Linux

**Create Files**

* touch <filename> (ex: touch file1.txt – it creates empty file named file1.txt)
* echo “Welcome to Linux” >> file2.txt – it creates the file2.txt with Welcome to Linux text inside the file
* We can also use editors (vi or nano) to create files
* vi editor comes by default in all Linux environments
  + vi file3.txt
* Install nano editor using below command
  + sudo yum install nano -y -- Redhat
  + sudo apt install nano -y -- Debian/Ubuntu
  + Once nano editor is installed, you can create the file using below command
  + nano file4.txt

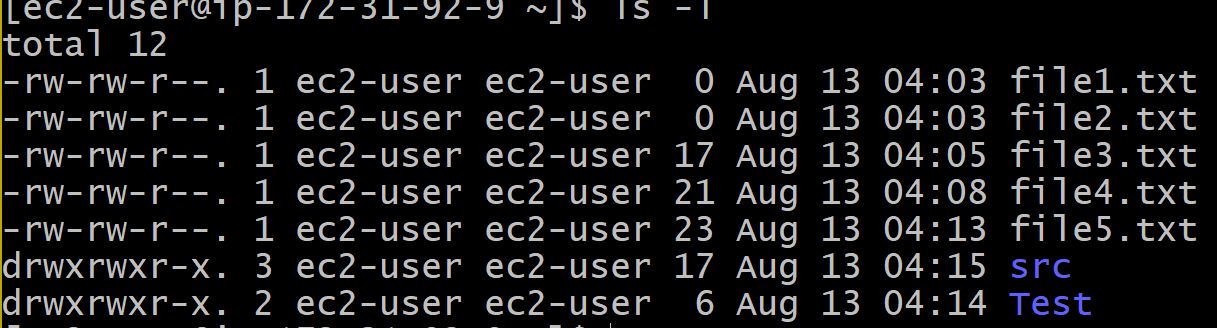
**List Files**

* ls – short listing of files and folders in current directory
* ls -l – Long listing where it shows files folders in a current Directory
* ls -lt -- Show the long list of files and folders in Ascending order of modified time stamp
* ls -l \*.html -- long listing of files that ends with .html
* ls -l test -- Long listing of files and folders from test directory
* ls -la -- Shows the hidden files and folders also
  + Single dot (.) − This represents the current directory.
  + Double dot (..) − This represents the parent directory.

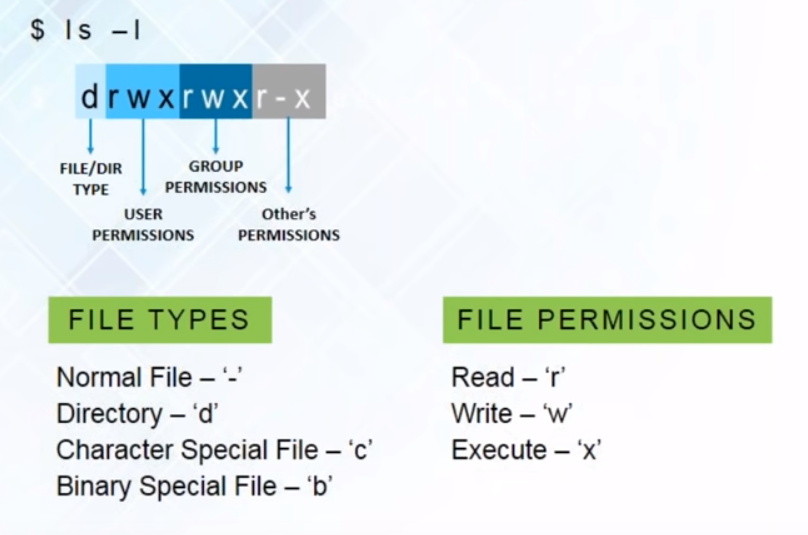
**Directories/Folders**

* mkdir Test -- Creates the directory named **Test**
* mkdir -p src/com/utils -- Creates the directory **src** then creates the directory **com** under **src** then creates the directory **utils** under **com**

**File Management**



* **First Column** − Represents the file type and the permission given on the file. Below is the description of all type of files.



