

Notes 2: The Linux Files System

File System

- The way files are stored and organized.

Child and Parent Directories.

- **Child Directory:** Anytime You go forward into a directory you are going to the child directory or subdirectory.
- **Parent Directory:** Anytime your going backwards, you are going to the parent dircetory.

Pathname

- Indicates the location fo the file in the filesystem.

Types of Pathnames:

- **Absolute Path**
 - The location of a file starting at the root of the file system.
 - Example: */home/adam/Downloads/Song1.wav*
- **Relative Path**
 - The location of the file starting from the current working directory or a directory that is located inside the current working directory.
 - Example: *Downloads/Song1.wav*

The Difference Between My Home and The Home.

- **My Home:**
 - Is the users home where all the users personal files go.
- **The Home:**
 - Is where all the users' directories are.

Bash Special Characters

Special characters are function like commands that tell the shell to perfomr a specific action without having to type the complete command.

- **.(Single Period):** Represents current directory
- **..(2 periods):** Represents the parent directory
- **~(Tilde):** Expands the current users home directory.
 - This *~/Downlaods* is the same as */home/adam/Downloads*
- **/ (forward slash):** The root directory and the shortest path in the system.
- **-(hyophen-minus):** Used to move to the previous working directory.
- **#(hash):** This is used for single line comments in shell scripting.
- **!(exclamation):** Used for repeating a command from history.
 - *!5* repeats the 5th command in history.

- **!!(2 exclamations):** Used for repeating the previous command.
 - `sudo !!` will repeat the previous command but will add `sudo` at the beginning of the command.

Bash Environment Variables

A Variable is a place to store data.

An Environment Variable stores values of a users environment and can be used in commands in the shell. The \$ is used to use variables stored in the environment variable

- **\$USER:** Stores current user username
- **\$HOME:** Stores the absolute path of current user homed irectory
- **\$PWD:** Stores the absolute path of the present working directory
- **\$OLDPWD:** Stores the absolute path of the previous working directory.

Bash User Variables

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

- **Rules:**
 - User variables can be any text string of up to 20 letters, digits, or underscore but can not start with a number.
 - They are case sensitive.
 - Values are assigned with an equal sign with no spaces.
 - the shell stores all values as text strings; Bash is essentially untyped.

The '\$' Drawback

If '\$' is used in a variable infront of a number it needs to be escaped with a '\' because '\$0' to '\$9' are reserved variables.

File System Commands

- **PWD**
 - *Usage:* Displays the absolute path of current working directory.
 - *Formula:* `pwd`
- **CD**
 - *Usage:* Changes the current working directory.
 - *Formula:* `cd + distenation(absolute or relative path).`
 - *Example:*
 - `"cd"`: takes you home
 - `"cd /usr/share/themes"`: takes you to a specified directory with absolute path
 - `"cd adam/Downloads"`: Goes to a directory with relative path from /home directory.
- **LS**
 - *Usage:* Lists files and directories.
 - *Formula:* `ls + option + directory to list`
 - *Example:*
 - `"ls --help"`: see all the optinos for ls command

- `"ls"`: List the current directory.
- `"ls -A"`: List files including hidden files in current directory.

- **Tree**

- *Usage*: Displays the recursive directory listing of files.
- *Formula*: `tree + option + directory`
- *Example*:
 - `"tree ~"`: Displays the home directory tree
 - `"tree -p ~/particles"`: Displays a tree of a directory with premissions.
 - `"tree -pughD ~/particles/"`: Displyas a tree of directory with full path prefix, with user and group owner, human readable file sizes, and the date of last modified.

- **EXA**

- *Usage*: Used the same way as `ls`. A modern replacement for `ls` written in rust.
- *Formula*: `exa + option + directory`
- *Example*:
 - `"exa -1 ~/particles"`: List all files in a given directory in a single line
 - `"exa -l ~/praticles"`: Long list all the files in a given directory.
 - `"exa -lh ~/Particles"`: Long list all files in a given directory with headers.