Muhammad Rizal Alfaridzi
1706019740

Overview: Directory - Files & Path

1. It has the standard named FHS (File Hierarchy Standard)
   It all started with (based on the hierarchy standard) :
   1.) **"/"** : Primary hierarchy root
   2.) **"/bin"** : Essential command binaries that need to be available in single user mode; for all users, e.g., cat, ls, cp.
   3.) **"/boot"** : Boot loader files, e.g., kernels, initrd.
   4.) **"/dev"** : Essential device files, e.g., /dev/null.
   5.) **"/etc"** : Host-specific system-wide configuration files. Contains configuration files required by all programs.
   6.) **"/home"** : Users' home directories, containing saved files, personal settings, etc.
   7.) **"/lib"** **:** Libraries essential for the binaries in /bin/ and /sbin/.
   8.) **"/media"** : Mount points for removable media such as CD-ROMs (appeared in FHS-2.3).
   9.) **"/mnt"** : Temporarily mounted system.
   10.) **"/opt"** : Optional application software packages
   11.) **"/sbin"**: Essential system binaries, e.g., fsck, init, route.
   12.) **"/srv"** : Site-specific data served by this system, such as data and scripts for web servers, data offered by FTP servers, and repositories for version control systems.
   13.) **" /tmp"**: Temporary files. Often not preserved between system reboots, and may be severely size restricted.
   14.) **"/usr"** : Secondary hierarchy for read-only user data; contains the majority of (multi-)user utilities and applications.
   15.) **"/proc"** : Virtual filesystem providing process and kernel information as files. In Linux, corresponds to a procfs mount. Generally automatically generated and populated by the system, on the fly.
   16.) **"/var"** : Variable files—files whose content is expected to continually change during normal operation of the system—such as logs, spool files, and temporary e-mail files.
   17.) **"/sys"** : Contains information about devices, drivers, and some kernel features.
   18.) **"/sbin"**: Essential system binaries, e.g., fsck, init, route.
   19.) **"/run"** : Run-time variable data: Information about the running system since last boot, e.g., currently logged-in users and running daemons. Files under this directory must be either removed or truncated at the beginning of the boot process; but this is not necessary on systems that provide this directory as a temporary filesystem (tmpfs).
   20.) **"/root"** : Home directory for the root user.

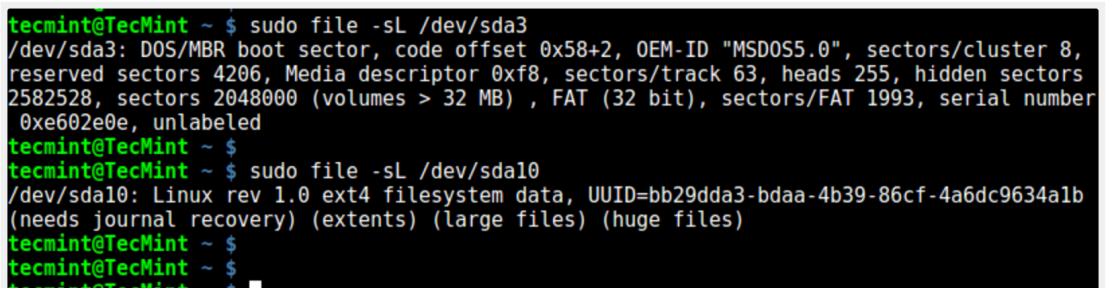
references : https://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard
https://www.geeksforgeeks.org/linux-file-hierarchy-structure/

2. – Absolute path : the specifying the location of a file or directory from the root directory(/).

- Relative Path : Relative path is defined as the path related to the present working directly(pwd). It starts at your current directory and **never starts with a /** .

3. in order to know the file type on GNU/Linux , File command can identifies a file type

```
$ sudo file -sL /dev/sda3
```



reference : https://www.tecmint.com/find-linux-filesystem-type/

4. absolute path is , /home/joko/surat

Part B: ProcFS

1.) The **/proc/partitions** file contains a table with major and minor number of partitioned devices. The **major** number corresponds to the device type (or driver) and can be found in **/proc/devices. The minor number** is a unique identification of an instance of this **device type.**

2.) - /proc/ioports : shows which I/O ports that are being used at the moment

- /proc/stat : Overall/various statistics about the system, such as the number of page faults since the system was booted.

- /proc/devices : shows a list of device drivers configured into the currently running kernel (block and character).

