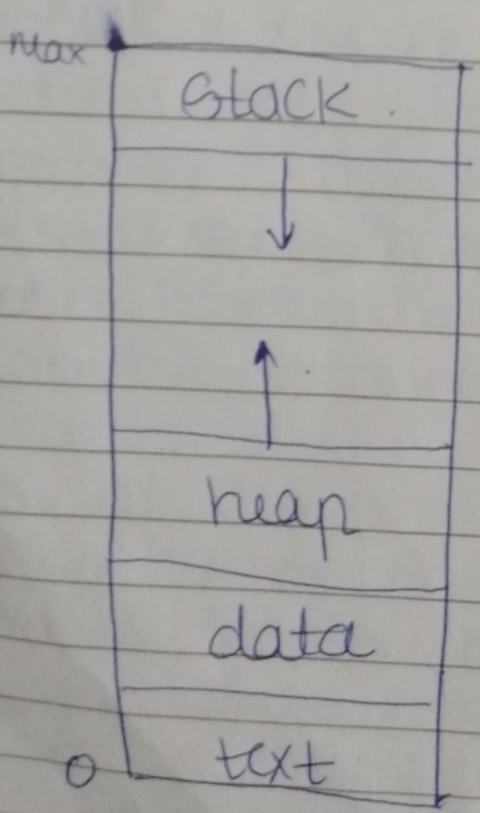
Q. Define process & draw the diagram of process in memory

Ans ① Process - a program in execution, process execution must progess in sequential fashion.

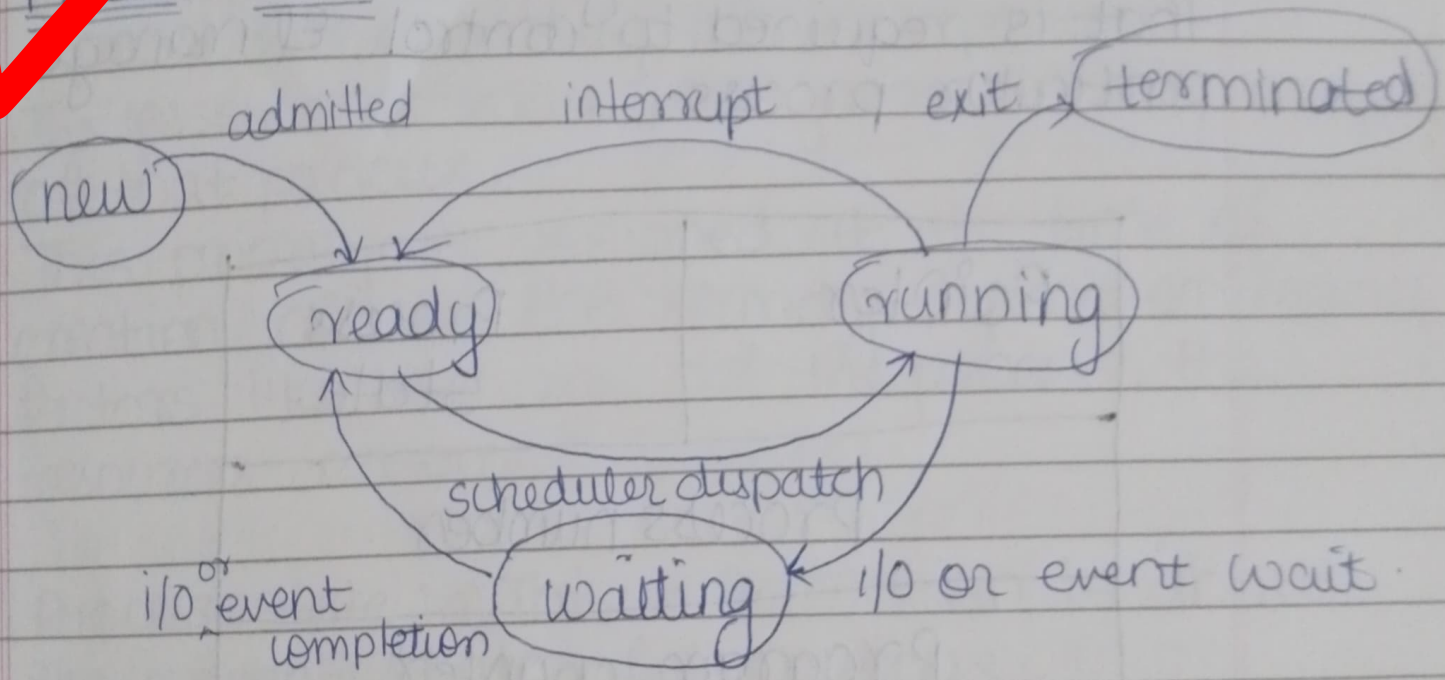
② A process is a program in execution. Process is also called as Job, task & unit of work.

