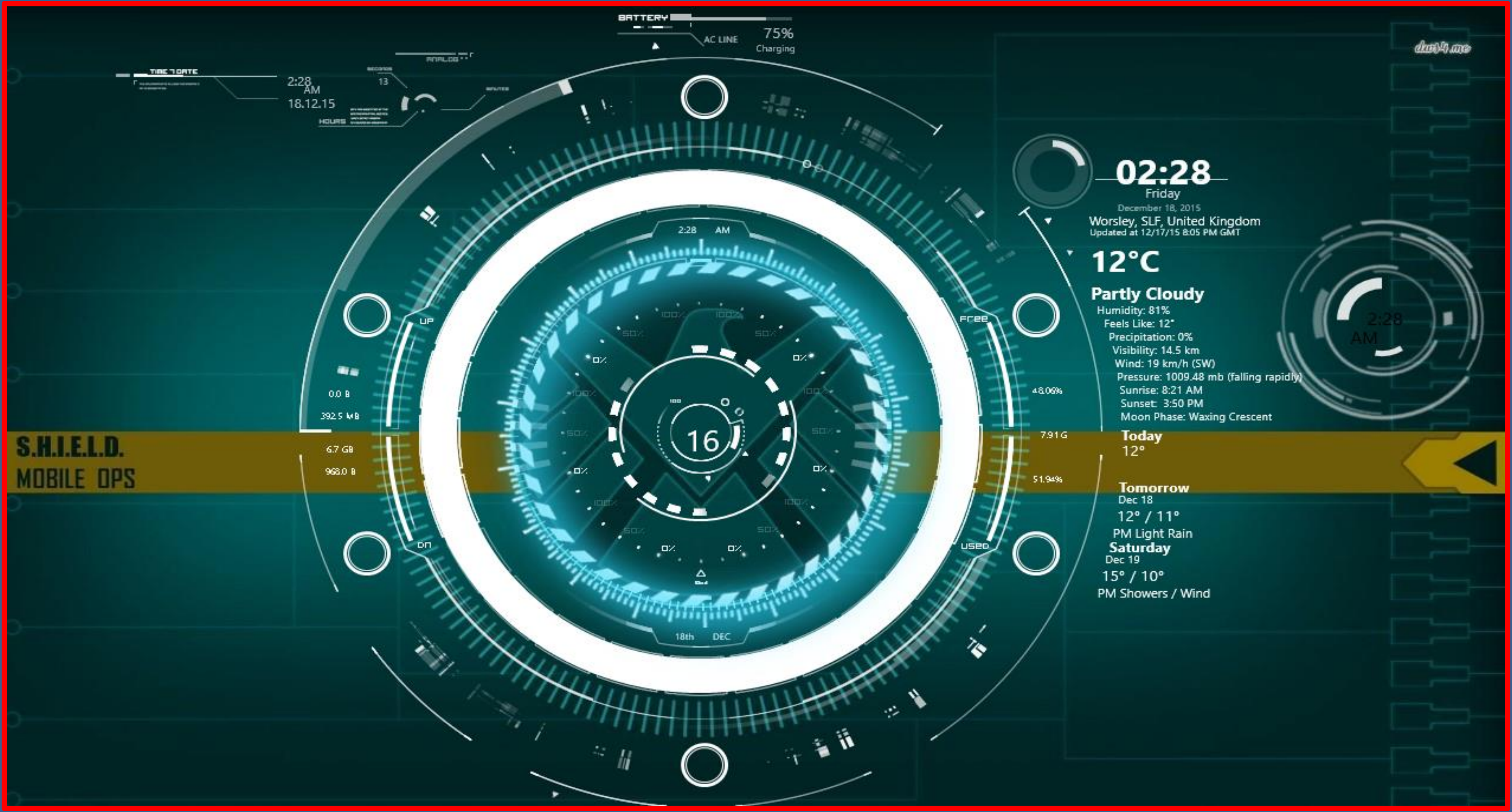
OPERATING SYSTEM



02:28
Friday
December 18, 2015
Worsley, SLF, United Kingdom
Updated at 12/17/15 8:05 PM GMT
12°C
Partly Cloudy
Humidity: 81%
Feels Like: 12°
Precipitation: 0%
Visibility: 14.5 km
Wind: 19 km/h (SW)
Pressure: 1009.48 mb (falling rapidly)
Sunrise: 8:21 AM
Sunset: 3:50 PM
Moon Phase: Waxing Crescent
Today
12°
Tomorrow
Dec 18
12° / 11°
PM Light Rain
Saturday
Dec 19
15° / 10°
PM Showers / Wind

OS: INTRODUCTION

Operating System:

A program that acts as an intermediary between a user of a computer and the computer hardware.

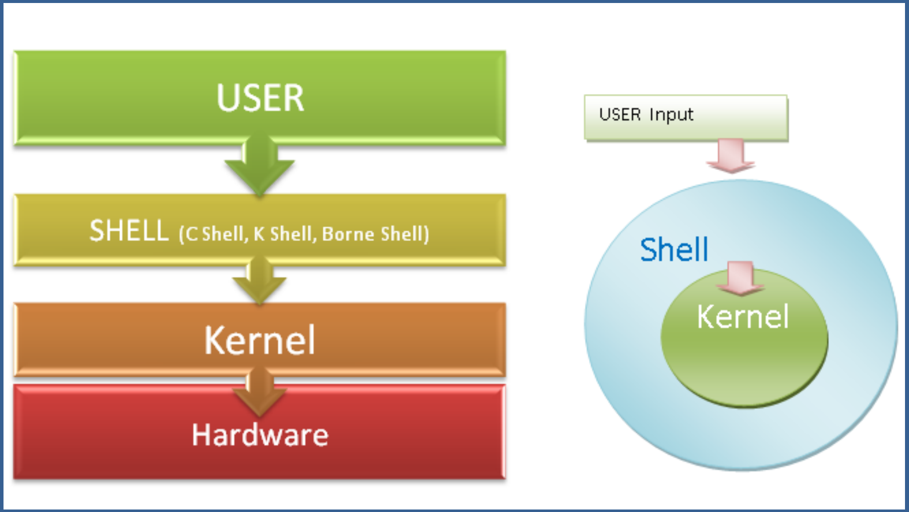
- **Resource allocator-** It Manages and allocates resources.
- **Control program–** Controls the execution of user

  programs and operation of I/O devices.
- **Device Manager-** It controls the work of devices used in computer.

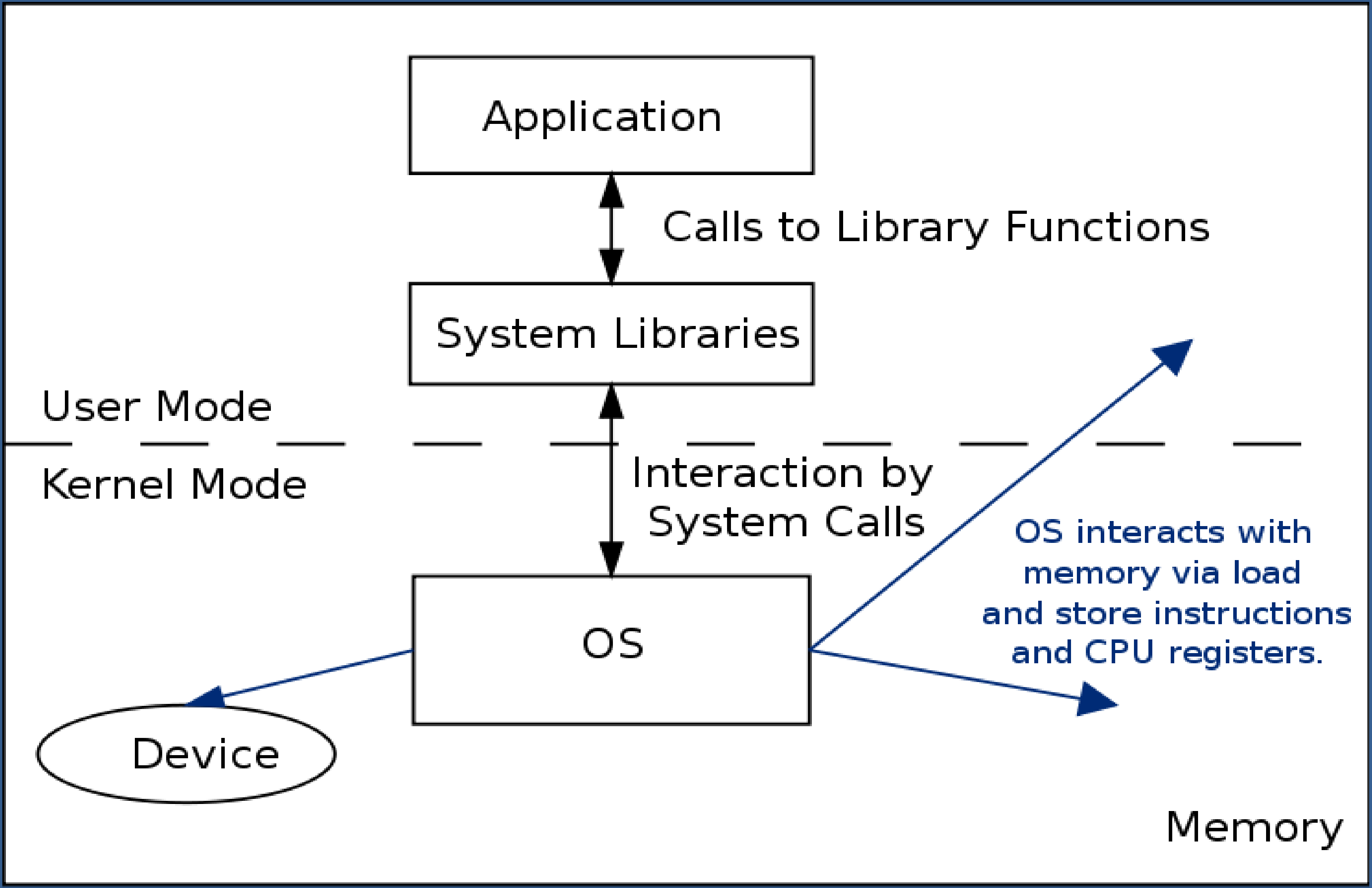
OS: INTRODUCTION.

