Notes 2: The Linux FS

Commands

cd(Change Directory)

- Description: Changes the current working directory.
- Usage: `cd [directory]`
- Examples:
  - `cd Documents`
  - `cd ..`
  - `cd /`

ls(List directory contents)

- Description: Lists files and directories in the current directories.
- Usage: `ls [options] [directory]`
- Examples:
  - `ls`
  - `ls -l`
  - `ls /etc`

pwd(Print working directory)

- Description: Displays the current working directory.
- Usage: `pwd`
- Examples:
  - `pwd`
  - `cd Documents && pwd`
  - `pwd -P`

mkdir(Make directory)

- Description: Creates a new directory.
- Usage: `mkdir [directory]`
- Examples:
  - `mkdir Documents`
  - `mkdir -p Parent/Child`
  - `mkdir -m 777 Temp`

rmdir(Remove directory)

- Description: Deletes an empty directory.
- Usage: `rmdir [directory]`
- Examples:

    - `rmdir Temp`
    - `rmdir -p Parent/Child`
    - `rmdir --ignore-fail-on-non-empty NonEmptyDir`

cp(Copy)

- Description: Copies files and directories.
- Usage: `cp [options] source destination`
- Examples:
    - `cp file1.txt file2.txt`
    - `cp -r directory1 directory2`
    - `cp -v file1.txt /tmp`

mv(Move)

- Description: Moves or renames files and directories.
- Usage: `mv [options] source destination`
- Examples:
    - `mv file1.txt file2.txt`
    - `mv directory1 /tmp`
    - `mv -i file1.txt directory1`

rm(Remove)

- Description: Removes files or directories.
- Usage: `rm [options] file/directory`
- Examples:
    - `rm file.txt`
    - `rm -rf directory`
    - `rm -i file.txt`

touch

- Description: Creates an empty file or updates file timestamps.
- Usage: `touch [options] file`
- Examples:
    - `touch file.txt`
    - `touch -d "last week" file.txt`
    - `touch -a -m -t 202203101200.00 file.txt`

cat(Concatenate)

- Description: Displays the contents of a file.
- Usage: `cat [options] file`
- Examples:
    - `cat file.txt`
    - `cat file1.txt file2.txt > merged.txt`

    - `cat -n file.txt`

Terms and Definitions

- **File System**
    - The way files are stored and organized.
- **pathname**
    - Indicates the location of a file name in a file system.
- **Absolute path**
    - The location of a file starting at the root of the file system.
- **Relative path**
    - The location of a file starting from the current working directory or a directory that is inside the current working directory.
- **What is the difference between YOUR HOME directory and THE HOME directory?**
    - "YOUR HOME directory" refers specifically to the home directory of the currently logged-in user, "THE HOME directory" refers to the parent directory containing all user home directories on the system.
- **parent directory**
    - The directory containing the current directory.
- **Child directory/subdirectory**
    - The directory located within the current directory.
- **Bash special characters**
    - Symbols with specific functions in the Bash shell, used for tasks like redirection, wildcard expansion, and command substitution, essential for efficient command-line operations in Linux (*, ?, |, <>, (), etc.).
- **Environment variables**
    - Dynamic named values that contain information used by operating system processes and applications, facilitating communication and configuration across the system. (Ex: PATH and HOME).
- **User defined variables**
    - Custom variables created and named by users within programming scripts or command-line environments to store and manipulate data temporarily.
- **Why do we need use $ with variables in bash shell scripting?**
    - This sign must be used with variables because it is used to access the value stored in a variable. When used with a variable name, it tells the shell to substitute the variable's value in its place. This is crucial for distinguishing variable names from regular text within the script. Without the $ symbol, bash would treat the variable name as literal text rather than substituting its value.