Name:
Date: / /

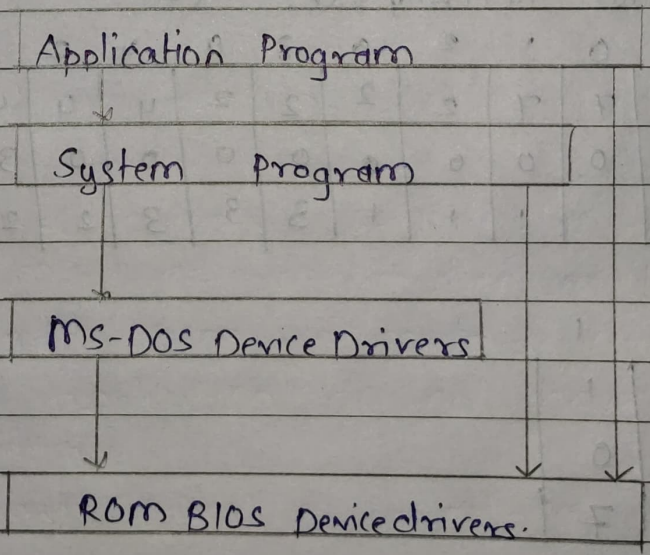
Name: Prasad Kashid.
Roll No: A75.
ID: TU4S2122010
@Pkashid

OS
Assignment 1.

Q1 Explain Operating System structure.

→ Operating System:
An Operating system (OS) is a set of programs that manage computer hardware resources and provides common services for application software.
It allows the user to interact with the system hardware.

a) Simple structure:
1) There are many operating system that have a rather simple structure. These started as small systems and rapidly expanded much further than their scope.
2) A common example of this is MS-DOS. It was designed simply for a niche amount for people.

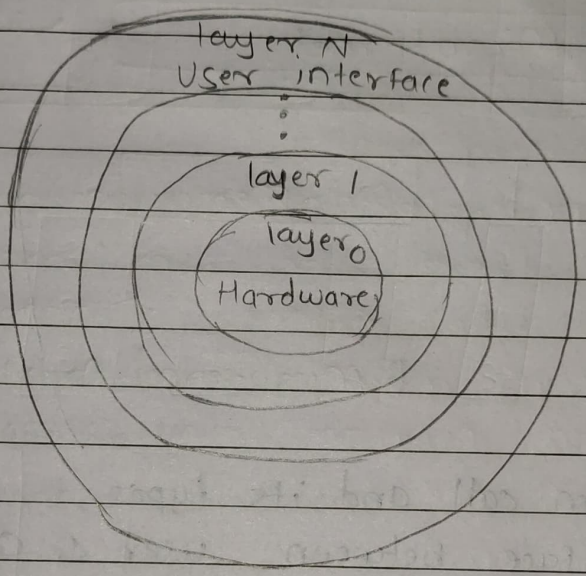


MS-DOS - structure.

Name :
Date : / /

b) layered structure:

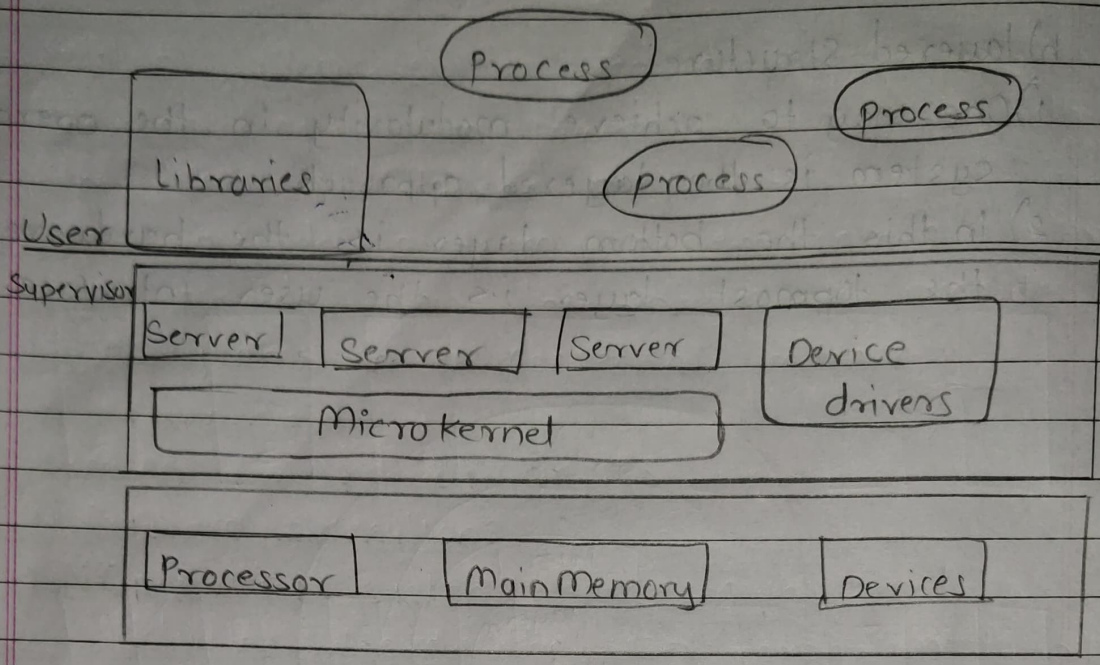
1) One way to achieve modularity in the operating system is the layered approach.
2) In this, the bottom layer is the hardware and the topmost layer is the user interface.



c) Microkernels:-

1) This structures the OS by removing all nonessential portions of the kernel and implementing them as system and user level program.
2) Generally they provide minimal process and memory management and a comm^n facility.
3) Communication between components of OS is provided by message passing

Name :
Date : / /



A Microkernel Architecture.

Q2 Explain System call and its types.

→ 1) The interface between user & OS is defined by the system calls.

2) System call is call for operating system to perform some tasks on behalf of user program.

3) The system call are functions used in the kernel itself.

4) Due to system call, code is executed in kernel and so there must be some mechanism to change the process mode from user mode to kernel mode.

5) In unix, User application do not have direct access to hardware

6) Application first request to kernel to get hardware access and access to computer resources.

7) During execution when application invokes a system

call it is interrupted process and the system switches to Kernel

8) Kernel saves process execution of interrupted process & determine purpose of call.

Types of System calls.

- Process Control
- Device Manipulation
- Communication
- File Manipulation
- Information Maintenance

1) Process Controls:-
End, abort, load, Execute, Create process, terminate process.

2) Device Manipulation:-
Request device, release device, read, write.

3) Comm^n:-
Create comm^n, delete comm^n, Send, receive.

4) File manipulation:-
Create File, delete File, open, close, read.

5) Information maintenance:-
get time, set time, get date, set date.

Name :
Date : / /

Q3 Explain System services and interfaces.

→ The operating system provides services to both the users and to the program.

1) It provides program an environment to execute
2) It provides users the services to execute the programs in a convenient manner.

Services Provided by an OS:-

1) Program Execution:

Operating system handle many kinds of activites from the user program to system program like printers, name servers, file server etc.

- loads a program into memory
- Executes the program.
- Handles program's execution.
- Provides a mechanism for process synchronization

2) I/O Operations:

An I/O subsystem comprises of I/O devices and their corresponding driver software.

- I/O operation means read or write operation with any file or any specific I/O device.
- Operating system provides the access to the required I/O device when required.

3) File System manipulation:

A file represent collection of related information. Computer can store files on the disk, for long-term storage purpose.

Name :
Date : / /

- Program needs to read a file or write a file.
- The OS gives the permission to the program for operation on files.
- Permission varies from read-only, read-write denied and so on.

4) Communication:-

Mutiple processes communicate with one another through communication lines in the network.

- Two process often requires data to be transferred between then.
- Both the process can be on one computer and on different computers, but connected through computer network.

5) Error handling:-

Error can occur anytime and anywhere. An error may occur in CPU, in i/o devices or in the memory hardware.

