Lab Manual
Objectives:
- Demonstrate an understanding of the following:
  - Basic computer hardware components
  - Computer operating systems
  - UNIX varieties and history
  - The Solaris operating environment
- Investigate the use of UNIX at an organization or institution
- Research UNIX powered web sites

Background:
This lab exercise will review UNIX computing environment terminology and help to reinforce concepts introduced in Chapter 1. The student will also investigate the use of UNIX at their institution or another organization and research web sites to see what organizations are using Solaris UNIX.

Tools / Preparation:
1) Before starting this lab, the student should review Chapter 1: The UNIX Computing Environment and the chapter quiz.
2) The student should contact someone, such as an Information Technology staff member, who is knowledgeable about the student’s network. This person should be able to discuss what servers and network operating systems are in use in the student’s institution, or another organization, to find if and where UNIX is being used.
3) A computer with a browser and access to the Internet will be needed.

Web Site Resources:
- Sun Microsystems Solaris – http://www.sun.com/solaris

Notes:

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________
1. Identify the four main hardware components of a computer:
   a)
   b)
   c)
   d)

2. In the blank spaces, write one of these three terms: kernel, shell, or file system.
   a) The __________ represents the user interface, which translates requests into actions.
   b) The __________ manages and allocates resources among users.
   c) The __________ provides command interpretation.
   d) The __________ organizes and stores data in a hierarchical structure.
   e) The __________ performs memory management.
   f) The __________ components are files and directories.
   g) The __________ controls disks, tapes, printers, terminals, communication lines, and other devices.

3. What are four benefits of the UNIX operating system (answers will vary)?
   a)
   b)
   c)
   d)

4. List the three major components of the UNIX operating system.
   a)
   b)
   c)

5. The kernel manages what three things?
   a)
   b)
   c)

6. What are the names of the three main UNIX shells and the default prompt for each?
   a)
   b)
   c)

7. The file system is made up of what three components?
   a)
   b)
   c)
8. List the various Desktop operating systems in use at the student’s institution and indicate the number of workstations installed. Check with the instructor or an Information Technology staff member to find out. (Answers will vary).

<table>
<thead>
<tr>
<th>Operating System Name and Version</th>
<th>Hardware Manufacturer</th>
<th>Number of Desktops Installed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9. Check with the instructor or an Information Technology staff member to find out if and where UNIX servers are used in the student’s institution or another organization. List the UNIX version, hardware manufacturer, and function the servers perform. (Answers will vary).

<table>
<thead>
<tr>
<th>UNIX Version</th>
<th>Hardware Manufacturer</th>
<th>Server Function</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10. A number of Fortune 500 companies run Solaris to power their websites and run their networks. Conduct a web search and list 5 of those companies.

____________________________________

____________________________________

____________________________________
Fundamentals of UNIX

Lab 2.1.8 – Accessing Your System

(Estimated time: 30 min.)

Objectives:
• Demonstrate an understanding of the following:
  o UNIX login ID requirements
  o UNIX password requirements
• Login to a UNIX system with CDE, GNOME, or KDE
• Change user password
• Exit or logout of the UNIX system properly

Background:
In this lab, the student will review the requirements for UNIX login IDs and passwords. The student will practice logging in to a UNIX system using the GUI login screen. The student will then change their password, exit or log out, and return to the login screen.

Tools / Preparation:
A. Before starting this lab, the student should review Chapter 2, Section 1 - User accounts.
B. The student will need a login user ID for example, user2. The student will also need a password assigned by their instructor.
C. A computer running the UNIX operating system with CDE, GNOME, or KDE is required.

Web Site Resources:
• Sun Microsystems – http://www.sun.com/solaris

Notes:

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________
Part 1 - Answer the following questions:

1. List at least 2 rules or requirements for UNIX login IDs.
   a) 
   b) 
   c) 

2. List at least 3 rules or requirements for UNIX passwords.
   a) 
   b) 
   c) 
   d) 
   e) 
   f) 
   g) 

Part 2 - Obtain a user name and password from the instructor. Student should be at a UNIX workstation with the CDE Login screen displayed. Perform the following steps to complete this lab:

Step 1. Login to the system.
   a. Login with the user name assigned by the instructor in the entry box provided.
   b. Enter the password assigned by the instructor.

