MODULE 2

Process

A process is a program in exicution. It is an active entity, but a program is a 'passive' entity. Attribute held by a process include memory, CPU etc. A process in memory is defined as,

fig (1) process in memory.

i) text section :

It is the compiled program code, read into memory when the program is launched.

ii) data Section :

It is maid up of the global static variables allocated and initilized

iii) heap Section :

used for dynamic memory allocation and is managed through malloc, free etc.

iv) Stack Section

used for local variable storage purpose

Program

A program is a piece of code which may be, single line or millions of lines to accomplish a particular task. A computes program is usually written by a computes programmer in a programing language

eg: # include <stdio.h>

Main ()

{

Printf ("hello");

return 0;

}

A process can be devided into 2

i) users process

2) System process

A user process exicute user code and system process exicute System code.

Process Life Cycle.

When a process executes it passes through different state. The state of the process defines the current activity of that process. Each process can have one of the following 5 states. at a time.

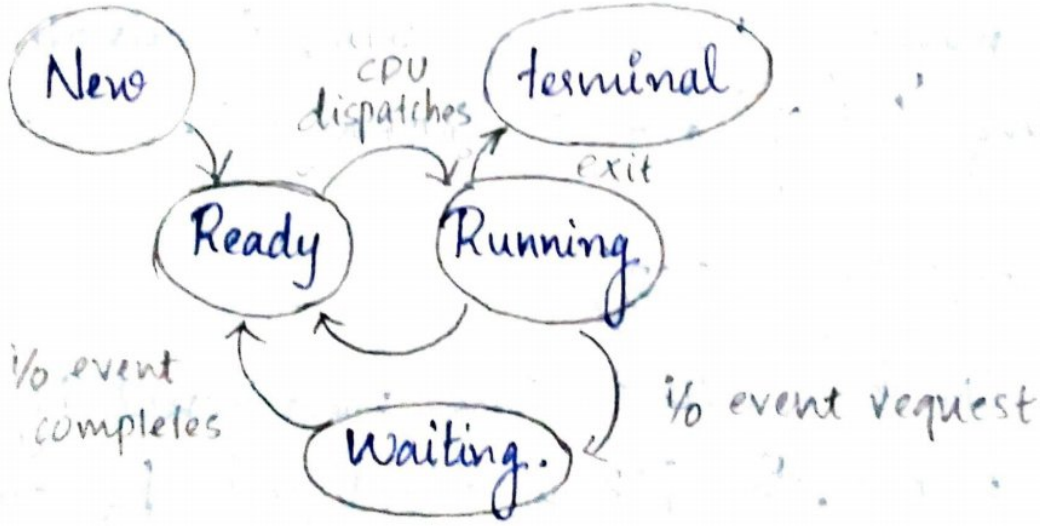
i) Newborn state / Start State

2) Ready

3) Running.

4) Waiting or Suspended State

5) Terminated / ~~Exited~~ state.
Exit



i) New State:

This is the initial state when a process is first started or created.

ii) Ready:

The process is ready for execution but wait for CPU time.

3) Running

One the process has been assign to a processor by the OS shedular, the process state is set to be running and the processor exicute its instructions

4) Waiting.

process moves to waiting state if it needs to prows wait for an i/p, o/p event to occur.

5) Terminated:

One the prows finishes its exiution, it will be terminated and it waits to be removed from main memory. We can forcefully terminate a prows by calling 'kill()' system call.

Process Control Block (PCB)

A PCB is a datastructure maintained by the OS. for every process. the PCB is identified by an integer value called (PID) ie Process id. A PCB keeps all information needed to keep track of a prows.

Process ID (PID)
-State
Pointer
Priority
Program Counter
CPU registor
I/O information
accounting informat".

* Process ID (PID):
  unique identifier for each of the process in the OS.
* Process state:
  the current state of the process
* pointer:
  a pointer is to a parent process.
* Priority:
  Priority value associated to the process
* Program counter (PC):
  A pointer to the address of next instruction to be executed.
* CPU registor:
  Various CPU registor where Process need to be stored for execution for running state.

* I/O "information":

includes a list of input output devices allocated to the process.

* accounting information:

Includes amount of CPU used for process exist. time limits, exist" id. etc.

