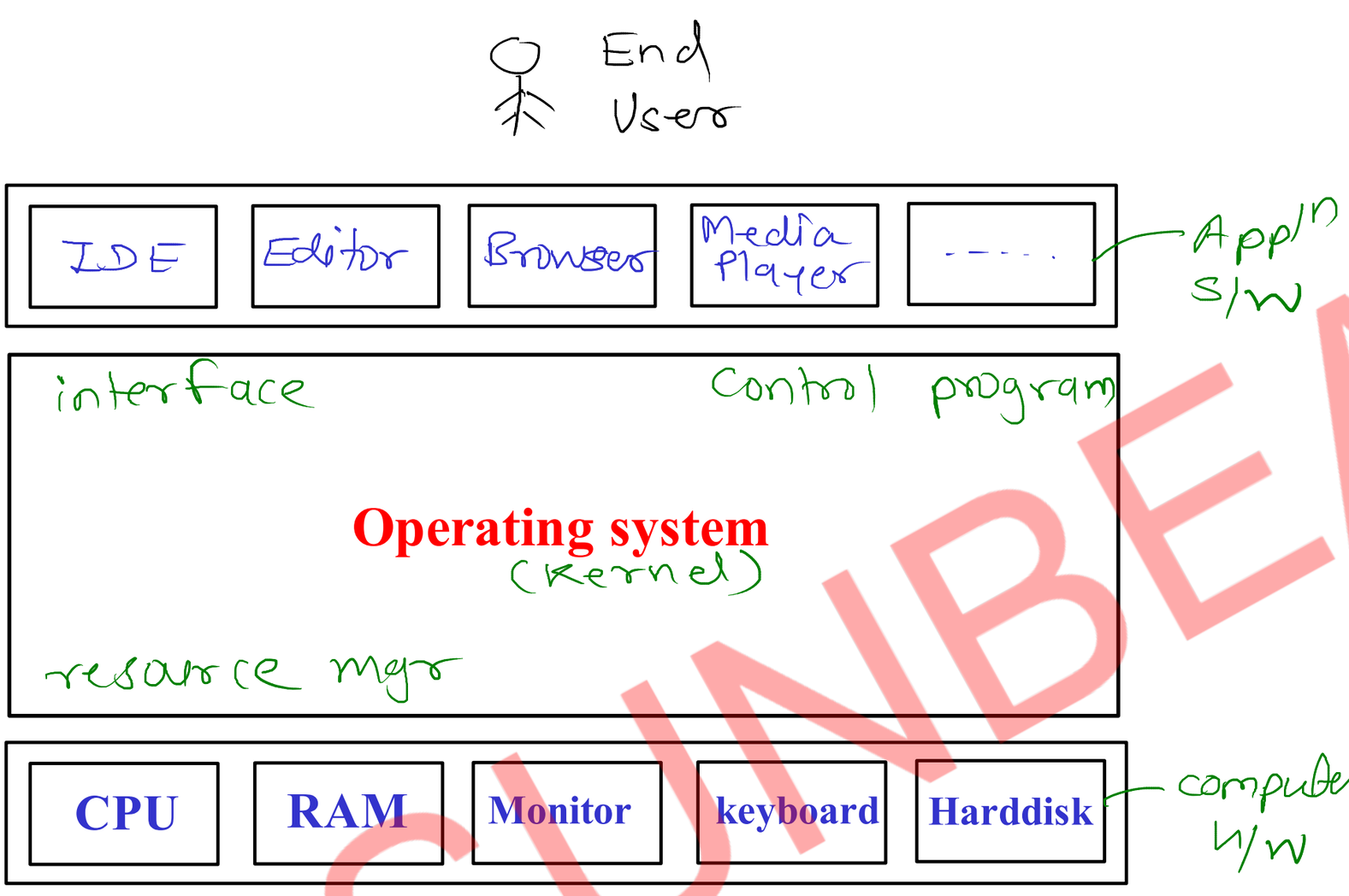
Operating System

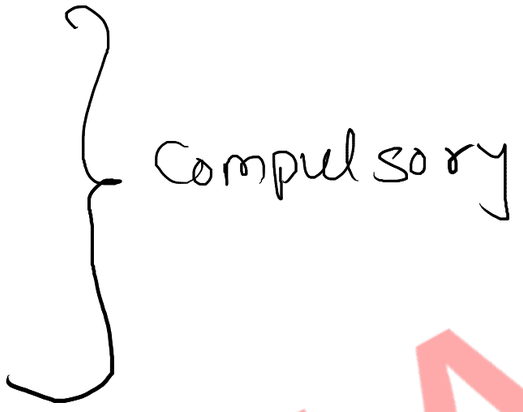


- interface betn end user & hardware
- interface betn appln s/w & computer h/w
- control program which is controlling execution of all the programs(appln) running on the system
- resource manager/ allocator which will allocate h/w resources to the programs