Operating system has following parts

- kernel is a computer program that manages input/output requests from software, and translates them into data processing instructions for the CPU
- Shell is a command line interpreter (CLI). It interprets the commands the user types in and arranges for them to be carried out.



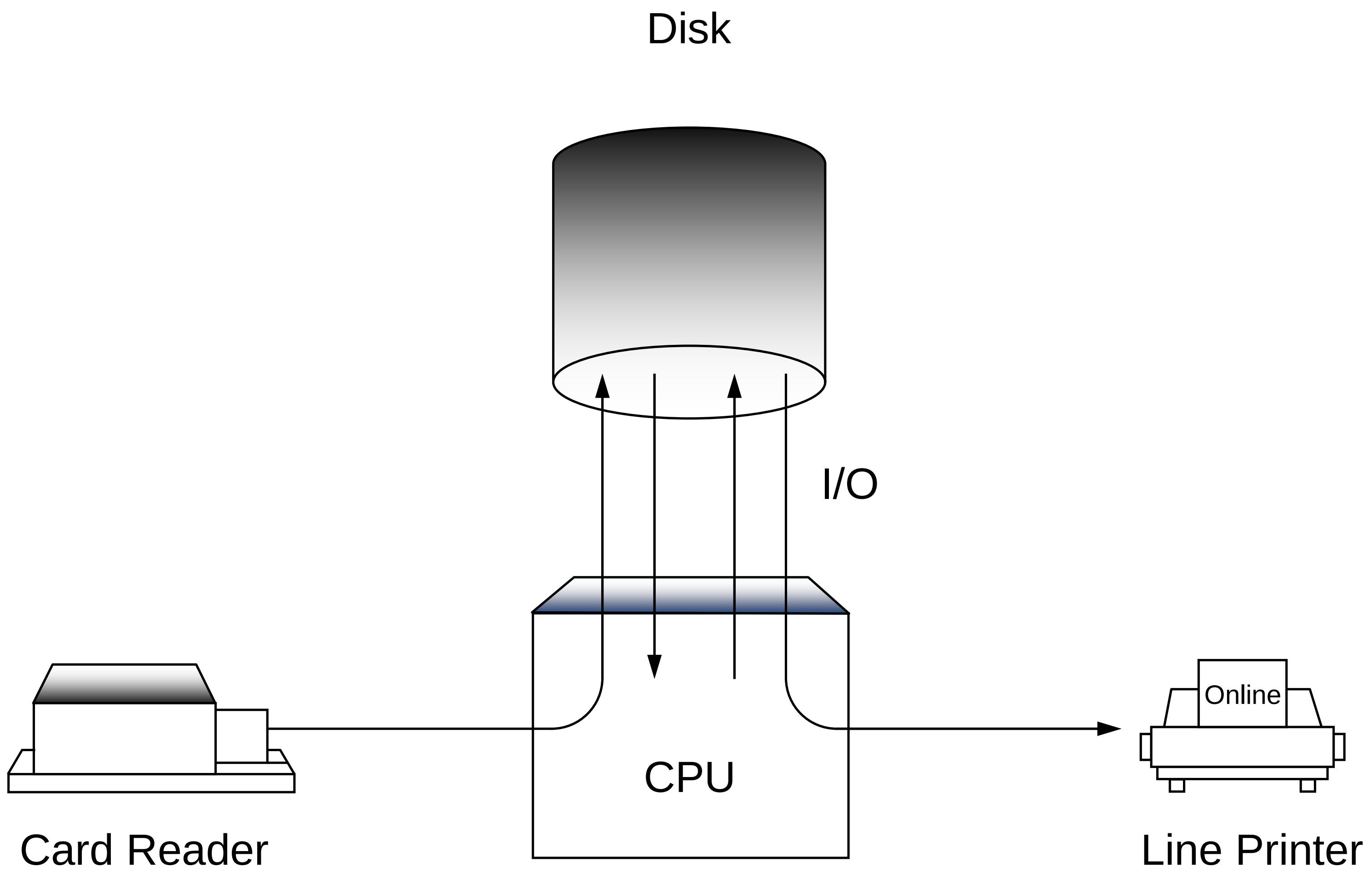
OS: INTRODUCTION


Application
Calls to Library Functions
System Libraries
User Mode
Kernel Mode
Interaction by System Calls
OS
OS interacts with memory via load and store instructions and CPU registers.
Device
Memory

Operating System : SPOOLING

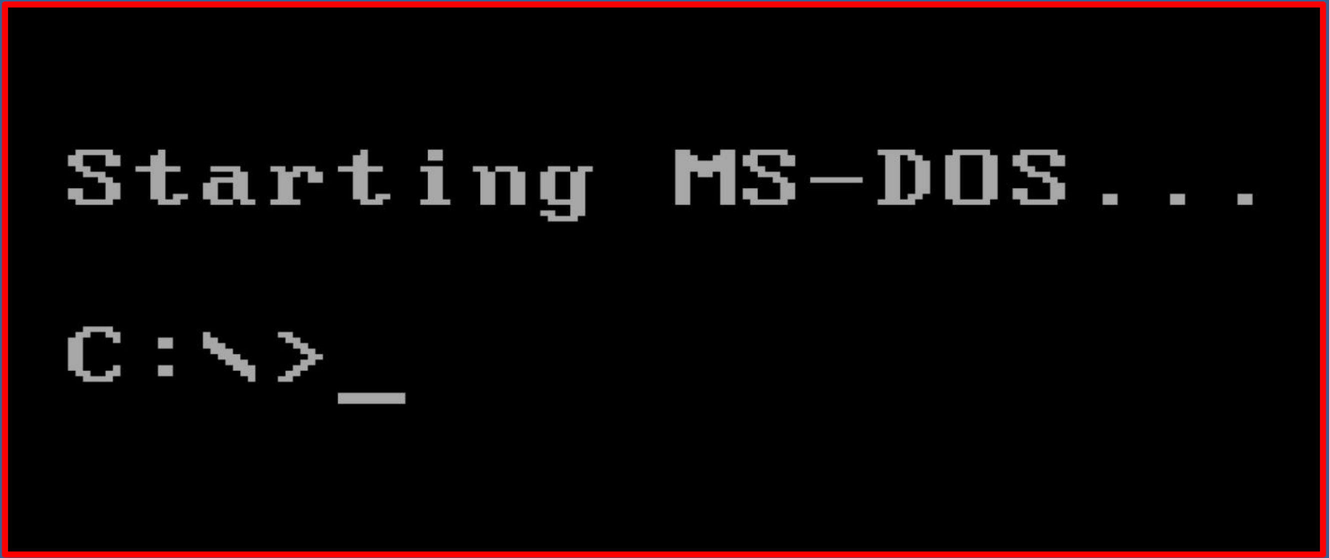
- SPOOL : **S**imultaneous **P**eripheral **O**peration **O**n-**L**ine
- A spool is a buffer that holds output for a device, such as a printer, that cannot accept interleaved data streams
- The spooler may be reading the input of one job while printing the output of a different job

Operating System : SPOOLING



OPERATING SYSTEM : TYPES

- A **single user operating system** only has to deal with the requests of the single person using the computer at that time.
- Eg. MS-DOS, Windows, Android etc..



