0.1.2 Basic Linux File Management

As a Linux administrator, much of your work will be done from a command line interface. If you are new to Linux, you may find it difficult to navigate the file system and make changes to text files. To help you make a quick transition to Linux and to help you be successful in this course, this lesson introduces a few basic command line interface pointers.

This lesson covers the following topics:

- Linux file system basics
- Create, copy, and move a Linux directory
- Remove a Linux directory
- File management commands
- The vi editor

Linux File System Basics

The container that holds Linux files is called a directory. The following rules apply to files and directories.

- File and directory names are case sensitive. The filenames testout.txt, TestOut.txt, and TESTOUT.txt are three different files.
- File and directory names can be made up of upper and lowercase letters, numbers, and the dot (.) and underscore (_) symbols.
- The dot (.) in a filename is not necessary. However, it can be helpful to use a dot based filename extension to identify file type.
- A directory can hold files and other directories
- Linux directories are arranged hierarchically as parent and child directories.
- The root directory does not have a parent directory and is represented by a single forward slash (/).
- The forward slash is also used as a delimiter when showing the directory path from the root directory to a file or directory. For example: /home/rtracy/bin/myaddresses.dat.

The following table describes basic commands you use to navigate within a Linux directory.

Command	Description	Examples
pwd	Displays the current working directory.	• If a user named Fred is currently in his home directory and types **pwd** at the shell prompt, **/home/Fred** is displayed.
cd	Changes the present working	• **cd directory1** changes to a directory named

	directory. • **cd ..** changes to the parent directory. • **cd ../..** to changes two levels up in the directory. • **cd /** changes to the root directory.	**directory1** within the current working directory. (This is a *relative* path.) • **cd /home/Fred/directory1** switches to **directory1** in Fred's home directory, regardless of the current working directory. (This is an absolute path.)
ls	Displays the contents of a directory. Options include: • **-a** displays all directory contents, including hidden content. • **-l** displays extended information, including the owner, modified date, size, and permissions. • **-R** displays the contents of a directory and all of its subdirectories. • **-d** displays directories but not files. • **-r** reverses the sort order.	• **ls -al** displays a long listing of all the contents in the current working directory, including hidden content. • **ls -d** displays only directories within the current directory. • **ls -R /etc** displays the contents of the /etc directory and all of its subdirectories.

Create, Copy, and Move a Linux Directory

The following table describes basic commands you use to manage a Linux directory:

Command	Description	Examples
mkdir	Creates a new directory. Use the **-p** option to create all directories within the specified path when that path does not already exist.	• **mkdir work_files** creates a directory named *work_files* in the current working directory. • **mkdir /home/Fred/work_files** creates a directory named *work_files* within the specified path.
cp	Copies directories. Copying leaves the source contents (directories and files) intact.	• **cp -r /temp /home/user** copies the entire **/temp** directory (with

	Use the **-r** or -R option to recursively copy subdirectories and files within the directory.	all of its files, subdirectories, and files in the subdirectories) to the **/home/user** directory.
mv	Moves or renames directories (and files). Moving directories removes the source directory and places it in the destination. Options include: • **-f** overwrites a directory that already exist in the destination directory without prompting. • **-i** prompts before overwriting a directory in the destination directory. • **-n** never overwrites files in the destination directory.	• **mv /temp/station ~/doc/** moves **station** from the **/temp** directory to the **~/doc** directory. • **mv /current /previous** renames the directory **current** to **previous**

Remove a Linux Directory

The following table describes basic commands you use to remove a Linux directory:

Command	Description	Examples
rmdir	Deletes an empty directory.	• **rmdir ~/Fred/work_files** deletes the **work_files** directory if it is empty.
rm	Removes the directory and file information from the file system, which makes the directories and files inaccessible. Options include: • **-i** prompts before removing. • **-r** removes directories, subdirectories, and files within them. • **-f** eliminates prompt for read-only files and avoids an exit code error if a file doesn't exist.	• **rm -rf /home/user/temp** deletes the temp directory with all its subdirectories and files without prompting. • **rm -r /home/user/*** deletes all directories and files in the **/home/user** directory.

File Management Commands

The following table describes Linux commands that can be used to manage files.

Command	Function	Examples
touch	If the file does not exist, **touch** creates a blank version of the file. If the file does exist, this command updates the file's modification and last accessed times.	• **touch myfile** makes a blank file named **myfile**.
cp	Copies files. Copying leaves the source file intact. • **-f** overwrites files that already exist in the destination directory. • **-i** prompts before overwriting a file in the destination directory.	• **cp /temp/document_ab.txt ~/doc/document.txt** copies **document_ab.txt** from the **/temp** directory to the **~/doc** directory and renames the file to **document.txt**. • **cp /temp/*.txt ~/doc** copies all text files from the **/temp** directory to the **~/doc** directory.
mv	Moves or renames files (and directories). Moving files erases the source file and moves it to the destination. • **-f** overwrites files that already exist in the destination directory. • **-i** prompts before overwriting a file in the destination directory. • **-n** never overwrites files in the destination directory.	• **mv /temp/document.txt ~/doc/document.txt** moves **document.txt** from the **/temp** directory to the **~/doc** directory. • **mv /temp/*.txt ~/doc/*.txt** copies all text files from the **/temp** directory to the **~/doc** directory.
rm	Removes a file or directory. Use the **-f** option to delete with a prompt. The **rm** command deletes a file or directory's inode, but it does not actually delete its data. To permanently remove data, use the **shred** command.	• **rm myfile** deletes a file in the current directory named **myfile**. • **rm /home/user/myfile** deletes **myfile** from the **/home/user** directory regardless of the current directory. • **rm -f /home/user/temp/*** deletes all files in the temp directory with prompts.

The vi Editor

The vi editor is a utility that creates and modifies text files. It is the defacto command line text editor included with most Linux distributions.

The vi editor uses the following operational modes:

- Command mode is the initial mode vi uses when started. It provides commands that can cut and replace text. It is also the mode from which you access the other vi modes.
- Command line mode is used to load files and to save files after editing them in the file system.
- Edit mode is the mode that vi uses to write and edit text in the file. It has two operation modes:
    - Insert mode adds text between the preceding and subsequent text.
    - Replace mode overwrites subsequent text.

The table below lists some of the most common vi commands:

Command	Function	Mode
vi	Starts vi. Type the command at the shell prompt.	N/A
vi [file_name]	Starts vi and immediately begins working on the named file (either a new file or an existing file). Type the **vi** command at the shell prompt.	N/A
Insert key **i** **s**	Enters insert mode from command mode.	Command
Esc key	Enters command mode from edit mode.	Insert/Replace
Delete key	Deletes text.	Insert/Replace
Insert key	Toggles between the insert and replace modes while in edit mode.	Insert/Replace
z	Exits without saving.	Command
:	Enters command line mode from command mode.	Command
w	Saves the current document.	Command line
w [file_name]	Names and save the file.	Command line
w! [file_name]	Overwrites the file.	Command line
q	Exits vi. This produces an error if the text was modified.	Command line
q!	Exits vi without saving.	Command

		line
wq or **exit**	Saves the document and exits vi.	Command line