Lesson 5: Managing Files

Objectives covered

○ **103.3 Performing basic file management (weight: 4)**

○ 104.5 Manage file permissions and ownership (weight: 3)

○ 104.6 Create and change hard and symbolic links (weight: 2)

○ 104.7 Find system files and place files in correct location (w:2)

Performing basic file management

File naming rules

- *Case sensitive*
- *Limit to 255 characters*
- *No need to use dot "."*
- *Character used as bellow conventions:*

Could use	Do not use
characters A-Z, a-z	Meta characters: \ / < > ! $ % ^ & *	{ } [ ] " ' ` ; ~
numbers 0-9	Space " "
dot "."	
dash "—"	
underscore "_"	

Creating empty files

touch filename

```
$ touch Project43.txt
$
$ ls
Everything  Life  Project42.txt  Project43.txt  Universe
$
$ touch Project44.txt Project45.txt Project46.txt
$
$ ls
Everything  Project42.txt  Project44.txt  Project46.txt
Life        Project43.txt  Project45.txt  Universe
```

Displaying files

ls [Options] [File]

Short	Long	Description
-1	N/A	List one file or subdirectory name per line
-a	--all	Display all file and subdirectory names, including hidden files' names
-d	--directory	Show a directory's own metadata instead of its contents
-F	--classify	Classify each file's type using an indicator code (*,/,=,>,@, or |)
-i	--inode	Display all file and subdirectory names along with their associated index number
-l	N/A	Display file and subdirectory metadata, which includes file type, file access permissions, hard link count, file owner, file's group, modification date and time, and filename
-R	N/A	Show a directory's contents, and for any subdirectory within the original directory tree, consecutively show its contents as well (recursively)

Displaying files using wildcards

* ? []

```
aaronkilik@tecmint ~/bin $ ls -l l*
-rw-r--r-- 1 aaronkilik aaronkilik  0 Oct  4 10:44 listopen.sh
-rw-r--r-- 1 aaronkilik aaronkilik 12 Oct  4 11:11 list.sh
-rw-r--r-- 1 aaronkilik aaronkilik  0 Oct  4 10:45 lost.sh
-rw-r--r-- 1 aaronkilik aaronkilik  0 Oct  4 10:44 lspace.sh
aaronkilik@tecmint ~/bin $
```

```
aaronkilik@tecmint ~/bin $ ls -l l?st.sh
-rw-r--r-- 1 aaronkilik aaronkilik 12 Oct  4 11:11 list.sh
-rw-r--r-- 1 aaronkilik aaronkilik  0 Oct  4 10:45 lost.sh
aaronkilik@tecmint ~/bin $
```

```
aaronkilik@tecmint ~/bin $ ls -l l[abdcio]st.sh
-rw-r--r-- 1 aaronkilik aaronkilik 12 Oct  4 11:11 list.sh
-rw-r--r-- 1 aaronkilik aaronkilik  0 Oct  4 10:45 lost.sh
aaronkilik@tecmint ~/bin $
```

```
aaronkilik@tecmint ~/bin $ ls
createbackup.sh  list.sh   lspace.sh       speaker.sh
listopen.sh      lost.sh   rename-files.sh topprocs.sh
aaronkilik@tecmint ~/bin $
aaronkilik@tecmint ~/bin $ ls [clst][io]?t*
listopen.sh  list.sh  lost.sh
aaronkilik@tecmint ~/bin $
```

Specify file type

file, type, ls -F

```
$ file Project42.txt
Project42.txt: ASCII text
$
$ file Everything
Everything: directory
```

```
$ type ls
ls is aliased to 'ls --color=auto'
$
$ type cd
cd is a shell builtin
$
$ type find
find is /usr/bin/find
```

```
$ ls -F
Everything/  Project42.txt  Project44.txt  Project46.txt
Life/        Project43.txt  Project45.txt  Universe/
```

Making directories

mkdir [Option] path-to-dir

```
$ mkdir /home/Christine/Answers/Galaxy/Saturn
$
$ ls -F Galaxy
Saturn/
```

```
$ mkdir Projects/42/
mkdir: cannot create directory 'Projects/42/': No such file or directory
$
$ mkdir -p Projects/42/
$
$ ls -F
Everything/  Life/          Project43.txt  Project45.txt  Projects/
Galaxy/      Project42.txt  Project44.txt  Project46.txt  Universe/
$
$ ls -F Projects
42/
```

Copying files and directories

cp [Option] Source Dest

Short	Long	Description
-a	--archive	Perform a recursive copy and keep all the files' original attributes, such as permissions, ownership, and timestamps.
-f	--force	Overwrite any preexisting destination files with same name as *DEST*.
-i	--interactive	Ask before overwriting any preexisting destination files with same name as *DEST*.
-n	--no-clobber	Do not overwrite any preexisting destination files with same name as *DEST*.
-R, -r	--recursive	Copy a directory's contents, and also copy the contents of any subdirectory within the original directory tree (recursive).
-u	--update	Only overwrite preexisting destination files with the same name as *DEST*, if the source file is newer.
-v	--verbose	Provide detailed command action information as command executes.

