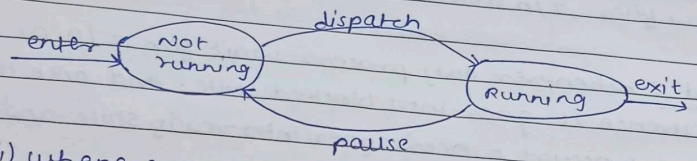
UNIT-3

Process Management. [12 - 16 M]

100) 1) Explain process state dgm. [4M]

→

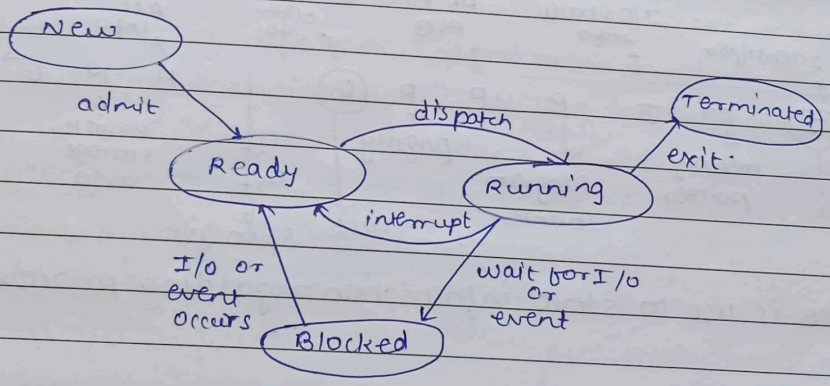
Two state process model



i) whenever any process is created, it goes into not running state.

ii) Once the process is dispatch it goes into Running state where various resources are allocated like CPU, memory, I/o devices files.

Five state process model



i) New: whenever any process is created it goes into new state and no resources are allocated.

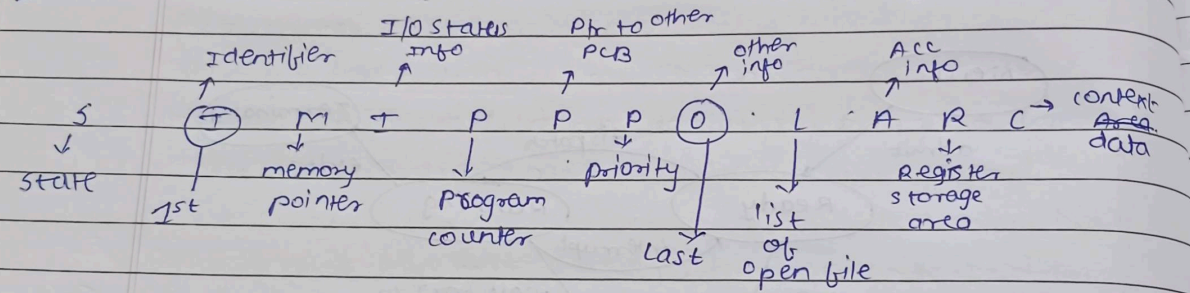
ii) Ready:- Once the process is ready for execution it goes into ready state and stored into ready queue. but still resources are not assigned.

iii) Running :- once the process is dispatch it goes into running state. whenever CPU becomes free Shortterm scheduler select the process from Ready queue. and assign it to the CPU and also other resource are allocated, CPU memory, files, I/o devices.

iv) Blocked :- whenever any process wait for I/o or any even occurence it goes into blocked state. And once the I/o or event occurs process goes into ready state and stored into ready queue.

v) Terminate :- once the process is sucessfully completed or terminated it goes into terminate state. Any resources are deallocated.

[OR] 2) Explain PCB (process control Block) [6m].



PCB is use to store information regarding process.

Identifier
State
memory pointer
I/o status info
Pgm counter
pointer to other PCB
priority
list of open files
Accounting info
Register storage area
context data
other info

dgm · PCB

1) Identifier :- Every process must have unique identifier abbreviated as Pid · If their are N number of processes then Pid becomes 1, 2, 3 · · · n. If Round robbin policy is followed than after n Pid becomes 1.

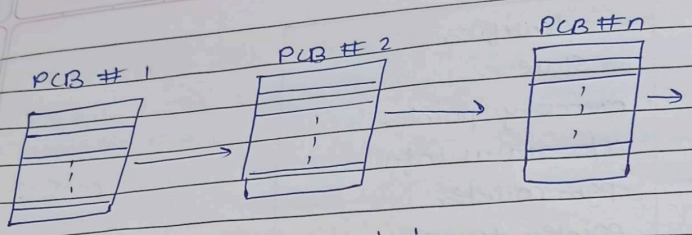
2) State :- This section stores the current state of the process like New, Ready, Running, Blocked, Terminated.

