PAGE No.
DATE / /

* Operating System * [Module 1].

- Computer System
- OS manages hardware resources of a computer system.
- Manage all the task in execution with respect to computer system.
- Manages hardware resources and has eye on execution unit
- Manages Jobs, Tasks, processes in computer system.
- Operating System is a Software. It is an interface between Hardware and user.
- Operating system is a resource manager
- It can take input from user and process that input and give output to the user.
- OS acts as an interface between Hardware and User.
- manages, processes and converts input to desire output.

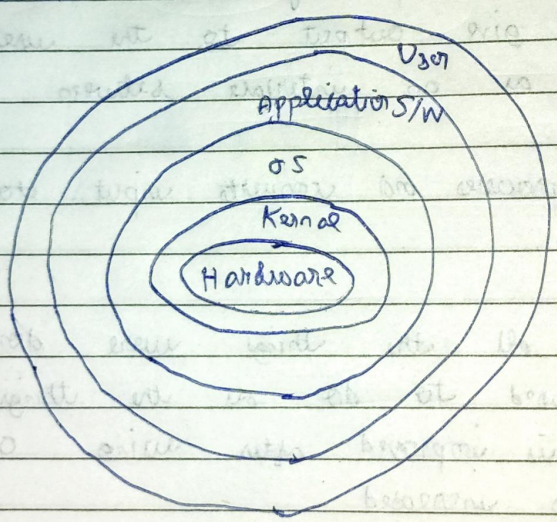
- Manually all the things were done before OS.
- User used to do all the things.
- Efficiency is improved after using OS.
- Speed gets increased
- System is now multi-tasked.

- OS is a Software.
- Software is Hardware dependent.
- Software gets installed in Hardware.
- Need of physical memory.
- When a program Runs into ram to produces intermidiate results which is stored in register.

- Cache is a high speed memory than a RAM.

* What is OS?
- Hardware Manager: It manages all the hardware resource or components of computer.
- Process Manager: It supervise all the task which is being executed by processor.

* How OS is different from other application software?
- OS is installed over hard-drive.
- Applications are also installed over OS. Application are runned inside OS.



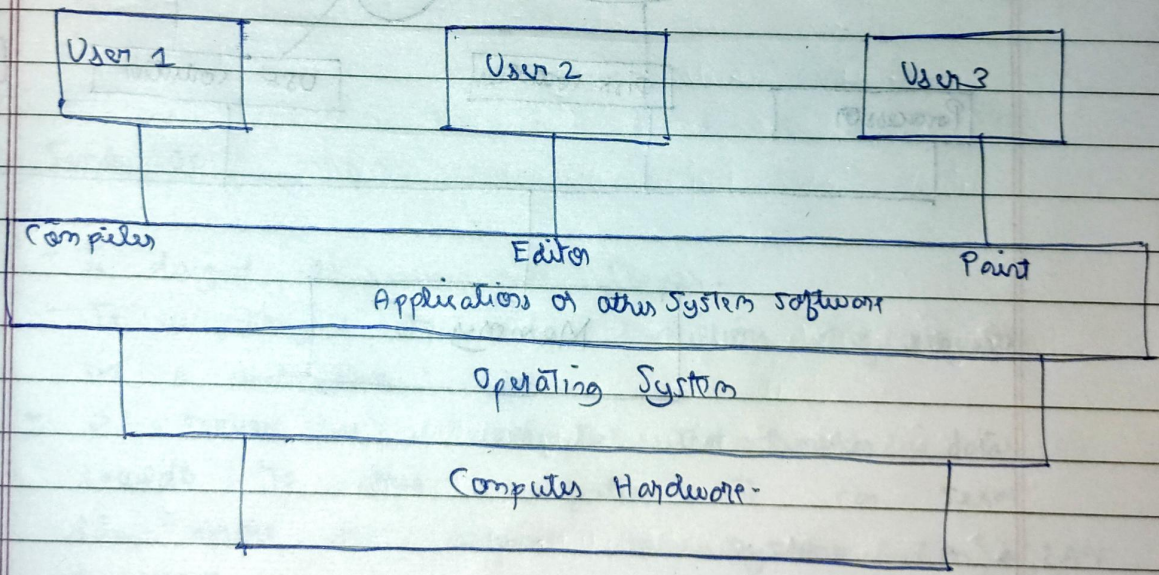
* Operating System layered Architecture.

OS runs over computer, Application run over OS.