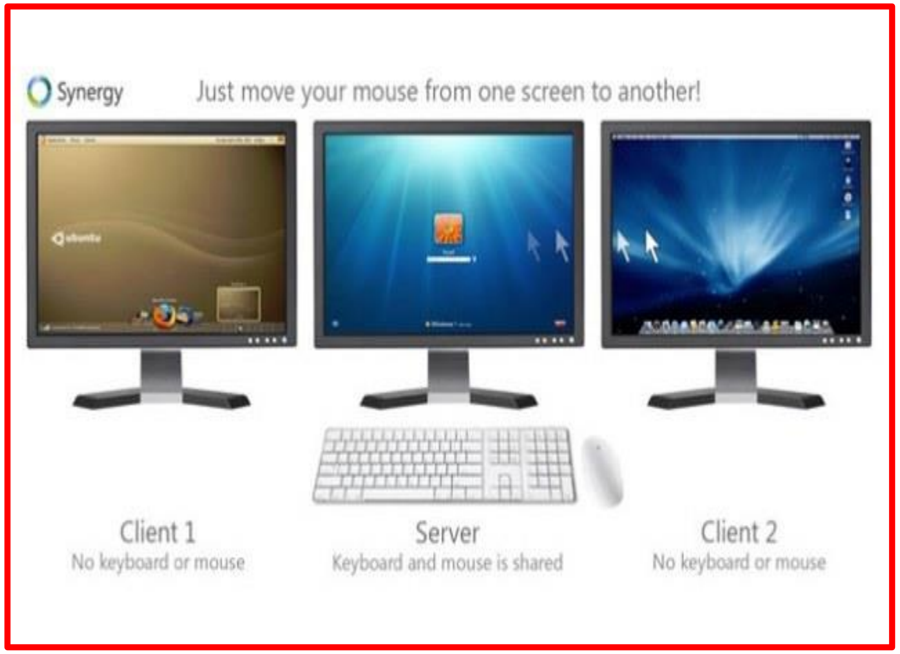
```
Starting MS-DOS...

C:\>_
```

OPERATING SYSTEM : TYPES

- **Multi-user operating** system allows lots of people to access the resources of the mainframe computer.
- A multi-user operating system 'slices up' the mainframes resources and divides it out to the different users.

Eg.– UNIX,LINUX etc.


Synergy
Just move your mouse from one screen to another!
Client 1
No keyboard or mouse
Server
Keyboard and mouse is shared
Client 2
No keyboard or mouse

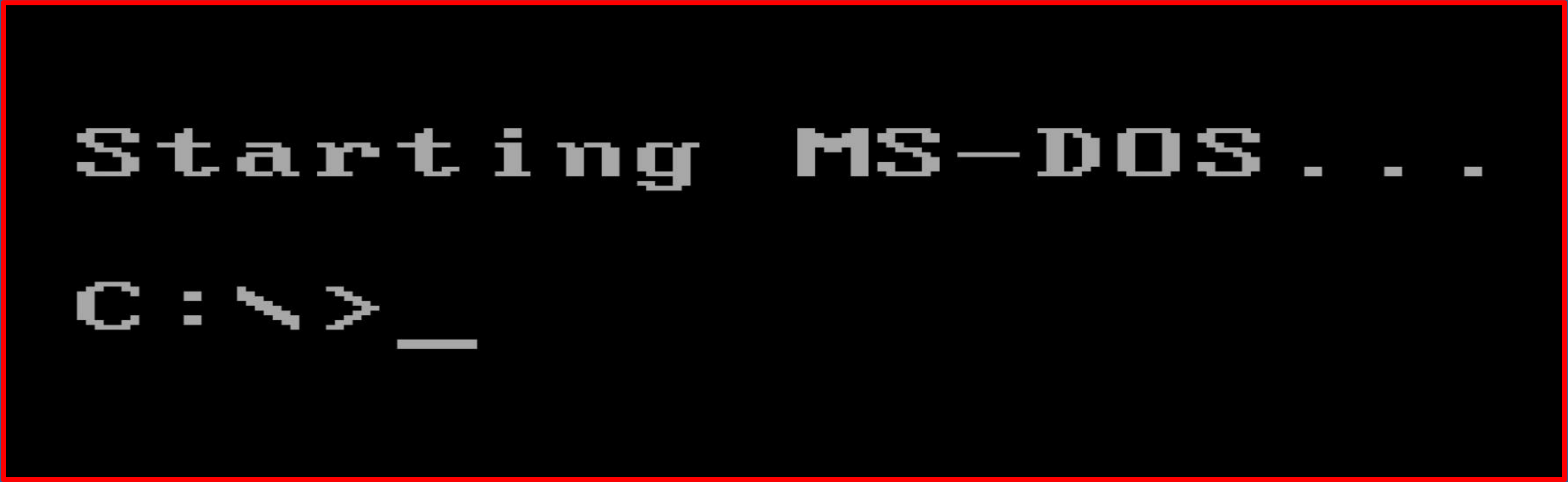
OPERATING SYSTEM : TYPES

- **Batch Processing operating system:** Batch processing is a technique in which Operating System collects one programs and data together in a batch before processing starts.
- OS keeps a number a jobs in memory and executes them without any manual information.
- Jobs are processed in the order of submission i.e first come first served fashion.

OPERATING SYSTEM : TYPES

- **Single-Tasking Operating System:**

  A single-tasking operating system can support only one task at a time. In such single tasking environment, the task execution is sequential. Eg. MS-DOS



```
Starting MS-DOS...

C:\>_
```

OPERATING SYSTEM : TYPES

- **Multi-tasking operating system:**

  Multi-tasking means to run more than one program at once. It is the operating system which manages this process.

- **Multiprocessing operating system:**

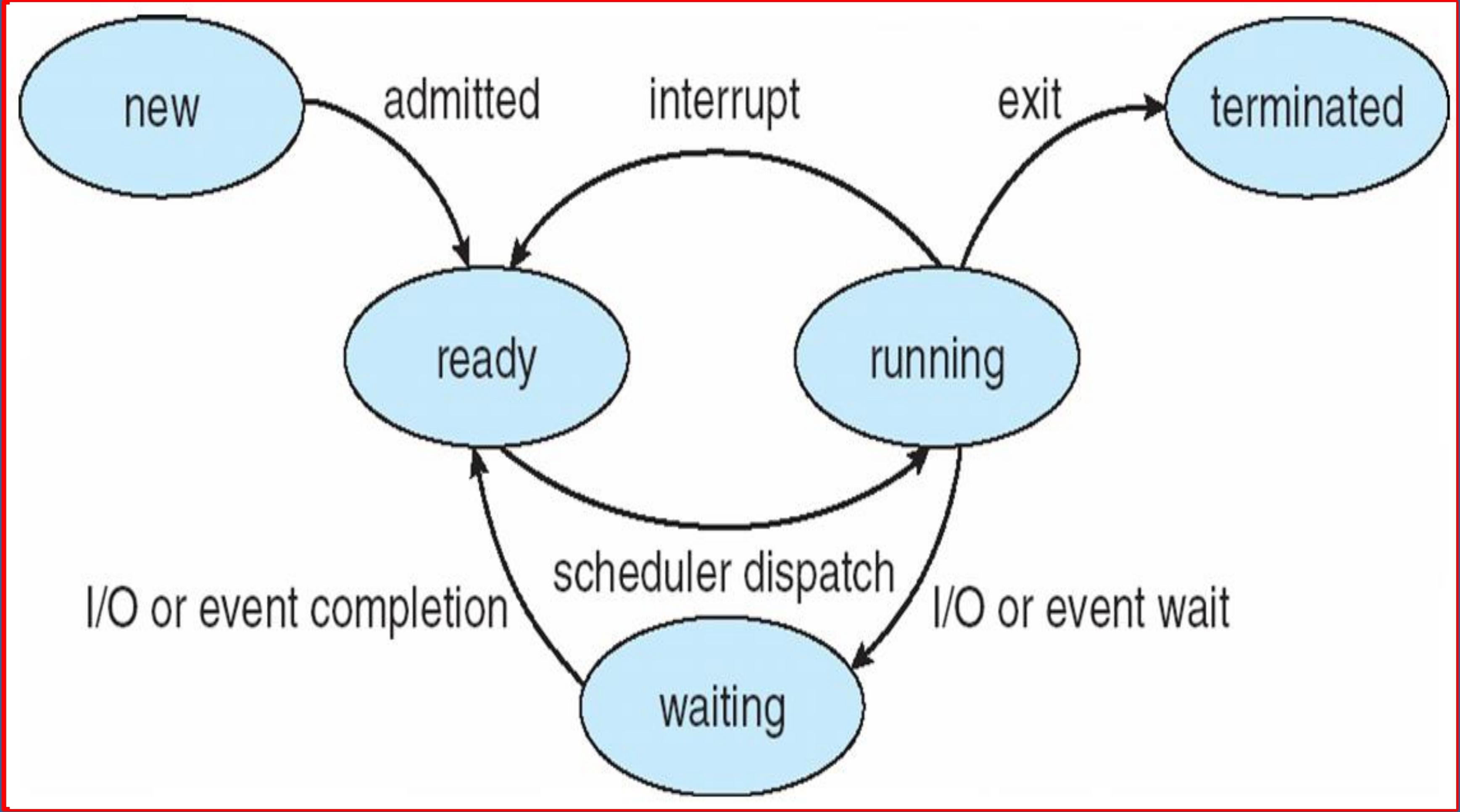
  Multiprocessor Operating System refers to the use of two or more central processing units (CPU) within a single computer system.

