Operating System Concepts

Introduction

OS Contents

- OS Concepts
- Linux commands
- Shell Scripts
- Linux System Call Programming

Schedules

- Lecture: 8.00 am to 1.30 pm (5 days)
    - Break1: 9:45 am (25 mins)
    - Break2: 12:00 pm (10 mins)
- Lecture: 2.30 pm to 4.15 pm (3 days)
- Lab: 4.40 pm to 7.00 pm (3 days)

Trainer

- Nilesh Ghule
- M.Sc. Electronics Science
- Experience: 20+ Years
- Subjects: Operating Systems, Device Drivers, Microcontrollers, Java, Advanced Java, Databases, Big Data.

Agenda

- Introduction
- What is OS?
- OS Functions
- Process management

- File management
- User interfacing

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
  - OS/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

Hardware abstraction

- OS hides all hardaware intricacies from the end user as well as user programs.
- Same program can work for different hardware components. e.g. Same Hello World program for CRT monitors, LCD monitors, or LED monitors.

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 (Portable Executable)
  - Linux -- ELF (Executable Linking Format)
- Programs are stored on disk (storage).

Process

- Progam under execution.
- Process execute in RAM.
- Process has multiple sections i.e. text, data, rodata, heap, stack. ... into user space and its metadata is stored into kernel space in form of PCB struct.
- 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 more than 5 kb.

File Management

File

- File is collection of data/information on storage device.
    - File = Contents (Data) + Information (Metadata)
    - The data is stored in zero or more Data blocks (in FS), while metadata is stored in the FCB (in filesystem).
- FCB is called as "inode" on UNIX/Linux. It contains
    - type: UNIX/Linux has 7 types of files
        - -: regular, d: directory, l: symbolic link, p: pipe, s: socket, c: char device, b: block device
    - size: number of bytes
    - links: number of hard links
    - mode (permissions): (u) rwx, (g) rwx, (o) rwx
    - user & group
    - time-stamps: modification, creation, access.
    - info about data blocks
- terminal> ls -l
    - type, mode, links, user, group, size, timestamp, name.
    - File Types
        - Regular file (-)
        - Directory file (d)

        - Link file (l)
        - Socket file (s)
        - Pipe file (p)
        - Character Special file (c)
        - Block Special file (b)
- terminal> stat filepath

File System

- Files are stored on storage device. Arrangement of files in storage device is called as "File System".
- e.g. FAT, NTFS, EXT2/3/4, ReiserFS, XFS, HFS, etc.
- File System logically divide partition into 4 sections.
    - Boot block/Boot sector
        - Contains programs/info required for booting of OS
        - Typically contains bootstrap program and bootloader program
    - Super block/Volume control block
        - Contains information of whole partition.
        - Capacity, Label.
        - terminal> df -h
            - Total number of data blocks/inodes.
            - Number of used/free data blocks/inodes.
            - Information of free data blocks/inodes.
    - Inode List/Master file table
        - Inodes (FCB) for each file
    - Data blocks
        - Stores data of the file.
        - Each file have zero or more data blocks.
        - Size of data blocks can be configured while creating file system
- File system is created by the format utility while formatting the partition.

  - Windows: format.exe
  - Linux: mkfs
    - terminal> sudo mkfs -t ext3 /dev/sdb1
    - terminal> sudo mkfs -t vfat /dev/sdb1
    - -t fs_type e.g. ext3, ext4, vfat, ntfs, ...
    - partition e.g. /dev/sdb1

- Disk/partition naming conventions

  - Windows:
    - Disks are named as disk0, disk1, ...
    - partitions are named as drives i.e. C:, D:, E:, ...
  - Linux:
    - Disks are named as /dev/sda, /dev/sdb, /dev/sdc, etc.
    - Partitions per disk are named as
      - sda partitions: sda1, sda2, sda3, ...
      - sdb partitions: sdb1, ...