Step 2. Changing a password.
   a. Open a GUI terminal window. At the command prompt ($), enter the command that will allow the password to be changed. What command was entered?
   b. Enter the student’s current login password when prompted. Enter the new password of abc when prompted. Was the password able to be changed to abc? Why was it able to be changed or why not?
   c. Enter another new password of abcdef when prompted. Was the password able to be changed to abcdef? Why was it able to be changed or why not?
   d. Enter the New password of unix123. **Note**: UNIX passwords are case sensitive. Reenter new password to confirm it. Was the password able to be changed? What message was received confirming this?
   e. Type exit or press the Ctrl and D keys at the command prompt ($) to close the terminal window.

Step 3. Logout of the system.
Objectives:
- Review the CDE front panel icons and menus
- Manage CDE windows
- Lock the Display
- Work with Workspaces
- Use the Workspace Menu

Background:
In this lab, the student will work with the standard UNIX graphical user interface (GUI) known as Common Desktop Environment or CDE. The student will become familiar with the Front Panel and use the mouse and keyboard to manage windows. The student will also practice locking the display, moving between workspaces, and using the Workspace menu.

Tools / Preparation:
- Before starting this lab, the student should review Chapter 2, Section 2 – Becoming Familiar with the Common Desktop Environment (CDE).
- The student will need the following:
  1. A login user ID for example user2. The student will need a password assigned by their instructor.
  2. A computer running the UNIX operating system with CDE.

Web Site Resources:
- Sun Microsystems – http://www.sun.com/solaris

Notes:

Conventions: When instructed to either single click, click, or double click use the left mouse button. When instructed to right click use the right mouse button. A left click or double left click typically selects or executes an icon while a right click typically opens a menu of options associated with the icon.
Perform the following steps to complete this lab.

**Step 1. Logging to CDE.**
Login with the user name and password assigned by the instructor in the CDE entry box.

**Step 2. Examining the CDE Front panel.**
The middle section has four workspace buttons and icons for Lock, GO and Exit. There are 5 icons to the left and right of the middle section. Move the mouse cursor over each of these for one or two seconds to determine what each one does. Write the answers below.

<table>
<thead>
<tr>
<th>Left Side Front Panel Icons (left to right)</th>
<th>Right Side Front Panel Icons (left to right)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Step 3. Displaying subpanel menus**
There is also a subpanel menu button with an arrow on it above each of the 10 icons listed in the previous step. Single click on each one of these icons from left to right and record the menu heading on each below. Double click on the dash in the upper right corner of each menu to close it.

<table>
<thead>
<tr>
<th>Left Side Front Panel Subpanels</th>
<th>Right Side Front Panel Subpanels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Step 4. Displaying System Information**
Single click on the Hosts subpanel menu and then click on the System Info icon. Answer the following questions (answers will vary). Double click the dash button to close the workstation information window.

- a. What is the workstation name? ______________________
- b. How much Physical memory (RAM) is installed in this workstation? _________
- c. What operating system is installed? ______________________

**Step 5. Minimizing the front panel**
At the top left corner of the CDE front panel is a window menu button with a dash on it. Single click this button and minimize the front panel. This button can also be double-clicked. Double click on the minimized front panel icon to restore it to its original size.

**Step 6. Opening and minimizing a window.**
Single click on the Text Note icon on the front panel to open a text editor window.

Once the text editor window has been opened, convert the window to an icon by minimizing it. The student can click on the dash button in the upper left corner and choose minimize or single click on the button with a dot on it in the upper right corner. Double click on the minimized icon to restore the window to its original size.

