



# ABC IT Education

WE'LL TAKE YOU FROM ZERO TO HERO IN A SNAP

# Linux Systems Administration

**Deleting, Copying, Moving,  
Renaming, compressing  
Files and Creating Archives**

# Manipulating Files

- ▶ In this lesson will cover how to delete, copy, move, rename and compress files, and also how to create file archives.
- ▶ **Deleting Files**
- ▶ To delete files, use the the 'rm' command, and 'rm filename' is the simplest form of the command. **\$ rm file1**
- ▶ To remove a directory and its contents recursively, use 'rm -r' and the name of the directory, **\$ rm -r dir1**
- ▶ If you want to do a forced removal, use 'rm -f'.
- ▶ Search patterns can also be used to delete multiple files at once, but remember to double check what you're going to delete with a 'ls' command before you actually run the 'rm' command against it.
- ▶ Doing a wildcard delete like 'rm .\*' will not remove 'dot' and 'dot\_dot' i.e. the present directory and the parent directory.

## Removing Files

`rm file` Remove file.

`rm -r dir` Remove the directory and its contents recursively.

`rm -f file` Force removal and never prompt for confirmation.

## Copying Files

- To copy files, use the 'cp' command.
- So to create a copy of a file, you just run 'cp source\_file destination\_file'.
- You copy a file or a series of files to a directory using 'cp' and then listing the file or files ending the line with the directory name.

- ▶ Use 'cp -i', to run cp in interactive mode, that way if the destination file exists, cp will prompt you before overwriting the file.
- ▶ If you use 'cp -r source\_directory destination', 'cp' will recursively copy source directory into the destination directory.
- ▶ Note that if the destination directory doesn't exist, it will create the destination directory with the contents of the source directory.

## **Copying Files**

```
cp source_file destination_file
```

Copy source\_file to destination\_file.

```
cp src_file1 [src_fileN ...] dest_dir
```

Copy source\_files to destination\_directory.

## **cp Options**

`cp -i`

Run in interactive mode.

`cp -r source_directory destination`

Copy src\_directory recursively to destination.

### **Moving/Renaming Files**

- ▶ The 'mv' command moves or renames files, so if you were to run 'mv source destination', 'move' will move the source into the destination, otherwise, source will be renamed destination.
- ▶ Like 'cp', 'mv' has an interactive mode, so if you were to run 'mv -i source destination' and destination existed, 'mv' will ask you if you want to overwrite the destination.

- ▶ Use the move command to rename a directory dir1 to firstdir. **\$ mv dir1 firstdir** and 'dir' has been renamed to 'firstdir'.
- ▶ Let's do the same thing with a file by renaming 'file1' to 'file1.renamed' **\$ mv file1 file1.renamed**
- ▶ Now let's move file 'file1.renamed' into directory 'firstdir'.
- ▶ **\$ mv file1.renamed firstdir**, the file is now in 'firstdir'.
- ▶ This is the contents of file1, this is the contents of file2.
- ▶ If file1 and file2 both exist and we run '**mv file1 file2**', that will rename file1 to file2, file1 will overwrite file2, and file1 is gone.
- ▶ To do this in interactive mode, use the '-i' option and the system says, 'file exist. You want to overwrite it or not'?

## **Moving and Renaming Files**

**mv**      Move or rename files and directories.

**mv source destination**

**mv -i source destination**

- ▶ **Sort text in files**
- ▶ We used the 'sort' command before in this course, but as a reminder, the sort command at its most basic level, sorts text alphabetically that appears in a file.
- ▶ Sort can be used with a '-k' option and a supplied field, so you can sort by a different field than the first field.
- ▶ The '-r' option sorts in reverse order

- ▶ The '-u' option removes duplicate lines and only provides unique results. **\$ sort -u sortfile**
- ▶ Let's look at the contents of 'sortfile', if we sort this file, it will sort on the first column.
- ▶ The file 'sortfile' has 2 duplicate lines, to remove those, you can run sort with a '-u' for unique. **\$ sort -u sortfile**
- ▶ We see that we only get 1 'lions' line and one 'Fish' line
- ▶ To reverse the sort, use the '-r' option, **\$ sort -r sortfile**
- ▶ And to sort by the second column and make it unique use the '-k' and the '-u' options **\$ sort -u -k2 sortfile**, and file is sorted by the second column

## Sorting Data

sort file

Sort text in file.

## **sort Options**

- k F      Sort by key. F is the field number.
- r          Sort in reverse order.
- u          Sort unique.

## **Creating Archives**

- We use the 'tar' command to bundle a group of files and/or directories together in an archive, as you may want to create a copy or back up of a group of files.
- You may also have several files that you want to transfer at once or transfer as a set, the 'tar' command will help.
- You'll notice that tar does not require a hyphen to proceed its arguments, so the hyphen or the dash is in brackets and may be used or omitted and 'tar' will still work

- So 'tar cf file.tar' is the same as 'tar -cf file.tar'.

## Creating a Collection of Files

`tar [-] c|x|t f tarfile [pattern]`

Create, extract or list contents of a tar archive using pattern, if supplied.

## tar Options

c	Create a tar archive.
x	Extract files from the archive.
t	Display the table of contents (list).
v	Be verbose.
z	Use compression.
f file	Use this file.

- Let's create a tar archive of `clsreports`' directory, so `tar`, '`c`' for create, '`f`' for the file we want to use, use '`cls.tar`' and then supply the directory that you want to archive.
- `$ tar cf cls.tar clsreports`**
- We can use '`t`' for a list the contents of an archive file
- `$ tar tf cls.tar`**
- And it shows the directory and all the files in the '`cls.tar`' file
- Copy the `cls.tar` file in the `/tmp` directory and extract it.
- `$ tar xf cls.tar`** and the contents are extracted into '`clsreports`'.
- To get a verbose listing of what '`tar`' is doing you can use '`v`' option, so we could have ran '`tar xvf cls.tar`' and it will list the files that it's extracting.

- ▶ To compress files and save space, use the 'gzip' command.
- ▶ To uncompress the file, you use the 'gunzip' command, and to view the contents of a zip archive, you can use 'gzcat' or 'zcat'.

## **Compressing Files To Save Space**

**gzip**      Compress files.

**gunzip**      Uncompress files.

**gzcat**      Concatenates compressed files.

**zcat**      Concatenates compressed files.



## ► Disk space used by file

- You use the 'du' command to display how much disk space is used by a file.
- 'du -k' display sizes in kilobytes, while 'du -h' display sizes in a human readable format for example if the file is 7 MB, it will say 7 M or 2.8 GB, 2.8 G
- Let's use the 'du' command to see how much space that the 'csp' file is using. It's using 12 K.

### Disk Usage

---

du      Estimates file usage.

du -k    Display sizes in Kilobytes.

du -h    Display sizes in human readable format.

- ▶ When you compress a file with 'gzip', it compresses it and adds the '.gz' extension to the filename.
- ▶ Let's try and see if that saved us any space, it did, 8 K, 12 K uncompressed, 4 K compressed.
- ▶ Uncompress the file with the 'gunzip' command.
- ▶ If you create a compressed tar archive, you can use the 'z' option for compression and for 'f' name the file 'cls.tgz'
- ▶ This is just a naming convention, but you don't have to name files this way as sometimes you'll see it - 'cls.tar.gz', but 'tgz' is short for 'tar' that's been compressed with 'gzip'.
- ▶ Let's create a compressed tar archive of the 'clsreports' directory. **\$ tar zcf cls.tgz clsreports**
- ▶ To work on a compressed file, you need to use 'z' option and it lists the files that are in that archive.