

Fundamentals of UNIX
Lab 6.1.10 – Basic Command Line File Management
(Estimated time: 30 min.)

Objectives:

- Review file and directory naming conventions
- Create new files with the `touch` command
- Create new directories with the `mkdir` command
- Remove files using the `rm` command.
- Remove directories using the `rm -r` command.

Background:

In this lab, you will work with file management commands from the command line. The guidelines for file and directory naming, which are known as naming conventions, will be reviewed. You will create a simple directory structure and then create some files in those directories. You will practice creating and removing both files and directories.

Knowledge of how to manage files and directories using commands is very important in building a solid foundation for further study of UNIX. Power users and administrators frequently create executable script files, which are an important tool for automating certain tasks such as backing up files or creating new user accounts. Script files are a series of UNIX commands and are similar to batch files used with other network operating systems.

Tools / Preparation:

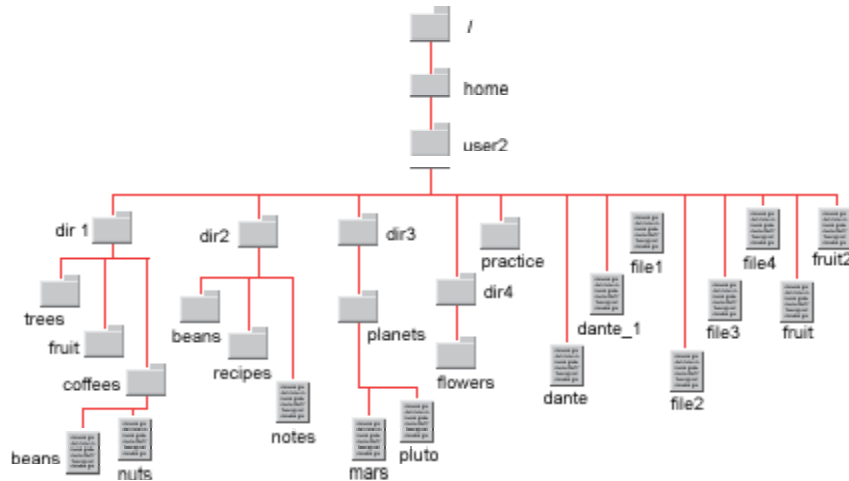
- a) Before starting this lab, review Chapter 6, Section 1 – Directory and File Management Using the Command Line.
- b) You will need the following:
 1. A login user ID (e.g. user2) and password assigned by your instructor.
 2. A computer running the UNIX operating system with CDE
 3. Networked computers in classroom

Notes:

Fundamentals of UNIX
Lab 6.1.10 – Basic Command Line File Management
Worksheet

Use the diagram of the sample Class File system directory tree to assist with this lab.

Class File Tree Structure



Step 1. Log in to CDE

Login with the user name and password assigned to you by your instructor in the CDE entry box.

Step 2. Access the Command Line

Right click on the **workspace** backdrop and click on **Tools**. Select **Terminal** from the menu to open a terminal window.

Step 3. Review UNIX File and Directory Naming Conventions

In this lab you will be creating files and directories so it is important to review the naming rules and guidelines for UNIX files and directories before we begin.

Maximum Length - The maximum length of files and directories combined is 255 alphanumeric characters. In general, it is desirable to keep file names as short as possible but still be meaningful.

Non-alphanumeric Metacharacters - Some non-alphanumeric or metacharacters are allowed - **Underscores** (`_`), **hyphens** (`-`), and **periods** (`.`), and can be used multiple times in a file or directory name (Feb.Reports.Sales is a valid file or directory name for example). While the shell will allow asterisks (`*`), question marks (`?`), tildes (`~`), brackets (`[]`), ampersands (`&`), pipes (`|`), quotes (`" "`), and dollar signs (`$`) to be used in a file name, this is **not recommended**, as these characters have special meaning to the shell. The semicolon (`;`), less than (`<`) and greater than (`>`) symbols are **not allowed**.

Fundamentals of UNIX
Lab 6.1.10 – Basic Command Line File Management
Worksheet – Cont.

File Name Extensions - File Names may contain one or more extensions. Extensions are usually appended to file by an application. Extensions are usually one to three characters that are appended to the end of a file name and are preceded by a period (.). You may choose to use this convention when naming files, but it is not a necessary part of a file name.

Directory Name Extensions - Directory names generally do not contain extensions, but there are no rules against it.

Case Sensitivity – UNIX file and directory names are case sensitive. **Project1** is not the same file as **project1**. You cannot have two files with the same name in the same directory. Use lower case letters as a general rule.

Examine the following file names and indicate whether they would be valid or recommended UNIX file or directory names and why or why not.

	File Name	Is this a UNIX file or directory name? (yes/no)	Why or Why Not?
a.	12345abcde678		
b.	Hobbies;2		
c.	Adcd-123		
d.	Sales*repts*2001		
f.	D.projects.bj.2001		
g.	Projects>1.bj-2001		

Step 4. Create Files with the touch Command

Every time you create a new word processing document or spreadsheet, you are creating a new file and should adhere to the file naming conventions previously mentioned. You must also have adequate permissions for the directory in which you are working to create files.

Using the **touch** command, you can create one or multiple files simultaneously. Some applications require files to exist before they can be written. The **touch** command is useful for quickly creating files to experiment with. You can also use the **touch** command to update the time and date that a file is accessed. This will reset the archive bit making the file available for backup again. Absolute and relative pathnames can be specified when creating files or directories.

