

Concept of Operating Systems & Administration

Concept of OS

- Architecture of Operating Systems
 - Introduction
 - Kernel Architecture
 - Interrupt Management
- Process Management
 - Process
 - Process Scheduling
 - CPU Scheduling (Preemptive and non preemptive)
 - Scheduling policies (Algorithm-FCFS, RR, PRIO, FAIR-SHARE, EDF)
- Memory & File System Management
 - Virtual Memory
 - Page Replacement Algorithm
 - Segmentation/ Paging
 - File System Organization
 - File System managers
 - Disk Management

Linux Operating System

- Introduction to Linux
- The Linux File System
- Working with Files and Directories
- Linux commands
- Shell Scripts

Learning OS

- step 1: End user
 - Linux commands
- step 2: Administrator
 - Install OS (Linux)
 - Configuration - Users, Networking, Storage, ...
 - Shell scripts
- step 3: Programmer
 - Linux System call programming
- step 4: Designer/Internals
 - UNIX & Linux internals

What is OS?

- Interface between end user and computer hardware.
- Interface between Programs and computer hardware.

- Control program that controls execution of all other programs.
- Resource manager/allocator that manage all hardware resources.
- Bootable CD/DVD = Core OS + Applications + Utilities
- Core OS = Kernel -- Performs all basic functions of OS.

OS Functions

- CPU scheduling
- Process Management
- Memory Management
- File & IO Management
- Hardware abstraction
- User interfacing
- Security & Protection
- Networking

Process Management

Program

- Set of instructions given to the computer --> Executable file.
- Program --> Sectioned binary --> "objdump" & "readelf".
 - Exe header --> Magic number, Address of entry-point function, Information about all sections. (objdump -h program.out)
 - Text --> Machine level code (objdump -S program.out)
 - Data --> Global and Static variables (Initialized)
 - BSS --> Global and Static variables (Uninitialized)
 - RoData --> String constants
 - Symbol Table --> Information about the symbols (Name, Size, section, Flags, Address) (objdump -t program.out)
- Program (Executable File) Format
 - Windows -- PE
 - Linux -- ELF
- Program are stored on disk (storage).

Process

- Program under execution
- Process execute in RAM.
- Process control block contains information about the process (required for the execution of process).
 - Process id
 - Exit status
 - 0 - Indicate successful execution
 - Non-zero - Indicate failure
 - Scheduling information (State, Priority, Sched algorithm, Time, ...)
 - Memory information (Base & Limit, Segment table, or Page table)
 - File information (Open files, Current directory, ...)
 - IPC information (Signals, ...)

- Execution context (Values of CPU registers)
- Kernel stack
- PCB is also called as process descriptor (PD), uarea (UNIX), or task_struct (Linux).
- In Linux size of task_struct is approx 4KB

User interfacing

- UI of OS is a program (Shell) that interface between End user and Kernel.
- Shell -- Command interpreter
 - End user --> Command --> Shell --> Kernel
- User interfacing (Shell)
 - Graphical User Interface (GUI)
 - Command Line Interface (CLI)

Example shells

- Windows
 - GUI shell: explorer.exe
 - CLI shell: cmd.exe, powershell.exe
- DOS
 - CLI shell: command.com
- Unix/Linux
 - CLI shell: bsh, "bash", ksh, csh, zsh, ...
 - ls /bin/*sh
 - echo \$SHELL
 - shell of current user can be changed using "chsh" command.
- GUI shell/standards
 - GNOME: GNU Network Object Model Environment (e.g. Ubuntu, Redhat, CentOS, ...)
 - KDE: Kommon Desktop Environment (e.g. Kubuntu, SuSE, ...)

Windows File Structure

- This PC
 - C:
 - Windows (System files & Utilities)
 - Windows kernel: ntoskrnl.exe
 - Windows drivers: *.sys, *.drv
 - Windows libraries: *.dll
 - taskmgr.exe
 - Program Files (User/Installed softwares)
 - Users
 - user1 (User profile)
 - Documents
 - Downloads
 - Desktop
 - Music
 - Videos
 - user2

- D:

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

Linux Admin

- In Linux, Admin is called as "super-user".
- Admin's login name is "root".
- Most of modern Linux, disable "root" login (for security).
- To execute commands with admin privileges use "sudo" (if approved by system admin).
 - cmd> sudo apt update
 - cmd> sudo apt install vim gcc python3 python3-pip
 - cmd> sudo snap install --classic code

Directory commands

- pwd -- print present working directory (current directory)
- cd -- change directory (syntax> cd dirpath)
- ls - list directory contents (syntax> ls dirpath)
- mkdir -- make directory (syntax> mkdir dirpath)
- rmdir -- remove empty directory (syntax> rmdir dirpath)

File commands

- cat
 - cat > filepath <-- create new file
 - cat filepath <-- display file contents
- rm
 - rm filepath <-- delete given file
 - rm -r dirpath <-- delete dir with all contents
- mv
 - mv filepath destdirpath <-- move given file into given dest directory
 - mv dirpath destdirpath <-- move given dir into given dest directory
 - mv oldname newname <-- rename given file
- cp
 - cp filename newfilename <-- copy file with new name/path.
 - cp filepath destdirpath <-- copy file into given dest dir with same name.
 - cp -r dirpath destdirpath <-- copy file into given dest dir with same name.
- touch
 - Create empty file (if file doesn't exists).
 - Modify access time of the file (if file exists).