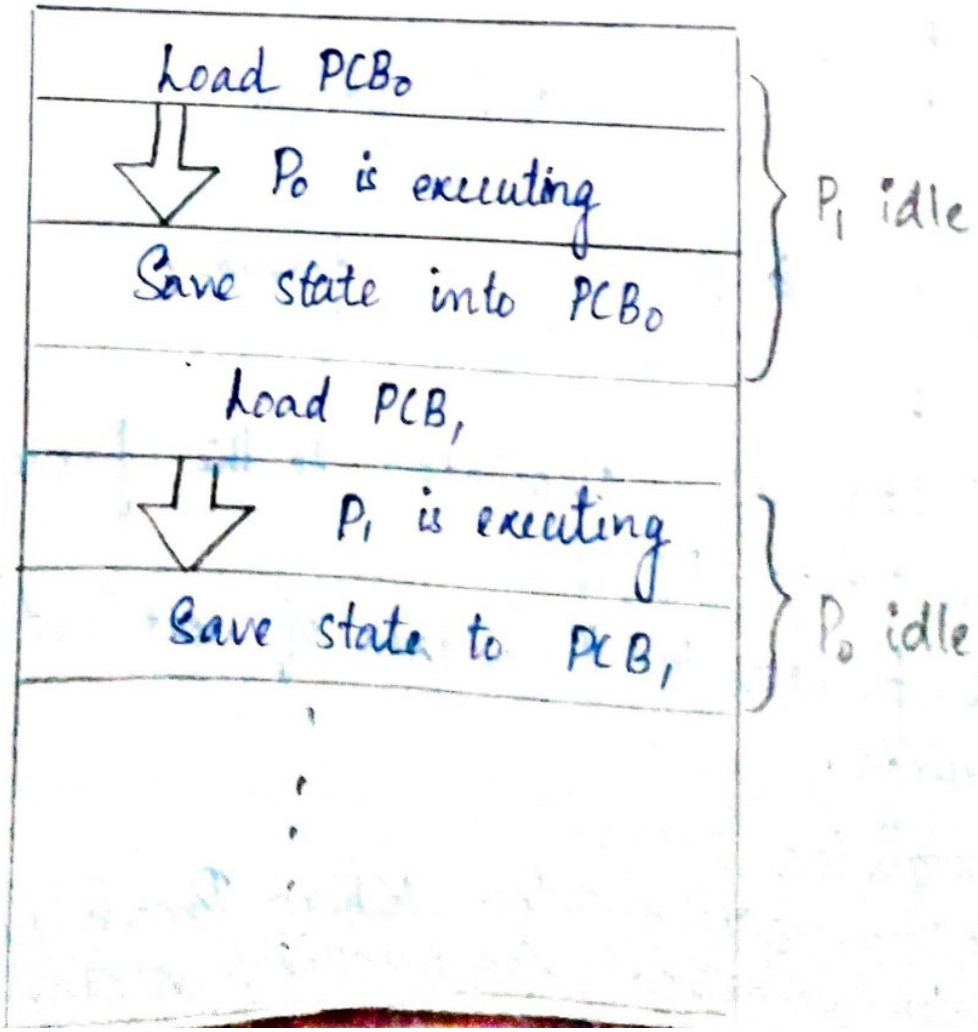
The PCB is maintain for a process throughout its life time and is deleted once the process terminates.

CPU Switches from process to Process.

Let P_0, P_1 are 2 process

Process P_0 PCB_0

Process P_0, PCB 1



TYPES OF PROCESS.

1) Parent Process.

It is an existing process from which a new process is being created.

2) Child Process.

The newly created process is called child process

3) ~~Daem~~ Dea

3) Daemon Process

It is process running in back ground

4) Orphan

If the parent process terminate before the completion of child process such process is called orphan process.

5). Zombie

If the child process terminates before the completion of Parent process, such process

? What is starvation

The process with small burst time is waiting for a long time to get executed by the CPU is known as starvation.

? What is aging?

Aging is the solution for the 'Starvation' That means gradually increasing the priority of a process.

Sheduling Queue.