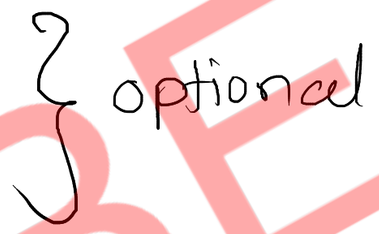
- CD/DVD/ISO — Core OS (Kernel) + Appln s/w + System Utilities

Functions of Operating System

1. Process Management
2. CPU Scheduling
3. Memory Management
4. File and IO Management
5. Hardware Abstraction

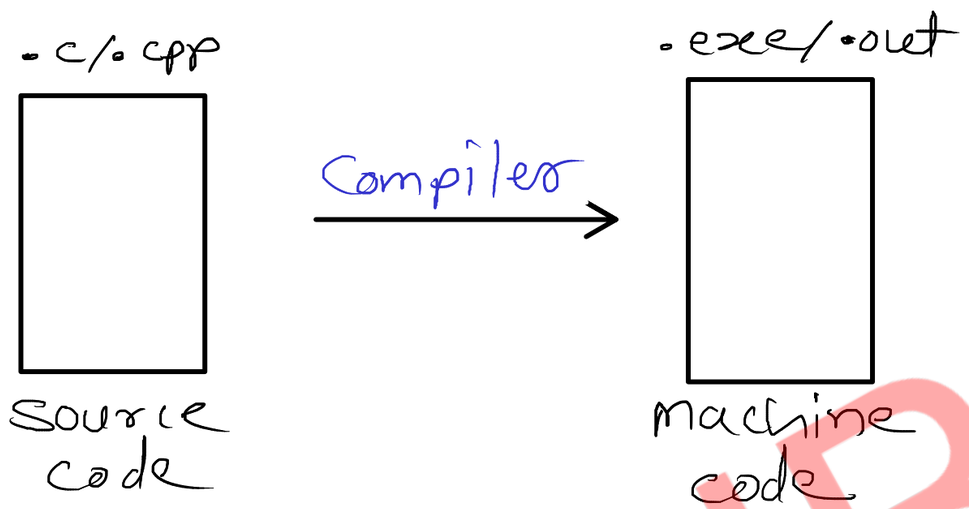


6. User Interfacing
7. Networking
8. Security and Protection



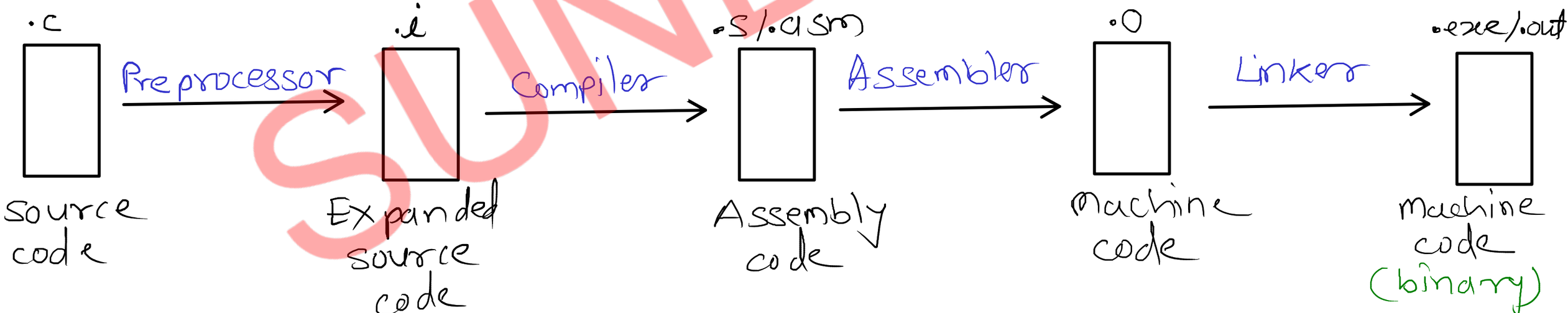
Process Management

Process - Program in execution
Program - set of instruction to the machine (CPU)