* **Second Column** − Represents the number of memory blocks taken by the file or directory.
* **Third Column** − Represents the owner of the file. This is the Unix/Linux user who created this file.
* **Fourth Column** − Represents the group of the owner. Every Unix/Linux user will have an associated group.
* **Fifth Column** − Represents the file size in bytes.
* **Sixth Column** − Represents the date and the time when this file was created or modified for the last time.
* **Seventh Column** − Represents the file or the directory name.

**Permissions on Files and Folders**

How do I remove the read permissions from others for file2 by using symbolic mode?

**#** chmod o-r file2

This example removes (-) the read (r) permission from others (o) for file2.

Here's another simple example:

How do I grant the read and write permissions to the group for file2?

**#** chmod g+rw file2

This one gives (+) read and write (rw) to the group (g) for file2.

How do I set permissions for a directory and all of its contents by using symbolic mode?

**#** chmod -R u=rwx,g+rw,o-rwx Resources

**Copy Files**

The **cp** command is the primary method for copying files and directories in Linux. Virtually all Linux distributions can use **cp**. The basic format of the command is:

cp [additional\_option] source\_file target\_file

For example:

cp my\_file.txt my\_file2.txt

This Linux command creates a copy of the my\_file.txt file and renames the new file to ***my\_file2.txt***.

By default, the **cp**command runs in the same directory you are working in. However, the same file cannot exist twice in the same directory. You’ll need to change the name of the target file to copy in the same location.

Additional Options

Additional options can be used in combination with the **cp** command:

* **–v  verbose**: shows the progress of multiple copied files
* **–p  preserve**: keeps the same attributes, like creation date and file permissions
* **–f  force**: force the copy by deleting an existing file first
* **–i  interactive**: prompts for confirmation, highly advised
* **–R** **recursive**: copies all files and subfolders in a directory
* **–u update**: copy only if source is newer than destination

### How to Copy File to Another Directory in Linux

To copy a file from the directory you’re working in to a different location, use the command:

cp my\_file.txt /new\_directory

You don’t need to rename the file unless there’s already one with the same name in the target directory.

To specify a path for the source file:

cp /etc/my\_file.txt /new\_directory

This lets you copy without having to change directories. The **cp**command will create the **/new\_directory** if it doesn’t exist.

To rename and copy a file to a different path:

cp my\_file.txt /new\_directory/my\_file2.txt

### Copy Multiple Files from One Directory to Another in Linux

You may need to copy more than one file at a time.

List each file to be copied before the target directory:

cp my\_file.txt my\_file2.txt my\_file3.txt /new\_directory

This example created a copy of all three files in the **/new\_directory** folder.

Use a wildcard to specify all files that share a string of characters:

cp /pictures/\*.jpg /new\_directory

This would find all the files with the **.jpg** extension in the **/pictures** directory, and copy them into the **/new\_directory**folder.

To**copy an** **entire folder and its subfolders and files**, use the **–R**option:

cp –R /documents /new\_directory

**Move Files**

The mv command (short from move) is used to rename and move and files and directories from one location to another. The syntax for the mv command is as follows:

mv [OPTIONS] SOURCE DESTINATION

Copy

The SOURCE can be one, or more files or directories, and DESTINATION can be a single file or directory.

* When multiple files or directories are given as a SOURCE, the DESTINATION must be a directory. In this case, the SOURCE files are moved to the target directory.
* If you specify a single file as SOURCE, and the DESTINATION target is an existing directory, then the file is moved to the specified directory.
* If you specify a single file as SOURCE, and a single file as DESTINATION target then you’re renaming the file .
* When the SOURCE is a directory and DESTINATION doesn’t exist, SOURCE will be renamed to DESTINATION. Otherwise if DESTINATION exist, it be moved inside the DESTINATION directory.

