System Calls

- Software interrupt is used to implement OS/Kernel services.
- Functions exposed by the kernel so that user programs can access kernel functionalities, are called as "System calls".
    - e.g. Process Mgmt: create process, exit process, communication, synchronization, etc.
    - e.g. File Mgmt: create file, write file, read file, close file, etc.
    - e.g. Memory Mgmt: alloc memory, release memory, etc.
    - e.g. CPU Scheduling: Change process priroty, change process CPU affinity, etc.
- System calls are specific to the OS:
    - UNIX: 64 syscalls e.g. fork(), ..
    - Linux: 300+ syscalls e.g. fork(), clone(), ...
    - Windows: 3000+ syscalls e.g. CreateProcess(), ...

Classification of OS

- OS can be categorized based on the target system (computers).
    - Mainframe systems
    - Desktop systems
    - Multi-processor (Parallel) systems
    - Distributed systems
    - Hand-held systems
    - Real-time systems

Mainframe systems

Resident Monitor

- Early (oldest) OS resides in memory and monitor execution of the programs. If it fails, error is reported.
- OS provides hardware interfacing that can be reused by all the programs.

Batch Systems

- The batch/group of similar programs is loaded in the computer, from which OS loads one program in the memory and execute it. The programs are executed one after another.
- In this case, if any process is performing IO, CPU will wait for that process and hence not utilized efficiently.

Multi-Programming

- In multi-programming systems, multiple program can be loaded in the memory.
- The number of program that can be loaded in the memory at the same time, is called as "degree of multi-programming".
- In these systems, if one of the process is performing IO, CPU can continue execution of another program. This will increase CPU utilization.
- Each process will spend some time for CPU computation (CPU burst) and some time for IO (IO burst).
    - If CPU burst > IO burst, then process is called as "CPU bound".
    - If IO burst > CPU burst, then process is called as "IO bound".

- To efficiently utilize CPU, a good mix of CPU bound and IO bound processes should be loaded into memory. This task is performed by an unit of OS called as "Job scheduler" OR "Long term scheduler".
- If multiple programs are loaded into the RAM by job scheduler, then one of process need to be executed (dispatched) on the CPU. This selection is done by another unit of OS called as "CPU scheduler" OR "Short term scheduler".

Multi-tasking OR time-sharing

- CPU time is shared amoung multiple processes in the main memory is called as "multi-tasking".
- In such system, a small amount of CPU time is given to each process repeatedly, so that response time for any process < 1 sec.
- With this mechanism, multiple tasks (ready for execution) can execute concurrently.
- There are two types of multi-tasking:
    - Process based multitasking: Multiple independent processes are executing concurrently. Processes running on multiple processors called as "multi-processing".
    - Thread based multi-tasking OR multi-threading: Multiple parts/functions in a process are executing concurrently.

Multi-user

- Multiple users can execute multiple tasks concurrently on the same systems. e.g. IBM 360, UNIX, Windows Servers, etc.
- Each user can access system via different terminal.
- There are many UNIX commands to track users and terminals.
    - tty, who, who am i, whoami, w

Desktop systems

- Personal computers -- desktop and laptops
- User convinence and Responsiveness
- Examples: Windows, Mac, Linux, few UNIX, ...

Multiprocessor systems

- The systems in which multiple processors are connected in a close circuit is called as "multiprocessor computer".
- The programs/OS take advantage of multiple processors in the computer are called as "Multiprocssing" programs/OS.
    - Windows Vista: First Windows OS designed for multi-processing.
    - Linux 2.5+: Linux started supporting multi-processing.
- Modern PC architectures are multi-core arch i.e. multiple CPUs on single chip.
- Since multiple tasks can be exeuted on these processors simultaneously, such systems are also called as "parallel systems".
- Parallel systems have more throughput (Number of tasks done in unit time).
- There are two types of multiprocessor systems:
    - Asymmetric Multi-processing
    - Symmetric Multi-processing

Asymmetric Multi-processing

- OS treats one of the processor as master processor and schedule task for it. The task is in turn divided into smaller tasks and get them done from other processors.

Symmetric Multi-processing

- OS considers all processors at same level and schedule tasks on each processor individually.
- All modern desktop systems are SMP.

Distributed systems

- Multiple computers connected together in a close network is called as "distributed system".
- Its advantages are high availability (24x7), high scalability (many clients, huge data), fault tolerance (any computer may fail).
- The requests are redirected to the computer having less load using "load balancing" techniques.
- The set of computers connected together for a certain task is called as "cluster". Examples: Linux.

Handheld systems

- OS installed on handheld devices like mobiles, PDAs, iPODs, etc.
- Challenges:
    - Small screen size
    - Low end processors
    - Less RAM size
    - Battery powered
- Examples: Symbian, iOS, Linux, PalmOS, WindowsCE, etc.

Realtime systems

- The OS in which accuracy of results depends on accuracy of the computation as well as time duration in which results are produced, is called as "RTOS".
- If results are not produced within certain time (deadline), catastrophic effects may occur.
- These OS ensure that tasks will be completed in a definite time duration.
- Time from the arrival of interrupt till begin handling of the interrupt is called as "Interrupt Latency".
- RTOS have very small and fixed interrupt latencies.
- RTOS Examples: uC-OS, VxWorks, pSOS, RTLinux, FreeRTOS, etc.

Linux File Structure

- Linux follows “/” (root) file system.
- “/” is a starting point of Linux file system.
- All your data is stored in this partition.
- / contains boot, bin, sbin, etc, root, home, dev, proc, mnt, media, opt
- In Linux everything is a file.
- Mainly there are two types of files in Linux
    - File
    - Directory (Folder)
- Linux Directories

- boot - files related to booting
  - vmlinuz - kernel Image
  - grub - boot loader
  - config - kernel configuration
  - initrd/initramfs - initail root file system
- bin - user commands in binary format
- sbin - all admin/system commands in binary format
- etc - configuration files
- root - home directory of root user
- home - it contains sub directories for each user with its name
  - devendra -> /home/devendra
  - sunbeam -> /home/sunbeam
  - osboxes -> /home/osboxes
- dev - it contains all device related files
- lib - shared program libraries required by kernel
- mnt - it is temporary mount point
- media - it is mount point for media eg cdrom
- opt - stores optional files of large softwares
- proc - virtual file system - it contains information about system or processes
- sys - entries of each block devices, subdirectories for each physical bus type supported, every device class registered with the kernel, global device hierarchy of all devices
- tmp - temporary files that may be lost on system shutdown
- usr - read only directory that stores small programs and files accessible to all users

Path

- It is a unique location of any file in the file system.
- It is represented by character strings with few delimiters (“/”, “\”, “:”)
- Types of path
  - There are two types of paths in linux
  - Absolute path
    - Path which starts with “/” is called as absolute path.
    - E.g. /home/devendra/MyData/Demos/demo01.sh
  - Relative path
    - Path with respect to current directory is called as relative path
    - E.g. MyData/Assignments/assign02.pdf

Types of files

- Regular file (-)
- Directory file (d)
- Link file (l)
- Socket file (s)
- Pipe file (p)

- Character Special file (c)
- Block Special file (b)