Introduction to Operating System :- Glavin

Topics :-
- What is OS
- OS Duties | Function | Services | Features
- What is Kernel? Different OS Structure Type & Kernel Architecture?
- Types Of OS
- Spooling
- System Call
- What is Booting & What happens when we turn on computer? BIOS VS UEFI | MBR vs GPT
- RAM Vs ROM

What is OS?
- OS is system program that acts as interface between user and computer hardware which controls execution of all kinds of programs and manages all resources of computer system
- First OS is ATLAS developed by Tom Kilburn professoor at Manchester University in 1962

  Primary Goals :- Windows | Mac : Machintosh
  - Convenience : Use Easiet way to any hardware
  - Throughput : Number of tasks executed per unit time (Linux)

OS Duties | Function | Services | Features :-

Resource Management | Resource Governance:-
- Manage all kinds of resources both hardware and software using Schedulers to achieve parallel processing

Process Management | Process Execution :-
- Load program into memory, execute it in CPU and Handle it using various CPU Scheduling
- Also Provide mechanism for process synchronization, IPC and Deadlock handling

Storage/File management or Manipulation:-
- Support Storage allocation, Disk scheduling using File System and Manage all file operations based on permission mode

Memory management :-
- Keep track of memory to identify which part being used by which Job(Process)
- Allocating and deallocating memory space of Primary(Main) Memory because size limits

Protection, Security & Privacy:-
- OS controls all access to system resources to Ensure that external I/O devices are protected from invalid access attempt
- Windows use Kerberos Security Protocol

I/O Device Management :-
- Keep tracks of all devices and Manage device communication between User and Device via their respective drivers
- Program that responsible for this task is known as I/O controller.

Communication :-
- Communication may be implemented either by Shared Memory or by Message Passing
- Handle routing, connection strategies
- Processes can be on one computer or on different computers, but they are connected through computer network

Error Handling | Error detection :-
- OS constantly checks errors and takes appropriate action to ensure correct and consistent computing

What is Kernel? Different OS Structure Type & Kernel Architecture :-
- Acts as bridge between applications and data processing which performed at hardware level using system call

Objectives of Kernel :
- Establish communication between user level application and hardware
- Decide state of incoming processes
- To control disk management, memory management, task management

Monolithic Architecture :-
- Entire OS working in kernel space so execution is fast and process run completely in single address space
- It is static single binary file
- Kernel provides various services using system calls
- If single service fails its lead to entire system failure and to add new services entire architecture needs to modify

Example :- MS DOS, Unix, Linux, Open VMS, XTS-400

Layered Architecture :-
- OS divide into number of layers where every layer has different functionality
- Bottom layer (layer 0) is hardware and topmost layer (layer N) is user interface
- Each layer can only uses function of lower level layer
- Change in one layer specification does not affect rest of layers
- Layer 1 to 6:- Hardware, CPU Scheduling, Memory Management, Process Management, I/O Buffer, User Program

Example :- Windows NT

Microkernel Architecture :-
- Kernel is broken down into separate processes, known as servers
- Some servers run in kernel space and other in user-space
- Servers invoke "services" from each other by sending messages via IPC
- If one server fails, other servers can still work efficiently.
- New services added in user space and does not require kernel to be modified which increased security and stability
- All kernel components live in a common shared address space Therefore no protection between OS and drivers

Example :- MacOS, Windows NT, Mach, L4, AmigaOS, Minix, K42

Modular Architecture :-
- System parts allocated as independent files called modules added to system at run time
- Requires small amount of time for load modules

Hybrid Kernel :-
- Combination of both monolithic kernel and mircrokernel which use speed and design of monolithic, modularity and stability of microkernel

Example :- Windows NT, Netware, BeOS

Types of OS :-
Single process :- MS DOS 1981
Multiprogramming :- Dijkstra 1960
Multitasking :- CTSS, MIT 1960

Batch Operating System | Batch Processing :-
- Set Similar kind of process(job) in same batch using FCFS manner for execution
- When Job request for I/O at time CPU is Idle which is major disadvantages(Non-Preemption)
- IBM create FORTRAN Alone with IBSYS 709X in 1960

