

**Fundamentals of UNIX**  
**Lab 10.2.4– Determining File System Permissions**  
***(Estimated time: 30 min.)***

**Objectives:**

- Display file system permissions
- Interpret permissions
- Determine User permissions for files
- Determine Group permissions for a files
- Determine Other (Public) permissions for files
- Determine File Permissions for an Executable file
- Create a new file with default permissions
- Create a new directory with default permissions

**Background:**

In this lab, you will become familiar with file system permissions. A major component in any comprehensive security policy, file system security determines who can get to what data and what they can do with it. System administrators set up file system security based on users, groups, and permissions. Directory and file permissions will be determined using the **ls (list)** command with the **-l (long)** option. You can determine the file type, permissions, owner and group with the **ls -l** command. You will display permissions on files and directories, interpret the results and evaluate the effect on various user categories.

**Tools / Preparation:**

- a) Before starting this lab, review Chapter 10, Section 2 – File System Permissions
- 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
  3. Networked computers in classroom with class file system installed

**Notes:**

---

---

---

---

---

---

---

---

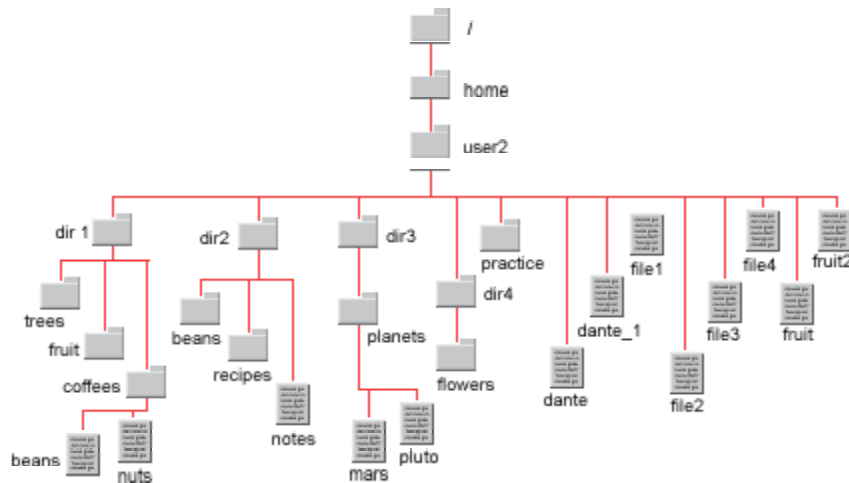
---

---

Fundamentals of UNIX  
Lab 10.2.4– Determining File System Permissions  
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. Display Permissions.**

Permissions control who can do what to files and directories in the file system. Directory and file permissions can be determined using the **ls (list)** command with the **-l (long)** option. The **ls -l** command will display a long listing of the contents of a directory. If the **-a** (all) option is included, all files, including hidden files and directories (those beginning with a dot) will be displayed.

**Fundamentals of UNIX**  
**Lab 10.2.4– Determining File System Permissions**  
**Worksheet – Cont.**

The following table provides a summary review of the information displayed with the `ls -l` command.

**Note:** When working with permissions, File type, Permissions, Owner, Group and File/Directory name are the most important pieces of information in the listing.

<b>File Type</b>	A dash (-) in the first position indicates a regular file. A <b>d</b> indicates directory
<b>Permissions</b>	3 Sets of permissions – User, Group, Others
<b>Links</b>	Links to other files and directories.
<b>Owner</b>	User (login) ID of user who created the file or directory unless ownership was assigned.
<b>Group</b>	Group name that owner belongs to as established by the system administrator
<b>Size</b>	File size in Bytes
<b>Modification Date / Time</b>	Month, Day, Year (if not current year) and Time the file was created or last modified.
<b>File Name</b>	File or Directory Name

a. The **first** position in the `ls -l` listing for a file or directory indicates the **file type**. Use the `ls -l` commands to list the files in your home folder. What is the file type of **dante** (File or Directory)? **File**. How can you tell? \_\_\_\_\_

b. The **next 9** positions in the `ls -l` listing represent the **permissions** of the file. The possible permissions for any file or directory are: (lower case) **r**, **w**, **x** or **dash (-)**. What are the permissions for **dante**? \_\_\_\_\_

**Step 4. Interpret Permissions.**

Use the table below as a review of permissions and answer the following questions. **Note: that the interpretation of permissions is different for files and directories.**

<b>Permission</b>	<b>Symbol</b>	<b>Plain File</b>	<b>Directory</b>
<b>Read</b>	<b>r</b>	File can be displayed or copied.	Contents can be listed with the <code>ls</code> command. (To Display a long listing ( <code>ls -l</code> ) you must also have Execute permission)
<b>Write</b>	<b>w</b>	File contents can be modified.	Files can be added or deleted. (To add or delete files you must also have execute permission)
<b>Execute</b>	<b>x</b>	File can be executed (shell scripts or executables only).	Allows the find command to search through a directory
<b>No Permission</b>	<b>-</b>	A dash (-) indicates permission is denied	A dash (-) indicates permission is denied

