Practical's

Assignment No. 1

Demonstration of Linux commands with attributes: - pwd, cd, ls, more, less, echo, clear, kill, ps, man, cal, date, who, who am I, wc, mkdir, rmdir, rm, sort.

1. **ls** (List) :
   - **-l:** Long format, displays detailed information about files, including permissions, owner, size, modification date, etc.
   - **-a:** All, shows hidden files starting with a dot (e.g., bashrc).

   Syntax $ ls
2. **cd** (Change Directory) :
   - **directory_name:** Change to the specified directory.
   - ...: Move one level up in the directory hierarchy.
   - -: Switch to the previous directory.

   Syntax $ cd ..
3. **pwd** (Print Working Directory):
   - Displays the current working directory's full path.

   Syntax $ pwd
4. **cp** (Copy):
   - **-r:** Recursive, copy directories and their contents.
   - **-i:** Interactive, prompt before overwriting existing files.

   Syntax $ cp source destination
5. **mv** (Move or Rename):
   - **-i:** Interactive, prompt before overwriting existing files.

   Syntax $ mv source destination
6. **rm** (Remove) :
   - **-r:** Recursive, remove directories and their contents.
   - **-f:** Force, suppress prompt before removing write-protected files.

   Syntax $ rm filename
7. **mkdir** (Make Directory):
   - **-p:** Create parent directories if they don't exist.

   Syntax $ mkdir directoryname
8. **rmdir** (Remove Directory):
   - Removes empty directories.

Syntax $ rmdir directoryname

9. **cat** (Concatenate):
   - Displays the contents of a file.

     Syntax $ cat filename

10. **less :**
    - A pager program for viewing file content interactively.

      Syntax $ less filename

11. **head and tail :**
    - Display the first or last few lines of a file.
    - **-n**: Specify the number of lines to display (e.g., **head -n 10 file.txt**).

      Syntax $ head –n no filename

      Syntax $ tail –n no filename

ls - The **ls** command lists files and directories within a system. Running it without a flag or parameter will show the current working directory's content.

To see other directories' content, type **ls** followed by the desired path. For example, to view files in the **Documents** folder, enter:

ls /home/username/Documents

Here are some options you can use with the **ls** command :

- **ls -R** lists all the files in the subdirectories.
- **ls -a** shows hidden files in addition to the visible ones.
- **ls -lh** shows the file sizes in easily readable formats, such as MB, G

$ ls :

more command is used to view the text files in the command prompt, displaying one screen at a time in case the file is large (For example log files). The more command also allows the user do scroll up and down through the page. The syntax along with options and command is as follows. Another application of more is to use it with some other command after a pipe. When the output is large, we can use more command to see output one by one.

$ more [-options] [-num] [+/pattern] [+linenum] [file_name]

echo command in Linux is a built-in command that allows users to display lines of text or strings that are passed as arguments. It is commonly used in shell scripts and batch files to output status text to the screen or a file.

- Syntax of `echo` command in Linux

$ echo [option] [string]

clear (clear screen) Before you start the next section, you may like to clear the terminal window of the previous commands so the output of the following commands can be clearly understood. At the prompt, type

$ clear

wc (word count) A handy little utility is the wc command, short for word count. To do a word count on science.txt, type wc -w science.txt To find out how many lines the file has, type wc -l science.txt University of Leicester | Tutorial Two 11 To find out how many characters the file has, type

$ wc -m science.txt

sort command read input from STDIN (the keyboard) and produces an alphabetically or numerically sort list. Type sort Then type in the names of some vegetables. Press Return after each one, and hit control-d after the last entry to return to the shell.

To see who is on the system with you, type who One method to get a sorted list of names is to type,

$ sort filename

man-There are on-line manuals which gives information about most commands. The manual pages tell you which options a particular command can take, and how each option modifies the behaviour of the command.

$ man

kill (terminate or signal a process) It is sometimes necessary to kill a process (for example, when an executing program is in an infinite loop) To kill a job running in the foreground, type ^c (control-c). For example, run sleep 100 ^c to kill a suspended or background process, type

$ kill %job number

ps (process status) Alternatively, processes can be killed by finding their process numbers (PIDs) and using

$ kill PID_number

***