Introduction to Operating Systems

1. **Basic Terminology & Definitions**
   System, kernel, shell, program, hardware, software, system software, user applications, elements of computing system, operating systems, main memory, secondary memory, system call, multiprogramming, multiprocessing, timesharing, uniprocessor system, multiprocessor systems, file, process, interrupt, batch processing systems

2. **Operating System Objectives and functions**
3. **Operating system structure**
   Monolithic and layered structure, microkernel architecture, UNIX system architecture, block diagram of system kernel, OS design hierarchy, General structure of OS control tables, characteristics of modern OS, SMP

4. **Summary of System calls**
   System calls for process management, device management, file management, time management, signaling, protection

swatimali@somaiya.edu

1. *Basic terminology and definitions*

Q. What is System, kernel, shell with reference to Operating System

Ans:
System: A system is a collection of various components interacting with each other to achieve a specific goal. In a Computer System, Hardware and Software work together for solving the task assigned by user.

Kernel :- Kernel is a fundamental part of operating system that is loaded into primary memory on starting up the computer. This core part of Operating System has control over operations of the computer. This manages the functioning of input & output requests from application software.

Shell: Shell command interpreter of the Operating system (OS). It interacts with OS by invoking System Calls.

Q. What is the role of hardware & software in a computing system?

Ans: Hardware includes the CPU, Memory and other communication Devices while the software includes the programs for using these devices.

Q. Differentiate between system software & application software

Ans:
System Software: System software can be defined as a program that runs computer hardware and software.

- Examples: Operating System, BIOS, firmware
- Special Types of System Software: Translators, Compilers, DBMS Programs, Other Diagnostic tools
- Operates at the lowest computer level
- Manages & Operates Hardware
- Interface between Application Software and Hardware
- Synchronizes & controls data flow between memory, secondary storage device, Display, printers etc.

Application software: Application softwares are defined as the softwares those run on system software to serve end users.

- Examples: Animation, Graphics, Microsoft Office, etc.
- Developed to perform single or multiple tasks

swatimali@somaiya.edu

Q. What are the elements of computing system?

Ans:- A Computing System Consists of -

Processor: For execution of instructions.
Memory Devices: For staring data and instructions (programs)
Input /Output /Communication Devices: Used for communicating with the real world. Receiving the date from the user, presenting data to the user or communicating with other devices.

Q. What is an operating systems

Ans: Operating System is collection of software that controls hardware for scheduling the execution of other programs, mange storage; input, output & Communication Devices and the CPU.

Q. **Compare main memory and secondary memory.**

Ans: Main Memory is the CMOS/ RAM in the Computer System. The program must be loaded in main memory to get executed along with it's data. Typical size of primary memory these days is 4 GB.

Secondary memory is typically hard disks in the computer. The OS partially resides in secondary memory too. Typical sizes of contemporary hard disk is 250GB

Q. What is system call?

Ans: Means of invoking Operating System functions is called System Calls. They are used for Process Management, File Management, Time Management, Memory and I/O Management.

Q. Differentiate between multiprogramming Timesharing , multiprocessing
Ans:
Multiprogramming & Timesharing:
The ability of OS to run multiple programs at a time is called multiprogramming. The time-sharing is the concept that helps the OS to achieve multiprogramming.

When CPU time is shared by multiple processes on the same processor it is called time sharing. With the help of timesharing, multiple processes are executed on same processor giving pseudo impression to the user of executing many processes simultaneously.

Multiprocessing :
The ability of OS to run multiple programs on multiple processors concurrently is called multiprocessing.

Swatimali@somaiya.edu

Multiprogramming can be done on one or more processors but multiprocessing can be done only on multiple processors.

Q. Define Uniprocessor system and multiprocessor systems giving example.
Ans:

Uniprocessor systems:- When only one processor exists for execution of processes, the computer system is termed as Uniprocessor System. The speed of operation is slower as compared to multiprocessing as each processes gets a time slice and time is also spent in context switching.
Multiprocessor systems:- In Multiprocessor environment many processors executes the processes in synchronization with each other, thus reducing the time for execution.