PAGE No.
DATE / /

* Kernel in OS

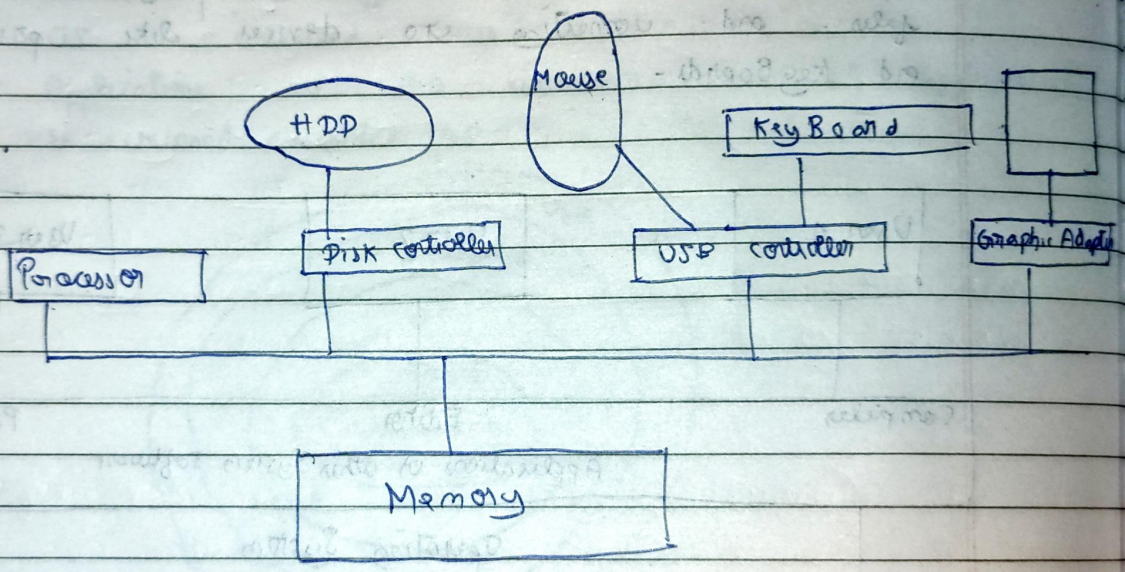
- A Kernel is the core part of an Operating System.
- It acts as a Bridge between software applications and the hardware of a computer.
- Kernel manages system resources, such as the CPU, memory and devices, ensuring everything works together smoothly and efficiently.
- It Handles tasks like running programs, accessing files and connecting to devices like printers and KeyBoards.



PAGE No.
DATE / /

* Basic computer organization required for OS:

- Examples of well-known OS
- Mobile OS : Android, iOS, Windows
- Embedded Systems
- Real time OS : HRT & SRT
- Desktop OS : Personal computer
- Server machine OS.



PAGE No.
DATE / /

* Booting is of two types:
  1) Cold Booting: When the computer is started after been switched off.
  2) Warm Booting: When the operating system alone is restarted after system crash or freeze.

* Real-Time Operating System (RTOS)

- A real-time operating system (RTOS) is a special kind of Operating System designed to handle tasks that need to be completed quickly and on time.
- Two types of RTOS
  1) HRT
  2) SRT

* Server OS

- It designed to run over Server.
- It is designed to handle millions of requests in a single time.
- A server is a computer that makes data available to other computers. It can serve data across the internet to systems on a LAN or WAN.
- Examples of Server OS are:
  1) Apache HTTP Server
  2) Microsoft IIS

PAGE No.
DATE / /

* Functions of OS

1) Process Management (Process scheduling Algorithms)
- It controls how processes are carried out and controls how your computer runs by handling the active processes.
- This includes stopping processes, setting which processes should get more attention and many more.
- Responsible for managing the Start, Stop and scheduling of processes which are programs running on System.

2) Memory Management

3) Device Management. (Manages multiple Devices HDD, Monitor, mouse, Printer, Speaker, Webcam).

4) Disk Management (Disk scheduling Algorithms)

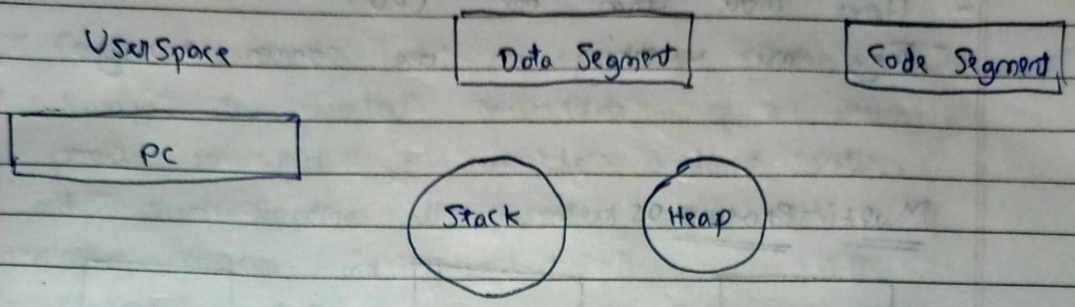
5) Network Management (Network Card/ controller)

6) File Management

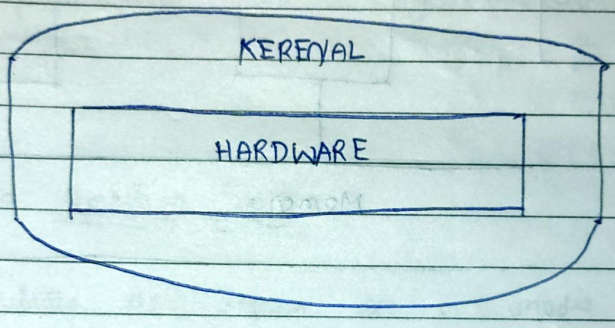
7) Security Management. (Firewall, Anti-virus, Anti-Spyware).