GCC (GNU C Compiler) (Toolchains)
- set of tools which will work on source code one by one

1. Preprocessor
2. Compiler
3. Assembler
4. Linker
5. Debugger
6. Utilities



Program

to read executable files
↳ readelf, objdump
↳ -h, -S, -t

Exe Header

- info about program
- type of program (CLI / GUI / library)
- info about remaining sections (start, end, size)
- address of entry point function
- Magic number - identity to file format
  - Windows (.exe) - Portable Executable - MZ
  - Linux (.out) - Executable Linking Format - ?ELF

(2 or 4 bytes)

Text (Code)

- instructions of program in machine code format

Data - all static & global variable (initialised) `int num1 = 10;`

BSS - all static & global variable (uninitialised) `int num2;`

RO Data - read only data e.g. `char *str = "sunbeam";`

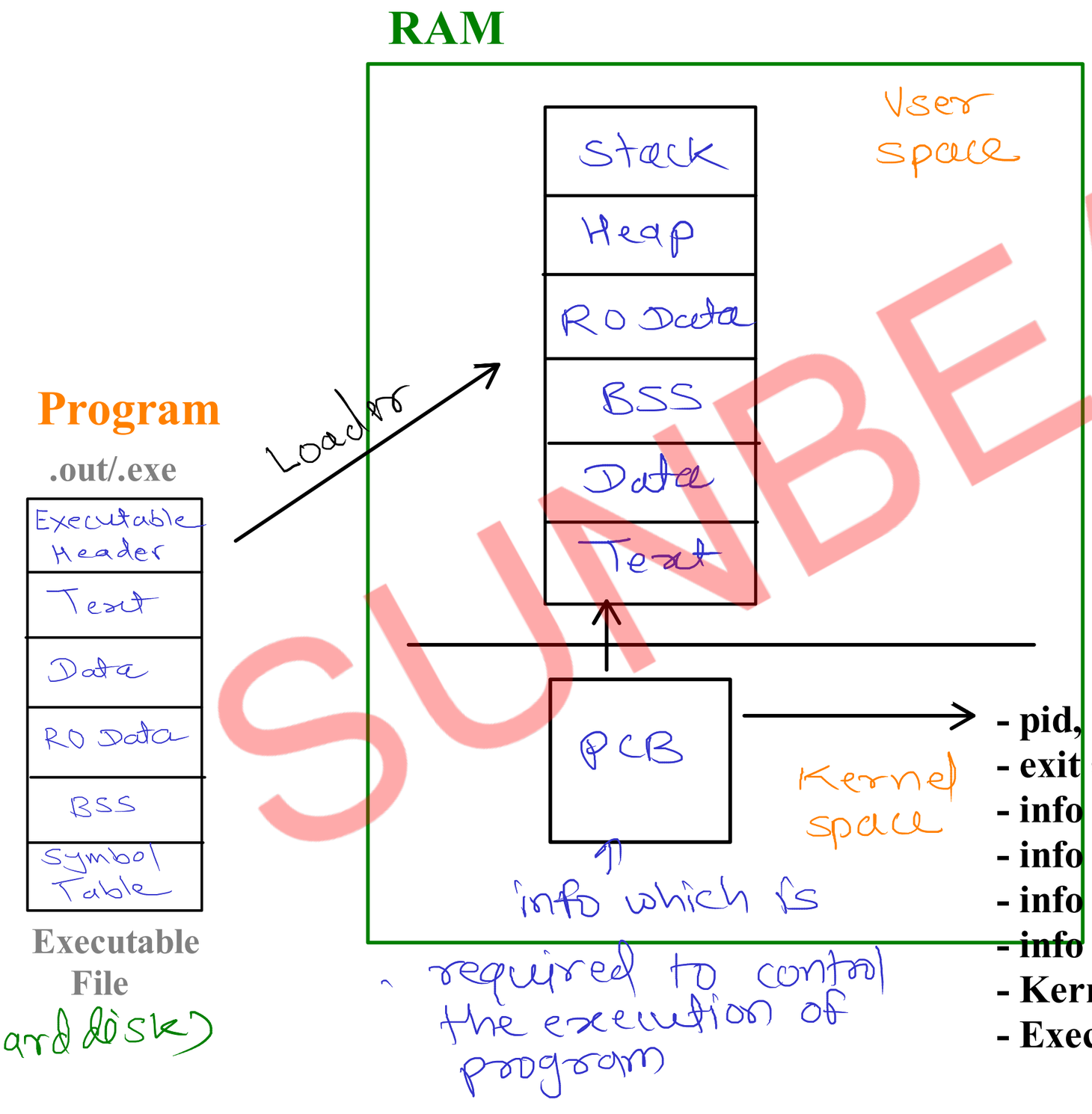
Symbol Table

- info about symbols

symbols
- variables — name, address, type, size, value, section
- functions - name, address, return type, no./type args.

Process

Process = All sections + PCB



Stack
- FAR of functions
  - return address
  - local variables
  - formal args

Heap
- dynamically allocated space

PCB/
task_struct/
process descriptor/
uarea

- 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

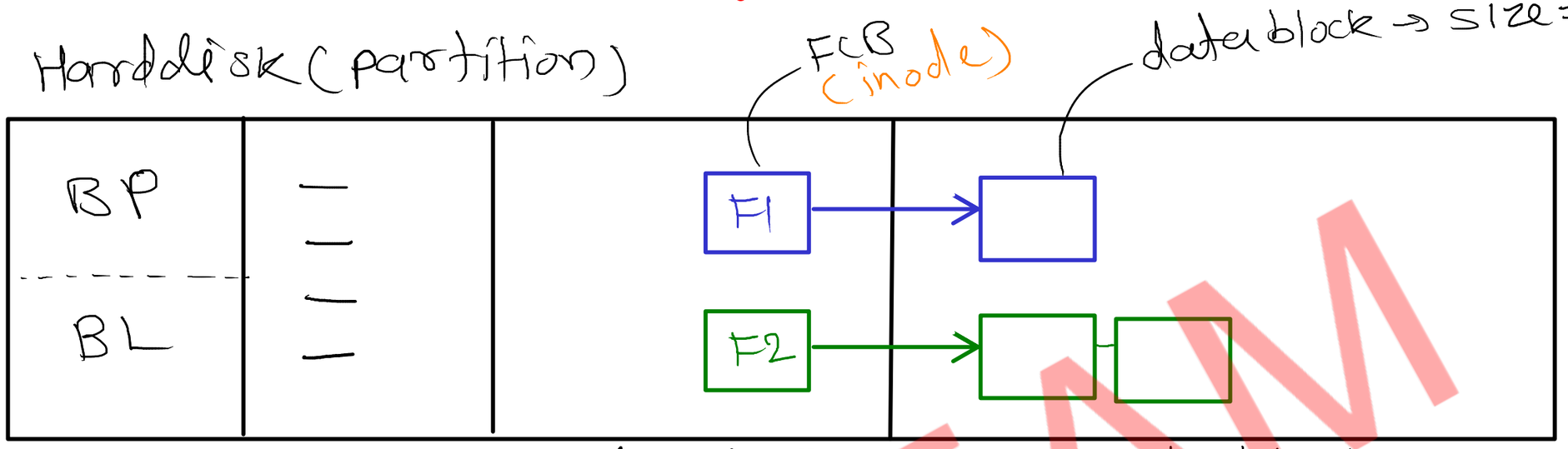
File – data + metadata

(Actual content)
(Data blocks)

(info about file)
- name
- size
- timestamps
  - create, modify, access
- user/owner
- group
- permissions
  r-read, w-write, x-execute
  - user/owner, group, others
- link count
- type
- info about data blocks of file

(File control block) (FCB)