3) Memory pointer :- It stores the address of memory where process is stored.

4) I/o status information :- It stores the outstanding i.e pending i/o request.

5) pgm counter :- It stores the address of next instruction

6) Pointer to other PCB :- It stores the address of PCB of another process · if their are multiple PCB's connected than it called PCB chain. as shown below.



dgm. PCB chain.

7) priority :- This section stores the priority process which is helpful in priority scheduling

8) List of open files :- This section stores the filename which are required to execute a process.

9) Accounting information :- This section stores the statistics like processes start time, end time, number of registers required amount of memory etc.

10) Register storage area :- This section stores the register which are required for process like SP, IR, AC etc.

11) context data :- This section stores the data which is currently required by process.

12) other information :- This section stores the other information like userid and password. So that authorised user can execute a process.

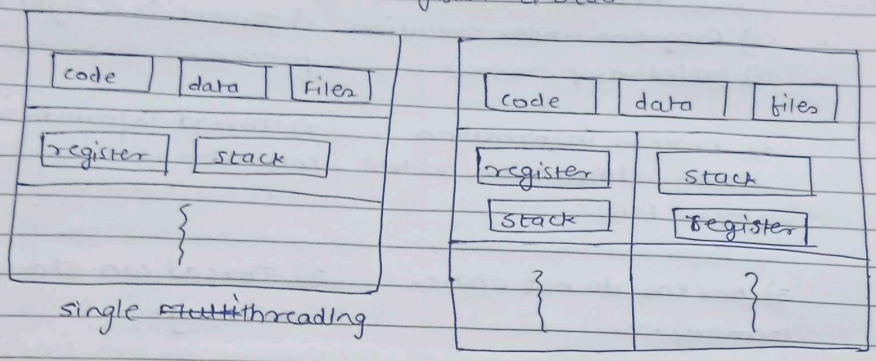
3) what is thread & multhreading ? what are it's advantage ?

→ i) A thread is a light weight process and basic unit of CPU utilization.

2) i) It is possible to create multiple threads in a single process.

ii) If only one thread is executing at a time is called as single threading and if multiple threads are executing in parallel it is called as multithreading

3) If their are multiple threads in a single process than all those threads can share code, data, files. But each thread having it's own register & stack.



single ~~Multi~~threading

Multithreading

* Advantages

R E R R U

Reliability | resource sharing | CPU utilization.

Economical

responsiveness

1) Reliability :- One process can contain multiple threads if any thread fails than it doesn't affect entire process.

2) Economical :- Thread creation and termination requires less time than process creation and termination Hence it is economical to create a thread instead of creating a new process.

3) resources sharing :- All the threads within single process can share the resources like CPU, memory, I/O, files, etc.

4) Responsiveness :- process gives response whenever it is completed but threads gives response at regular intervals

5) CPU utilization :- If multiple threads are executing in parallel than it ~~enter~~ increases CPU utilization.

4) Difference between Process & Thread

Process	Thread
1) Program under execution is called as process.	1) Thread is a light weight process.
2) Process information stored into process control block (PCB).	2) Thread information stored into Thread control block (TCB)
3) Process do not share memory.	3) Thread can shared memory.
4) memory is isolated	4) Threads are not isolated
5) Process creation ~~time~~ requires more time.	5) Thread creation requires less time.
6) Proces termination requires ~~more~~	6) Thread
7) Process gives the responce once it is completed.	7) Thread gives responce at regular interval
8) It is less economical to create a process.	8) It is more economical to create a process.
9) It require more time for context switching	9) It requires less time for context switching

10) P

Time
5)
→
100%

10) Process is less efficient in terms of communication.	10) Thread is more efficient in terms of communication
11) process is heavy weight	11) Thread is light weight.
12) If one process fails than it may block the execution of other processes.	12) If one thread fails than it doesn't affect on other threads.
13) If changes are made to parent process than it does affect child processes.	13) If changes are made to one thread than it may affect on other threads.
14) To call a process, system call is required.	14) To call a thread system call is not required.
15) process do not share data with other process	15) One thread can share data with other thread

VIMP

5) what is a process. How it is different from program.

→ 1) A process is a program under execution to which various resources are allocated.

100%

2) process is an instant of a program.

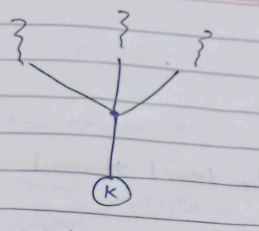
Program	Process.
1) It is a set of instructions.	1) It is a program under execution.
2) It is a passive entity.	2) It is an active entity
3) Program written by programer	3) Process executed by processor
4) Program resides on paper or hard disk	4) process resides into main memory.
5) Program doesn't required any resources expect harddisk	5) process requires various resource, memory, I/O, file, CPU.
6) program doesn't have any state.	7) process having various state like new, ready, running, terminated blocked.