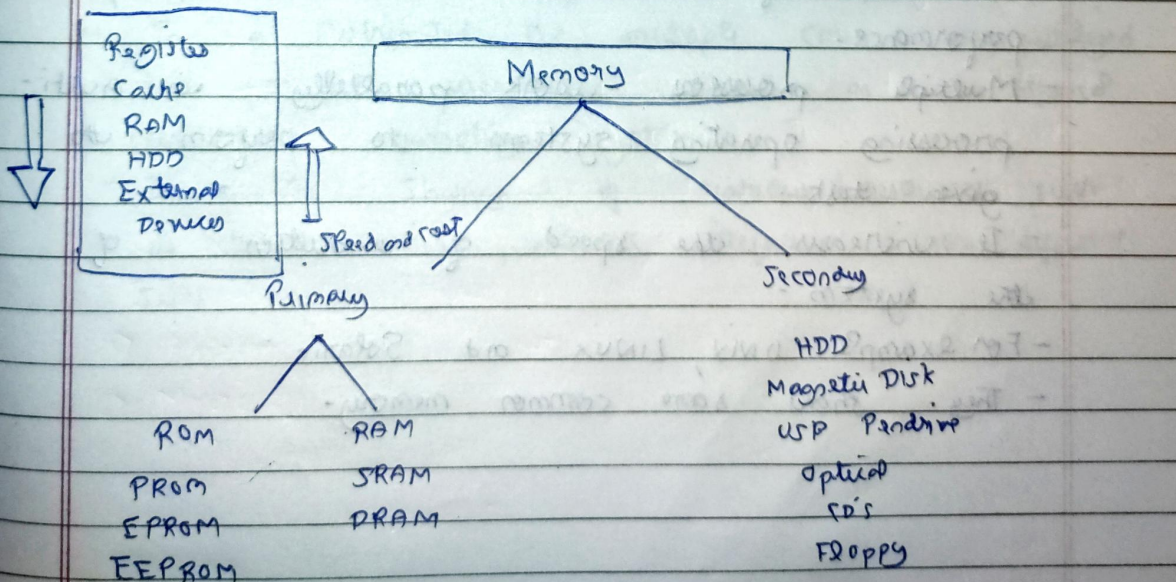
PAGE No.
DATE / /

* User Space and Kernal Space



Kernal Space

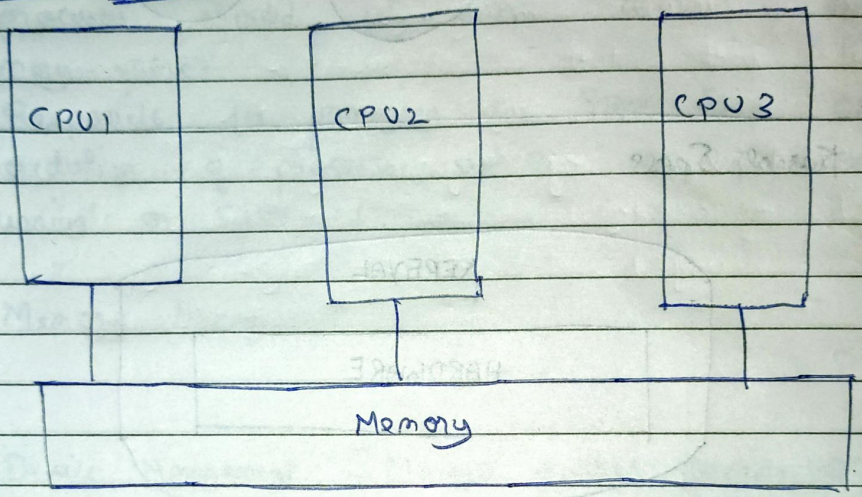




PAGE No.
DATE / /

* Multi-processing Operating System.
- Have more than one CPU.
- They are working on one common memory

Multi-Processor OS



- A multi-processing operating system is defined as a type of operating system that makes use of more than one CPU to improve performance.
- Multiple processors work parallelly in multi-processing operating systems to perform the given task.
- It increases the speed of execution of the system.
- For example UNIX, LINUX and Solaris.
- They share some common memory.

PAGE No.
DATE / /

* Context switching

- Context switching in an operating system involves saving the context or state of a running process so that it can be restored later, and then loading the context or state of another process and run it.
- Context Switching refers to the process / method used by the system to change the process from one state to another using the CPUs present in the system to perform its job.

* Distributed Operating System -

- A Distributed OS refers to a model in which applications run on multiple interconnected computers, offering enhanced communication and integration capabilities compared to a network OS
- In a Distributed OS, multiple CPUs are utilized but for end user, it appears as a typical centralized operating system.
- It enables sharing of various resources such as CPUs, disks, network interfaces across different sites