Q. Define interrupt.

Ans: Interrupt is a request by any device for service. The device needing resources send signal to the processor which intern directs operating system to perform the requested task.
The interrupts could be hardware or software interrupts.

Q. What are batch processing systems?

Ans: The batch operating system maintains files that contains commands to be executed in a sequence. Typically these commands do not need human intervention to complete their execution.
The batch processing is classical OS characteristic, but in modern days too, the periodic executions are scheduled as batch commands.

swatimali@somaiya.edu

Q. what are the operating system objectives?

Ans: Operating systems has got three main objectives:-

1. Convenience:- The OS must make the computer convenient to use.
2. Efficiency:- The OS must make efficient use of computing resources.
3. Ability to evolve:- The OS must be designed and implemented in such a way as to provide some means to introduce new functionalities without interfering with OS services.

Q. What are the functions carried out by an Operating Systems?

Ans: the operating systems provides following main functionalities-

1. The OS functions as User/Computer interface
2. The OS functions as resource manager

Q. What are the issues handled by OS as resource manager

Ans: The operating system is responsible for managing all the computing resources . These resources combinely store, process and move the data.

1. Managing Processor:- The OS functions are executed by processor and also the user programs. To manage the balance, OS gives control to processor and then depends on OS to execute instructions which give control back to OS. This way, processor keeps on executing interleaved sequence of user programs and OS programs. The OS decides how much CPU time should be awarded to execution of user program. If the system has more than one processor, then the OS takes decision for all processors. There are more than one policies those decide which processor should run OS in case of multiprocessor system.

2. Managing Memory: The main part of OS that resides in main memory is kernel; rest OS resides in secondary memory. The remaining part of Main memory contains user data, user programs and the OS programs that are invoked by kernel to process user requests. The allocation of main memory to various user and system programs is managed by operating system functions for memory management and memory management hardware in the processor.

3. Managing I/O:- The OS decides which program to execute next, and if the program needs to use I/O then its OS that decides when the program gets that access. Also, if required, the OS can take the I/O from the program before it completes execution.