Command Format: **touch filename(s)**

- From your **home** directory, change to the **practice** directory using a relative pathname. What command did you use? _____
- Enter the **pwd** command to verify what directory you are currently in. What was the response?

- Use the **touch** command to create a file in this directory called **newfile**. What command did you use? _____
- Use the **touch** command to create another new file in this directory called **filenew**. What command did you use? _____

Fundamentals of UNIX
Lab 6.1.10 – Basic Command Line File Management
Worksheet – Cont.

- e. Enter the command to display a **long** listing of the files in the practice directory. Are the files you created listed? _____
- f. Who is the **owner** of files? _____
- g. What is the **group** associated with the files? _____
- h. What is the **date** and **time** created? _____
- i. What is the **size** of the files? _____
- j. Use the **file** command to determine the file type for **newfile**. What kind of file is it?

- k. Create 3 files at the same time with the **touch** command; **new1**, **new2** and **new3**. What command did you use? _____
- l. Enter the command to display a **long** listing of the files in the practice directory. Are the three new files you created present? _____

Step 5. Create New Directories with the `mkdir` Command

The **mkdir** (make directory) command is used to create **directories** or **folders**. Directories can contain other directories, which are referred to as subdirectories and they can contain files. Directories can be created using either an absolute or a relative pathname. You can specify more than one directory name on the same line to create more than one new directory. You must have the appropriate permissions to create a directory. (Permissions are covered later)

The **mkdir -p** (parent) option can be used to create parent directories while creating lower level directories. You can create multiple levels of directories including all the directories in a pathname simultaneously. If you use the **-p** option and specify a directory in the pathname that does not exist, it will be created.

Command Format: **mkdir** **[-p]** **directory_name(s)**

- a. From your **home** directory, change to the **practice** directory using a relative pathname. What command did you use? _____
- b. Use the **mkdir** command to create a subdirectory in this directory called **newdir**. What command did you use? _____
- c. Enter the command to display a **long** listing of the files and directories in the practice directory. Is the subdirectory you created listed? _____
- d. Who is the **owner** of the directory? _____
- e. What is the **size** of the file? _____
- f. Use the **file** command to determine the file type for **newdir**. What kind of file is it?

Fundamentals of UNIX
Lab 6.1.10 – Basic Command Line File Management
Worksheet – Cont.

- g. How else could you tell this was a directory if it did not have the characters “dir” in the name?

- h. Use the **mkdir** command with the **-p** option to create a hierarchy of three new directories with the names of: **high**, **medium** and **low**. Medium should be a subdirectory of high and low should be a subdirectory of medium. What command did you enter?

- i. Use the **ls** command with the **-R** (recursive) option to see all directories and subdirectories. Are all of the new directories listing in the proper order? _____

Step 6. Remove Files with the rm Command

The **rm** command can remove a single file or multiple files. You can remove several files at once by specifying their names after the **rm** command or you can use the asterisk (*) and question mark (?) metacharacters (wildcards). Files that are deleted on a UNIX system are permanently deleted and cannot be recovered unless you are using the CDE graphical interface. The **rm** command can be used with the **-i** (interactive) option, which prompts the user before removing files. Use the **rm -i** command as a precaution to avoid accidentally deleting files.

Command Format: **rm [-i] filename(s)**

- a. Use the **rm** command to remove the **newfile** you created earlier from the practice directory. What command did you use? _____
- b. Enter the command to display a **long** listing of the files in the practice directory. Is the file you created gone? _____
- c. Use the **rm** command with the **-i** (interactive) option to remove the **filenew** you created earlier from the practice directory. What did the interactive option do?

- d. Remove the 3 files you create earlier named **new1**, **new2** and **new3**. Use the question mark (?) wildcard to remove all three with one command. What command did you use?

- e. Why could you not have used the **rm new*** command? _____
- f. Enter the command to display a **long** listing of the files in the practice directory. Are the three files gone? _____

Step 7. Remove Directories with the rm -r Command

The **rm -r** (recursive) command is used to remove directories. It will remove the directory being targeted including all subdirectories and files in it. When the **rm** command is used with the **-r** option it can remove a single directory (empty or not) or an entire section of the directory tree. The **rm** command can be used with the **-i** (interactive) option, which prompts the user before removing the directory.

Command Format: **rm -r [-i] directory_name(s)**

- a. Remove the **newdir** subdirectory you created earlier. What command did you use?

Fundamentals of UNIX
Lab 6.1.10 – Basic Command Line File Management
Worksheet – Cont.

- b. Enter the command to display a **long** listing of the files and directories in the **practice** directory. Is the subdirectory you created gone? _____
- c. Change to the **medium** subdirectory you created earlier. What command did you enter?

- d. Remove the **low** subdirectory you created earlier. What command did you use?

- e. Change back to the **practice** subdirectory using a relative pathname and shortcuts. What command did you enter? _____
- f. Remove the **high** and **medium** subdirectories with one command. What command did you use?

Step 8. Practice What You Have Learned

Practice using the **touch**, **mkdir**, and **rm** commands by creating a simple three-level directory tree within the practice directory. Try to use meaningful directory names. Remember you can create an entire directory structure with one command. Create multiple files in each of the directories. Remember you can create multiple files with one command. Remove the files and then remove the directories so you have no files or directories in your practice directory when finished.

Step 9. Close the Terminal Window and Logout

Double click on the dash button in the upper left corner of the screen, then click the **EXIT** icon on the front panel.