Redirection

- By default every process opens 3 file descriptors
  - fd = 0 -> standard input to a program
  - fd = 1 -> standard output to a program
  - fd = 2 -> standard error to a program
- You can redirect each of these independently.
- According to direction of redirection, there are three types
- Input redirection
  - "<" is used for input redirection
- Output redirection
  - ">" is used for input redirection
- Error redirection
  - "2>" is used for input redirection

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.
- E.g.
  - who | wc

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/

grep

- 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
- 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.

VI Editor

- sudo apt-get install vim
- VI editor works in two modes
    - command mode
    - insert mode
- press i - to go into insert mode
- press Esc - to go into command node
- VI editor commands:
    - w - write/save into file
    - q - quit vi editor
    - yy - to copy current line
    - nyy - copy n lines from current line
    - m,ny - copy fomr mth line to nth line
    - dd - to cut current line
    - ndd - cut n lines from current line
    - m,nd - cut fomr mth line to nth line
    - press p - to paste copied line on next line of current line

Simple OS structure

- Small Operating systems like MS-DOS or few embedded OS follow a very simple structure.
- DOS operating system is made up of three files only.
    - COMMAND.COM <- command interpreter
    - MSDOS.SYS <- kernel
    - IO.SYS <- device drivers

Layered structure (architecture)

- OS is divided into multiple layers, so that each layer depends on the lower layer and provide functionality to the upper layer.
- Example: Windows, UNIX, Linux, etc.
- Windows OS have following layers
    - applications
    - system call APIs
    - system call implmenetation
    - Kernel Executive : File Mgr, Memory Mgr, Process Mgr, Scheduler, Thread Mgr, etc.
    - IO Subsystem
    - Device Drivers
    - Hardware Abstraction Layer

Monolithic Kernel

- Multiple kernel source files are compiled into single kernel binary image. Such kernels are "monolithic" kernels.
- Since all functionalities present in single binary image, execution is faster.
- If any functionality fails at runtime, entire kernel may crash.
- Any modification in any component of OS, needs recompilation of the entire OS.
- Examples: BSD Unix, Windows (ntoskrnl.exe), Linux (vmlinuz), etc.

Micro-kernel

- Kernel is having minimal functionalities and remaining functionalities are implemented as independent processes called as "servers".
    - e.g. File management is done by a program called as "file server".
- These servers communicate with each other using IPC mechanism (message passing) and hence execution is little slower.
- If any component fails at runtime, only that process is terminated and rest kernel may keep functioning.
- Any modification in any component need to recompile only that component.
- Examples: Symbian, MACH, etc.

Modular Kernel

- Dynamically loadable modules (e.g. .dll / .so files) are loaded into calling process at runtime.
- In modular systems, kernel has minimal functionalities and rest of the functionalities are implemented as dynamically loadable modules.
- These modules get loaded into the kernel whenever they are called.

- As single kernel process is running, no need of IPC for the execution and thus improves performance of the system.
- Examples: Windows, Linux, etc.

Hybrid Kernel

- Mac OS X kernel is made by combination of two different kernels.
- BSD UNIX + MACH = Darwin

Linux - OS Structure

Linux components

- Linux kernel has static and dynamic components.
- Static components are
  - Scheduler
  - Process management
  - Memory management
  - IO subsystem (core)
  - System calls
- Dynamic components are
  - File systems (like ext3, ext4, FAT)
  - Device drivers
- Static components are compiled into the kernel binary image.
  - They are kernel components.
  - The kernel image is /boot/vmlinuz.
- Dynamic components are compiled into kernel objects (*.ko files).
  - They are non-kernel components.
  - They are located in /lib/modules/kernel-version.