swajmal@somaiya.edu

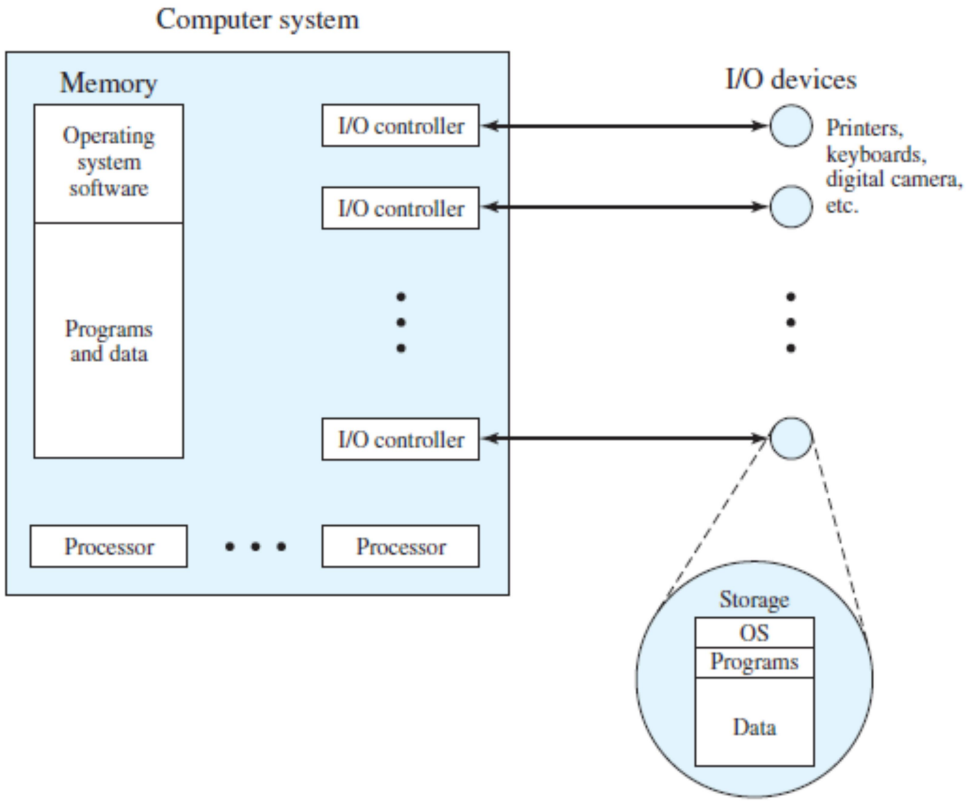

Figure 2.2 The Operating System as Resource Manager

Image courtesy: "Operating systems", William Stallings, Pearson education, 6th edition.

The operating system manages all these resources by using time and space multiplexing concepts.

swatimali@

3. *Operating system structure*

Q. What are different OS structures?
Ans: The OS has had following structure since inception it's concept

1. Monolithic: The OS was developed as a big program which contained functions written for OS functionalities. All the functions could call each other and modification in any module would require recompiling of whole OS and reinstallation of the same. There was no concept of data hiding , encapsulation or protecting data from any function. The modification in one module had chances of introducing errors in other modules.

   Drawbacks:

   1. The entire OS used to reside in main memory and thereby it left very less space for user programs.
   2. In this structure all the functions are executed in kernel mode.

2. Layered: The operating system functions were divided into modules which were arranged in form of layers. In this structure, only the required layer was modified and other were not affected with the same. The interfaces between any two layers were clearly defined and minimal information could flow from one module to another. Modification in one layer required compilation of the only that module.

   Drawbacks:

   1. The information had to pass through all the layers sequentially even if all the layers wouldn't need access to the same.
   2. Also, The entire OS used to reside in main memory and thereby it left very less space for user programs.
   3. In this structure most the functions are executed in kernel mode.

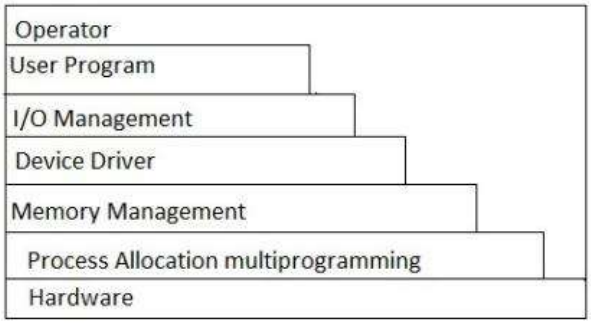

fig:- layered Architecture

Image courtesy: http://www.sciencehq.com/wp-content/uploads/layered-architecture.jpg

svatumali@sriaya.edu

3. Microkernel architecture: This is modern operating system architectures.

   Here the entire OS need not reside in main memory and thereby it leaves more space for user programs. The OS partially resides in main memory and secondary memory. The part of OS that always resides in main memory is called kernel and rest of the part residing in secondary memory is invoked by special managers in kernel as and when needed.

   In this structure, Most of the functions run in user mode and only high privileged operations are executed in kernel mode.