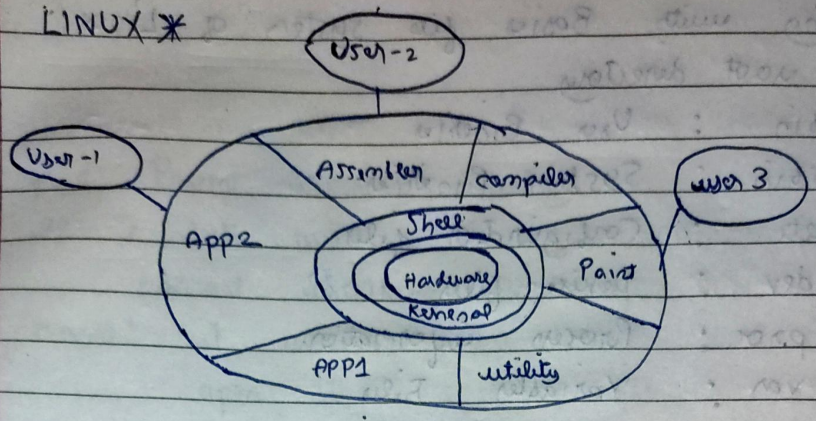
PAGE No.
DATE / /

* Batch Operating System

- A Batch operating system (BOS) is a type of computer system that allows multiple users to work on a system at the same time without communicating directly with each other.
- This is done by keeping users in seperate batches or groups so that they cannot interact directly.
- This makes it useful for applications where users need to work on different parts of project without interfering with each other.

PAGE No.
DATE / /

* LINUX *



Linux OS

- Linux is an Open-source operating System.
- It is availble free to use and user can modify it according to their need.
- The founder of Linux is Linus Torvalds. It is availble since 1991.
- An Open source community is working behind the updation and upgradation of linux code.
- Features of Linux:
  1) No cost / Low cost
  2) Multi-Tasking
  3) Security
  4) Multi-user
  5) Stable and scalable
  6) Networking
  7) CLI as wwel as GUI.

- Working with Basic file system of Linux

-/ is root directory

1) /bin : User Binaries
2) /sbin : System Binaries
3) /etc : Configuration files
4) /dev : Device files
5) /proc : Process information
6) /var : Variables Files
7) /tmp : temporary files
8) /usr : User Programs
9) /home : Parent directory of user friendly directory
10) /boot : Boot Loader Files
11) /opt : Apps
12) /lib : System Libraries

* Ubuntu for Windows

1) $ pwd → Prints present working directory
/home/malkeet

2) $ nano abc.txt
$ ls

3) ls : It lists out all the files and directory of current working directory.

$ touch file1.txt
$ ls

$ touch file2.txt
$ ls

$ mkdir

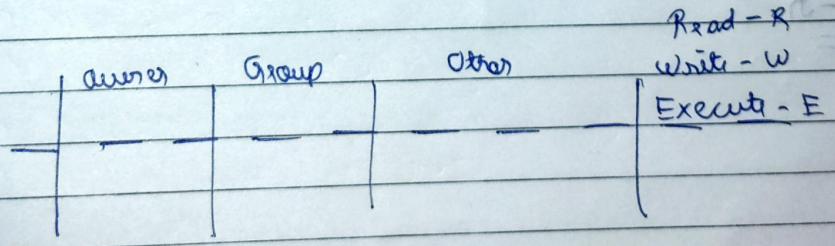
1) pwd: Present working directory
2) ls: It lists out all the files and directories of current working directory.
3) nano: it actually run the nano editor and open the specified file.

4) touch: It is used to create a new file

5) mkdir: It is used to create new directory.
6) chmod: to give and revoke file and directory permission
7) rm: Used to remove a file.

$ ls -1
↓
Give details about files in directories.

$ whoami
↓
Tells the currently working user.



chmod u+x file2.txt.

u - owner
g - group
o - other
a - all users

$ chmod

↓

It is used to give read, write and execute permission

$ chmod a+wx file2.txt

ls -l

$ chmod o-wx file2.txt

$ chmod a-x file2.txt

8) cd :- This command is used to change the directory.

9) cd.. :- Goes one step directory Back.

10) rmdir : Used to remove the directory

11) rm : to remove file and recursive directory

12)

rm -r

==

$ sudo adduser user2

- This command is used to create a user.

- sudo is known as root privileges.

$ sudo chown user2 file1.txt

- The chown command in Linux is used to change the owner of a file, directory or link.
- Chown is short for "change owner".

cd /etc/
↓
ls
↓
sudo nano sudoers

* Display content of the file to the screen.

$ nano file.txt
↓
$ cat file.txt
↓
$ head -5 file.txt (Prints first 5 lines of file)
↓
$ tail -5 file.txt (Prints last 5 lines of file)

$ head file.txt (Prints first 10 lines)

$ Tail file.txt (Prints last 10 lines)

* 1) chown: To change the owner of file

2) SW: to switch the user from current to any specified user.

3) cat: to display content of file on console.

4) head: to display top 10 lines of the file on console. By default it will print first 10 lines.

5) tail: to display bottom 10 lines of the file on console.
By default it will print last 10 lines.

6) sudo: to give some specific privileges to the users other than root.

$ cat file.txt

$ cat > file3.txt

hello
hii
Bye
You

$ cat file3.txt

$ head -5 file.txt

* Redirection (>>)

- Output of one command is used as an input of one command.

$ tail -5 file.txt > file5.txt

PAGE No.
DATE / /

* System call

- A system call is a programmatic way in in which a computer program request a service from the Kernel of the Operating system it is executed on.
- A System call is a way for programs to interact with the Operating System.
- A computer program makes a system call when it requests the Operating System's Kernel.
- It is a mechanism used by programs to request services from the Operating System (OS).

