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.
- 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, ...

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, ...)

Computer structure

- CPU: Genral purpose processor for program/OS execution
- Memory: RAM
- Storage: Disk

- IO: Keyboard, Monitor
- Connected by "bus".
- Each IO device has a "dedicated" "internal" processing unit -- IO device controller.

Computer IO (Input Output)

- Synchronous IO: CPU is waiting for IO to complete.
  - Hw technique: Polling
- Asynchronous IO: CPU is not waiting for IO to complete (doing some other task)
  - Hw technique: Interrupts
  - OS maintains a device status table to keep track of IO devices (busy/idle) and processes waiting for those IO devices.

Interrupt Processing

- IO event is sensed by IO device controllers.
- It will be conveyed to CPU as a special signal - Interrupt.
- CPU pause current execution and execute interrupt handler.
- "Interrupt handler" will get address of "ISR" (from IVT) and execute ISR.
- When ISR is completed, execution resumes where it was paused.

Hardware vs Software interrupt

- Hardware -- interrupts from hardware peripherals.
- Software interrupt
  - Special instructions (Assembly/Machine level) when executed, current execution is paused, interrupt handler is executed and then the paused execution resumes.
  - Arch specific:
    - 8085/86: INT
    - ARM 7: SWI
    - ARM Cortex: SVC
  - Also called as "Trap" in few architecture.

Interrupt Controller

- Convey the interrupts from various peripherals to the CPU.
- Also manage priority of the interrupt (when multiple interrupts arrives at same time).
- e.g. 8085/86 <-- 8259, Modern x86 processors (apic), ARM-7 (VIC), ARM-CM3 (NVIC), ...

CPU Scheduling

- Modern OS are time-sharing systems i.e. CPU time is allocated to each process and after that time the next process is scheduled on the CPU.
- Timer Hardware (PIT/SysTick) is used periodically to generate the interrupt.
- When interrupt is arrived, interrupt handler is executed which in turn invokes ISR.
- Then interrupt handler invokes scheduler.
- Scheduler check if time allocated to current process is completed and if completed then decide the next process to be executed (using some scheduling algorithm).

- The selected process's execution context is then restored into CPU by the dispatcher and the next process continues to execute.
- The whole process is also referred as "Context Switch".

Hardware abstraction layer

- Most important feature of OS.
- It hides hardware details from end-user as well as from user programs.
- HAL Provides reusable code to interact with hardware.
- HAL layer of any OS is always arch depedent. Most of the code is written in assembly language.

Networking

- Networking feature enable computers (processes) to communicate with each other.
- Even though important (nowadays), networking is optional feature of OS.
- For networking computers are connected to each other in LAN, MAN or WAN.
- Computers are connected with different topologies e.g. bus, star, ring, mesh, ...
- Networking feature internally use "sockets" IPC mechanism.
- Socket is communication end-point.

Security and Protection

- Security is securing system from "external" threats e.g. virus, trojans, worms, hacking, etc.
  - Security is optional feature and is not implemented in many OS.
  - Security is usually provided by "Anti-virus" application.
  - Windows 10+ comes with Windows Defender, which is handling security aspects.
- Protection is protecting system (programs & files) from internal threats/elements.
  - Dual mode protection: CPU can differentiate whether code belongs to OS or user application.
  - IO protection: Only OS should be able to perform IO operations. User programs should use system calls to perform IO. This is feasible when IO instructions are privileged instructions (they can only be executed in kernel mode).
  - Memory protection: One process should not access memory of another process directly, so that one process cannot disturb execution of another process. This is implemented using MMU.'
  - CPU protection: If a process goes in infinite loop, the whole system should not hang. This is done using Timer hardware.