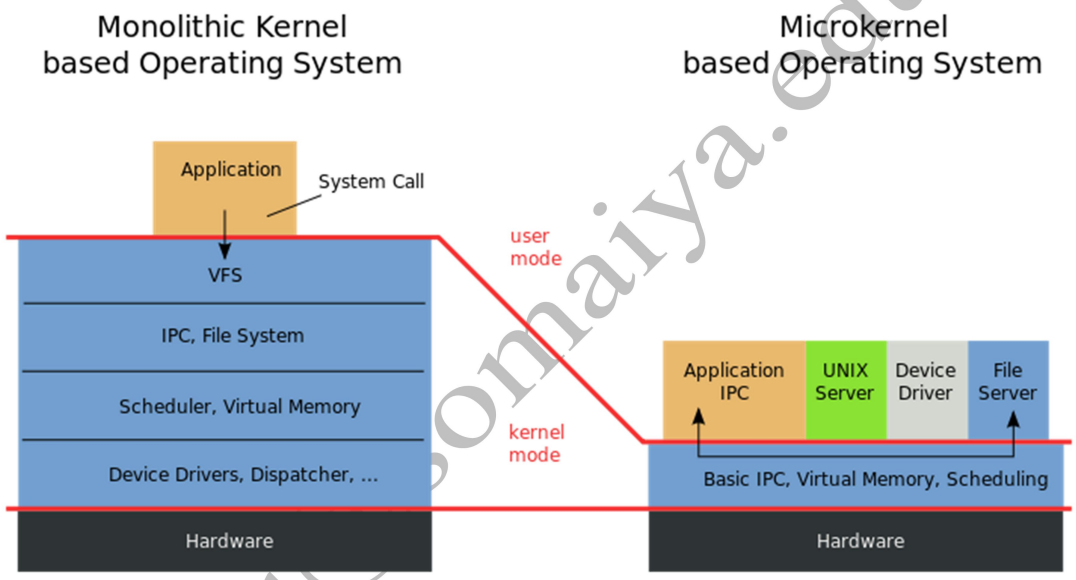

Image courtesy: https://en.wikipedia.org/wiki/Microkernel

swatimalik@somaiya.edu

Q. Explain the Unix system architecture

Ans: The Operating system works as an interface between hardware and user by providing common services to programs and hiding complexities from the user. The OS provides services as kernel and the utilities and user programs have clear separation from the OS. The programs like shell and various editors interact with OS by invoking different system calls.

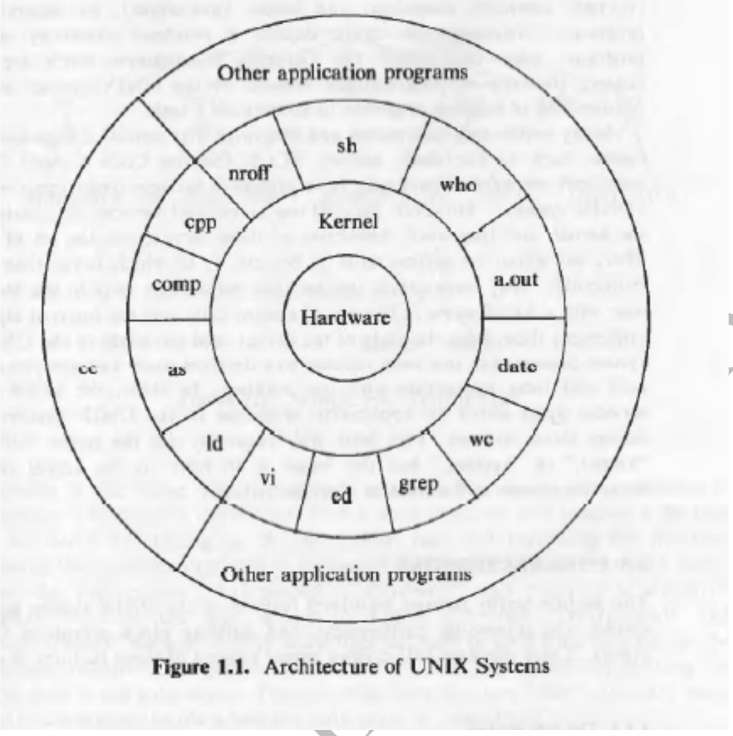

Figure 1.1. Architecture of UNIX Systems

Image courtesy: "The design of Unix operating system" Maurice j Bach , PHI

Q. Draw and explain Block diagram of OS system Kernel

Ans: The operating system is typically divided into three main layers: hardware level, kernel level and user level.