PROCESS MANAGEMENT

- A process is a program in execution. It is a unit of work within the system. Program is a passive entity, process is an *active* entity.

- Process needs resources to accomplish its task
  - CPU
  - memory
  - I/O

Diagram of Process State

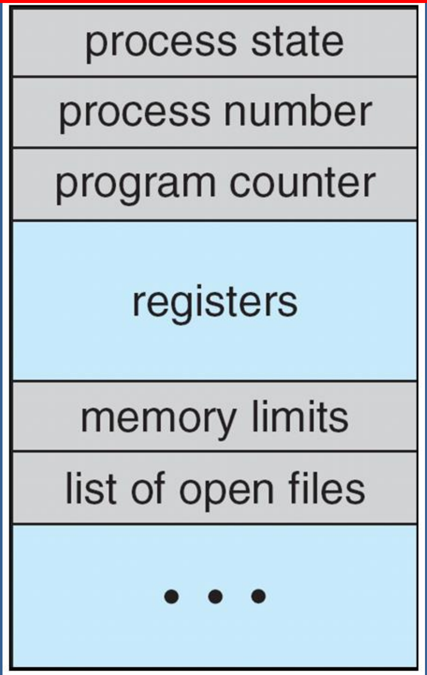


PROCESS SCHEDULING

- For Maximizing CPU use, quickly switch processes onto CPU for time sharing.
- **Process scheduler** selects among available processes for next execution on CPU.
- Maintains **scheduling queues** of processes
  - **Job queue** – set of all processes in the system
  - **Ready queue** – set of all processes residing in main memory, ready and waiting to execute.
  - **Device queues** – set of processes waiting for an I/O device.

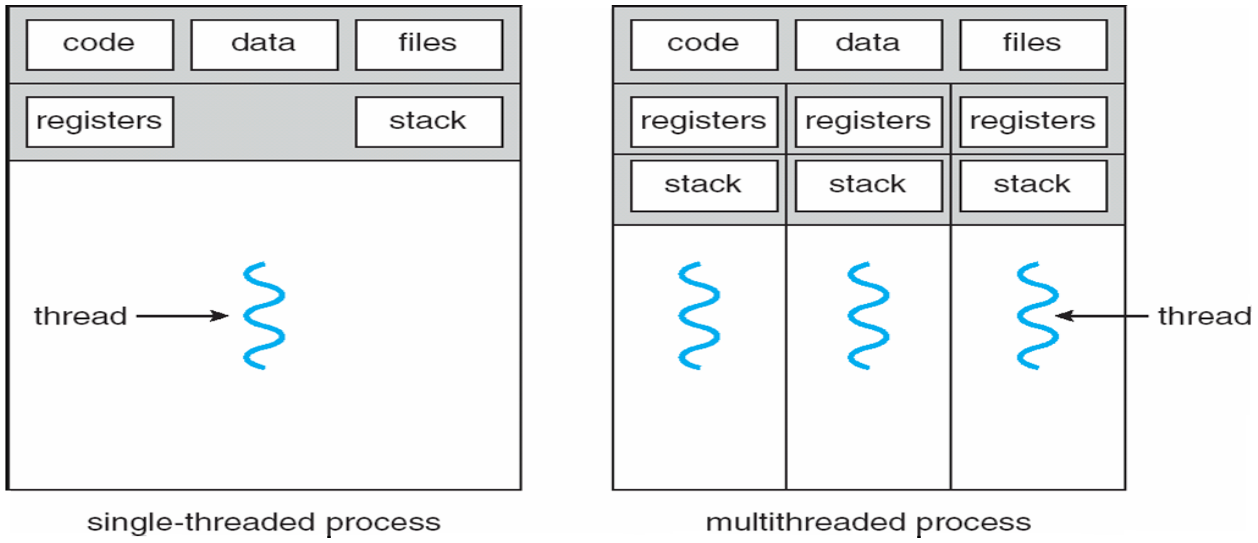
PROCESS CONTROL BLOCK (PCB)

- It is used for storing the collection of information about the Processes and this is also called as the Data Structure which Stores the information about the process.



THREADS

- Thread is a light weight process that uses same address space for the execution.
- As each thread has its own independent resource for process execution, multiple processes can be executed in parallel by increasing number of threads.



single-threaded process   multithreaded process

SCHEDULING CRITERIA

- CPU utilization – keep the CPU as busy as possible
- Throughput – number of processes that complete their execution per time unit
- Turnaround time – amount of time to execute a particular process
- Waiting time – amount of time a process has been waiting in the ready queue
- Response time – amount of time it takes from when a request was submitted until the first response is produced.

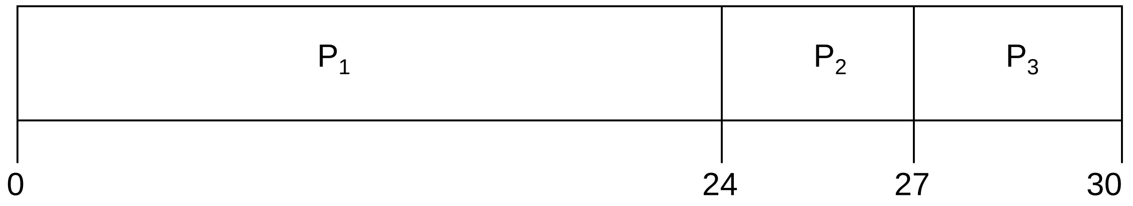
Scheduling Algorithm Optimization Criteria

- Max CPU utilization
- Max throughput
- Min turnaround time
- Min waiting time
- Min response time

FIRST-COME, FIRST-SERVED (FCFS) SCHEDULING

Process	Burst Time
P1	24
P2	3
P3	3

- **Suppose that the processes arrive in the order: P1 , P2 , P3**
  The Gantt Chart for the schedule is:



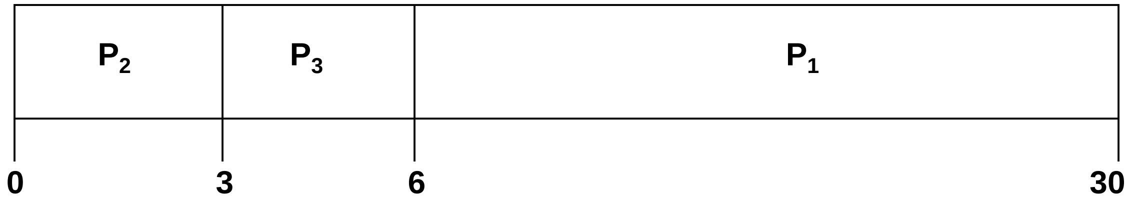
- **Waiting time for P1 = 0; P2 = 24; P3 = 27**
- **Average waiting time: (0 + 24 + 27)/3 = 17**

Note :

Burst Time is actually time that is required to complete execution of particular task or process.

FCFS SCHEDULING (CONT.)

