Basic Linux Commands

Most used commands:

- **man** (manual page, e.g., man rm)
- **cd** (change directory)
- **ls** (list contents of directory, see man ls for more options, e.g., ls -1, just print file names)
- **mv** (move/rename file or directory)
- **cp** (to copy files)
- **clear** (clears terminal window)
- **rm** (remove file or directory)
- **echo** (print something to stdout)
- **cat** (print an entire file to stdout)

Man Command

The first command you should learn in Linux is “man”. Using this command you can get the usage and description of all Linux commands. For example, if you want to know about “ls” command and its options, just execute “man ls” command in the terminal to list its usage and description.

```
Syntax: man <command></command>
```

```
$man ls
```

Touch And Cat Command

- Touch command is used to create any type of file in Linux systems with “0” size. As a developer , when working with Linux you might want to create files in the server. You can make use of touch command to do that.

```
Syntax: touch
touch demo.txt
```

- Cat command is used to view the contents of a file. You cannot edit the contents of the file using cat. It just gives a view of the file. cat doesn’t support scrolling using keyboard.

```
Syntax: cat
cat demo.txt
```

Sort Command

Sort is used to sort the contents of a file. Create a file named test.txt and copy the following contents on to the file to test the sort command.

```
$cat test.txt
1 mike level intermediate jan
10 lucy level beginer mar
45 Dave level expert dec
4 dennis start beginner jul
7 Megan employee trainee feb
58 Mathew Head CEO nov
```

In the above example, second column has the names. So if you want to sort the names alphabetically use “-k” flag with the column location. It would be “-k2”.

Sort Command Example

```
Syntax: sort
sort -k2 test.txt

$sort -k2 test.txt
45 Dave level expert dec
4 dennis start beginner jul
10 lucy level beginer mar
58 Mathew Head CEO nov
7 Megan employee trainee feb
1 mike level intermediate jan
```

Exercise:

The first column has numbers. If you want to sort numbers, use "-h" flag. If the numbers are in different column, you can use the "-k" flag along with "-h" flag.

```
$sort -h test.txt
```

The last column has months. You can sort a file based on month using "-M" flag.

```
$sort -k5 -M test.txt
```

Grep Command

Grep command is used for searching specific string patterns in a file as well as the standard output (STDIN). We will look into few file based operations.

```
Syntax: grep ""
        grep "Mathew" test.txt
```

```
$grep "dennis" test.txt
4 dennis start beginner jul
```

The above command gives the output including the sub-string. If you want to search for individual words, you need to add “-i” flag to the grep command.

Grep Command Example

Also you can search for a string or a pattern in multiple files using a single grep command

```
$grep "dennis" test1.txt test2.txt test3.txt
```

You can also use regular expressions for matching the string.

Sed Command

sed is a text-editor which can perform editing operations in a non-interactive way. Sed command gets its input from a standard input or a file to perform the editing operation on a file. Sed is a very powerful utility and you can do a lot of file manipulations using sed.

I you want to replace a text in a file by searching it in a file, you can use the sed command with substitute “s” flag to search for the specific pattern and change it.

```
Syntax: sed 's///' test.txt
```

For example, lets replace “mike” in test.txt file to “michael

```
sed 's/mike/michael/' test.txt
```

Sed Command Example

You can use any character as a delimiter for substitution. For example, if you want to make changes to a url, you need to have a different delimiter because the url already have slashes. So you can substitute like the following.

```
$echo "http://www.example.uk/main.html" | sed 's_uk/main_com/index_'
http://www.example.com/index.html
```

You can also replace a line by matching a string pattern in the line. “-c” flag is used for replacing text using sed. Lets replace the first line in our test.txt file using the following command.

```
$sed '/1 mike/c 1 michael start beginner mar' test.txt
1 michael start beginner mar
10 lucy level beginer mar
45 Dave level expert dec
```

Tar Command

tar command is used to create and extract archive files. “-cf”, “-xf” and “-zvxf” flags are used for creating and extracting archives.

```
Syntax: tar  &lt;file/folder name&gt;
```

Lets create a tar archive out of test.txt file

```
$tar -cf test.tar test.txt
$ls
test.tar  test.txt
```

Find

find command is used for finding files. You can find the files using its name with “-name” flag.

```
Syntax: find -name test.txt
```

You can also find folder using its name by using”/ -name” flag.

```
$find / -name passwd
/etc/cron.daily/passwd
/etc/pam.d/passwd
/etc/passwd
/usr/share/lintian/overrides/passwd
```

Chmod Command

chmod command is used for changing the read/write/execute permissions of a file. Permissions are represented in numbers as follows.

```
4 - read permission

2 - write permission

1 - execute permission

- no permission
```

To give all permissions on test.txt file, you can use the following chmod command.

```
$chmod 755 test.txt
```

Tips and Tricks

Some of these tips also depend on how the shell is configured. Let’s begin!

- **Using tab for autocompletion**
- **Switch back to the last working directory**

```
$cd -
$cd ~
```

- **List the contents of a directory**

```
$ll
```

- **Running multiple commands in one single command**

  ```
  $command_1; command_2; command_3
  ```

- **Running multiple commands in one single command only if the previous command was successful**

  ```
  $command_1 && command_2
  ```

- **Easily search and use the commands that you had used in the past**

  Imagine a situation where you used a long command couple of minutes/hours ago and you have to use it again. Problem is that you cannot remember the exact command anymore.

  Just use the keys ctrl+r to initiate reverse search and type some part of the command. It will look up into the history and will show you the commands that matches the search term.

  ```
  $ctrl+r search_term
  ```

- **Unfreeze your Linux terminal from accidental Ctrl+S**

  You probably are habitual of using Ctrl+S for saving. But if you use that in Linux terminal, you'll have a frozen terminal.

  Don't worry, you don't have to close the terminal, not anymore. Just use Ctrl+Q and you can use the terminal again.

  ```
  $ctrl+Q
  ```

- **Move to beginning or end of line**

  You can use Home and End keys here of course but alternatively, you can use Ctrl+A to go to the beginning of the line and Ctrl+E to go to the end.

- **Reading a log file in real time**

  In situations where you need to analyze the logs while the application is running, you can use the tail command with -f option.

  ```
  $tail -f path_to_Log
  ```

  You can also use the regular grep options to display only those lines that are meaningful to you:

  ```
  $tail -f path_to_log | grep search_term
  ```

  You can also use the option F here. This will keep the tail running even if the log file is deleted. So if the log file is created again, tail will continue logging.

- **Reading compressed logs without extracting**

  Server logs are usually gzip compressed to save disk space. It creates an issue for the developer or sysadmin analyzing the logs.

  So you get zless, zcat, zgrep etc and you don’t even have to explicitly extract the compressed files