* System call Types

File related calls

1. Read(), Write(), Delete(), Open(), Close(), Create()

2. Process related calls: New(), FORK(), Exit(), Wait(), Running(), etc.

3. Device related calls: Read(), ioctl etc

4. Information related: getpid, gettime, sysdate, etc

5. Communication related: wait(), Signal(), status, etc

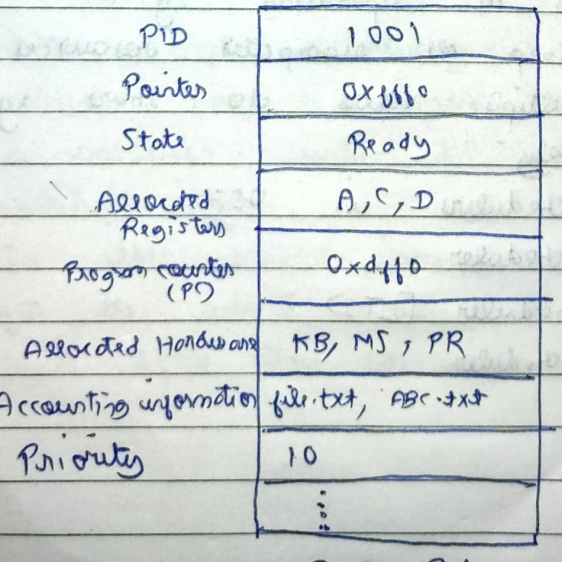
PAGE No.
DATE / /

* Process

- In OS, a process is a sequence of instructions that are executed in a specific order.
- A process is essentially a running program that represents the fundamental unit of work that must be implemented in a system.
- It is information or code or data which help the processor to execute or complete the user task.

1. Code Segment: It consist of compiled code or instructions to be executed.
2. Data Segment: It consist of data required for the execution.
3. Information Segment: It have metaData about the system variables or system which helps in process execution.

* Process Control Block (PCB)



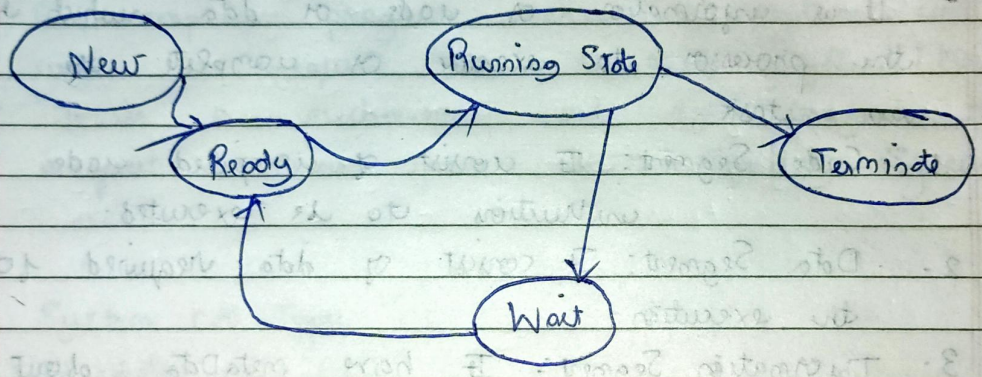
PCB P1

- A process control Block (PCB) is a data structure used by the Operating system to manage information about a process. The process control keeps track of many important piece of information needed to manage processes efficiently.

PAGE No.
DATE / /

* Process Life Cycle

1. New State (Creation)
2. Ready State (Ready Queue)
3. Running State - Processor is executing the process.
4. Terminated State
5. Wait State



Process Life Cycle.

* Scheduler

- A scheduler is a software that helps schedule the processes in an Operating System.
- It helps to keep all computer resources busy and allows multiple users to share system resources effectively.
- Types of Schedulers in OS:

1) Short-term scheduler
2) Long-term scheduler (LTS)
3) Medium-term Scheduler

PAGE No.
DATE / /

* Scheduling Algorithms

1) Pre-Emptive Scheduling
2) Non-Pre-emptive scheduling

Pre-emptive Scheduling

- In this preemptive scheduling, a higher priority task takes precedence over a lower priority tasks, even if the lower priority task is already running.
- For example, if a high priority process enters the queue and is allocated to the CPU, the current process switches from the running queue to ready queue. The lower priority task waits for some time and continues its executing process when the higher priority job gets completed.

Non-preempting Scheduling

- Non-preemptive scheduling is a CPU scheduling method where a process keeps control of a resource until it ends or becomes a waiting state.
- In this method, new process must wait for the current process to finish its CPU cycle before they can begin.

PAGE No.
DATE / /

1) FCFS (First Com First Serve)
   - Waiting time of process = CPU Allocation - Arrival time
   - Avg Waiting Time = Sum WT of All process / no of processes.

2) SJF (Shortest Job first) - Shortest process first with smallest Burst time.

3) Round Robin
   - Round Robin is a CPU scheduling algorithm where each process is cyclically assigned a fix time slot.
   - It is the preemptive version of the First come First serve CPU Scheduling algorithm.
   - Giving CPU to each process every time named as quantom time.

PAGE No.
DATE / /

* Memory Management:

- Fixed size partition
- Variable size partition

- The memory of computer system can be divided into block of fixed size or variable size
- This process is known as Fixed size partitioning and variable size partitioning respectively.
- Fixed size partitioning: Here the memory is divided into fixed size of Block like 2 Bytes, 4 Bytes etc.
- The process which will get the memory must be more than the partitioning size.