- Hardware level: The hardware level handles interrupts and low level communication for the machine. The interrupts are processed by special kernel functions which are invoked in the context of the process currently being executed by the processor.

- Kernel level: the kernel level contains the hardware control device drivers those manage the different hardware devices. The devices also include I/O devices connected to system. Generally these devices read and write data into systems which are saved in terms of files. The device drivers could be serial or block drivers. The block drivers have access to buffer cache module that caches the frequently/recently used information.

swayam...ya.edu

The processes those access memory/files are managed by process control subsystem. This subsystem manage communication amongst processes, scheduling them on the processors and manage memory allocated to the processes.

All the high privileged processes are executed in kernel mode.

- User level: User can invoke the system functions through system calls. The system has predefined system calls for file, time, device, memory, communication, security, i.e. all kinds of managements. System calls access file system and process control subsystem on user's behalf.

  The users can also develop programs using system defined libraries which in turn can invoke system calls to execute what user program intends to. In short, one has to execute system calls directly or indirectly to use the computing machine .

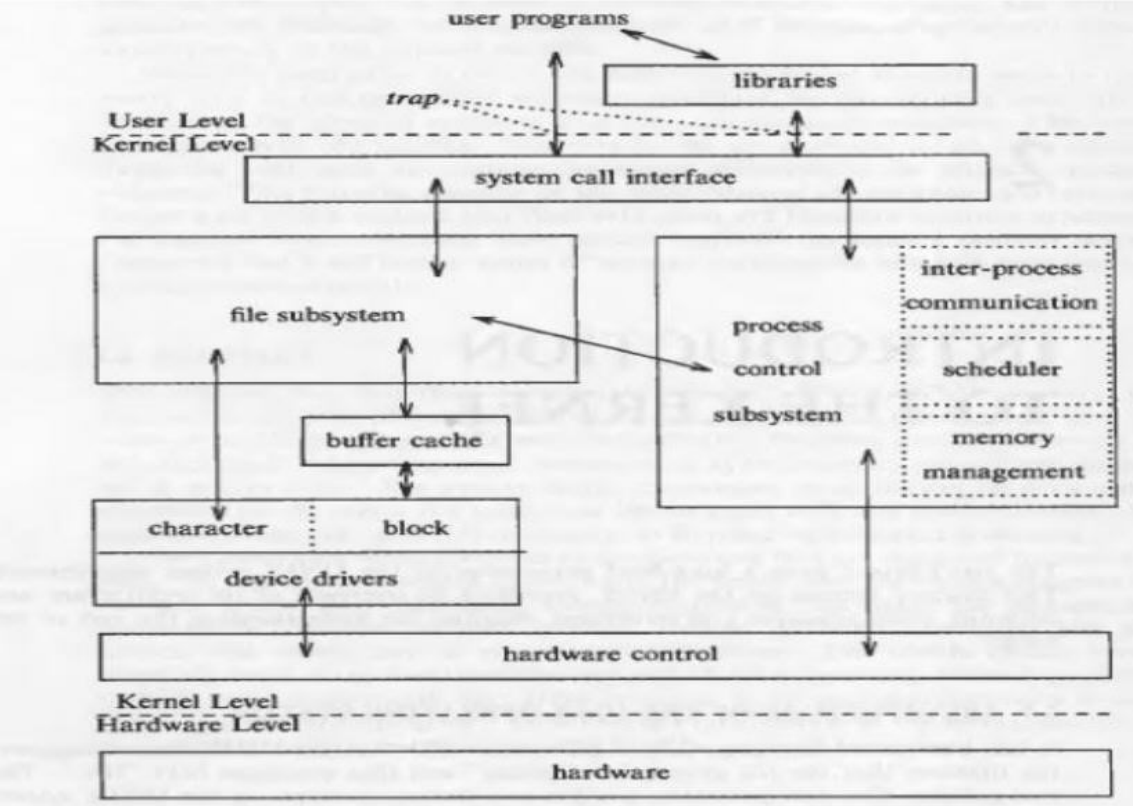

Figure 2.1. Block Diagram of the System Kernel

Image courtesy: "The design of Modern Operating System", Maurice Bach, PHI, 2nd edition.

