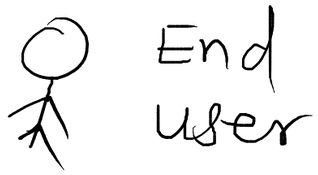
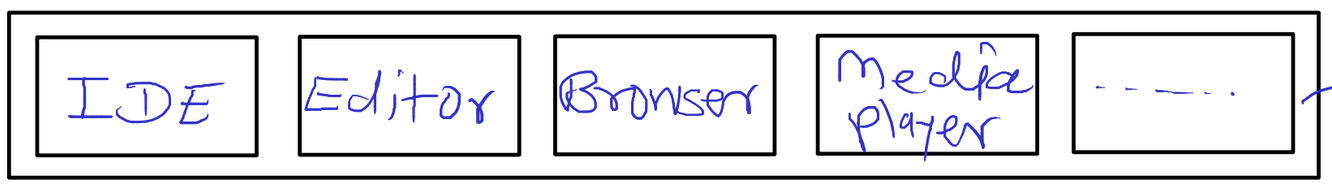
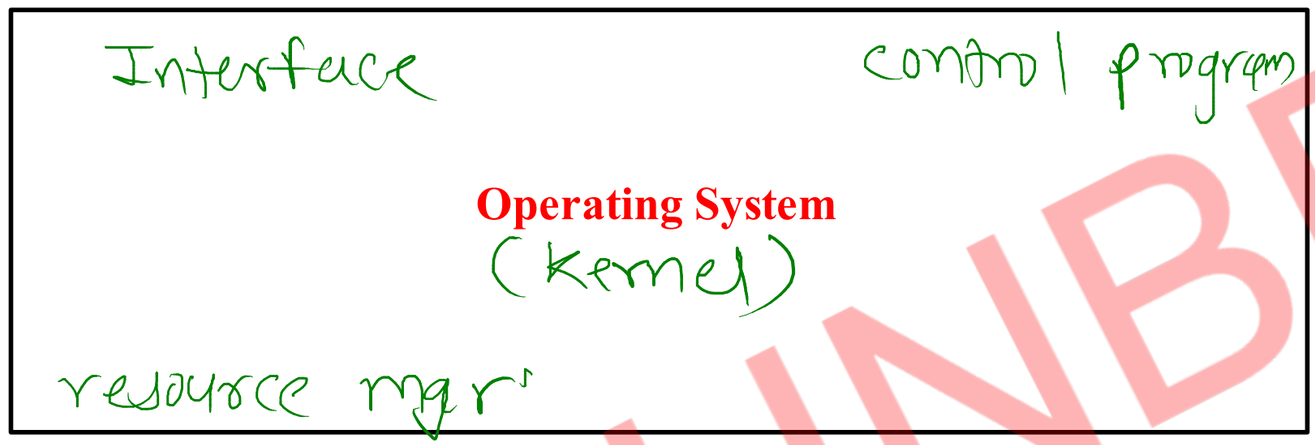
# Operating System





- interface betn end user and computer h/w
- interface betn Appln s/w & computer h/w
- control program which controls execution of all appln/programs running on the top it.
- resource manager/ allocator which allocates h/w resource to all the programs one by one.

- CD/DVD/ISO - Core OS (kernel) + Appln s/w + System Utilities

# Functions of Operating System

1) Process Management
2) CPU scheduling
3) Memory Management
4) File & IO Management
5) Hardware Abstraction
6) User Interfacing
7) Networking
8) Security & Protection