File System



↓
programs which are responsible for booting
1) Bootstrap program
2) Bootloader

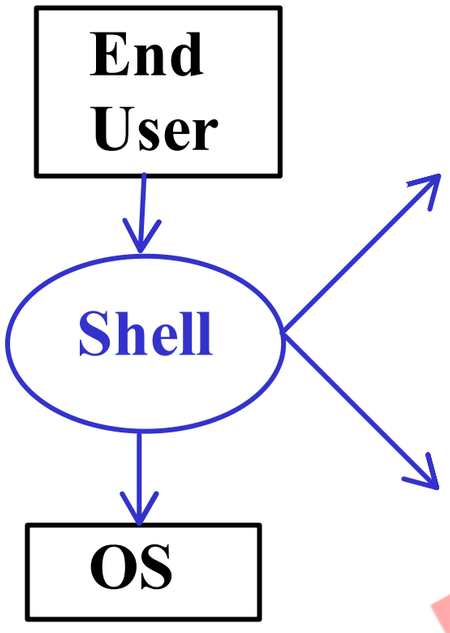
↓
info about volume/partition
- label
- size
- filled / empty
- list of free data blocks
- partition start & end

File system - Organising 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)

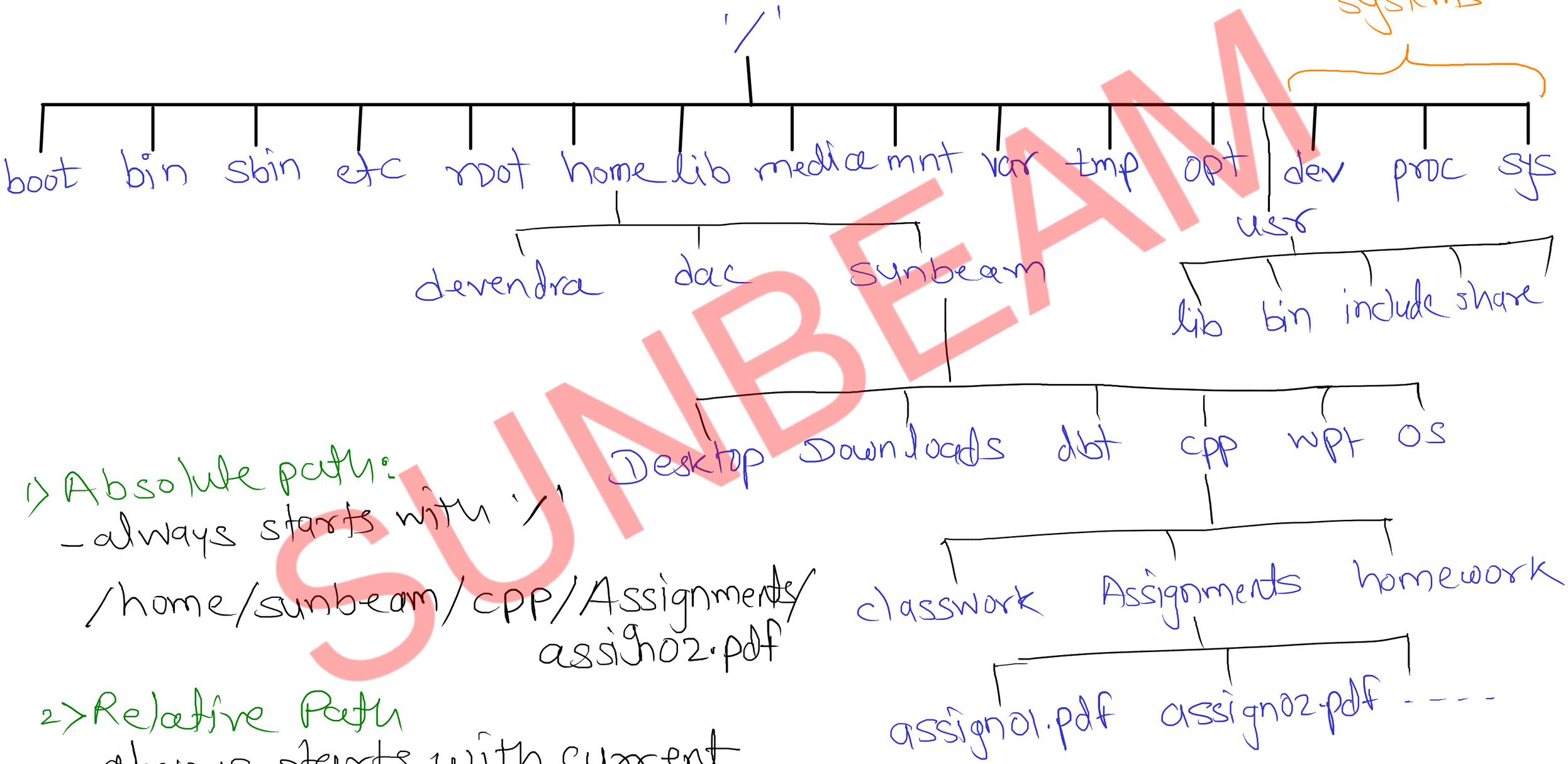
Windows - explorer.exe
Linux - KDE(Komman Desktop Env)
- GNOME(GNU Network Object Model Env)