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
    - lib - shared program libraries required by kernel
    - etc - configuration files
    - root - home directory of root user
    - home - it contains sub directories for each user with its name
        - nilesh -> /home/nilesh
        - sunbeam -> /home/sunbeam
        - osboxes -> /home/osboxes
    - usr - read only directory that stores small programs and files accessible to all users
    - dev - it contains all device related files
    - 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
    - mnt - it is temporary mount point
    - media - it is mount point for media eg cdrom
    - opt - stores optional files of large softwares
    - tmp - temporary files that may be lost on system shutdown

User interfacing

- UI of OS is a program (Shell) that interface between End user and Kernel.
- Shell -- Commmand 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, ...)

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/nilesh/MyData/Demos/demo01.sh
    - Relative path
        - Path with respect to current directory is called as relative path
        - E.g. MyData/Assignments/assign02.pdf

Linux Commands

- env -- display all env variables

- echo $VAR -- display given env variable
- which programname -- find the program in all directories mentioned in PATH variable and display its path if found.
- pwd -- prints current/present working directory
- man command -- show help of the command
    - press "q" to exit.
- ls -- shows contents of current of current directory
    - -l: long listing (details of file/subdirectories)
    - -i: show inode number
    - -R: recursive listing
    - -S: sorted order of size (desc sort)
    - -r: reverse order used with sorting
    - -a: hidden + non-hidden + . + ..
    - -A: hidden + non-hidden
- ls dirpath -- shows contents of given directory
    - ls /
    - ls /home
    - ls /dev
- mkdir dirpath -- create new directory
    - mkdir /tmp/movies
    - mkdir /tmp/songs
    - cd /tmp
    - mkdir movies/hw
    - cd movies
    - mkdir bw
    - mkdir -p /tmp/songs/classic/gazal
    - ls -R /tmp/movies
    - ls -R /tmp/songs
- cat > filepath
    - enter file contents and press ctrl+d to exit.
    - create new file if doesn't exist
    - truncate if file exist

- cat filepath
    - display file contents
    - -n: show line numbers
- cat >> filepath
    - append to file
- tac filepath
- rev filepath
- cp srcfilepath destdirpath
    - cp hw/inception.txt bw
- cp srcfilepath destfilepath
    - cd hw
    - cp inception.txt inception2.txt
- cp -R srcdirpath destdirpath
    - cp -R /tmp/movies /home/nilesh/mar-24/dac/os/day01
- mv filepath destdirpath
    - mv /home/nilesh/mar-24/dac/os/day01/a.out /tmp
    - ls /tmp/*.out
    - /tmp/a.out
- mv dirpath destdirpath
    - ls /tmp/songs
    - mv /tmp/songs/ /home/nilesh/mar-24/dac/os/day01
    - ls /tmp/songs
- mv filename newfilename
    - cd /home/nilesh/mar-24/dac/os/day01/movies/bw
    - ls
    - mv inception.txt incept.txt
        - file rename
    - cat incept.txt
    - mv incept.txt .incept.txt
    - ls -A
    - cat .incept.txt

    - mv .incept.txt incept.txt
    - ls
- rmdir dirpath
    - can delete only empty directories
    - rmdir /home/nilesh/mar-24/dac/os/day01/movies
        - Error
- rm filepath
    - delete file
    - cd /home/nilesh/mar-24/dac/os/day01
    - rm movies/bw/incept.txt
- rm -R dirpath
    - delete directories with all its contents
    - rm -R movies
- touch filepath
    - If file exists, change its timestamp.
    - If file doesn't exist, create empty file
- head -n filepath
    - Display first n lines of file
- tail -n filepath
    - Display last n lines of the file
- alias c=clear
    - "c" is shortcut for "clear" command
    - c -- clear screen
    - alias
    - unalias c

IO Redirection

- Input redirection
    - command < filepath
    - Example: wc < in.txt
- Output redirection

  - command > filepath
  - Example: history > out.txt
- Error redirection
  - command 2> filepath
  - Example: ls /abcd > err.txt
- Multiple redirection operators can be used in single command
  - Example: ls /home /abcd > out.txt 2> err.txt