③ A process is defined as "an entity which represents the basic unit of work to be implemented in the system".

④ A process includes:
- program counter
- stack
- data section



→ Process state.



① As a process executes, it changes state.
- new: The process is being created.
- ready: The process is waiting to be assigned to a process.
- running: Instructions are being executed.
- waiting: The process is waiting for some event to occur.
- terminated: The process has finished execution.

3

→ Syntax for sleep, kill, wait, ps.

① sleep:
sleep NUMBER[SUFFIX] ....
sleep option OPTION.

② kill
kill pid.

③ wait
wait.

④ PS
$ ps [option]

4

→ ps command

① It is used to display the characteristics of a process. This command when executed without options, it lists the processes associated with a user at a particular terminal.

② Syntax: $ ps [options]

③ Example: $ps

④ Each line in the output shows PID, the terminal with which the process is associated, the cumulative processor time that has been consumed since the process has been started & the process name.

⑤ Options:

a) -f :- It is used to display full listing of attributes of a process. It includes UID (user ID), PID (Parent ID), C (amount of CPU time consumed by the process) & STIME (chronological time that has elapsed since the process started)
Example: $ p -f

b) -u : shows the activities of any specified user at any time.
Example: $ ps -u abc.

c) -a : It shows the processes of all the users.
Example: $ ps -a.

d) -e : It displays processes including users &

system processes.
example: $ ps -e.

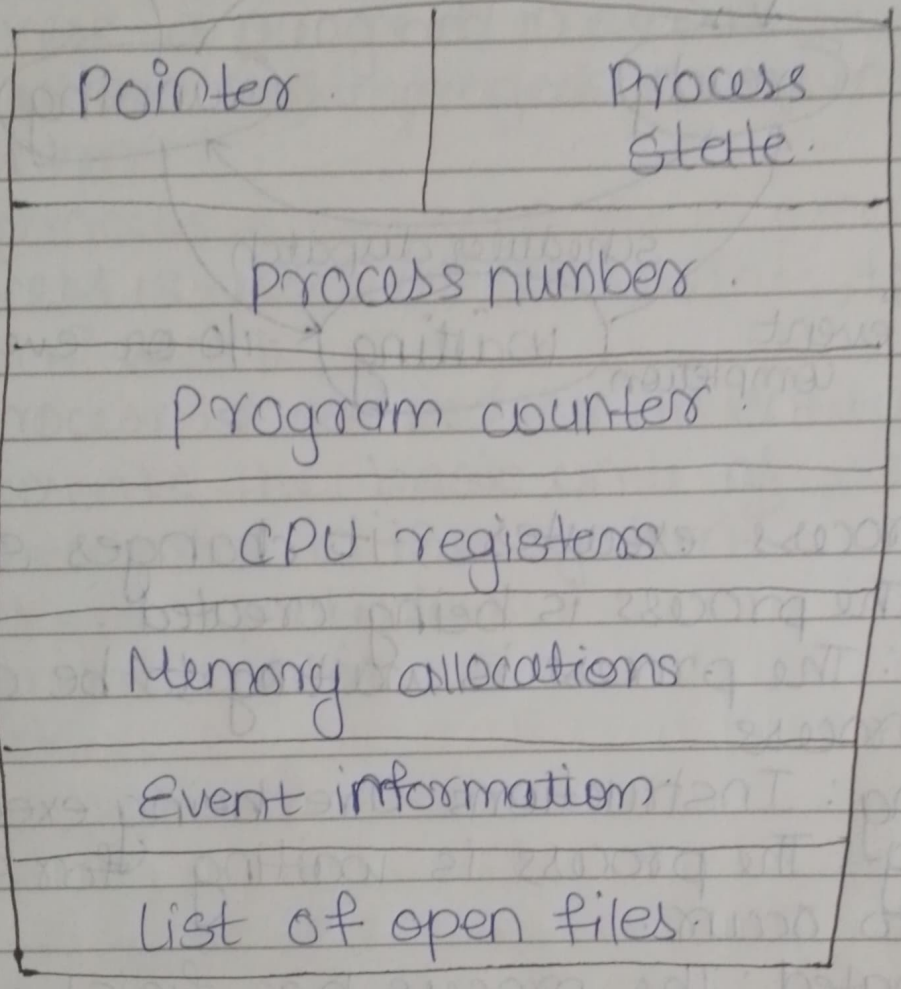
→ PCB (Process control Block)

① Each process is represented in the operating system by a Process control Block (PCB) is also called as Task control Block.