Moving files and directories

mv [Option] Source Dest

Short	Long	Description
-f	--force	Overwrite any preexisting destination files with the same name as *DEST*.
-i	--interactive	Ask before overwriting any preexisting destination files with the same name as *DEST*.
-n	--no-clobber	Do not overwrite any preexisting destination files with the same name as *DEST*.
-u	--update	Only overwrite preexisting destination files with the same name as *DEST* if the source file is newer.
-v	--verbose	Provide detailed command action information as the command executes.

deleting files and directories

rm [Option] File

Short	Long	Description
-d	--dir	Delete any empty directories.
-f	--force	Continue on with the deletion process, even if some files designated by the command for removal do not exist, and do not ask prior to deleting any existing files.
-i	--interactive	Ask before deleting any existing files.
-I	N/A	Ask before deleting more than three files or when using the -r option.
-R, -r	--recursive	Delete a directory's contents, and also delete the contents of any subdirectory within the original directory tree (recursive).
-v	--verbose	Provide detailed command action information as command executes.

Compressing/decompressing files

Compress command	Uncompress command	Notation
gzip	gunzip	Just compress one file at a time, replace the original file with the compressed file with a .gz file extension
bzip2	bunzip2	Just compress one file at a time, replace the original file with the compressed file with a .bz2 file extension
xz	unxz	Just compress one file at a time, replace the original file with the compressed file with a .xz file extension
zip	unzip	Can compress multiple file at a time, does not replace the original file(s), create a new file with .zip file extention

Viewing content of compressed file

Command	Equivalent	Description
bzcat	bzip2 -dc	Used to display bzip2 compressed files.
xzcat	xz --decompress --stdout	Displays the contents of xz compressed files.
zcat	gunzip -c	Used to display gzip compressed files. Some Unix-like systems have a gzcat command instead.

Archiving files

cpio

Short	Long	Description
-I	N/A	Designates an archive file to use.
-i	--extract	Copies files from an archive or displays the files within the archive, depending upon the other options employed. Called copy-in mode.
N/A	--no-absolute-filenames	Designates that only relative path names are to be used. (The default is to use absolute path names.)
-o	--create	Creates an archive by copying files into it. Called copy-out mode.
-t	--list	Displays a list of files within the archive. This list is called a table of contents.
-v	--verbose	Displays each file's name as each file is processed.

Archiving files

tar

Short	Long	Description
-c	--create	Creates a tar archive file. The backup can be a full or incremental backup, depending on the other selected options.
-u	--update	Appends files to an existing tar archive file, but copies only those files that were modified since the original archive file was created.
-g	--listed-incremental	Creates an incremental or full archive based on metadata stored in the provided file.
-z	--gzip	Compresses tar archive file into a tarball using gzip.
-j	--bzip2	Compresses tar archive file into a tarball using bzip2.
-J	--xz	Compresses tar archive file into a tarball using xz.
-v	--verbose	Displays each file's name as each file is processed.
-d	--compare --diff	Compares a tar archive file's members with external files and lists the differences.
-t	--list	Displays a tar archive file's contents.
-W	--verify	Verifies each file as the file is processed. This option cannot be used with the compression options.

Extracting tarball

tar

Short	Long	Description
-x	--extract --get	Extracts files from a tarball or archive file and places them in the current working directory
-z	--gunzip	Decompresses files in a tarball using gunzip
-j	--bunzip2	Decompresses files in a tarball using bunzip2
-J	--unxz	Decompresses files in a tarball using unxz

Archiving files

dd if=InputDev of=outputDev [Operands]

Operand	Description
bs=*BYTES*	Sets the maximum block size (number of *BYTES*) to read and write at a time. The default is 512 bytes.
count=*N*	Sets the number (*N*) of input blocks to copy.
status=*LEVEL*	Sets the amount (*LEVEL*) of information to display to STDERR.

LEVEL values:
- **none**   only display error messages
- **noxfer**   does not display final transfer statistics.
- **progress**   displays periodic transfer statistics

Question...