



Linux Essentials

Dr. Hatem

Eng. Somaya Yasser

Pwd & Ls Commands

- Pwd stands for Print Working Directory.
- It prints the path of the working directory, starting from the root
- Ls: lists directory contents of files and directories
- ls [option] [directory]
- -l option: known as a long format that displays detailed information about files and directories.
- -a option: Represent all files Include hidden files and directories in the listing.



Cd & mkdir Commands

- Cd: the **change directory** command.
- It is used to move efficiently from the current working directory to different directories in our System.
- **cd [directory]**
- Mkdir: **create directories (also referred to as folders in some operating systems)**
- **can create multiple directories at once.**
- **mkdir [directories ...]**



Man, help & whatis commands

- `man` command in Linux is used to display the user manual of any command that we can run on the terminal.
- `$ man [COMMAND NAME]`
- Help displays information about shell built-in commands.
- `$ help [COMMAND NAME]`
- `whatis` is used to get a one-line manual page description.
- `whatis [command_name]`

Creating files commands: touch, cat

- **touch** command: It is used to **create a file without any content**. The file created using the touch command is **empty**.
- **touch file_name**
- **Cat**: It reads data from the file and gives **its content as output**. It helps us to **create, view, and concatenate** files.
- To **view** single file ---- **cat file_name**
- **Create a file and add content** ----- **cat > newfile_name**

Cp & mv Commands

- is used to copy files or groups of files or directories.
- cp Source Destination
- cp Source Directory

- Mv: Renaming a file or directory.
- Mv: Moving a file or directory to another location(cut)

- mv [source_file_name] [Destination_file_name] ----- rename
- mv [source_file_name] [Destination_path] ----- cut



Removing files & folders

- The `rmdir` command is useful when you want to remove the empty directories.
- `rmdir <directory>`
- `rm` command is used to remove objects such as files, directories.
- To remove file: `rm FILE`
- To remove directory: `rm -r folder`

Nano Text Editor

- To check it's already installed use this command `nano --version`
- For installation `sudo apt install nano`
- to `open` Nano and to `edit` a certain file is: `nano filename`
- Press `CTRL + O` to `save` the changes made in the file and continue editing.
- To `exit` from the editor, press `CTRL + X`.
- If there are changes, it will ask you whether to save them or not. Input Y for Yes, or N for No, then press Enter. But if there are no changes, you will exit the editor right away.

Nano Text Editor (cont')

- To **search** in the text, press **CTRL + W**. Insert your value and press Enter.
- To **select** text, go to the beginning of the desired text and press **ALT + A**.
- Press **ALT + 6** to **copy** the selected text to the clipboard.
- **CTRL + K** , It **cuts** the entire selected line to the cut buffer (similar to clipboard).
- **CTRL + U** , To **paste** the text from the **cut** buffer into the selected line.
- **CTRL + A** , Lets you jump to the **beginning** of the line.
- **CTRL + E** , Lets you jump to the **end** of the line.

Nano Text Editor (cont')

- ➡ CTRL + G, Help window will pop out and show you all the available commands.
- ➡ CTRL + R, Opens a file and inserts it at the current cursor position.
- ➡ CTRL + T, Browse which file to open.
- ➡ Nano +number of line filename. To go to line number



Redirection operator

1. **Overwrite** Redirection:

- Overwrite redirection is useful when you want to **store/save** the **output of a command** to a file and replace all the existing content of that file.
- for example, if you run a command that gives a report, and you want to save the report to the existing file of the previous report you can use overwrite redirection to do this.
- **">"** standard output
- **"<"** standard input



Examples of overwrite

- `cat > file.txt`
- `cat < file.txt`
- `ls -al > filename`
- `cat filename1 > filename2`
- `Tr char1 char2 < filename`

Append Redirection

- With the help of this Redirection, you can **append** the output to the file without compromising the existing data of the file.
- “>>” standard output
- `cat >> file.txt`
- `echo “this is the second line” >> file.txt`

Error Redirection

- The standard error can also be redirected so that error messages do not clutter up the output of the program.
- `command 2> file`
- `eccho "gfg" 2>file.txt`
- `2>&` is used both standard error and standard output get redirected to the same file.
- `command > file 2>&`
- `eccho "gfg" >error.txt 2>&1`
- `echo "gfg" >error.txt 2>&1`

Piping operator

- ➡ The pipe is used to combine two or more commands, and in this, the output of one command acts as input to another command, and this command's output may act as input to the next command, and so on.

- ➡ `command_1 | command_2 | command_3 | | command_N`

- ➡ List all files and directories and give them as input to `grep` command using piping in Linux

```
ls | grep file.txt
```

- ➡ `$ ls -l | more`
- ➡ `$ cat sample2.txt | head -7 | tail -5`
- ➡ `ls | grep 'file' > geeks.txt`
- ➡ `$ cat contacts.txt | sort`
- ➡ `$ cat contacts.txt | sort | uniq`

Filtering

- filters are the set of commands that take input from standard input stream , perform some operations and write output to standard output stream i.e. Stdout.
- **grep Command**: It is a pattern or expression matching command. It searches for a pattern or regular expression that matches in files or directories and then prints found matches.
- **\$grep "pattern to be matched" filename**
- **sort Command**: sorts lines of text alphabetically or numerically, default sorting is alphabetical.
- **sort filename**

Filtering (cont')

- **More** command: It displays the text file contents on the terminal with **paging** controls
- **\$more filename**
- **head** : Displays the **first n lines** of the specified text files. If the number of lines is not specified then by **default prints first 10** lines.
- **head [-number_of_lines_to_print] [path]**
- **tail**: Displays the **last n lines** of the specified text files. If the number of lines is not specified then by default prints **last10 lines**.
- **Tail [-number_of_lines_to_print] [path]**

Filtering (cont')

- `uniq`: Removes duplicate lines.
- `uniq [path]`
- `wc`: gives the number of lines, words and characters in the data.
- `Wc [path]`
- `Sed`: print data from line to another line.
- `Sed -n 'line1,line2p' filename`