- Suppose that the processes arrive in the order:
  P2 , P3 , P1
- The Gantt chart for the schedule is:



- Waiting time for P1 = 6; P2 = 0; P3 = 3
- Average waiting time: (6 + 0 + 3)/3 = 3

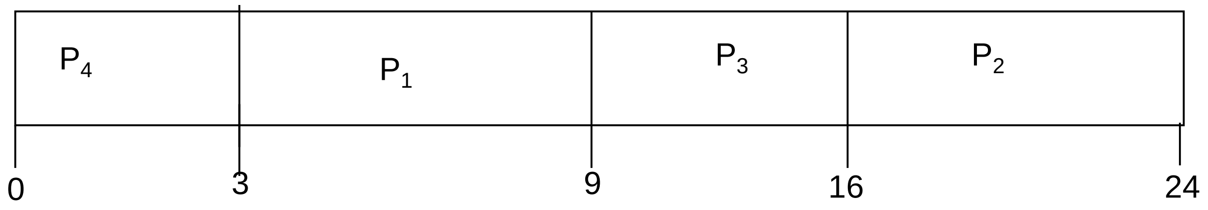
SHORTEST-JOB-FIRST (SJF) SCHEDULING

- Associate with each process the length of its next CPU burst
  - Use these lengths to schedule the process with the shortest time
- SJF is optimal – gives minimum average waiting time for a given set of processes

EXAMPLE OF SJF

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

- SJF scheduling chart



- Average waiting time = (3 + 16 + 9 + 0) / 4 = 7

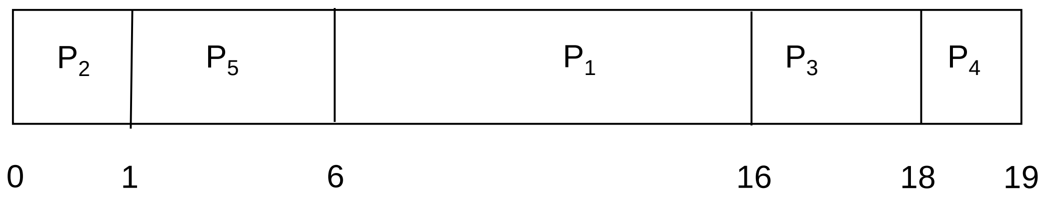
PRIORITY SCHEDULING

- A priority number (integer) is associated with each process
- The CPU is allocated to the process with the highest priority (smallest integer ≡ highest priority)

- Problem : Starvation – low priority processes may never execute
- Solution : Aging – technique of gradually increasing the priority of processes that wait in the system for a long period of time.

EXAMPLE OF PRIORITY SCHEDULING

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	4
P4	1	5
P5	5	2



- Priority scheduling Gantt Chart

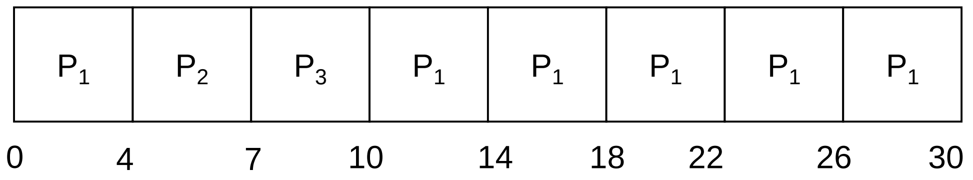
ROUND ROBIN (RR)

- Each process gets a small unit of CPU time (time quantum q), usually 10-100 milliseconds. After this time has elapsed, the process is preempted and added to the end of the ready queue.
- Timer interrupts every quantum to schedule next process

Example of RR with Time Quantum = 4

Process	Burst Time
P1	24
P2	3
P3	3

- The Gantt chart is:



- Typically, higher average turnaround than SJF, but better response
- q should be large compared to context switch time
- q usually 10ms to 100ms, context switch < 10 usec

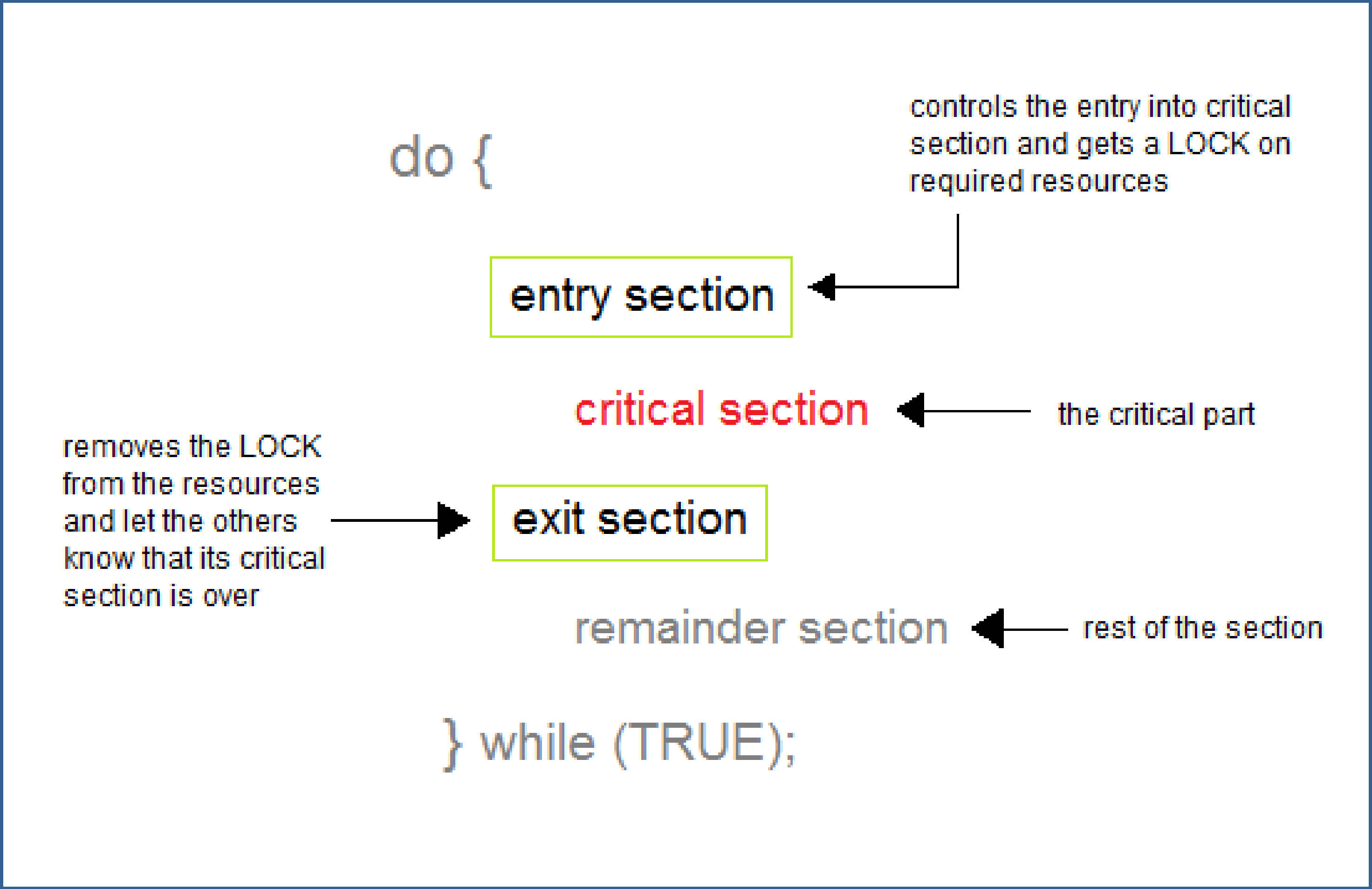
PROCESS SYNCHRONIZATION

- Process Synchronization means sharing system resources by processes in a such a way that, Concurrent access to shared data is handled thereby minimizing the chance of inconsistent data.

CRITICAL SECTION PROBLEM

- A Critical Section is a code segment that accesses shared variables and has to be executed as an atomic action. It means that in a group of cooperating processes, at a given point of time, only one process must be executing its critical section. If any other process also wants to execute its critical section, it must wait until the first one finishes.

CRITICAL SECTION PROBLEM


do {
controls the entry into critical section and gets a LOCK on required resources
entry section
critical section
the critical part
removes the LOCK from the resources and let the others know that its critical section is over
exit section
remainder section
rest of the section
} while (TRUE);

MUTEX LOCKS

- It provides support for critical section code .
- Is a strict software approach, here in the entry section of code, a LOCK is acquired over the critical resources modified and used inside critical section, and in the exit section that LOCK is released.

