Module 08
Managing Files and Directories

Exam Objective

2.4 Creating, Moving, and Deleting Files

Objective Description

Create, move and delete files and directories under the home directory

Introduction

Introduction

- In this chapter we will discuss how to manipulate files and directories.
- Some Linux distributions have GUI-based applications that allow you to manage files, but it is advantageous to know how to perform these operations via the command line.
- Note that everything in Linux is case sensitive so pay attention to capitalization:
  - The `hello.txt` file is different from the `HELLO.txt` and `Hello.txt` files.

Globbing

Globbing

- *Glob characters* are often referred to as *wild cards*.
- These are symbol characters that have special meaning to the shell.
- Globs are powerful because they allow you to specify patterns that match filenames in a directory:
    - Instead of manipulating a single file at a time, you can easily execute commands that will affect many files.

Globbing - The Asterisk *

- The asterisk character is used to represent zero or more of any character in a filename.
- For example, suppose you want to display all of the files in the `/etc` directory that begin with the letter `t`:

```
sysadmin@localhost:~$ echo /etc/t*
/etc/terminfo /etc/timezone
```

- The pattern `t*` matches any file in the `/etc` directory that begins with the character `t` followed by zero or more of any character.

Globbing - The Question Mark ?

- The question mark character matches exactly one character, no more and no less.
- Suppose you want to display all of the files in the `/etc` directory that begin with the letter `t` and have exactly 7 characters after the `t` character:

```
sysadmin@localhost:~$ echo /etc/t???????
/etc/terminfo /etc/timezone
```

- The asterisk and question mark could also be used together to look for files with three-letter extensions by running the `echo /etc/*.???` command:

```
sysadmin@localhost:~$ echo /etc/*.???
/etc/blkid.tab /etc/issue.net
```

Globbing - Brackets [ ]

- Brackets are used to match a single character by representing a range of characters that are possible match characters.

- For example, `echo /etc/[gu]*` will print any file that begins with either a `g` or `u` character and contains zero or more additional characters:

```
sysadmin@localhost:~$ echo /etc/[gu]*
/etc/gai.conf /etc/groff /etc/group /etc/group- /etc/gshadow /etc/gshadow-
/etc/ucf.conf /etc/udev /etc/ufw /etc/update-motd.d /etc/updatedb.conf
```

- Brackets can also be used to a represent a range of characters by using the – character (i.e., any letter between and including `a` and `d`):

```
sysadmin@localhost:~$ echo /etc/[a-d]*
```

Globbing - Exclamation Point !

- The exclamation point is used in conjunction with the square brackets to negate a range.

- For example, the command `echo /etc/[!a-t]*` will display any file that **does not** begin with an `a` thru `t`:

```
sysadmin@localhost:~$ echo /etc/[!a-t]*
/etc/ucf.conf /etc/udev /etc/ufw /etc/update-motd.d /etc/updatedb.conf /etc/vim
/etc/wgetrc /etc/xml
```

Globbing - Listing With Globs

- When the `ls` command sees a directory as an argument, it will display the contents of the directory, not just the directory name.

- Use the `-d` option, which tells the `ls` command to display the name of directories, instead of their contents:

```
sysadmin@localhost:~$ ls -d /etc/e*
/etc/encript.cfg /etc/environment /etc/ethers /etc/event.d /etc/exports
```

Copying Files and Directories

Copying Files

- The `cp` command is used to copy files. It requires a source and a destination.
- The structure of the command is as follows:

```
cp [source] [destination]
```

- The *source* is the file to be copied. The *destination* is where the copy is to be located.
- The following command will copy the `/etc/hosts` file to your home directory:

```
sysadmin@localhost:~$ cp /etc/hosts ~
sysadmin@localhost:~$ ls
Desktop    Downloads  Pictures  Templates  hosts
Documents  Music      Public    Videos
```

Copying Files - Verbose Mode

- The `-v` option will cause the `cp` command to produce output if successful.
- The `-v` option stands for *verbose*.
- An example of the `-v` option used with the `cp` command:

```
sysadmin@localhost:~$ cp -v /etc/hosts ~
`/etc/hosts' -> `/home/sysadmin/hosts'
```

- When the destination is a directory, the resulting new file will have the same name as the original file.
- If you want the new file to have a different name, you must provide the new name as part of the destination.