- The OS constantly checks for possible errors.
- The OS takes an appropriate action to ensure correct and consistent computing.

Name :
Date : / /

Q4. Process State model and transition.

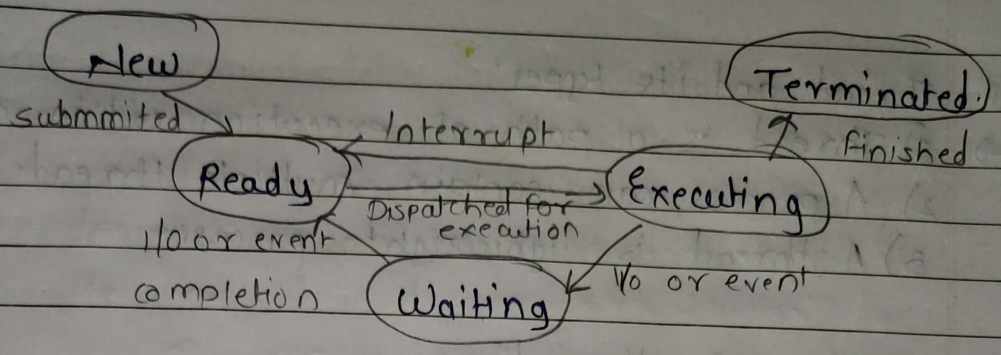
→ 1) A process is a program under execution that consists of a number of a elements including. program code and a set of data.

States :-

1) Running
2) Ready
3) Block / Waiting.
4) New
5) Exit / Terminate.

2) A process is executed in main memory.
3) To complete execution, program needs many resources and it complete for it.
4) From computation context point of view, a process is defined by CPU state, memory contents.
5) A CPU state is defined by the content of various register such as Instruction register, Program counter, Stack pointer (SP).
6) A small amount of data is stored in CPU register
7) Heap is reserved memory area for dynamically allocation of memory to the program at run time.

Name :
Date : / /



1) New :-
When a process is created it is in new state.

2) Ready :-
In Ready state, process is ready to run but waiting for CPU.

3) Execution :-
It is said to be running, if CPU allocated & executing.

4) Waiting :-
It can't continue execution and waiting for event to happen.

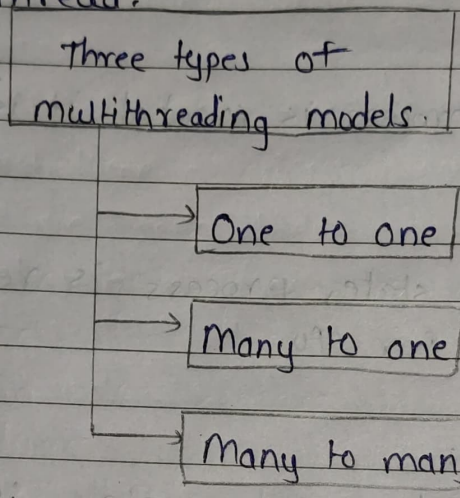
5) Terminated :-
After execution of process it is terminated.

Name :
Date : / /

Q5. Thread and its types:

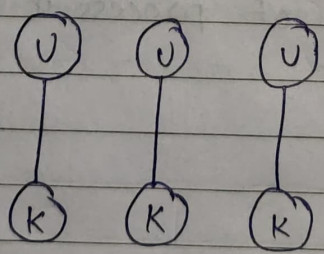
→ 1) Thread is a path of execution within a process.
2) A process can contain multiple thread.
3) A thread is lightweight process.

Types of thread:



a) One to One model:-

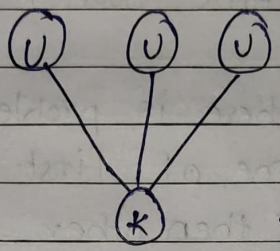
1) In this model relationship between user level thread and kernel level thread is one to one.
2) It contains single user-level thread to a single kernel-level thread.
3) It is needed to create kernel thread for every new creation of user thread.