From my own perspective, /proc/stat is one of the most important entry, because we can check all of the conditions / status of our system (such as number of page fault)

reference : https://www.tldp.org/LDP/Linux-Filesystem-Hierarchy/html/proc.html

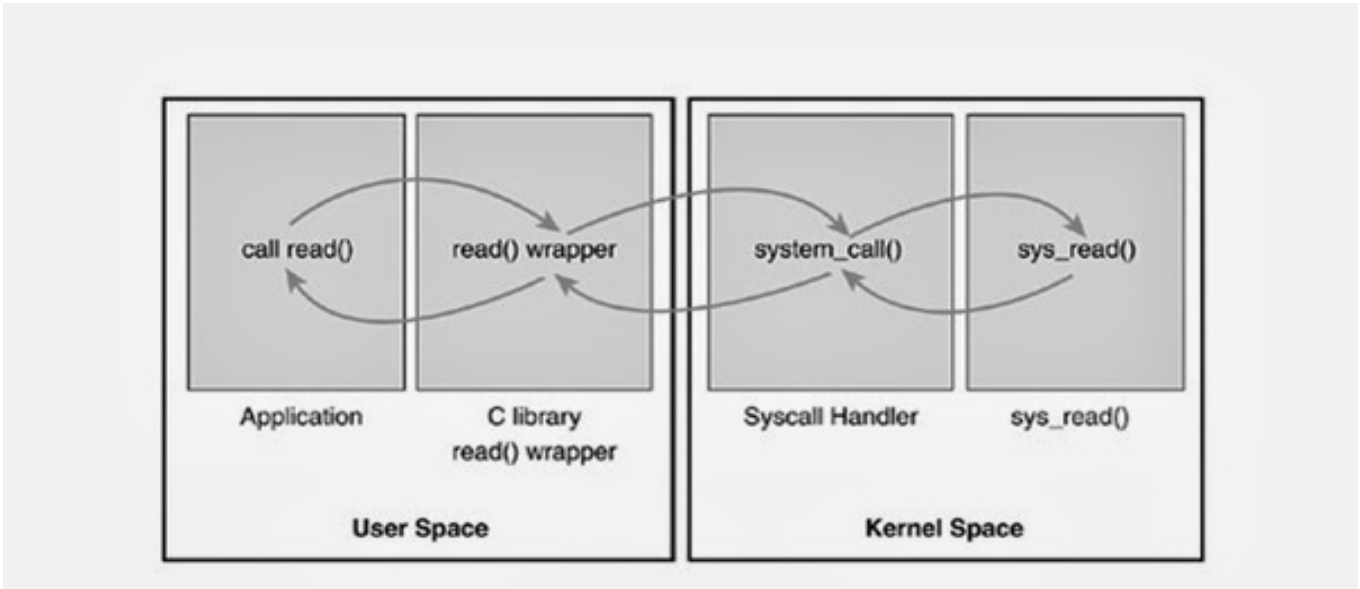
Part C: System Calls

1.) On the user space, the user process makes a call to the wrapper function in the C library.

A software interrupt takes place to switch the system to the kernel mode in order to user-space applications can execute kernel code directly.

Each system call is assigned to a syscall number (a unique number to reference a specific syscall). Before the system switches to kernel mode, it stores the syscall number and system call parameters in some registers. When a user-space process executed, the syscall number identified which syscall instruction that wanted to be executed.

After syscall number is put to the registers, the system call handler is called (an interrupt). Then, the handler reads the syscall number from the register to verifies and calls that specific syscall and runs in the kernel-space. After the syscall executed, the code will be stored in some registers. (see figure below for illustrasion)



2.) **Gettimeofday()** obtain the current time, expressed as seconds and microseconds since the Epoch, and store it in the timeval structure

***getrusage*() (int getrusage(int who, struct rusage *r_usage);**) function shall provide measures of the resources used by the current process or its terminated and waited-for child processes. If the value of the *who* argument is RUSAGE_SELF, information shall be returned about resources used by the current process. If the value of the *who* argument is RUSAGE_CHILDREN, information shall be returned about resources used by the terminated and waited-for children of the current process. If the child is never waited for (for example, if the parent has SA_NOCLDWAIT set or sets SIGCHLD to SIG_IGN), the resource information for the child process is discarded and not included in the resource information provided by *getrusage*().

```
muhammad.rizal71@badak:~$ ./time 1000000000
2241151
muhammad.rizal71@badak:~$ ./time 0
0
muhammad.rizal71@badak:~$ ./time 100
1
muhammad.rizal71@badak:~$ ./time 10
0
```

```
muhammad.rizal71@badak:~$ ./checkusage
CPU Time: 0.000000 sec user, 0.000000 sec system
```