* Internal Fragmentation

- Internal fragmentation is a phenomenon that occurs when a memory block is allocated to a process that is smaller than the amount of memory requested.
- This results in unused space within the memory Block, which is then considered internal fragmentation.

* Variable size partitioning

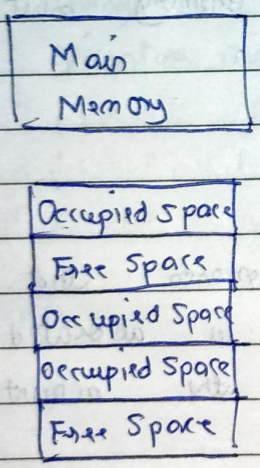
- Here the memory is divided into variable size Blocks in which Blocks may 2 Bytes, 3 Bytes or any size.
- In variable size partitioning during contiguous memory allocation some time total memory available can not given to the new process which leads to external fragmentation.

PAGE No.
DATE / /

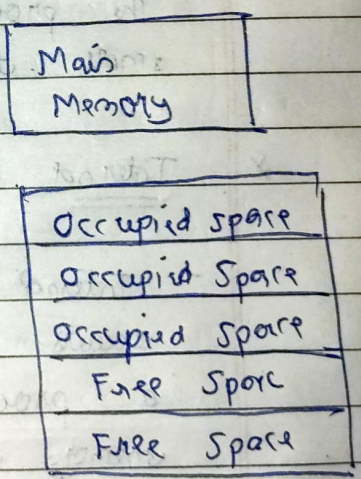
* Compaction

- Compaction is a technique to collect all the free memory present in the form of fragments into one large memory, which can be used to run other processes.
- It does that by moving all the process towards one end of the memory and all the available free space towards the other end of the memory so that it becomes contiguous.

Before compaction



After Compaction



PAGE No.
DATE / /

* Process scheduling Revision

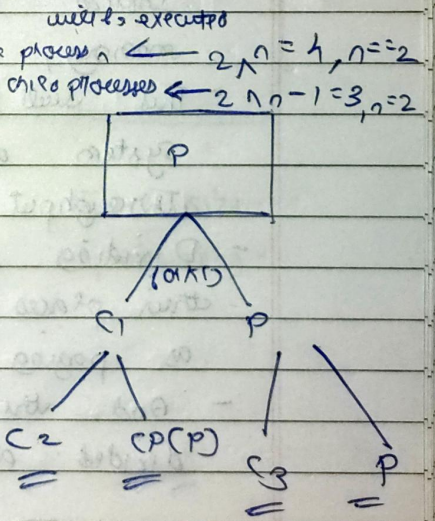
- Process creation using fork
- Wait pid and exec system calls (fork-exec-wait)
- Examples on process creation
- Parent and child processes
- Orphan and Zombie processes.

* Process creation using fork

- fork() : It is method / system call which is used to create child process of the current process executed.
- What a child process does: It does exactly the same for which you have designed the parent process.

```
$ nano program.c
$ gcc -o Program Program.c
$ ./Program
$
```

```
# include <stdio.h>
int main()
{
fork();
fork();
printf("Hello Aditya\n");
return 0;
}
```



Four time Hello Adi will be printed.

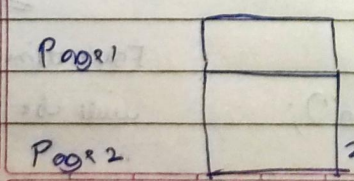
PAGE No.
DATE / /

* Orphan and Zombie process in OS

- A zombie process is a process that has finished execution but still has an entry in the process table.
- An Orphan process is a process that is still running but its parent processes has finished or terminated.