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

Linux File Structure

Administrator – Super User
admin – root

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



1) Absolute path:
- always starts with '/'

/home/sunbeam/cpp/Assignments/assign02.pdf

2) Relative Path
- always starts with current working directory

cpp/Assignments/assign02.pdf

ls -l command's Output

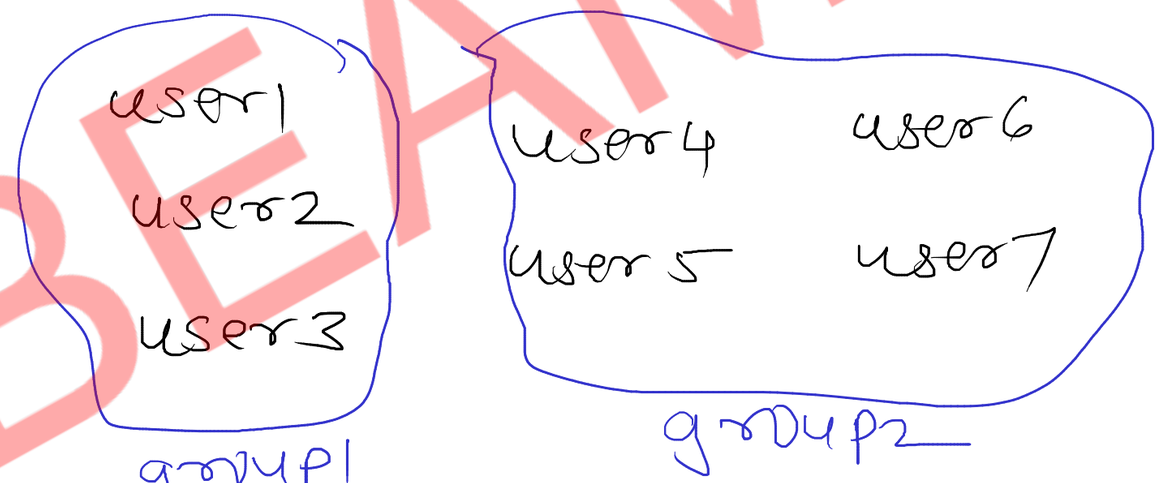
Permissions | user/ Owner | size (bytes) | name

- rw-rw-r-- 1 sunbeam sunbeam 0 Oct 7 09:02 file2.txt

type | linkcount | group | timestamp

Types of file :

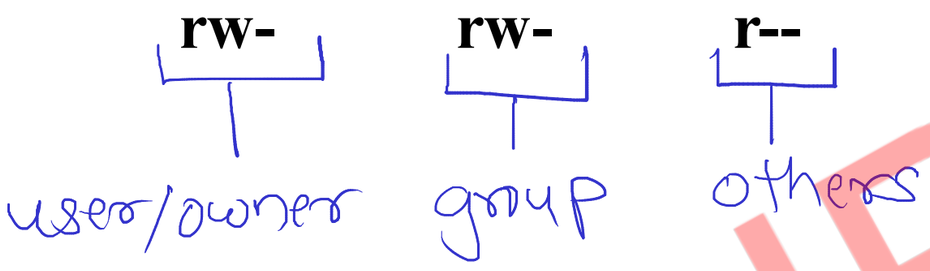
1) Regular file (-)
2) Directory file (d)
3) Link file (l)
4) pipe file (p)
5) Socket file (s)
6) Character special file (c)
7) Block special file (b)



Logged in with - user1
created file with name - file.txt
owner of file.txt - user1
group of file.txt - group1
permissions : rw- rw- r--
rw- : user1
rw- : user2 & user3
r-- : user4 to user7

File Permissions

r - read
w - write
x - execute



Permissions in Numeric Format

1 — present
0 — absent

0664