To move a file or directory, you need to have write permissions on both SOURCE and DESTINATION. Otherwise, you will receive a permission denied error.

For example, to move the file file1 from the current working directory to the /tmp directory you would run:

mv file1 /tmpCopy

To rename a file you need to specify the destination file name:

mv file1 file2Copy

The syntax for moving directories is the same as when moving files. In the following example, if the dir2 directory exists, the command will move dir1 inside dir2. If dir2 doesn’t exist, dir1 will be renamed to dir2:

mv dir1 dir2Copy

## Moving Multiple Files and Directories

To move multiple files and directories, specify the files you want to move as the source. For example, to move the files file1 and file2 to the dir1 directory you would type:

mv file1 file2 dir1Copy

The mv command also allows you to use pattern matching. For example, to move all pdf files from the current directory to the ~/Documents directory, you would use:

mv \*.pdf ~/DocumentsCopy

## mv Command Options

### Prompt before overwriting

By default, if the destination file exists, it will be overwritten. To prompt for confirmation, use the -i option:

mv -i file1 /tmpCopy

mv: overwrite '/tmp/file1'?

Copy

To overwrite the file type y or Y.

### Force overwriting

If you try to overwrite a read-only file, the mv command will prompt you whether you want to overwrite the file:

mv -i file1 /tmpCopy

mv: replace '/tmp/file1', overriding mode 0400 (r--------)?

Copy

To avoid being prompted use the -f options:

mv -f file1 /tmpCopy

This option is especially useful when you need to overwrite multiple read-only files.

### Do not overwrite existing files

The -n option tells mv never to overwrite any existing file:

mv -n file1 /tmpCopy

If a file1 exists the command above will do nothing. Otherwise it will move the file to the /tmp directory.

### Verbose output

Another option that can be useful is -v. When this option is used, the command prints the name of each moved file:

mv -i file1 /tmpCopy

renamed 'file1' -> '/tmp/file1'

**Remove Directories/Folders**

* rmdir test -- Removes the directory named **test**. only if it is empty
* rm -r <Dir-Name> -- Removes the directory along with sub directories and files in it (Recursively)

**Manage Users:**

* To add a user execute adduser user1
* You have to become a root user to add the new user, to become root user execute
  + sudo su
* When you are creating the user in Ubuntu System, it will prompt for the password, Retype Password and additional information
  + You can enter password and Retype password and skip other information by pressing enter button
* When you create user in Redhat system, it simply creates the user without prompting anything then you need to execute below command to set the password
  + passwd user1
* Whenever we create the user, user home directory will created under /home with username
* To exit from root user, type exit
* To swath the user i.e to login as different user, execute
  + su – user1

**Working with tar files**

* tar format is equivalent to zip format in windows
* The command to compress is
  + tar -cvzf mybackup.tar.gz file1.txt file2.txt test
* To untar or Uncompress
  + tar xvzf mybackup.tar.gz

**Processes**

* To see what processes are running, execute below command
  + ps -ef
* To see a specific process, execute below command
  + ps -ef | grep jenkins
* To list down top processes (in terms of resource consumption)
  + top

**Editors**

1. Vi Editor (Available by Default)

<https://www.guru99.com/the-vi-editor.html>

2. Nano Editor

<https://linuxize.com/post/how-to-use-nano-text-editor/>

Install nano editor

sudo yum install nano -y

* **Absolute/Relative Path**
  + Absolute path is the path referred from root location
  + Relative path is referred from current directory

Commands

* pwd – Present Working directory

Packages

**tree** package – it shows the list of files and folders in tree format

Redhat – sudo yum install tree -y

Debian/Ubuntu – sudo apt install tree -y

wget – to download anything from internet, we use wget package

sudo yum install wget -y

sudo apt dist-upgrade

:set number – sets numbers in vi editor

free -g

nproc

find 🡪 finds the name of the file

grep 🡪 Search for a specific pattern in files