Copying Files - Avoid Overwriting Data

- The `cp` command can be destructive to existing data if the destination file already exists.

- With the `-i` (interactive) option, the `cp` will prompt before overwriting a file (`y` (yes) or `n` (no)):

```
sysadmin@localhost:~$ cp -i /etc/hosts ~/example.txt
cp: overwrite `/home/sysadmin/example.txt'? n
```

- The `-i` option requires you to answer `y` or `n` for every copy which could be tedious if there are a lot of files.

- If you want to automatically answer `n` to each prompt, use the `-n` option. It essentially stands for "no rewrite".

Copying Directories

- Using the `cp` command to copy directories will result in an error message:

```
sysadmin@localhost:~$ cp -n /etc/skel/.* ~
cp: omitting directory `/etc/skel/.'
cp: omitting directory `/etc/skel/..'
```

- However, the `-r` (recursive) option to the `cp` command will have it copy both files and directories.

Be careful with this option. The entire directory structure will be copied. This could result in copying a lot of files and directories!

Moving Files

Moving Files

- To move a file, use the `mv` command.
- The syntax for the `mv` command is much like the `cp` command:

```
mv [source] [destination]
```

- When a file is moved, the file is removed from the original location and placed in a new location.
- **Note**: If you don't have the right permissions, you will receive a "`Permission denied`" error message.

Moving Files - Renaming Files

- The `mv` command is not just used to move a file, but also to rename a file.
- The name of the file will change only if a destination file name is also specified.
- If a destination directory is not specified, the file will be renamed using the destination file name and remain in the source directory.
- For example, the following commands will rename the `newexample.txt` file to `myexample.txt`:

```
sysadmin@localhost:~/Videos$ mv newexample.txt myexample.txt
```

Moving Files - Additional mv Options

- Like the `cp` command, the `mv` command provides the following options:

Option	Meaning
`-i`	Interactive: Ask if a file is to be overwritten.
`-n`	No Clobber: Do not overwrite a destination files' contents.
`-v`	Verbose: Show the resulting move.

- **Important:** There is no `-r` option as the `mv` command will by default move directories.

Creating Files and Directories

Creating Files

- To create an empty file, use the `touch` command as demonstrated below:

```
sysadmin@localhost:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates
Videos
sysadmin@localhost:~$ touch sample
sysadmin@localhost:~$ ls -l sample
-rw-rw-r-- 1 sysadmin sysadmin 0 Nov  9 16:48 sample
sysadmin@localhost:~$ ls
Desktop    Downloads  Pictures  Templates  sample
Documents  Music      Public    Videos
```

Making Directories

- To create a directory, use the `mkdir` command:

```
sysadmin@localhost:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates
sample.txt
sysadmin@localhost:~$ mkdir test
sysadmin@localhost:~$ ls
Desktop    Downloads  Pictures  Templates   test
Documents  Music      Public    sample.txt
```

Removing Files and Directories

Deleting Files

- To delete a file, use the `rm` command:

```
sysadmin@localhost:~$ ls
Desktop    Downloads  Pictures  Templates  sample
Documents  Music      Public    Videos
sysadmin@localhost:~$ rm sample
sysadmin@localhost:~$ ls

Desktop  Documents  Downloads  Music  Pictures  Public  Templates
Videos
```

- Using the `rm` could cause problems when deleting multiple files by using glob characters.
- As a precaution, users should use the `-i` option when deleting multiple files.

Deleting Directories

- The `rm` command can be used to delete directories. However, the default usage (no options) of the `rm` command will fail to delete a directory:

```
sysadmin@localhost:~$ rm Videos
rm: cannot remove `Videos': Is a directory
sysadmin@localhost:~$
```

- To delete a directory, use the `-r` (recursive) option to the `rm` command:

```
sysadmin@localhost:~$ rm -r Videos
```

- **Important:** When a user deletes a directory, all of the files and subdirectories are deleted without any interactive question. It is best to use the `-i` option with the `rm` command.