

Notes 2: The Linux File System

- [Presentation](#)
- [Article](#)

File System

The way files are stored and organized.

pathname

A pathname indicates the location of the file in the file system (like an address)

Absolute path

Absolute path is the location of a file starting at the root of the file system. *Example:*

`/home/john/Downloads/song.mp3`

Relative path

Relative path is the location of a file starting from the current working directory or a directory that is located inside the current working directory. *Example:* `Downloads/song.mp3`

The difference between YOUR HOME directory and THE HOME directory:

THE HOME directory is the parent directory of all the home directories. This is where all the users' home directories are. The absolute path of this directory is `/home`. Noticed that it starts at the root. YOUR HOME directory is your user's personal directory where all your files are located. Every user has their own home directory just like in an apartment complex they all residents have their own apartment while sharing the common areas. You have total ownership of your home directory but outside of the home directory only the root user can make changes. An example absolute path, assuming that user name is maria53, would be

`/home/maria53`

Parent directory

A directory containing one or more directories and files.

Child directory or subdirectory

A better name for this is a subdirectory or subfolder. This is a directory inside another directory.

Bash special characters

Special characters are function like commands that tell the shell to perform a specific action without having to type the complete command. These special characters make working on the command line more efficient.

- `.` (**single period**): represents the current directory.

- **.. (2 consecutive periods)**: represents the parent directory.
- **~ (tilde character)**: expands the current user's home directory. It is like a variable that the shell uses to store the absolute path of the user's home directory. This ~/Downloads is the same as typing /home/maria53/Downloads
- **/ (one forward slash)**: as mentioned earlier, this is the root directory and the shortest path in the system. This is the beginning of the directory tree. There is nothing before it and everything after it.
- **- (hyphen-minus)**: is used to move to the previous current working directory.
- **# (hash or number sign)**: This is used for single line comments in shell scripting.
- **! (single exclamation mark)**: used for repeating a command from the history. For example !5 will repeat the 5th command in the command history. To view the entire command history type history.
- **!! (2 consecutive exclamation marks)**: are used for repeating the previous command. For example, !! will repeat the previous command while, sudo !! will repeat the previous command but will add sudo at the beginning of the command. This is useful for times when we forget to type sudo when performing administrative tasks.

Environment variables

Environment variables store values of a user's environment and can be used in commands in the shell. These values can be unique to the user's environment which makes them ideal when writing commands that you want to use regardless of which user is using the computer. To see a list of your environment variables type `env`. To use the value stored in an environment variable you must prepend the variable name with a `$`. Here are some useful environment variables:

```
* ` $USER ` = stores the current's user username
* ` $HOME ` = stores the absolute path of current's user home directory
* ` $PWD ` = stores the absolute path of the present working directory.
* ` $OLDPWD ` = stores the absolute path of the previous current working
directory
```

User defined variables

User defined variables are created by the user and exist only in the script and subshell that runs the script. They allow you the user to temporarily store data and use it throughout the script.

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

The `$` symbol is used to reference the value stored in a variable in bash shell scripting.