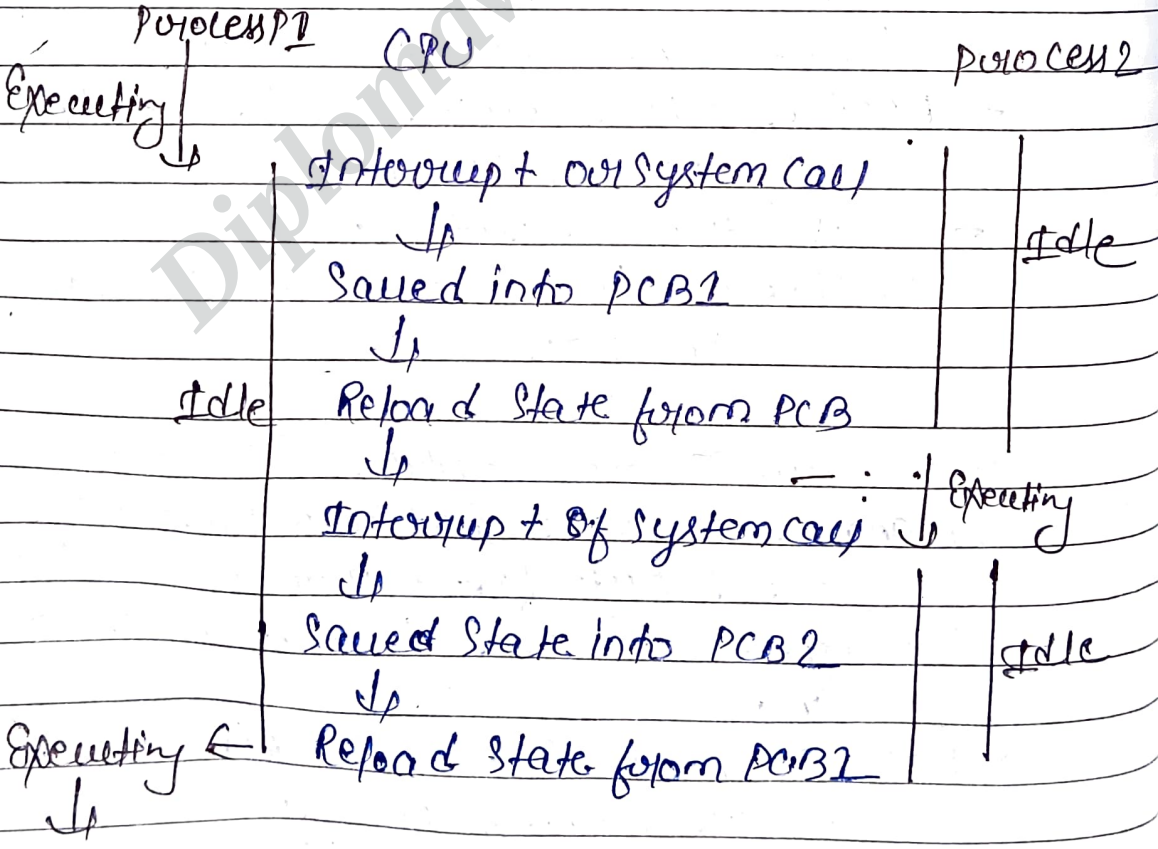
3. Medium Term Scheduling (Swapping Scheduler)
- manages process that are temporarily swapped out from memory at disk (swapped out) and brought back (swapped in).
- Helps improves CPU utilization and manages process memory usage.
- Mainly used in system that support swapping.

* Context Switch

- A context switch is the process where the CPU switches from executing one process to another.
- It involve saving the current state (context) of the running process and loading the saved state of the next process to run.

It includes:

- CPU registers (including program counter)
- process state
- memory management info (Sometimes).
- All the data needed to resume the process later exactly where it left off.



Types of Scheduling Algorithm.

1. FCFS (first come, first served)

For the following set of process scheduling using FCFS policy determine the average waiting time.
Assume that the process arrived in the order P1, P2, P3, P4.

process	Burst Time (ms)	WT	CT	TAT
P1	8	0	8	8
P2	15	8	23	23
P3	10	23	33	33
P4	7	33	40	40

Avg = 16

Step 1: Calculate waiting Time for each process
- Waiting Time (WT) of P1 = 0 (because start)
- WT of P2 = Burst time of P1 = 8
- WT of P3 = " " " P1 + Burst time of P2 = 8 + 15 = 23
- WT of P4 = Burst time of P1 + Burst time of P2 + Burst time of P3 = 8 + 15 + 10 = 33

Step 2: Average waiting Time

$$\frac{WT_{P1} + WT_{P2} + WT_{P3} + WT_{P4}}{4} = \frac{0+8+23+33}{4}$$

$$= \frac{64}{16} = \underline{16}$$

Completion Time Calculation

- CT of P1 = Arrival time + Burst time = 0+8 = 8
- CT of P2 = CT of P1 + Burst time of P2 = 8+15 = 23
- CT of P3 = CT of P2 + Burst time of P3 = 23+10 = 33
- CT of P4 = CT of P3 + Burst time of P4 = 33+7 = 40.

* Turn Around time.

Turnaround Time = Completion Time - Arrival Time.

Assuming all process arrive at time 0 (Since Arrival times were not given)

Using the completion time from before:

Turnaround time

P1 = 8-0 = 8
P2 = 23-0 = 23
P3 = 33-0 = 33
P4 = 40-0 = 40.

Sangam
unit-04 end.