VIMP
6) what is multithreading explain different multithreading models.
100%

→ 1) whenever multiple threads in single process are executing in parallel is called as multithreading.

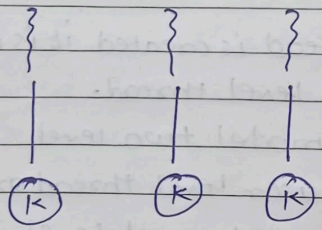
2) Multithreading models.

i) ~~One~~ Many - to - One



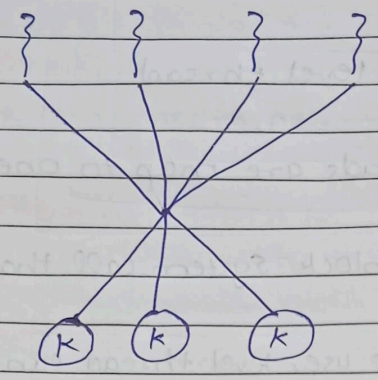
a) multiple user level threads are map to one kernel level thread.
b) If thread generate a block system call than entire process stop.
c) At any time only single user level thread can access kernel level thread. i.e true concurency cannot be achieve.
d) This model was use in Solaris O.S.

ii) one-to-one

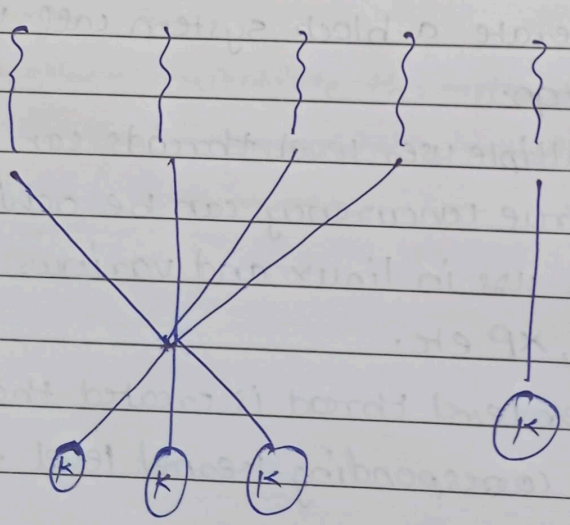


a) ~~multiple~~ one, user level threads are map to one kernel level thread.
b) If thread generate a block system call than entire process will not stop.
c) At any time multiple user level threads can access kernel level threads. i.e True concurency can be achieve.
d) This model was use in linux and various version of windows like ~~↑98~~ windows, xp etc.
e) If any new user level thread is created than it is compulsory to create corresponding kernel level thread.

iii) many - to - many



a) multiple user level threads are maped to less or equal number of kernel level threads.
b) If thread generate a block system call it will not stop entire process.
c) —11—
d) If any new user level thread is created it is not compulsory to create kernel level thread.
e) another variation of this model two level threading in which multiple user level thread are maped less or equal number of kernel level threads plus one user level thread is maped to one kernel level thread.

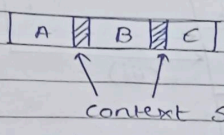


f) This model was use in IRIX, Irub4, Unix, HP, etc.

Q7 Explain context switch

→ 1) Transfering a control of CPU from one process to another process is called as context switch.

eg:-

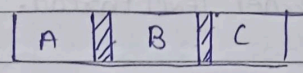


2) As shown is above diagram two context switch are present A to B & B to C.

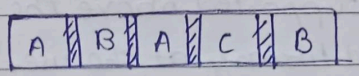
2) during context switch CPU does no useful work i.e CPU becomes idle.

3) whenever context switch occurs state of old process will be stored in PCB and state of new process will be loaded from PCB.

4) Number of context swithing is more than non-pre-emptive scheduling.