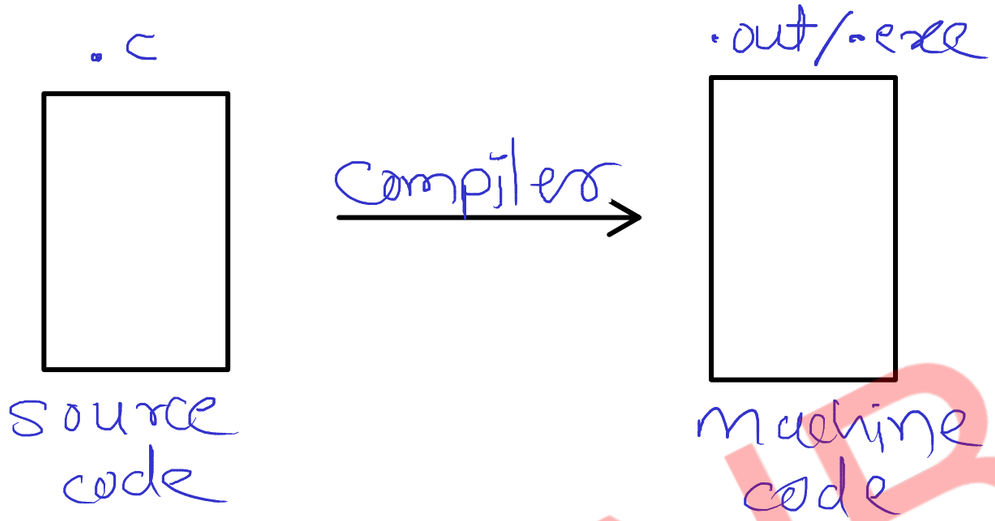
Compulsory

optional

# Process Management

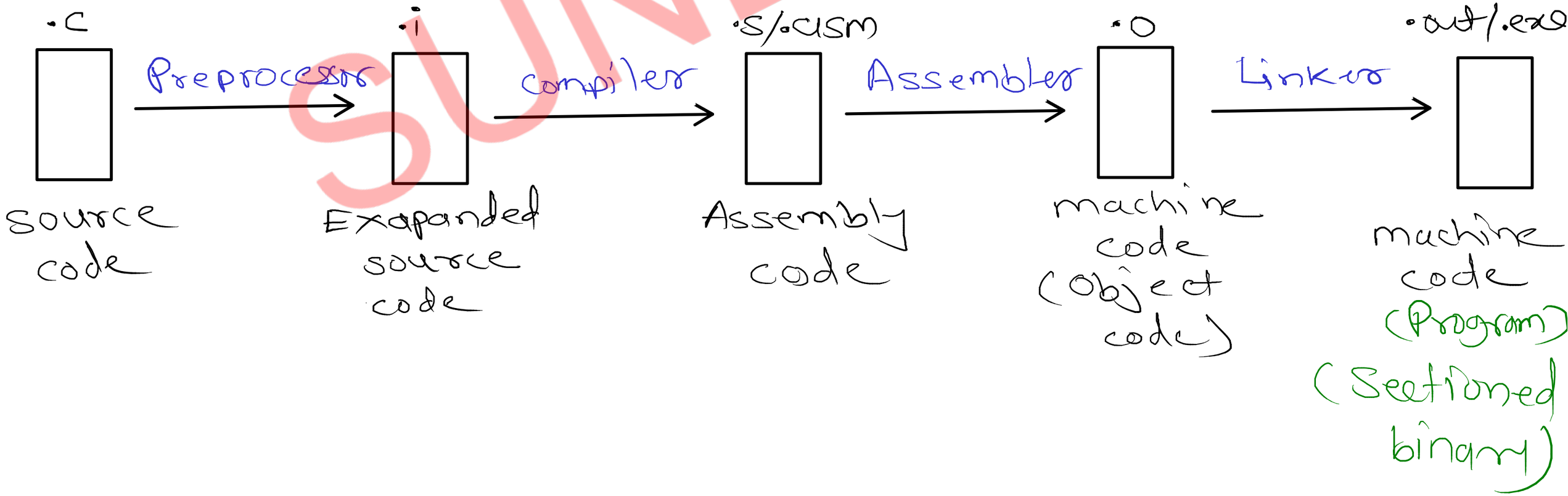
Process — Program in execution

Program — set of instructions to machine (CPU)




GCC — (GNU compiler collection)
- set of tools (tool chain)

1> Preprocessor
2> Compiler
3> Assembler
4> Linker
5> debugger
6> ----



# Program

tools – objdump / readelf


## Exe Header

- info about application
- info about remaining sections of executable
  - name, start, end, size
- type of application (CLI / GUI / Library)
- address of entry point function
- magic number – identify to file format
  - Windows – Portable Executable (PE) – MZ
  - Linux – Executable Linking Format (ELF) – 0x7F ELF

## Text (Code)

- instructions of the program in machine code format are stored.