Q. Explain the OS design hierarchy

Level	Name	Objects	Example Operations
13	Shell	User programming environment	Statements in shell language
12	User processes	User processes	Quit, kill, suspend, resume
11	Directories	Directories	Create, destroy, attach, detach, search, list
10	Devices	External devices, such as printers, displays, and keyboards	Open, close, read, write
9	File system	Files	Create, destroy, open, close, read, write
8	Communications	Pipes	Create, destroy, open, close, read, write
7	Virtual memory	Segments, pages	Read, write, fetch
6	Local secondary store	Blocks of data, device channels	Read, write, allocate, free
5	Primitive processes	Primitive processes, semaphores, ready list	Suspend, resume, wait, signal
4	Interrupts	Interrupt-handling programs	Invoke, mask, unmask, retry
3	Procedures	Procedures, call stack, display	Mark stack, call, return
2	Instruction set	Evaluation stack, microprogram interpreter, scalar and array data	Load, store, add, subtract, branch
1	Electronic circuits	Registers, gates, buses, etc.	Clear, transfer, activate, complement

Gray shaded area represents hardware.

Image courtesy: “Operating systems”, William Stallings, Pearson education, 6th edition

The operating system is interface between hardware and user. To do so, it has been divided into modules those run OS resource management programs and also serve the user requests. The above figure explains the OS’s division into different layers.

Q. What are the different control tables that OS maintains?

Ans: The Operating systems maintains four main tables such as,

- Memory table
- File table
- I/O table
- Process table

Q. General structure of OS control tables

Ans:

swatinali@s

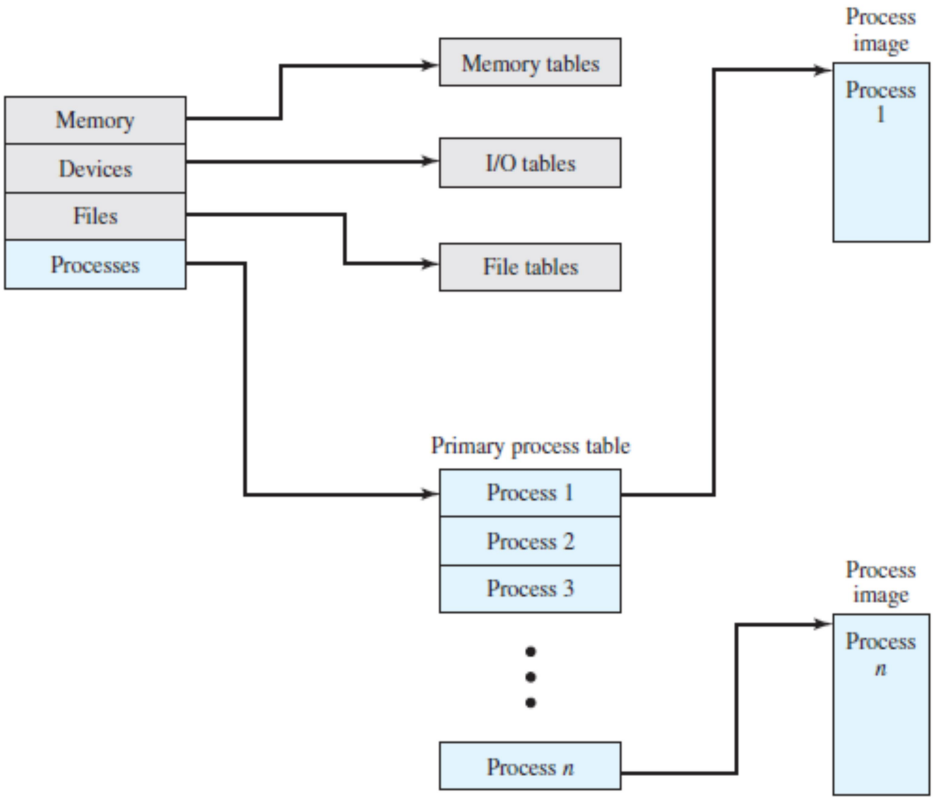

Figure 3.11 General Structure of Operating System Control Tables

Image courtesy: "Operating systems", William Stallings, Pearson education, 6th edition