Example :-
User ---> Set Job in ROM device ---> Go to operator
---> Operator group similar job and execute
---> One time one process execute (No particular time for output)

```
   ---> Output is write to disk and get to user from operator
   ---> User use output in own system for process



Multiprogramming :- IDLENESS
 - Increases CPU utilization by keeping multiple jobs(process) in main memory so that
   CPU always has one to execute
 - NonPre-emptive scheduling means if we give one process to cpu, cpu will complete that
   entire process and only after complete process pick another process
 - Sometime cpu will go for I/O operation at that time cpu will pick up another process
   from job pull So cpu IDLE time is zero



Multitasking | Time-Sharing :- Responsiveness
 - Sharing small portion of time or time quantum among multiple processes by CPU Scheduling
 - CPU executes multiple process using switching which Reduce CPU IDLE time
 - It is pre-emptive means Each process has given particular time in which it complete
   execution otherwise it will be switching(Context Switch) to another process by
   CPU Scheduling but process still available if not complete in within given time
 - Fast switching Security arise with program data, data communication and its reliability
 - Response time should be minimal



MultiProcessing :-
 - In uni-processor system, only one process can execute at time but in multi-processing
   multiple processes can execute at same time
 - Computer have more than one processor its called multiprocessing so multiple processe
   executed simultaneously to achieve Parallel processing
 - Multiple processors share computer bus, sometimes clock, memory and peripheral devices
 - Used to get more work done in shorter period of time,  increased reliability



Multi-Threading :-
 - Single process have multiple code segments(threads) so it can running concurrently
   within "Context" of that process
 - Increase Responsiveness, less costly


Real Time operating System(RTOS):-
 - Fixed time Constraint system where time constraint matter lot and can't be delays
 - Time taken by system to respond input and display required updated information
   is termed as response time
 - VxWorks, QNX & RTLinux are real-time operating system
 - Palm OS is mobile operating system for Personal Digital Assistant(PDA)
 Example :- Embedded systems, Robotics, Scientific utilities, etc.

 Types :-
  Hard-Real Time System :- Critical tasks needs to complete on time there no place for delays
  Soft-Real Time System :- Less restrictive with limited utility than hard-real time system


Distributed Operating System | Cluster :-
 - Multiple Processors serve Multiple applications to multiple users using Central Processor
 - Processor communicate with each other through various communication lines like network
   connection which referred as loosely coupled
 - OS distribute computation logics among several physical processors
 - Processors do not share memory instead each processor has its own local memory, storage
 - Reduction of delays in processing and loading on host computer
 - Availability, Load balancing, Scalability
 - If Processors connect with local network (Single network) like Single server it is
   called "Cluster Operating System"


Embedded Systems :-
 - Work on fixed machine like A.C.
 - You can't customally change its functionality
```

Network Operating System :-
- Runs on server computer and provides server capability to manage application, user data and other networking functionality
- Purpose to allow shared file and printer access among multiple computers in network typically local area network (LAN), private network or to other networks.
- Centeralized Server so security managed and software upgrade easily and enable remote access
- Dependency central server, High cost, Regular maintenance and update are some disadvantages

Spooling(Simultaneous Peripheral Operations Online) :-
- Different Pheripheral devices(I/O Device) can run simultaneously
- Provide buffer area in secondary memory to save CPU time and user not need to wait for I/O

System Call :-
- To use system functionality we need to go in kernel mode because we only can access user mode
- System call is mechanism using which user program request kernel to access all system functionalities

Type of System Call :-
E.x. :- Fork, exec, getpid, getppid, wait, exit

File	:- Open, read, write, close, create file
Device	:- Read, write, Repoisition, iocte(manipulates underlying device parameter of special files) fcnte (access point for several advanced operations on file descriptors)
Security	:- chmod(Chagen Privilege Mode), umask
Process Control	:- load, abort, execute, fork, wait, signal, allocate
Information	:- getpid, attribute, get system data
Inter-Process Communication	:- Pipe, shmget(Get Shared Memory), create/delete connection

Fork :-
- In UNIX Fork() Create child process of parent process
- Returns process ID which is generally process id of child process
- Return value 0 :- Child process , 1 :- Parent process, -1 : if child process not created
- Formula :- 2N-1 (Parent process) | 2N (Child process)