**Data** – static & global variables (initialised) int arr[] = {...};

**BSS** – static & global variables (uninitialised) int arr[];

## RO Data (Read Only)

- string constants

char *ptr = "sunbeam";

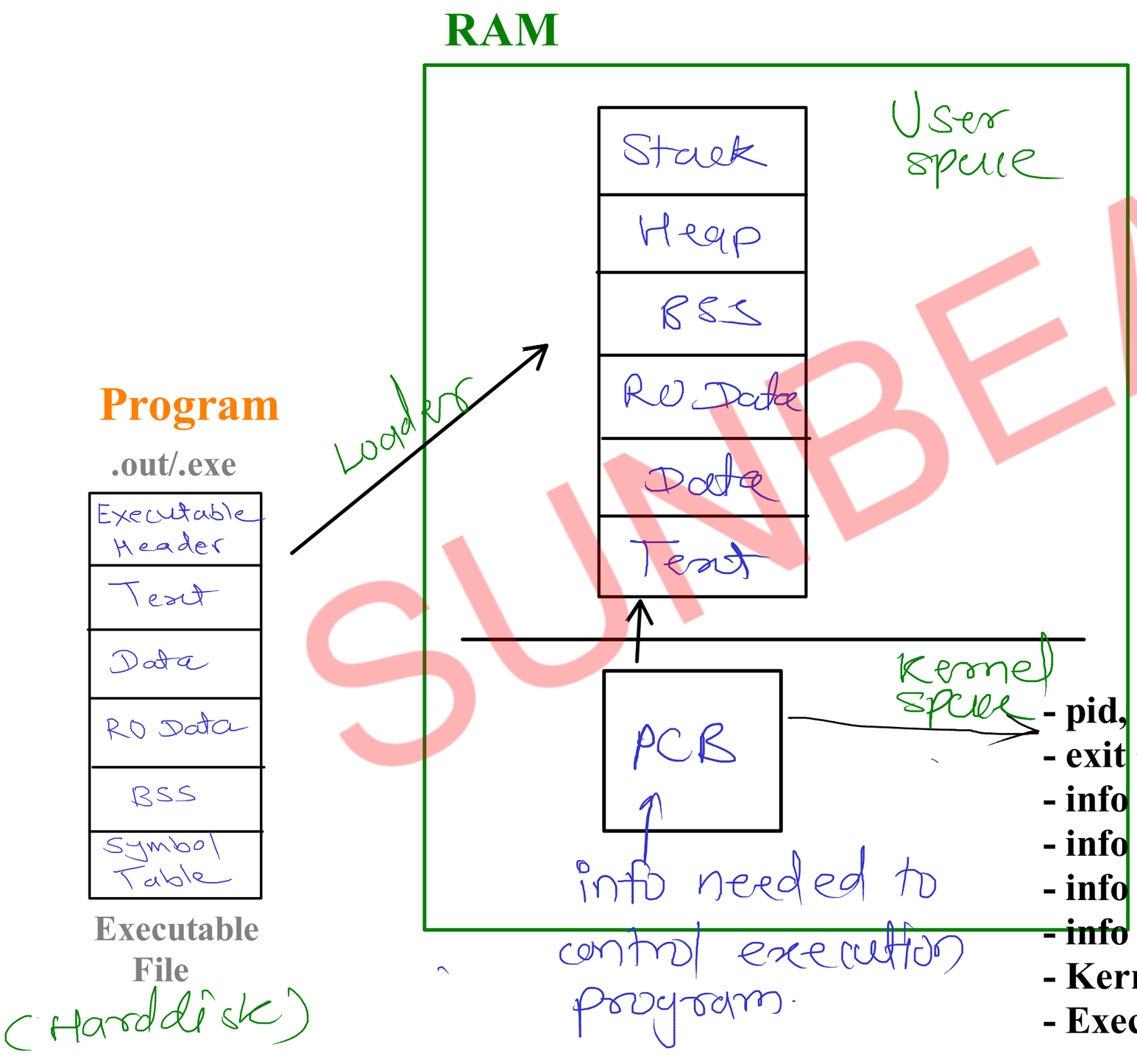
## Symbol Table

- info about symbols
- symbols are variables & functions of the program

symbols
- variables – name, type, size, address, section ----
- functions – name, address, return type, no./type of arg ----

