# NOTES 5

## File system

- Description = The method and structure used to store and organize files on a disk. It defines how data is stored and retrieved, allowing users to create, delete, and manage files and directories.
- usage = It is used to manage files and directories, providing a hierarchical structure for data organization. Linux uses the hierarchical directory structure (the one that looks like tree.)
- Example = linux root file system is denoted by `/` and directories like `/home`, `/etc` and `/usr` are some part of this structure.

## Pathname

- Description = A pathname is a string that specifies the location of a file or directory in the file system. It can be absolute or relative.
- usage = Pathnames are used to navigate the file system and access files or directories.

## Absolute path

- Description = An absolute path is a complete path from the root directory to a specific file or directory. It starts with the `/` character.
- usage = It is used when you want to specify the exact location of a file or directory, regardless of the current working directory
- Example = `home/john/Downloads/song.mp3`

## Relative path

- Description = A relative path is a path that is relative to the current working directory. It does not start with a `/`.

- usage = It is used for convenience when you want to access files or directories without specifying the full path.

- Example = `Downloads/song.mp3`

## The difference between your home directory and the home directory

- Description = Your home directory is a personal directory assigned to a specific user, while the home directory refers to the general concept of user home directories in the system.
- usage = Your home directory is where your personal files and configurations are stored, typically located at `/home/username`. The home directory concept applies to all users on the system.
- Example = For the user `prisha`, her home directory is `/home/prisha`, while the home directory for all users is `/home`.

## parent directory

- Description = The parent directory is the directory that contains the current directory. It is one level up in the hierarchy.

- usage = It is used to navigate back to the previous level in the directory structure.
- Example = If you are in `/home/user/documents`, the parent directory is /`home/user`.

## child directory or subdirectory

- Description = A child directory (or subdirectory) is a directory that is located within another directory (the parent directory)
- usage = used to organize files and directories hierarchically.
- Example = `/home/user/documents`, `reports` can be a child directory, making the path `/home/user/documents/reports`

## Bash special characters

- Description =Bash special characters are characters that have special meanings in the shell. They control the behavior of commands and scripts.
- usage = Common special characters include `*` (wildcard), `?` (single-character wildcard), `~` (home directory), `$` (variables), and & (background process).
- Example = he command `ls *.txt` lists all files ending with `.txt` in the current directory, using `*` as a wildcard.

## environment variables

- Description = dynamic values that affect the behavior of processes on a system. They are used to store system settings and user preferences.
- usage = They can be used to configure the shell environment, such as `PATH`, `HOME`, and `USER`
- Example = The command `echo $HOME` prints the home directory of the current user.

## user defined variables

- Description = User-defined variables are variables created by the user to store data for use in scripts or commands.
- usage = They allow users to store temporary data or parameters for scripts.
- Example = define a variable with `my_var="Hello World"` and access it with `$my_var`.

## Why do we need use $ with variables in bash shell scripting?

- Description = The `$` symbol is used to reference the value of a variable in Bash. It tells the shell to retrieve the value stored in that variable.
- usage = Using `$` allows you to access and manipulate the value of the variable in commands and scripts.