SEMAPHORES

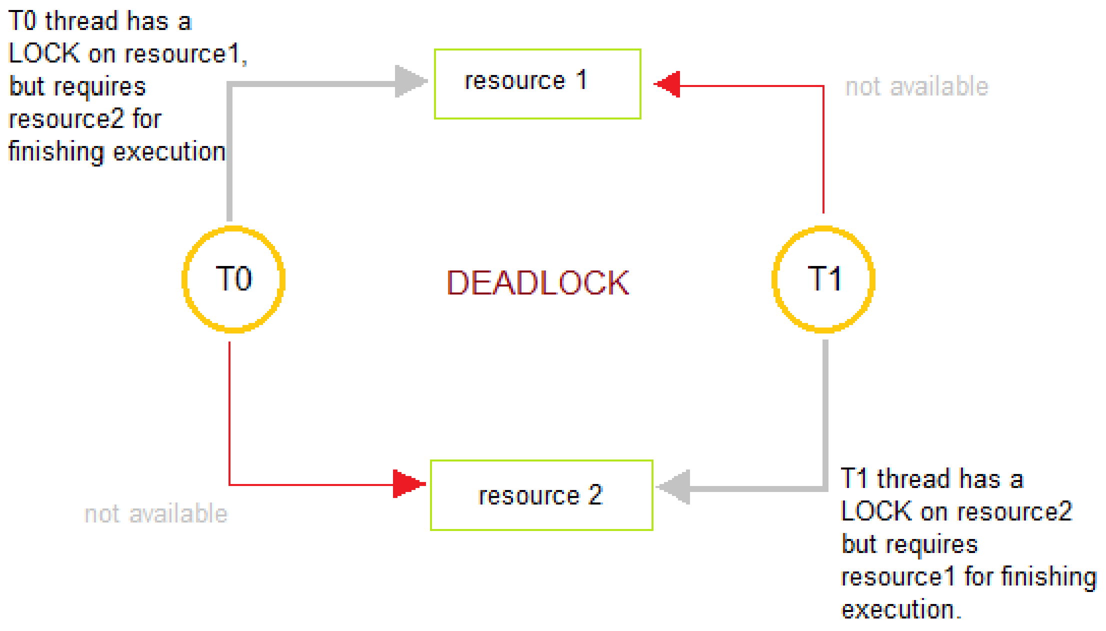
- It is a technique for managing concurrent processes by using the value of a simple integer variable. This integer variable is called semaphore.
- Semaphores are mainly of two types:
  - Binary Semaphore A binary semaphore is initialized to 1 and only takes the value 0 and 1 during execution of a program.
  - Counting Semaphores A synchronizing tool and is accessed only through two low standard atomic operations, wait and signal designated by P() and V() respectively

LIMITATIONS OF SEMAPHORES

- Priority Inversion is a big limitation os semaphores.
- Their use is not enforced, but is by convention only.
- With improper use, a process may block indefinitely. Such a situation is called Deadlock.

THE DEADLOCK PROBLEM

- Deadlocks are a set of blocked processes each holding a resource and waiting to acquire a resource held by another process.


T0 thread has a LOCK on resource1, but requires resource2 for finishing execution
resource 1
not available
T0
DEADLOCK
T1
resource 2
not available
T1 thread has a LOCK on resource2 but requires resource1 for finishing execution.

CONDITIONS FOR DEADLOCK

There are four conditions that are necessary to achieve deadlock

- **Mutual Exclusion** - At least one resource must be held in a non-sharable mode; If any other process requests this resource, then that process must wait for the resource to be released.
- **Hold and Wait** - A process must be simultaneously holding at least one resource and waiting for at least one resource that is currently being held by some other process.
- **No preemption** - Once a process is holding a resource (i.e. once its request has been granted), then that resource cannot be taken away from that process until the process voluntarily releases it.
- **Circular Wait** - A set of processes { P0, P1, P2, . . ., PN } must exist such that every P[i] is waiting for P[(i + 1)

HANDLING DEADLOCK

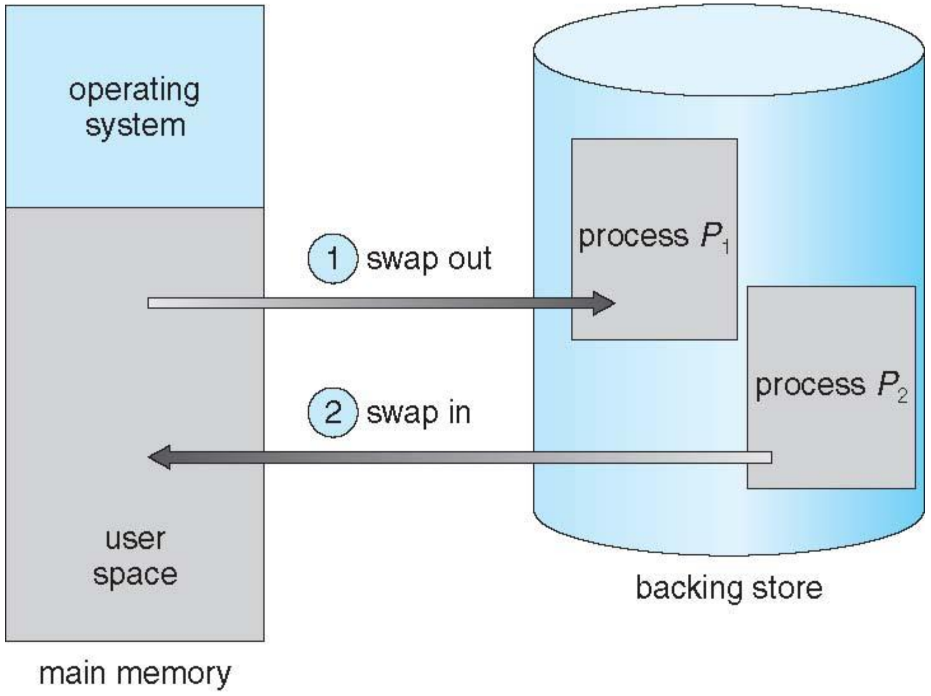
- Deadlocks can be prevented by avoiding at least one of the four conditions.
  - **Mutual Exclusion**
  - **Hold and Wait**
  - **No Preemption**
  - **Circular Wait**
- Deadlock can be Avoided by using banker’s algorithm.

MEMORY MANAGEMENT

- All the programs are loaded in the main memory for execution. Sometimes complete program is loaded into the memory, but some times a certain part or routine of the program is loaded into the main memory only when it is called by the program, this mechanism is called **Dynamic Loading**.

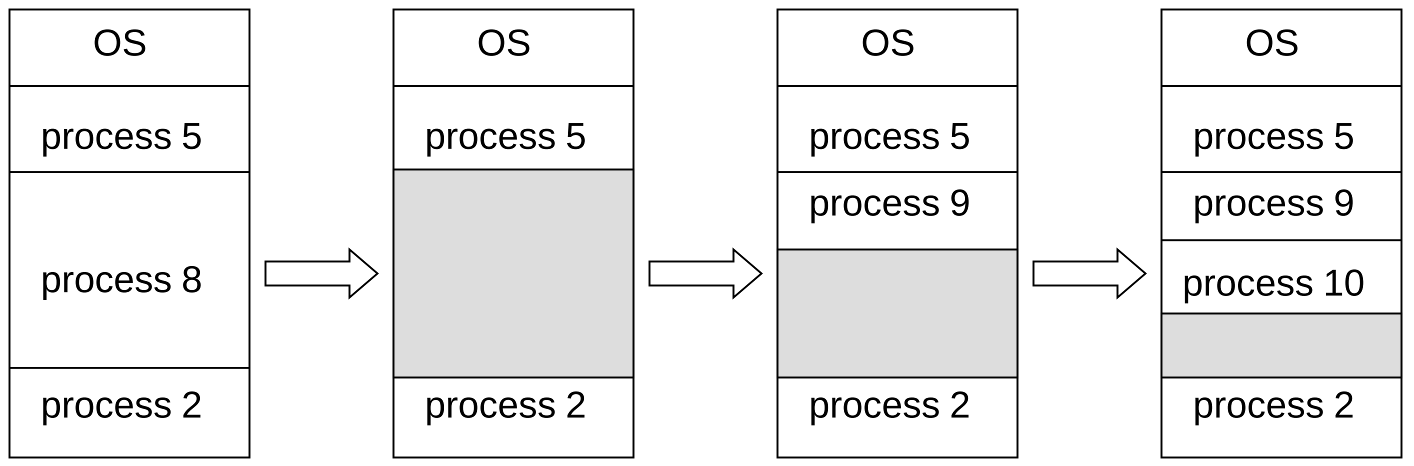
SWAPPING

Swapping is the process of bringing in each process in main memory, running it for a while and then putting it back to the disk.