**Fundamentals of UNIX**  
**Lab 10.2.4– Determining File System Permissions**  
**Worksheet – Cont.**

- a. What is the meaning of the **r** permission for a file? \_\_\_\_\_
- b. What is the meaning of the **r** permission for a directory? \_\_\_\_\_
- c. What is the meaning of the **x** permission for a file? \_\_\_\_\_
- d. What is the meaning of the **w** permission for a file? \_\_\_\_\_
- e. What is the meaning of the **w** permission for a directory?  
\_\_\_\_\_
- f. What is the meaning of the **dash (-)** in place of a permission? \_\_\_\_\_

**Step 5. Determine User Permissions for a File.**

The nine permissions are divided in to three sets of three permissions each. Each set of three permissions is always in the sequence of **r (read)**, **w (write)**, and then **x (execute)**. If a permission is not allowed a **dash (-)** will be in its place. The first set of three permissions is the user permissions and these determine what the **owner** can do.

- a. Who is the owner of the dante file? \_\_\_\_\_
- b. What are the first two characters of the user permissions? \_\_\_\_\_
- c. In the user set of permissions, what is in the third position? \_\_\_\_\_ What permission does it take the place of and prevent? \_\_\_\_\_
- d. Why is this character in that position? \_\_\_\_\_
- e. List at least four things the user (owner) can do to the file with the permissions listed?  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Step 6. Determine Group Permissions for a File.**

The system administrator assigns every user to a primary group. The group that the file owner is a member of is assigned along with the owner when a file is created. The second set of 3 permissions determines what the members of the primary **group** can do.

- a. The owner of the dante file is a member of what primary **group**? \_\_\_\_\_
- b. What is first character of the **group** permissions? \_\_\_\_\_
- c. What will this allow other members of the group to do with the file? \_\_\_\_\_
- d. Why are dashes in the second and third positions in place of the **w** and **x**?  
\_\_\_\_\_

**Fundamentals of UNIX**  
**Lab 10.2.4– Determining File System Permissions**  
**Worksheet – Cont.**

**Step 7. Determine Other (Public) Permissions for a File.**

The last set of characters, called **others** permissions, are the permissions everyone else has. **Others** refers to anyone who is neither the file owner nor a member of the group that owns the file, but who has access to the system.

- a. What permissions do people other than the owner and group have to the **dante** file?
- \_\_\_\_\_

**Step 8. Determine File Permissions for an Executable file**

Executable files such as UNIX utilities and script files require the **x** (Executable) permission in order for anyone to run the command or script.

- a. From your **home** directory display the long directory listing for the **pwd** command file in the **/usr/bin** directory. What command did you use? \_\_\_\_\_
- b. What are the permissions for the file? \_\_\_\_\_
- c. What are the **User** permissions? \_\_\_\_\_
- d. What are the **Group** permissions? \_\_\_\_\_
- e. What are the permission for **Others**? \_\_\_\_\_
- f. Why do none of the user categories have **w (write)** permission?
- \_\_\_\_\_

**Step 9. Create a New File with Default Permissions.**

New files are created with **default** permissions. Use the **touch** command to create a new file in your **practice** directory.

- a. From your **home** directory create a new file called **newfileperms** in your **practice** directory. What command and pathname did you use? \_\_\_\_\_
- b. From your home directory list the contents of your **practice** directory to see the permissions of **newfileperms**. What command and pathname did you use? \_\_\_\_\_
- c. What are the default permissions that were assigned to this file? \_\_\_\_\_
- d. Who is the **owner**? \_\_\_\_\_ Who is the primary **group**? \_\_\_\_\_
- e. Could a member of the primary users **group** rename this file?
- \_\_\_\_\_

**Step 10. Create a New Directory with Default Permissions.**

New directories are also created with a different set of default permissions. Use the **mkdir** command to create a new directory in your **practice** directory.

- a. From your **home** directory create a new file called **newdirperms** in your **practice** directory. What command and pathname did you use? \_\_\_\_\_
- b. From your **home** directory list the contents of your **practice** directory to see the permissions of **newdirperms**. What command and pathname did you use? \_\_\_\_\_

**Fundamentals of UNIX**  
**Lab 10.2.4– Determining File System Permissions**  
**Worksheet – Cont.**

- c. What are the default permissions that were assigned to this directory? \_\_\_\_\_
- d. Who is the **owner**? \_\_\_\_\_ Who is the primary **group**? \_\_\_\_\_
- e. Could a member of the primary users group add a file in this directory?  
\_\_\_\_\_

**Step 10 – Remove Files and Directories Created in this Lab**

**Remove all files** and **directories** created in you home directory during this lab.

**Step 11. 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.