Q. What are the characteristics of modern OS?

Ans:-

The modern operating system is identified with following four characteristics.

1. Microkernel architecture: Microkernel architecture has come up with minimal amount of modules that constitute microkernel, thereby has left more space for the user programs. The rest of the modules are loaded into the main memory as and when needed.
2. Symmetric multiprocessing: The system contains multiple processors. These processors in system can execute any process, there no processor exclusively dedicated to OS. This way, the available process load can be distributed well amongst all the processors thereby achieving efficiency, availability and provision for incremental growth. All processors share the same main memory and I/O subsystems which are connected very well with each other.
3. Distributed OS: The distrubted OS gives the illusion of single main memory, secondary memory space and file system, too. .
4. Multithreading:- The modern OS supports multithreading thereby striking the perfect balance between processes and threads. The multiple

swatimali@

processes can run multiple threads from multiple processes in parallel. Threads of one process too can run concurrently.

5. Object oriented design : Object oriented design helps in building the modular extensions to micro kernel. The object oriented feature helps in system integrity and modularity .

swatimali@somaiya.edu

4. System calls

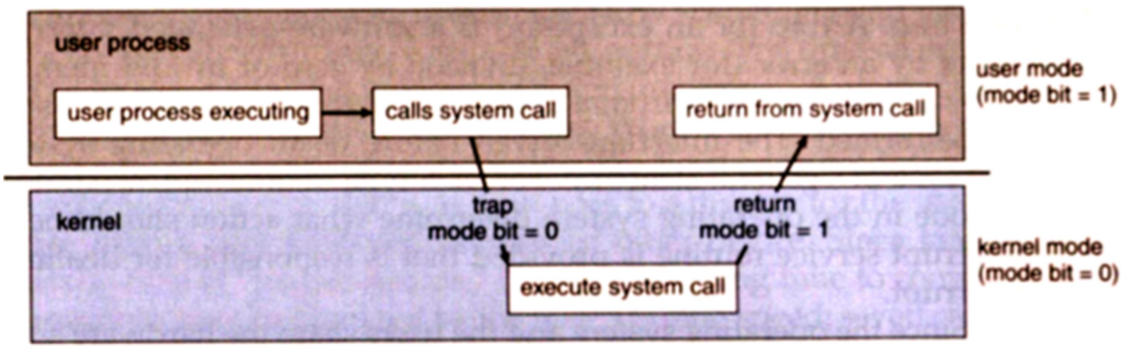

Image courtesy:http://faculty.salina.k-state.edu/tim/ossg/Introduction/sys_calls.html

Q. List System calls for process management

Ans:- Following are the various system calls those are used for process management.

- fork ()- to create a new process
- exec() – to execute a process
- wait () - to make the process to wait
- exit () - to terminate the process
- getpid()-To find the unique process id
- getppid () - To find the parent process id
- nice () - To change currently running process's priority

Q. List system calls for file management

- Create()- to create a new file
- Open()- to open an existing file
- Read()- to read file contents
- Write()- to write into a file
- Close()- to close an open file
- Seek()- to change the file offset position while reading or writing
- *Get()* and *set() – retrieve and set* file attributes.

Q. List system calls for device management,

- Open()- To open an I/O connection.
- Close()- To close the connection with associated device.
- Read() – to read data from device associated with opened device id.
- Write()- to write into the device associated with opened device id.

swatimalomaiya.com