CONTIGUOUS ALLOCATION

In contiguous memory allocation each process is contained in a single contiguous block of memory. The free blocks of memory are known as holes. The set of holes is searched to determine which hole is best to allocate.



DYNAMIC STORAGE-ALLOCATION PROBLEM

First-fit: Allocate the first hole that is big enough

Best-fit: Allocate the smallest hole that is big enough; must search entire list, unless ordered by size

Worst-fit: Allocate the largest hole; must also search entire list

FRAGMENTATION

Fragmentation occurs in a dynamic memory allocation system when most of the free blocks are too small to satisfy any request. It is generally termed as inability to use the available memory.

FRAGMENTATION ISSUES

External Fragmentation

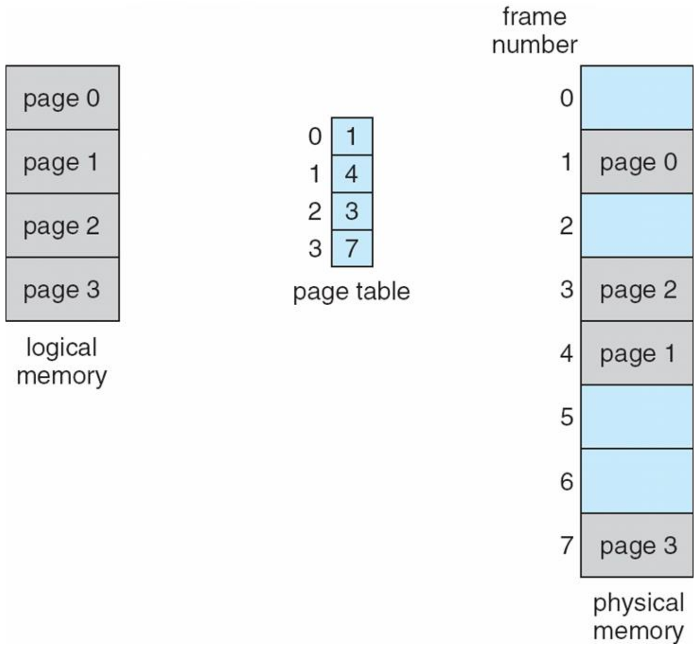
- total memory space exists to satisfy a request, but it is not contiguous
- Reduce external fragmentation by compaction

Internal Fragmentation

allocated memory may be slightly larger than requested memory

PAGING

Paging is a memory management mechanism that allows the physical address space of a process to be non-contagious.



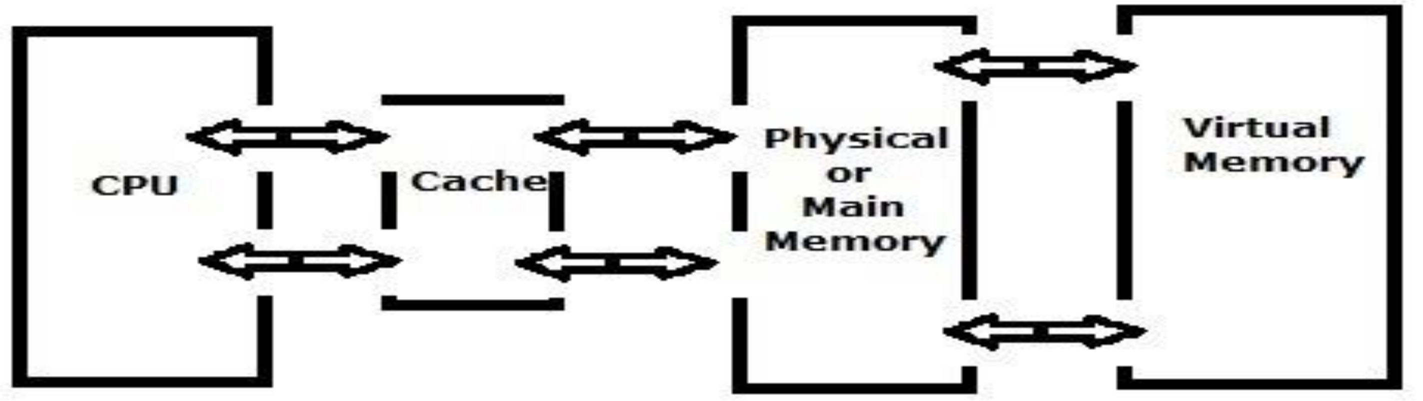
SEGMENTATION

Memory-management scheme that supports user view of memory

Segmentation allows breaking of the virtual address space of a single process into segments that may be placed in non-contiguous areas of physical memory.

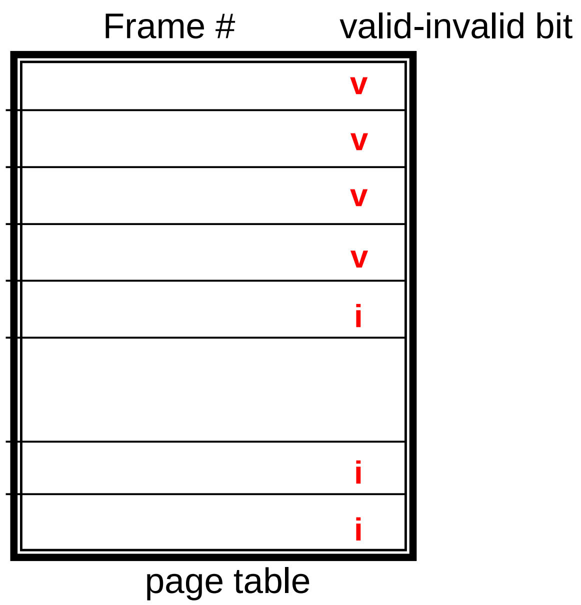
VIRTUAL MEMORY

- It provides the separation of user logical memory from physical memory.
- A technique that allows the execution of processes which are not completely available in memory.



DEMAND PAGING

- Virtual memory is commonly implemented by demand paging
- It is a type of swapping in which pages of data are not copied from disk to RAM until they are needed.



PAGE FAULT

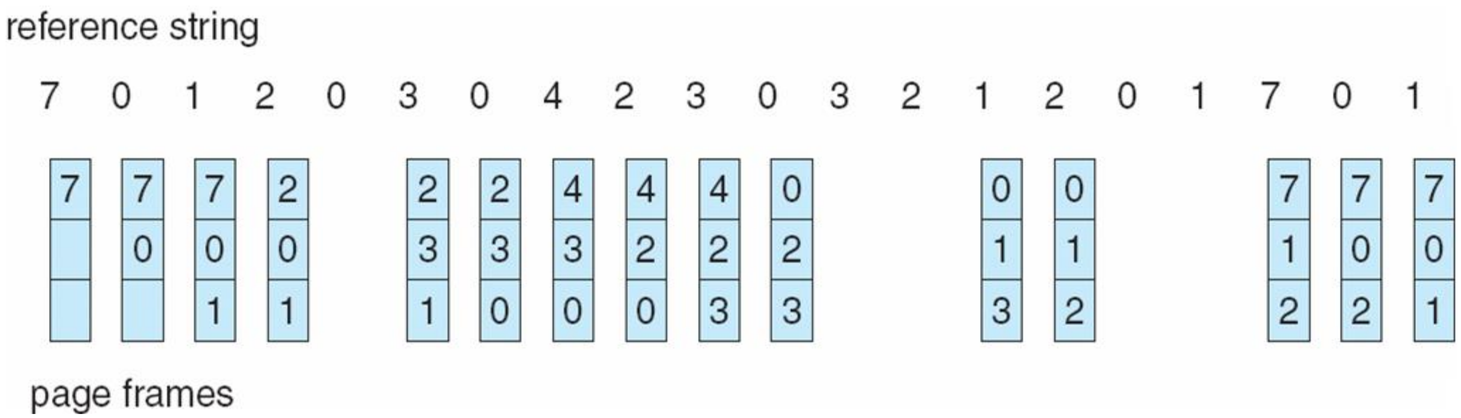
- An interrupt that occurs when a program requests data that is not in memory.
- The interrupt triggers the operating system to fetch the data from a virtual memory and load it into RAM.