② When a process is created, operating system creates a corresponding PCB & released whenever the process terminates.

③ A PCB stores descriptive information pertaining to a process, such as its state, program counter, memory management information,

information about its scheduling, allocated resources, accounting information, etc. that is required to control & manage a particular process.



6

→ Process Creation.

① When a new process is to be added to those currently being managed, the OS builds the data structures that are used to manage the process & allocates the address space in main memory to the process.

PAGE No.
DATE / /

② This is creation of a new process.
③ Parent process create children processes, which inturn create other processes, forming a tree of processes.

→ Process Termination:

① According to the condition, process may be terminated normally or forcibly by some other process.
② When a process completes its task ef invokes a system call Exitprocess() in windows ef exit() in UNIX which tells the O/S that it is finished is called normal termination.
③ A process causing abnormal termination of some another process. For this the process invokes system call TerminateProcess() in windows ef kill() in UNIX telling the s/w to kill another process.

Q7(i) add delay in the sleep.
→ sleep 10

(ii) To terminate a process.
→ kill 0

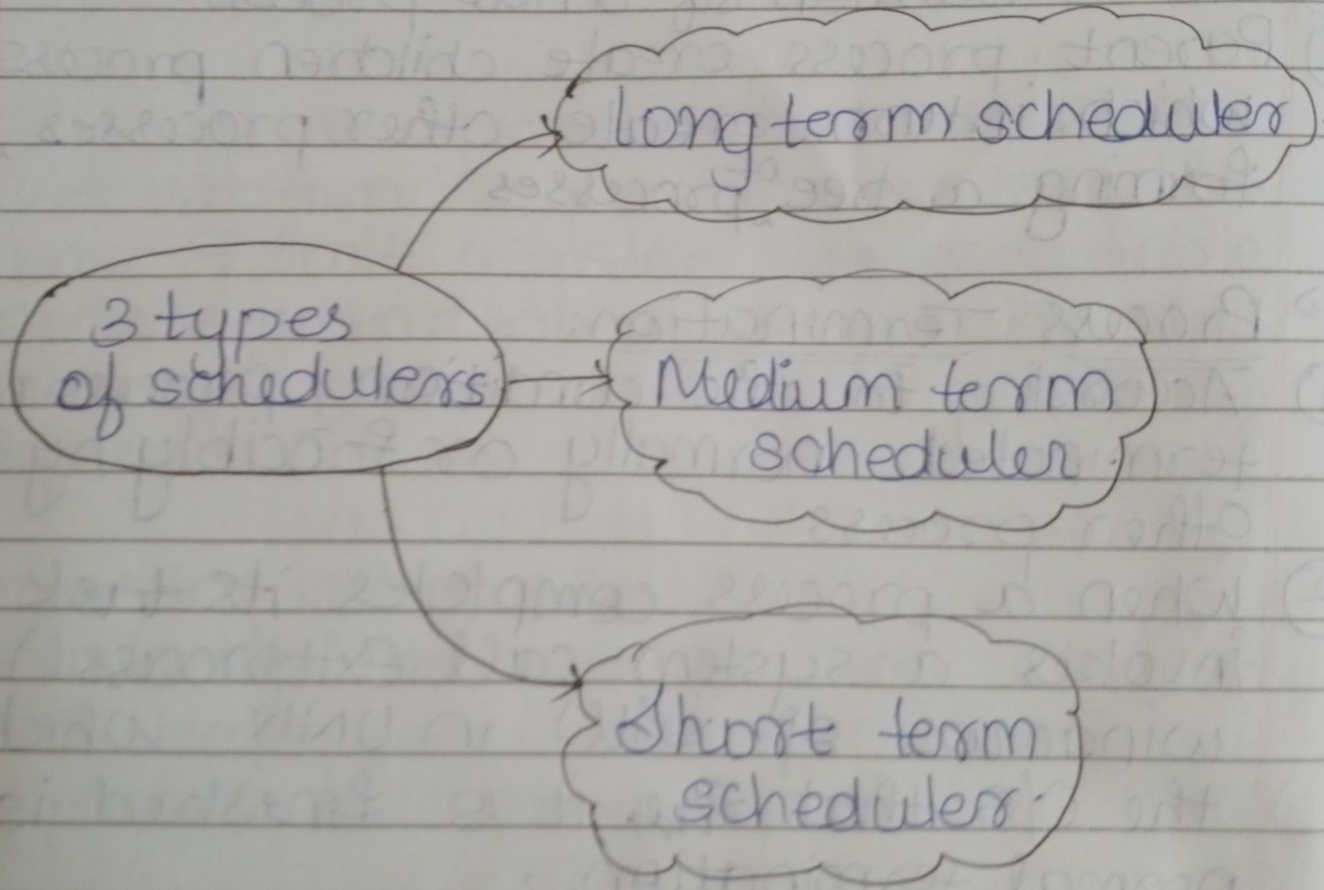
in windows. Eg kill() in UNIX telling the s/w to kill another process.