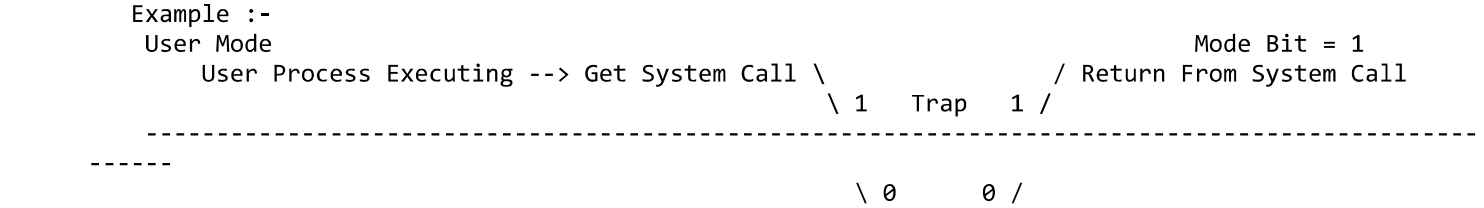
Exec :- exec replace current process with another (different) one

Wait :-
- Used by parent process to determine termination of child process
- Parent process uses Wait() system call and gets exit status of child process as well as PID of child process which is terminated

Dual-mode operation :-
- User mode
- Kernel mode | supervisor mode | system mode | privileged mode

Mode bit :- 0 : Kernel, 1 : User

Example :-



What is Booting & What happens when we turn on computer? BIOS VS UEFI | MBR vs GPT :-
- Booting is process of loading OS
- When computer power is turned on then cpu will move to BIOS which stored in EPROM to perform start-up procedures

- BIOS run POST(power-on self test) process which check all hardware if any error beeps are hered which known as POST beep codes
- BIOSes following ACPI(Advanced Configuration and Power Interface) create tables that describing devices and manage power in computer
- On POST process successful BIOS load MBR(Master Boot Record) to RAM
- MBR load bootstrap loader which loads OS, Kernel and much more in RAM
- init is last step of kernel boot sequence which decied inital state of OS
- In Linux Bootloader called GRUB(GRand Unified Bootloader) OR LLIO(Linux Loader-default)

Types of booting :-
Hard Booting - Power On system
Soft Booting - Restart / Ctrl + Alt + Del twice

BIOS VS UEFI :-
- Both boot pc before booting OS but UEFI is modern solution
- CMOS(Complementary Metal-Oxide-Semiconductor) used for battery-backed memory where BIOS stores various setting on motherboard
- It’s replaced with flash memory (also referred to as EEPROM)
- MS-DOS PCs released in 1980s had a BIOS
- BIOS has limited functionality like boot upto 2.1 TB, run in 16bit processor mode and has 1 MB of space to execute
- In 2007 PC Manufacturer agreed for UEFI
- UEFI Firmware can boot from drive 2.2 TB Or larger, and use GPT Partition Scheme
- UEFI can run in 32-bit or 64-bit processor mode and has more address space to execute

MBR VS GPT(GUID Partition Table) :-
- Partition structure defines how information is structured on partition and also code that is used during startup if partition is bootable.
- GPT is newer standard with many advantages while MBR for compatibility
- MBR introduced by IBM in 1983 which located in beginning of drive that contains bootloader for installed OS and drive logical information
- If linux OS than GRUB Bootloader located in MBR
- MBR works with upto 2TB and supports upto 4 primary partitions
- GPT associated with UEFI and GUID Partition Table provide every partition on your drive has globally unique identifier
- GPT allows upto 128 partitions, stores multiple copices of bootloader
- Provide cyclic redundancy check(CRC) to check data is intact, if data corrupted, GPT attempt to recover data

NTFS VS ETL VS FAT32 VS exFAT :-
- NTFS(New Technology File System) use with windows by default
- EFL(Extensible Firmware Interface) usually harddisk drive or solid-state drive that is used by computers having UEFI
- FAT(File Allocation Table 32)
- exFAT(Extended File Allocation Table)

RAM(Random Access Memory) Vs ROM (Read-Only Memory):-
Ram :-
- Volatile, Faster, loading OS and executing program
- For 32-bit Max Ram size :- 4 GB AND 64 : 16GB

Types :-
1)DRAM - Dynamic RAM must be continuously refreshed otherwise all contents are lost
2)SRAM - Static RAM faster, need less power, more expensive but need refreshed like DRAM
3)SDRAM - Synchronous Dynamic RAM run at very high clock speeds
4)DDR – Double Data Rate Provide synchronous Access Memory

ROM :-
- Non-volatile so Write once read any no. of time
- Used in start-up process of computer

Types :-
1)PROM :- Programmable ROM type of ROM is written or programmed only once
2)EPROM :- Erasable Programmable ROM reprogrammable using ultraviolet light
3)EEPROM :- Electrically Erasable Programmable ROM using electric charge(Flash Memory)
4)MASKROM :- Contents can be programmed only by an integrated circuit manufacturer