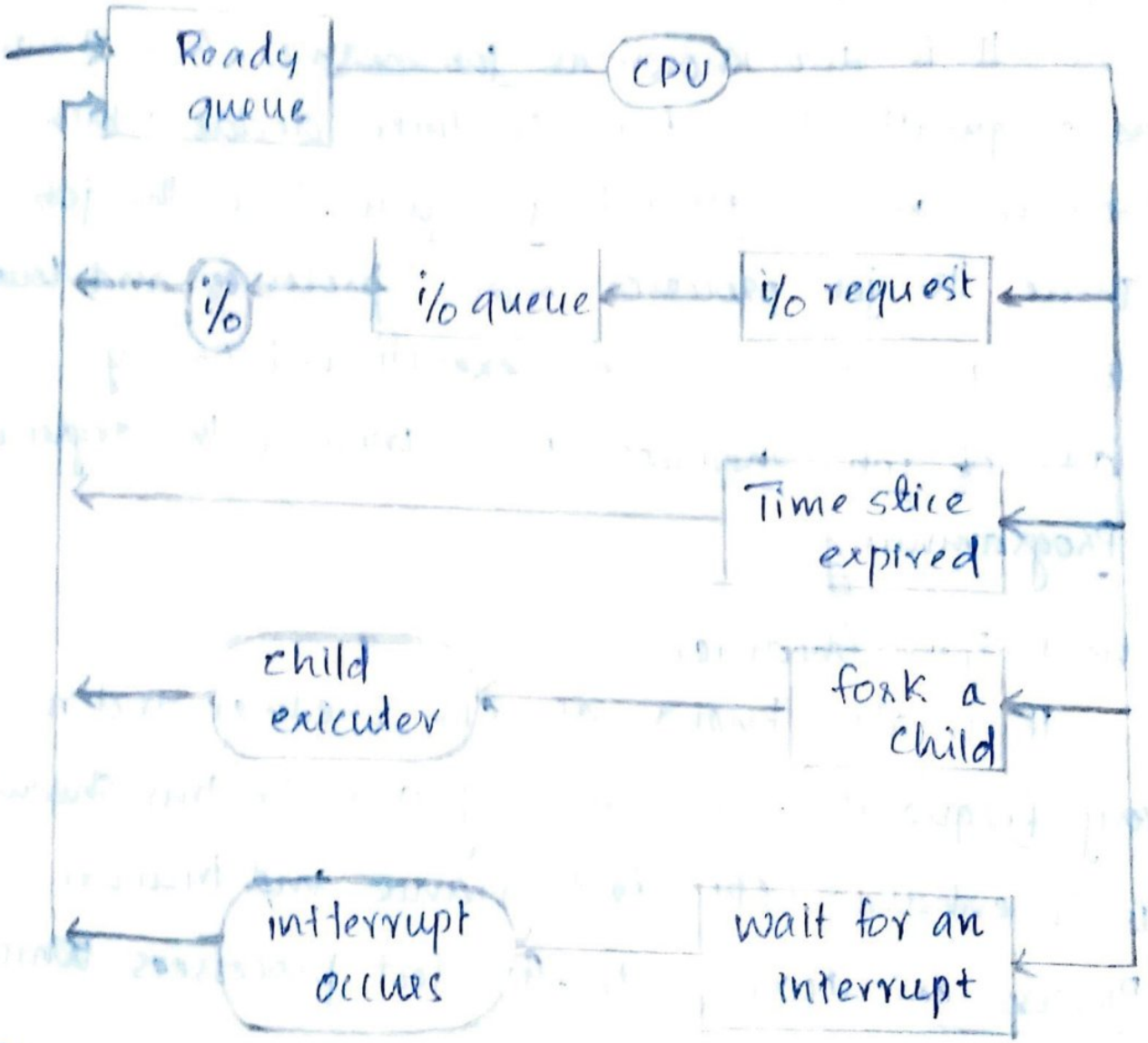
* All process, upon ext. entering into a system are stored in a "Job queue".
* Process in the ready state are placed in the "ready queue".
* Process waiting for a device to become available are placed in "device queue". there are unique device queue available for each I/o device.
* A new process is initially put in the 'Ready Queue' It waits in the ready queue until it is selected for execution by this CPU (dispatches) Once the process is assigned to the CPU and is executing one of the following several events can occur.

1) The process could issue an I/o request and then be Place in I/o queue

2) The process could create a new such process and waits for its termination

3) The process could be removed forcibly from the CPU as a result of an interrupt and be placed into ready queue.

In the first 2 cases, the process eventually switches from the waiting state to the ready state and is then placed into ready queue. A process continues this cycle until it terminates, at which time it is removed from all queue and has its PCB and resources deallocated.



⇒ Shedular

Shedular are special system software which handle process sheduling in various ways. Their main task is to select the job to be submitted into the system and to decide which process to run. Shedulars are of 3 types.

1) Long term shedular

2) Short term Shedular

3) Medium term Sheduler

1. Long term Sheduler.

It is also known as job Sheduler. Also it runs less frequently. Long term sheduler decide which program must get into job queue. From the job queue, the job processor selects processes and louds them into memory for execution. Primary aim of job Sheduler is to increase the degree of Programming.

2. Short term Sheduler

It is also known as CPU Sheduler and runs very frequently. The primary aim of this sheduler is to enhance CPU performance and increase Process exicution rate. It select processes which are ready to execute.

3. Medium term Sheduler.

This sheduler removes the process from memory by swapping. This process is swapped out and later swaped in by the medium term Sheduler. So it is a process swapping Sheduler.