# Process

## Process = sections + PCB




PCB
- task_struct
- uarea (UNIX)
- process descriptor

- pid, ppid
- exit status
- info sched (Priority, Algo, state)
- info mem (base and limit, segment/page table)
- info file (opened file, current directory)
- info IPC (signal...)
- Kernel stack
- Execution context

# File Management

File – collection of data/information

File = data + metadata

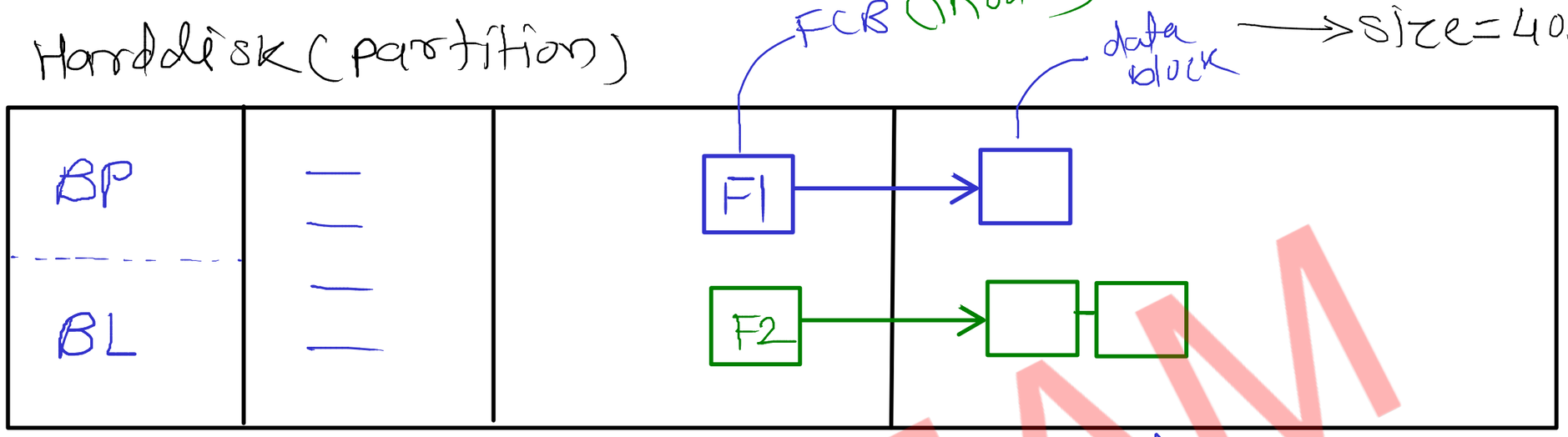
(Actual contents) (information about file)

(data block)

- name
- size
- type
- time stamp
  (create, modify, access
- owner/group
- permissions
  (r-read, w-write, x-execute
  -user/group/other
- link count
- info about data blocks

(File Control Block) (FCB)

# File System




→ size = 4096 bytes (default)
↓
configurable
↓
at the time formating

(boot Boot block) sector
↓
Booting related programs
1) Bootstrap program
2) Bootloader

Volume control block (superblock)
↳ info about volume/partition
- label
- size
- filled/empty
- info about free data blocks of partition

Master file Table (inode list)

