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

- Progam 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

Process

- Process is program in execution.
- 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.
- PCB contains
    - id, exit status,
    - scheduling info (state, priority, time left, scheduling policy, ...),
    - files info (current directory, root directory, open file descriptor table, ...),
    - memory information (base & limit, segment table, or page table),
    - ipc information (signals, ...),
    - execution context, kernel stack, ...

Linux commands

- which
    - which command
        - display the location of command executable.
- whereis
    - whereis command
        - display the location of command executable and also manual paage location.

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)
- cd
    - cd ~ - change working directory to home directory
    - cd - - change working directory to old working directory
    - cd .. - change working directory to parent directory

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.

Redirection

- for every command input is taken from terminal, output is printed on terminal and error is also printed on terminal
- Standard streams (by defualt for every process, three files are opened)
    - stdin
    - stdout
    - stderr
- There are three types of redirections
    - input redirection
        - input will be taken from file instead of stdin
        - to do input redirection '<' symbol is used
        - command < file
    - output redirection
        - output will be written into file instead of stdout
        - to do output redirection '>' or '>>' symbol is used
        - command > file
            - older content of file will be over written
        - command >> file
            - content will be appneded into file at the end

- error redirection
    - error will be written into file instead of stderr
    - to do output redirection '2>' or '2>>' symbol is used
    - command 2> file
        - older content of file will be over written
    - command 2>> file
        - content will be appneded into file at the end

Pipe

- Using pipe, we can redirect output of any command to the input of any other command.
- Two processes are connected using pipe operator (|).
- Two processes runs simultaneously and are automatically rescheduled as data flows between them.
- If you don't use pipes, you must use several steps to do single task.
- command1 | command2
    - output of command1 will be given as input to command 2
- E.g.
    - who | wc

Shell meta characters

- '*' - zero or more occurances of any character
- '?' - one occurance of any character

Regular Expressions

- Find a pattern in text file(s).
- Regular expressions are patterns used to match character combinations in strings.
- A regular expression pattern is composed of simple characters, or a combination of simple and special characters e.g. /abc/, /ab*c/
- Pattern is given using regex wild-card characters.
    - Basic wild-card characters
        - $ - find at the end of line.
        - ^ - find at the start of line.
        - [ ] - any single char in give range or set of chars
        - [^] - any single char not in give range or set of chars
        - . - any single character
        - zero or more occurrences of previous character
    - Extended wild-card characters
        - ? - zero or one occurrence of previous character
        - one or more occurrences of previous character
        - {n} - n occurrences of previous character
        - {,n} - max n occurrences of previous character
        - {m,} - min m occurrences of previous character
        - {m,n} - min m and max n occurrences of previous character
        - () - grouping (chars)
        - (|) - find one of the group of characters

grep

- Regex commands
  - grep - GNU Regular Expression Parser - Basic wild-card
  - egrep - Extended Grep - Basic + Extended wild-card
  - fgrep - Fixed Grep - No wild-card
- Command syntax
  - grep "pattern" filepath
  - grep [options] "pattern" filepath
    - -c : count number of occurrences
    - -v : invert the find output
    - -i : case insensitive search
    - -w : search whole words only
    - -R : search recursively in a directory
    - -n : show line number.