Context Switch

It is a mechanism to store and restore the state of process in PCB [process control Block] so that process execution can be resumed from the same point at a later time. It is an essential part of multi tasking as.

CPU

P_1 execute P_1
→ Save PCB_1
← Restore PCB_2

P_2 execute P_2
→ Save PCB_2
← Restore PCB_3

P_3 execute P_3
→ Save PCB_3
← Restore PCB_4

P_1 execute P_1

When the sheduler switches the CPU from executing one process to execute another the state from current running process is stored into PCB. After this the state of the next process is loaded from its

own PCB and used to set PC register etc. At that point, the second process can start executing.

"Switching the CPU to another process requires saving the state of old process and loading the saved state. for the new Process. This task is known as content switch".

4. ⇒ CPU Sheduling ~~and~~ or process Sheduling.

"The act of determining which process is in the ready state ~~of old process~~ and which should be moved to running state is known as process Sheduling"

The prime aim of process sheduling system is to keep the CPU busy all the time and to deliver minimum response time for all programs.

There are 2 types of Sheduling.

1) Pre-emptive
2) Non-Preemptive

1) Pre-emptive Sheduling:

In this sheduling server can switches to Processing a new request before completing the currently running process.

2) Non-Pre-emptive:

In this scheduling once the CPU has been allocated to a process, the process keeps the CPU until the termination of the process. or entering into the waiting stage.

⇒ Dispatcher:

The dispacher is the module that gives control of the CPU to the process selected by the short term scheduler. It involves

1) Switching context
2) Switching to user mode
3) Jumping to proper location in the user program to rest out that program from where it left last time.

The despatcher should be as fast as possible be cause it is invoked during every process switch.

" The time taken by dispatcher to stop one Process and start another process is known as dispatch latency"

CPU sheduling is take place during the following situation

1) Process swithes from running state to waiting state
2) Process switches from running to redy state.
3) Process switches from waiting to ready state
4) When a process terminates.

Scheduling Criteria.

There are many different criteria to check when selecting a best scheduling algorithm.

They are

i) CPU utilization

We want to keep the CPU as busy as possible CPU utilization may range from 0% to 100%.

2) Through Put

It is the total no. of process completed per unit time this may range from 1.10 process per second to 1 process per hour depending on the proces

3) Turn around Time (TAT)

It is the interval from the time of submission of the process to the time of completion of the process ie, It is the amount of time taken to execute a particular process.

4) Waiting Time (WT)

The Sum of period Spent waiting in the ready queue.

5) Load Average

It is the avg. number of process residing in the ready queue waiting for their turn to get into the CPU

6) Response time :

It is the interval from the submission of a request to the first response.

maximize ⇒ CPU utilization, throughput.

minimize ⇒ Turn around time, waiting time, response time.

7) Arrival Time

The time at which the process become ready to run.

8) Burst Time (BT) or Service Time (ST)

The time spend for execution in CPU ie it is the processing time

9) Completion Time (CT)

The time at which the process finishes its execution in CPU ie it is the finished time.

$$TAT = CT - AT$$

$$WT = TAT - BT$$

$$\text{Avg } WT = \frac{\text{total } WT}{\text{Total no of Process}}$$

$$\text{Avg } TAT = \frac{\text{total } TAT}{\text{total no. of Process.}}$$

Sheduling algorithm:

A process sheduler sheduleres different proces to be assigned to the CPU based on perticular sheduling algorithm.

Different sheduling algorithms are:

1) FCFS (First Come First Serve)
2) SJFS (Shortest Job First Serve)
3) Priority Sheduling
4) Round Robin (RR) Sheduling

i) FCFS

This is the simplest CPU sheduling algorithm Here the process that request the CPU first is allocated first. The implementation of FCFS is managed with FIFO queue.

* Jobs are executed on First Come First Serve manner
* It is a pre-emptive, non-pre-emptive sheduling algorithm
* Easy to understand and implement
* Implementation is based on FIFO queue
* Poor in Performance as avg. wait time is high.

* Eg:

Process	Arrival time (AT)	Burst time (BT)
P_0	0	6
P_1	2	10
P_2	4	3
P_3	7	8

Process	AT	BT	CT	TAT = CT - AT	WT = TAT - BT
P_0	0	6	6	6	0
P_1	2	10	16	14	4
P_2	4	3	19	15	12
P_3	7	8	27	20	12

$$\text{Avg TAT} = \frac{\text{Total TAT}}{\text{Total no of process}} = \frac{55}{4} = 13.75$$

$$\text{Avg WT} = \frac{\text{Total WT}}{\text{Total no. of Proces}} = \frac{28}{4} = 7$$