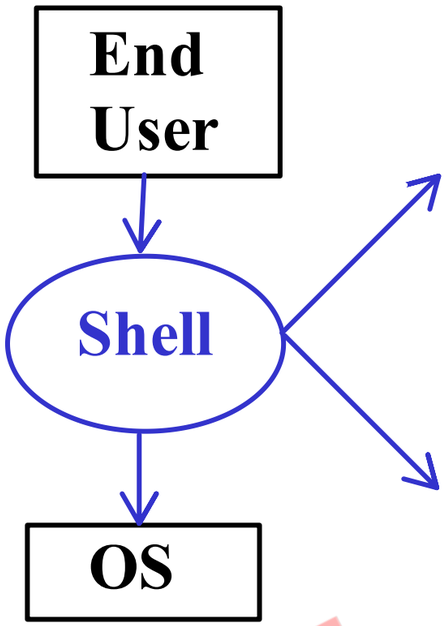
Data blocks

File system – Orgnising files on harddisk partitions

# User Interfacing

## Shell - intermediate between End user and OS

## Shell - Command Interpretor




**CLI (Command Line Interface)**

**Windows - cmd.exe, powershell.exe**
**Linux - bsh(sh), bash, csh, ksh, zsh**

**GUI (Graphical User Interface)**

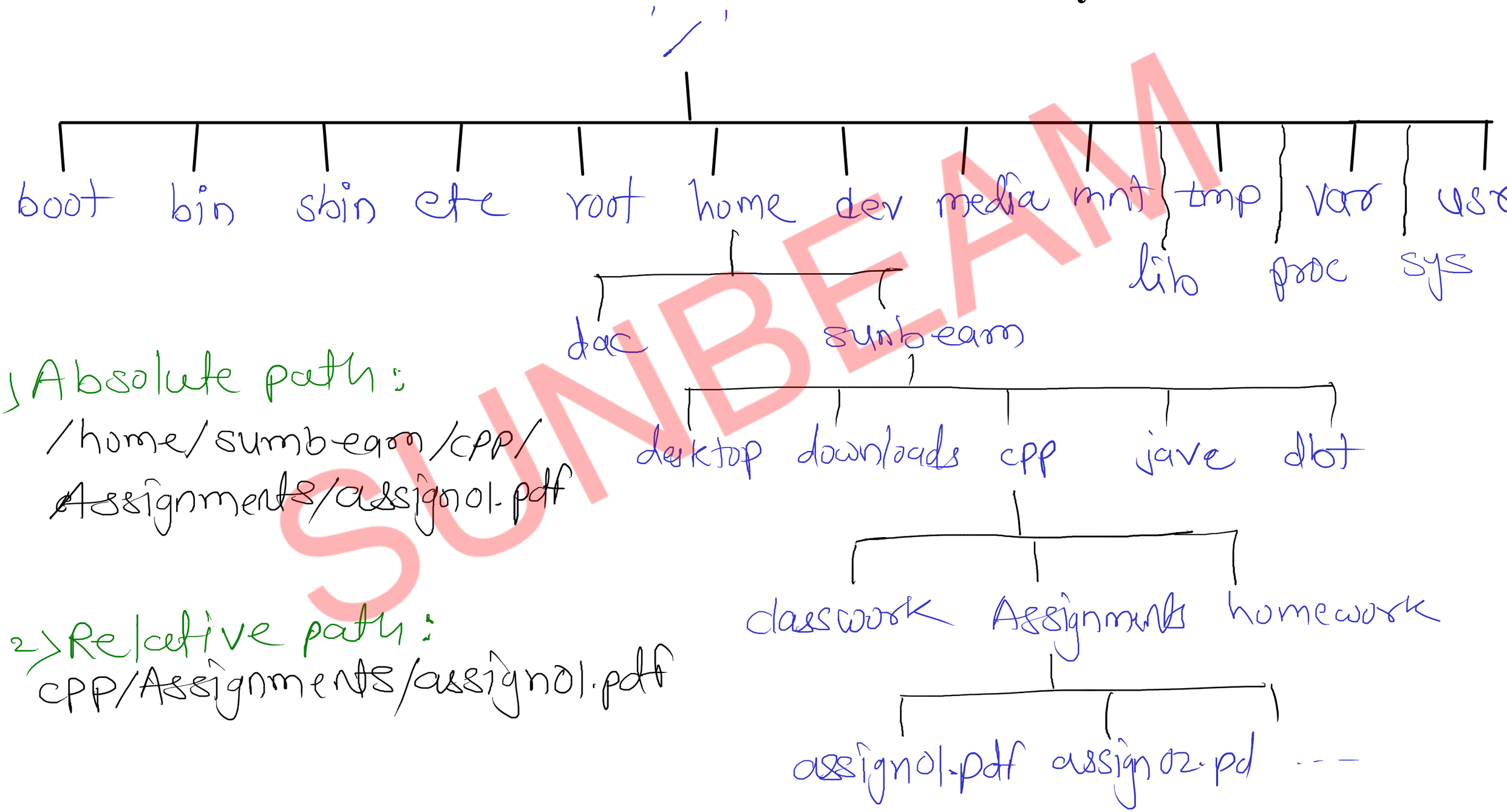
**Windows - explorer.exe**
**Linux - KDE(Komman Desktop Env)**
**- GNOME(GNU Network Object Model Env)**