PAGE REPLACEMENT

- when a process requests for more pages and no free memory is available to bring them in
- find some pages that are not being used right now, move them to the disk to get free frames. This technique is called Page replacement

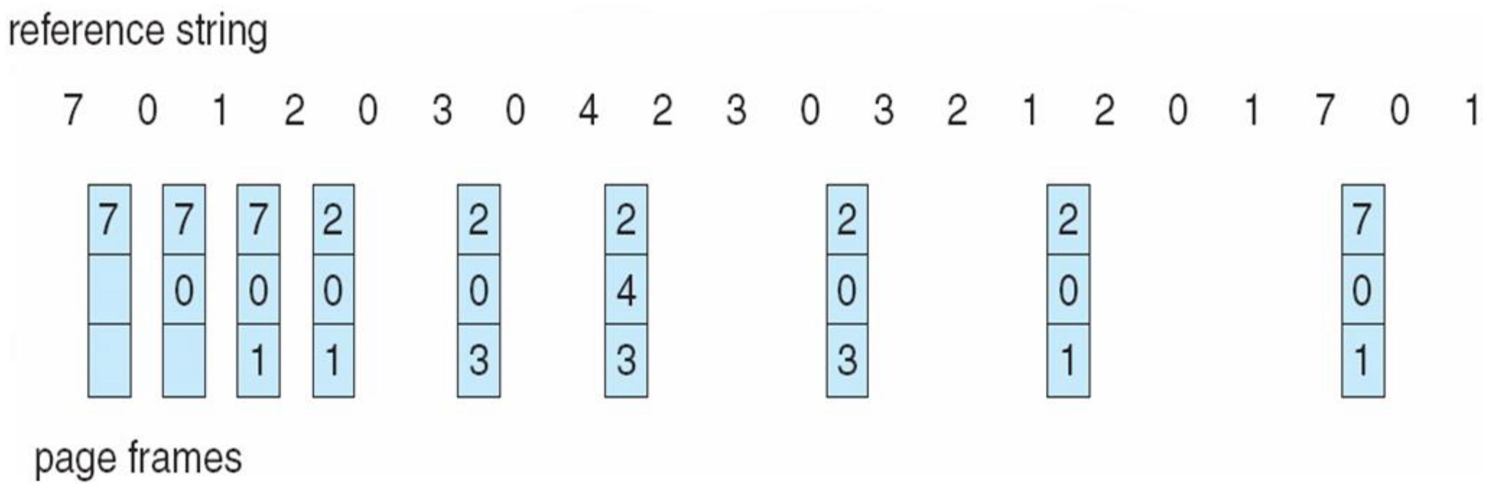
FIFO PAGE REPLACEMENT

- A very simple way of Page replacement is FIFO (First in First Out)
- Its not an effective way of page replacement but can be used for small systems.



OPTIMAL ALGORITHM

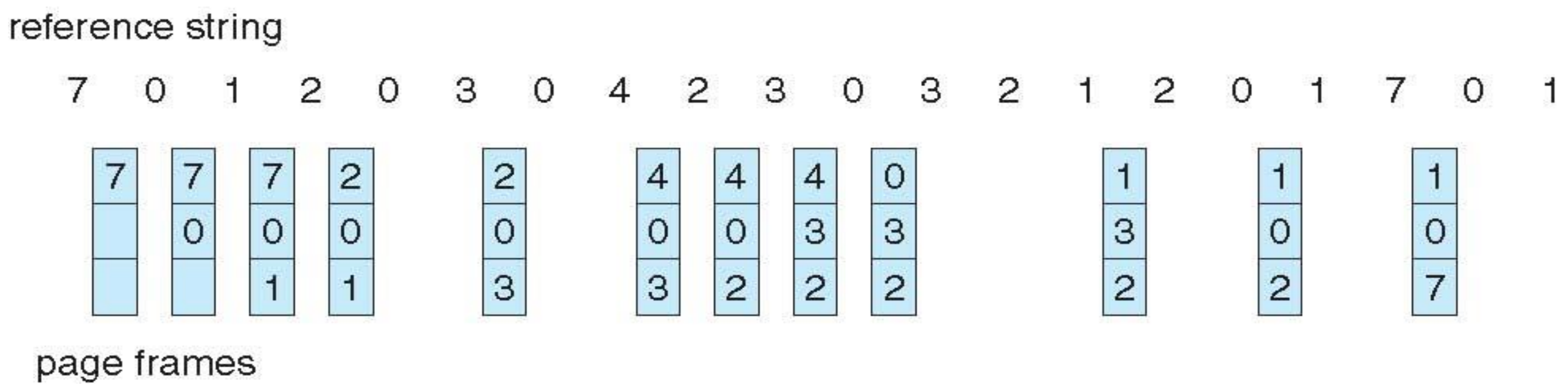
- Replace page that will not be used for longest period of time



- Not used practically
  – Can’t read the future

LEAST RECENTLY USED (LRU) ALGORITHM

- Use past knowledge rather than future
- Replace page that has not been used in the most amount of time



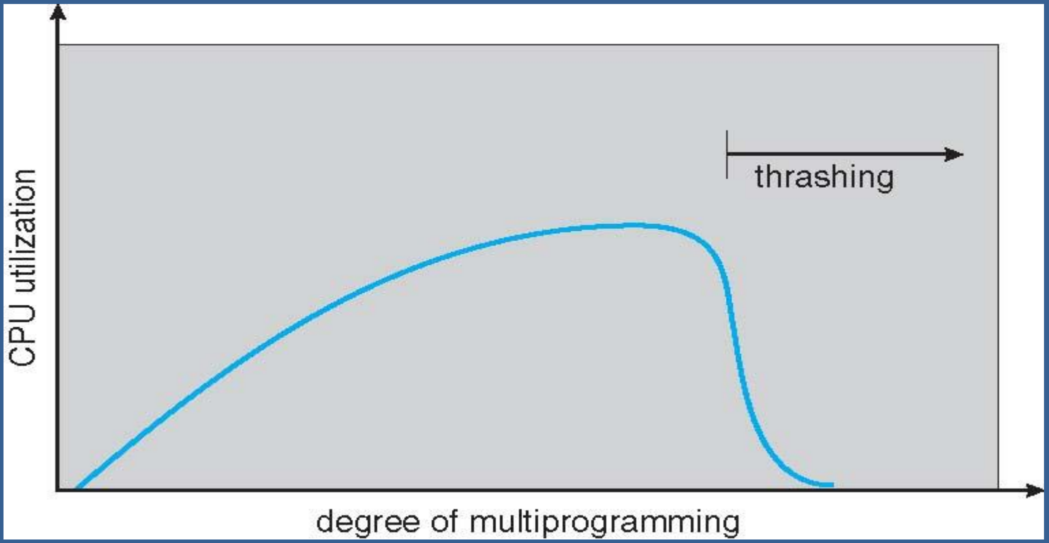
- 12 faults – better than FIFO but worse than OPT

LRU APPROXIMATION ALGORITHMS

- **Counter implementation**
  - Every page entry has a counter; every time page is referenced through this entry, copy the clock into the counter
  - When a page needs to be changed, look at the counters to find smallest value

THRASHING

- **Thrashing** is a condition in which excessive paging operations are taking place. A system that is **thrashing** can be perceived as either a very slow system or one that has come to a halt.
- This leads to low CPU utilization



OPERATING SYSTEM : EXAMPLES

- **Windows :** Its widely used O.S. Its associated with Microsoft corp

  Inbuilt browser : Internet explorer

  First version : windows1.0

  Latest version: windows10

  MICROSOFT EDGE browser use in window **10**.

OPERATING SYSTEM : EXAMPLES

- **Macintosh:** Mac O.S. is associated with Apple corporation.

  First version :1.0

  Latest version: Yosemite(10.10)

  Inbuilt browser: Safari

  Different versions of MAC: (Cheetah, puma, Jaguar, panther, Tiger, Leopard, Snow leopard, Lion, Mountain lion, Mavericks)

OPERATING SYSTEM : EXAMPLES

Linux:

Linus Torvalds has developed Linux O.S.

It is not associated with any company .

It is an open source software

First version :1.0

Examples: Red hat, Fedora, Ubuntu etc.

OPERATING SYSTEM : EXAMPLES

UNIX:

- UNICS(Uniplexed information and computing system)
- It is written in ‘c’ programming language.
- UNIX is multi user OS contains shell Programming.

OPERATING SYSTEM : EXAMPLES

- **Android :** It is associated with Google corporation.

First version: Alpha(1.0)

Latest version: Marshmallow(6.0)

Different versions of Android:
Alpha(1.0),Cupcake, Donut, Eclair, Froyo, Jellybean, Lollipop(5.2),Marshmallow(6.0)

Thank You
shift
{
[
}
]
4
\