→ Schedulers.

① A scheduler in an operating system is responsible for managing the execution of processes. It determines which process gets the CPU next based on priority or other scheduling algorithms.

② Schedulers are of 3 types.

4.1


3 types of schedulers
long term scheduler
Medium term scheduler
Short term scheduler

→ context switch

① When CPU switches to another process, the system must save the state of the old process & load the saved state for the new process. This task is is called context switch.

② CPU switching from one process to another process is called a context switch.

8.5

PAGE No.
DATE / /

→ IPC (Inter-Process Communication)

① IPC is a mechanism that allows different process to communicate with each other in an operating system.

② It helps in sharing data, synchronizing activities, & coordinating resources between processes.

③ It's like a language that processes use to talk to eachother.

④ Models of inter-process communication are:

(i) Shared memory model.

(ii) Message passing model.

q.1

→ Shared memory model-

(1) Two processes exchange data or information through sharing region. They can read & write data from & to this region.

(2) A region of the memory residing in an address space creating a memory segment can be accessed by all processes who want to communicate.

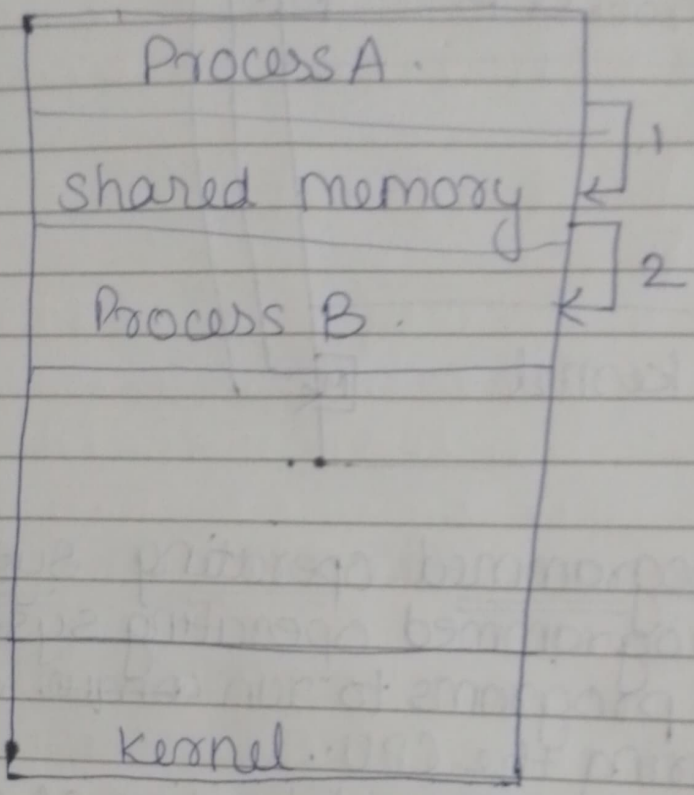
(3) All process using shared memory segment should attach to the address space of shared memory

(4) These processes are not under the control of o/s.

(5) These processes are also responsible for ensuring that they are not writing to the same location simultaneously.

(6) After establishing shared memory

segment, all access to shared memory system segment are treated as routine memory access & without assistence of kernel.



→ Message passing model

9.2

① In message passing model the data is exchanged in the form of messages.
② It allows the processes to communicate & synchronize their action without sharing the same address space.
③ Communication requires sending & recieving message through kernel.
④ The processes that want to communicate with each other must have a link between them.