**- In Linux, default shell is bash(Bourne Again Shell)**
**- echo $SHELL**
**- to change shell - chsh**

admin - administrator
root - superuser

# Linux File Structure

- **Linux follows root "/" file structure**
- **In Linux file -> file and folder -> directory**




1) Absolute path :

/home/sumbeam/cpp/Assignments/assign01.pdf

2) Relative path :

cpp/Assignments/assign01.pdf

permissions of file | user/owner | size | name

**-rw-rw-r-- 1 sunbeam sunbeam 0 May 20 09:30 file1.txt**

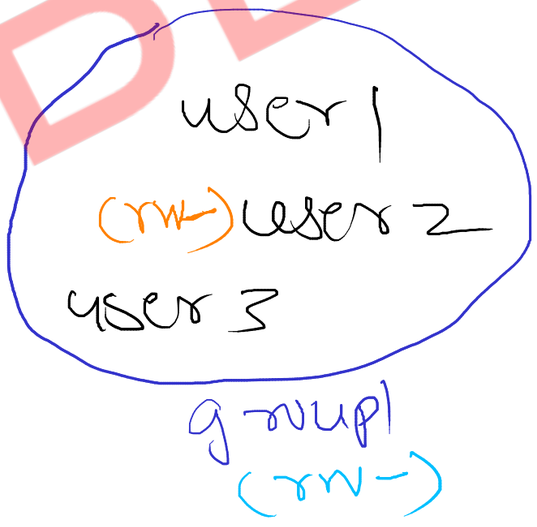
type of file | link count | group | timestamp

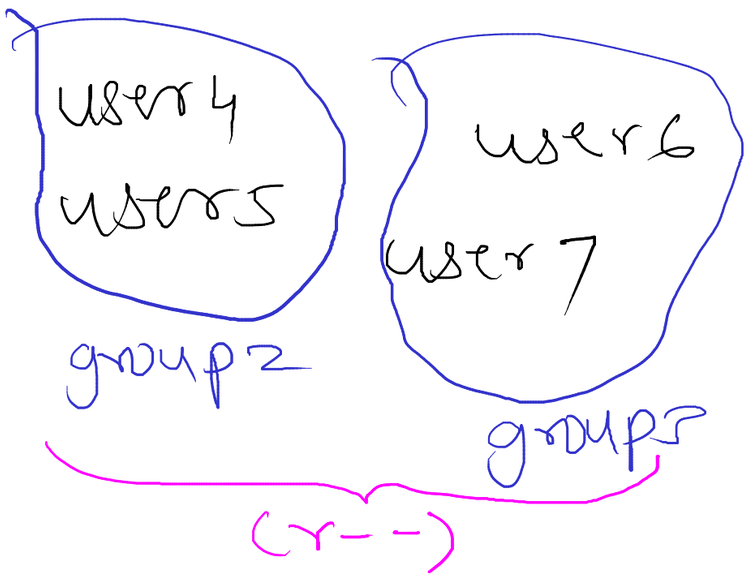
Types of files :

1) Regular (-)
2) Directory (d)
3) Link (l)
4) pipe (p)
5) socket (s)
6) char special (c)
7) block special (b)

Permission types – r-read, w-write, x-execute

levels – user/owner, group, other



```
chmod +x file.txt

    + : add        r
    - : remove     w
                   x
```

000-0
001-1
010-2
011-3
100-4
101-5
110-6
111-7

---

```
chmod g-x file.txt

    + : add        r        u - user/owner
    - : remove     w        g - group
                   x        o - others
```

---

```
rw-   rw-   r--
110   110   100
---   ---   ---
 6     6     4
```

664

```
rw-   rw-   r--
rwx   r--   r--
111   100   100
---   ---   ---
 7     4     4
```

744