Name :
Date : / /

b) Many to one model :-

1) In this model relationship between user level thread & kernel level thread is many to one.
2) It contains many user-level threads to a single kernel level thread.

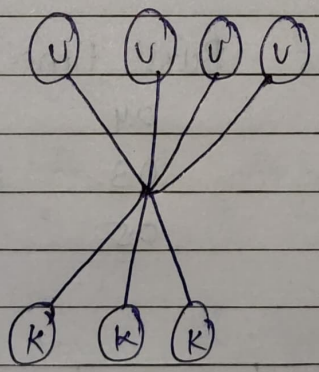


← Multiple User threads

← Single kernel threads.

c) Many to Many model :-

- Many to many association exist between user level thread and kernel level thread. It means than more the number of user level thread are allied to equal or less number of kernel level thread.
- The necessity of altering code in both kernel and user spaces leads to a level of complexity present in one to one and many to ~~one~~ one model.



User level thread

Kernel level thread.

Name :
Date : / /

Q6. Scheduling algorithms:-

→ a) FCFS

1) First Come First serve scheduling algorithm schedules job according to arrival.
2) The job that arrived first will get CPU first.
3) Lesser arrival time, sooner job will get CPU for execution.
4) But in FCFS there is problem of starvation. If the burst time of first process is longest among all job then other jobs have to wait unHill first job is done.

Advantages:-

1) Simple.
2) Easy.

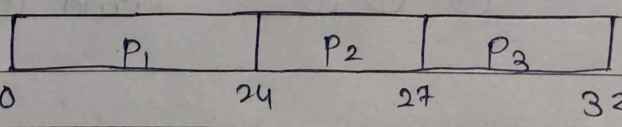
Disadvantages:-

1) Starvation.
2) Higher avg waiting time.

Example:-

Process	Burst time.
P_1	24
P_2	03
P_3	05

Gantt chart:-



Avg Turnaround time = (3+8+32)/3 = 14.33
Avg Waiting time = (0+3+8)/3 = 3.66.

Name :
Date : / /

b) Round Robin :

1) It is a most popular algorithm of OS.
2) It focus on time sharing.
3) In RR, every process executed in cyclic way.
4) Time slice is defined as time quantum.
5) Each process is in ready queue is assigned the CPU for that time quantum.

c)

Advantages :-

1) No starvation.
2) Fair allocation of CPU.

Disadvantages :-

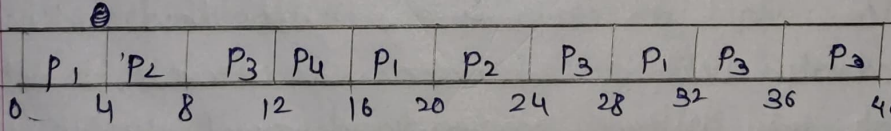
Deciding a perfect time quantum is hard.

Ex :-

Process	Burst time.
P1	11
P2	7
P3	15
P4	4

Time quantum = 4

d)



Avg Turnaround-time = $\frac{30 + 23 + 37 + 16}{4}$ = 26.5

Avg Waiting-time = $\frac{19 + 4 + 22 + 12}{4}$ = 14.25

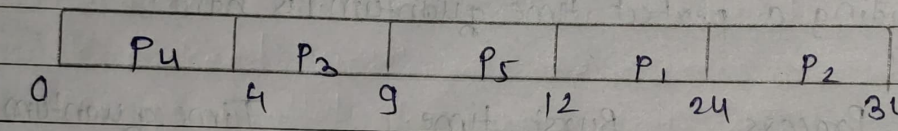
Name :
Date : / /

1) Priority Scheduling:-

1) In priority scheduling, each process has a priority which is integer value assigned to it.
2) lower the number higher the priority is.
3) Higher number has low priority.

Ex:-

Process	Burst Time	Priority
P_1	12	4
P_2	10	5
P_3	5	2
P_4	4	1
P_5	3	3



Avg Turnaround time = $\frac{24+34+9+4+12}{5}$

= 16.6.

Avg waiting time = $\frac{12+24+4+0+9}{5}$ = 9.8.