* Paging

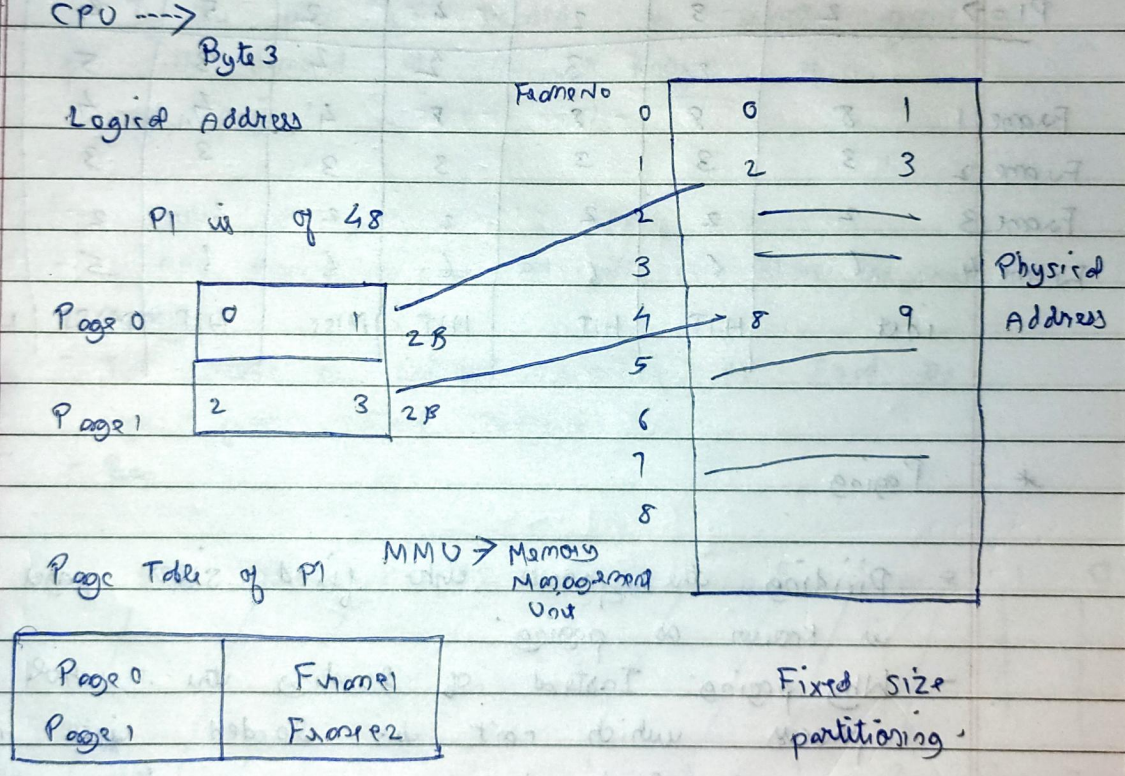
- The process of retrieving process in the form of pages from the secondary storage into the main memory is known as Paging.
- The Basic purpose of paging is to separate each processes into pages.
- Additionally frames will be used to split the main memory.
- Instead of loading entire program in main memory, just load the needed one.
- This will increase the efficiency of your system and it will increase the throughput.
- Dividing process into number of pages ~~and~~ on their places which is on Hardrive is known as paging.
- And the sections to which they are divided are known as Pages.



P1 of size 3B can

- The program is divided into pages and memory is divided into frames.

* To support paging the size of pages must be equal to the size of frames.
* All the pages must be divided into equal size.



Fixed size partitioning.

PAGE No.
DATE / /

* Page replacement Algorithms
1) LRU
2) MRU

- If the page is not in the frame, its known as miss of the page and it's a page fault.

P1 ->	2	3	2	4	3	5	4	7	4
		2	3	2	4	3	5	4	7
Frame 1	8	8	8	5	4	4	4	4	4
Frame 2	3	3	3	3	3	3	3	3	3
Frame 3	2	2	2	2	2	2	2	2	2
Frame 4	1	1	1	1	1	1	5	5	7
	init	HIT	HIT	HIT	MISS	HIT	MISS	HIT	MISS

* Paging

- Dividing the process into fixed sizes pages is known as paging.
- Why paging: Instead of loading the whole process which can't be loaded into main memory OS loads few page of the process into main memory according to frames avaible and other pages loaded from storage device on demand of CPU.

PAGE No.
DATE / /

* Paging Table

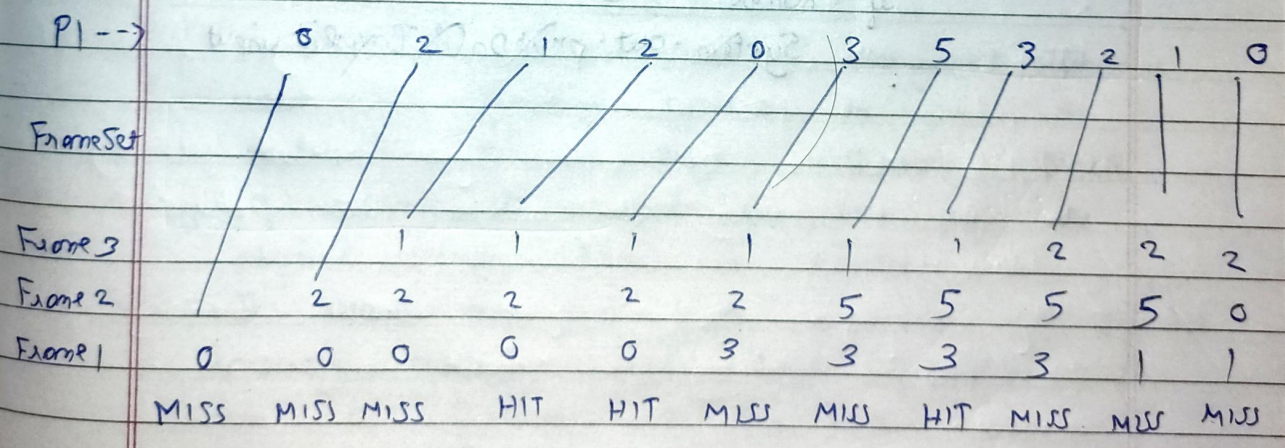
- It is a special table maintained by MMU (Memory management unit) to map the logical address of the demanded page with the physical address of the page in main memory where actually the page is placed.

* Demand paging

- If the page is loaded in memory frame as per the demand of CPU i.e known as demand paging.