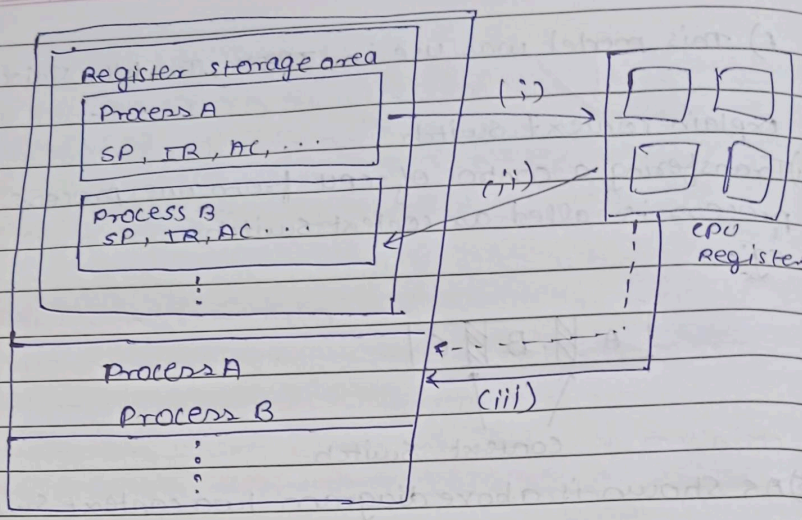


non-pre-emptive scheduling



pre-emptive scheduling.

5) working of context switch.



i) As shown in above diagram process a is currently executing is shown by ----->
ii) whenever process B arrives, state of process A will be stored (shown by (i)) and state of process B will be loaded (shown by (ii)).
iii) Once the state of process B is loaded, CPU is assign to process B (shown by (iii)).

[2m] Q8 Explain user level & kernel level thread.

→ user level thread:-
1) These threads are supported above the kernel and implemented by thread library at the user level.
2) A Thread library provides the support for thread creation scheduling and management without any support from kernel.
3) Those thread is generally fast to create and manage.

Kernel level thread

i) These threads are supported directly by OS kernel.

ii) The kernel perform thread creation scheduling and management.

iii) These thread is general slow to create and manage.

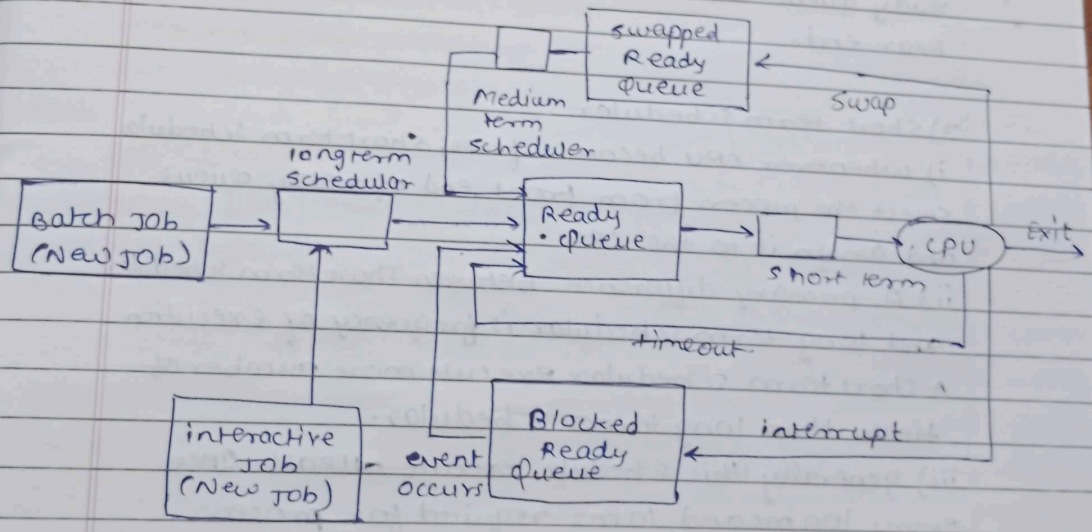
Q9 what are the difference types of schedulers?

→ Three types:-

i) long term scheduler

ii) mid term scheduler

iii) short term scheduler.



1) long term scheduler

i) Their is limitation on the number of process to be stored in a ready queue

ii) This scheduler select good mixture of jobs from two sources. i.e Batch Job and interactive job.

iii) generally their are two types of process i.e CPU bound and I/O bound. CPU bound processes perform computation and I/O bound processes perform I/O related activities. Hence it is responsibility of long term scheduler to select proper processes.

2) mid term scheduler
i) processes which are ready but swapped are stored into swapped ready queue
ii) whenever memory becomes free medium term scheduler select the process from swapped ready queue and stored into Ready queue. through Rear end.

3) Short term scheduler
i) whenever CPU becomes free, short term scheduler select the process from front end of ready queue and assign it to the CPU.
ii) A primary difference between short term scheduler and long term scheduler is frequency of execution A short term scheduler execute more number of times than long term scheduler.
iii) generally this scheduler execute atleast once every 100 ms and 10 ms required for process selection, therefore CPU is idle for 9%.

$$\frac{10}{100+10} = 0.09.$$

$$= 9\%.$$