* Page Fault:

- If CPU demands a page and it's not there in main memory.
- That time an interrupt generated to load the page from virtual memory.
- ~~Bug~~



Hit Ratio = 3/11

Miss Ratio = 8/11

[illegible]

P1 →	0	2	1	2	0	3	5	3	2	1	0
Frame 4						3	3	3	3	3	3
Frame 3			1	1	1	1	1	1	1	1	1
Frame 2		2	2	2	2	2	2	2	2	2	0
Frame 1	0	0	0	0	0	0	5	5	5	5	5
	MISS	MISS	MISS	HIT	HIT	MISS	MISS	HIT	HIT	HIT	MISS

* Belady's Anomaly:

- Belady's anomaly is the phenomenon in which increasing the number of page frames results in an increase in the number of page faults for certain memory access patterns.

* Virtual Memory

- Virtual Memory is a memory management technique used by Operating systems to give the appearence of a large, contiguous Block of memory to applications, even if the physical memory (RAM) is limited.
- It allows the system to compensate for physical memory shortages, enabling larger applications to run on Systems with less RAM.

- It is Space in HardDrive

- Hardware required for paging (virtual + cache)

- Translation look aside buffer.
  - It is a page table represented in cache memory to fasten the process of finding physical address for a given logical address.
- In paging we load only few pages in main memory of a particular ~~process of~~ process.

* Virtual Memory

- It is a memory space in Hard-drive, which works like physical memory to entertain large processes whose size is bigger than RAM.
- In virtual Memory process is divided into fixed sized partitions known as page.
- OS loads the process pages from virtual memory to physical memory on demand of CPU i.e known as Demand paging.
- While working with virtual memory pages are saved as per their logical address.
- While loading the pages from virtual memory to physical memory the pages get physical address.
- If a page loaded from virtual memory to physical memory i.e known as swap-in process.
- If the page is shifted from physical memory to logical memory by the replacement process that is known as Swap out memory.

PAGE No.
DATE / /

* What is Shell?

- Shell is an interface between user and the Kernel.
- The user can interact with Kernel by using shell commands or shell script / program.
- It takes input from user and passes it to the Kernel.

* What are different Shells in Linux?

cd / etc

etc$ ls

cat /etc/shells

X = 100
Y = 200

echo $X
O/P :- 100

nano p1
#! /bin/bash
echo "Enter your name"
read str1
echo Welcome, $ str1

*
Echo "Enter num1"
read num1
if [$num1 -eq 5]
then
Echo "Number is equal to 5"
else
Echo "Number is not equal to 5"
fi

PAGE No.
DATE / /

*
```
echo Enter Num 1
read Num1
echo Enter Num 2
read Num2
if [ $Num1 -gt $Num2 ]
then
    echo Num1 is greatest
else
    echo Num2 is greatest
fi
```

*
```
echo Enter Num 1
read Num1
echo Enter Num2
read Num2
echo Enter Num3
read Num3
if [ $Num1 -gt $Num2 ]
if [ $Num1 -gt $Num3 ]
then
    echo Num1 is greatest
else
    echo Num3 is greatest
fi
else
if [ $Num2 -gt Num3 ]
then
    echo Num2 is greatest
else
    echo Num3 is greatest
fi
fi
```

PAGE No.
DATE / /

* While loop in Shell programming

Syntax:

```
while [ condition ]
do
statement
increment
```

```
#! /bin/bash
a=0
while [ $a -lt 10 ]
do
echo $a
a=`expr $a + 1`
done
```

O/P:

```
0
1
2
3
4
5
6
7
8
9
10
```

PAGE No.
DATE / /

* cat file1 | grep ^h.n$
cat file1 | grep ^h.*n$

- Piping allows us to sort and filter the content

cat file2 | grep ^h

Piping allows us to filteration among the result of the command.
Eg cat file2 | grep ^h

* Redirection (>)

It allows you to use output of the command as an input of other command.

* Tilde (~)

Used for going back to the user's home directory

* Dollar Symbol ($)

It is also used to matching the end of the string for a line in a file.

* Caret (^)

It is used to match the starting of the string within given file.

```
echo Number of Param, $#
echo Script Name, $0
echo Hello, $1
echo Hii, $2
echo OK, $3
echo Bye, $*
```

Command line arguments

- Special parameters in Shell while passing command line arguments

1. $#: No. of parameters passed
2. $0: Script Name
3. $i: where i can be any number, it will return that parameter
4. $*: Give all parameters passed.

* Regular expression

- ls | grep ^s: It will filter the pipeline and get the words which are starting with s.

- ls | grep ^n.*f$: It will filter the pipeline and get the line which are starting with n and ending with f

- ls f*: It will return all the file starting with f.

PAGE No.
DATE / /

* Arithmetic expression -

```
#!/bin/bash

echo Enter a Number
read Num1
echo Enter a Number
read Num2
Res=`expr $Num1 \* $Num2`
echo Result, $Res
```

Basic Arithmetic Operators
1) + Addition
2) - Subtraction
3) * Multiplication
4) / Division
5) % Modulo
6) += increment by constant
7) -= Decrement by constant

* Network Commands

- Telnet : It is used to take the control of remot Server.

- SSH: It is also a used to take control of remote Server / machine.

- FTP: File Transfer protocol
  It allows user to send or recieve file from remote server / machine.

- STFP: Secure File Transfer protocol.