**Step 7. Maximizing a window**
Maximize the Text editor window with the dash menu button in the upper left corner or the box button in the upper right corner. Click on the box button again to restore the window to its original size or use the dash menu button.
Step 8. Sizing a window.
Move the cursor over the any edge or corner of the Text editor window. A small bar with an arrow will be seen. While the bar and arrow are displayed, click and hold the left mouse button and drag the side of the window to enlarge it.

Step 9. Moving a window
Single click on the heading bar, that is the top bar, of the Text editor window with the left mouse button and hold the button down. Drag the window to a new location. Note the two numbers in parenthesis indicating screen position X and Y coordinates. Developers use these coordinates to bring up a new window at a specific location on the desktop. The student can also use the dash menu button and select move.

Step 10. Working with overlapping windows.
With the Text editor window open click on the Calendar icon from the front panel. Note that the Calendar overlaps or covers the Text editor window. Calendar is now the foreground window or process. Click on any visible portion of the window to bring the Text editor window back to the foreground. If not, click on the Calendar dash menu and select Lower from the menu. This will lower the Calendar window to the background and raise the Text editor window to the foreground.

Step 11. Close both Calendar and Text editor windows.

Step 12. Locking the user’s workstation
If user needs to leave the workstation for a while and wants to leave the desktop as it is, the user can lock that workstation. When the user returns they will need to provide a password to unlock the workstation. A system administrator can also unlock any workstation with the root or superuser password.

   a. Lock the workstation and then unlock it. What icon was used to lock the workstation?

Step 13. Working with Workspace buttons
Workspace buttons allow the user to have multiple desktops. The user can have one set of applications open in one workspace and another set of applications open in another. The user can click on each workspace button to move from one workspace to another. Each workspace has a different backdrop. The workspace buttons are numbered one through four. The buttons can be renamed by right clicking on one of the buttons. Workspace one is opened by default.
   a. While in workspace one, open a Text editor window.
   b. Click on workspace button number two and open a Calendar window.
   c. Click on workspace button number three and open a Printer jobs window.
   d. Click on workspace button number four and open a Mail window.
   e. Click each workspace button to move between the four windows.
   f. Close the application window running in each of the four workspaces.
   g. Right click on button number one and rename it to the student’s first name and press enter.

Step 14. Working with the Workspace menu
A user can access the Workspace menu by right clicking anywhere on the backdrop of the workspace. The options shown are similar to those available with the subpanel menus except that all applications are available.
   a. Right click on the backdrop and click Hosts from the menu displayed.
   b. Click on Workstation Info. This is the same information seen earlier.

Step 15. Logging out of CDE
Exit from the current CDE session and confirm that it is to be logged out of.
Objectives:
- Customize the student’s workspace with CDE Style Manager
- Locate common applications with Application Manager
- Work with Subpanels to add and remove icons
- Work with Front Panel to add and remove icons
- Work with the Workspace Desktop to add and remove icons

Background:
In this lab, the student will work with CDE Style Manager, Application Manager, Subpanel menus, Front Panel, and the Desktop to customize their workspace environment. There are many attributes that Style Manager gives the student control over including: Color, Font, Backdrop, Keyboard, Mouse, Beep, Screen, Window, and Startup. Applications Manager provides a common location for applications icons and Subpanels are the pull-up menus on the Front panel. The Front panel is the primary graphical user interface and the workspace desktop is the backdrop for icons and menus. Together, the panel and desktop provide ways to add and remove applications to customize the desktop environment. The student can add and delete new buttons and menu options for launching their most frequently used applications.

Tools / Preparation:
- Before starting this lab, the student should review Chapter 2, Section 3 – Customization Your Workspace and Section 4 – Working with Subpanels.
- The student will need the following:
  1. A login user ID, for example user2. The student will also need a password assigned by the instructor.
  2. A computer running the UNIX operating system with CDE.

Web Site Resources:
- Sun Microsystems – http://www.sun.com/solaris

Notes:

________________________________________________________
________________________________________________________
________________________________________________________
________________________________________________________
________________________________________________________
________________________________________________________
CDE Mouse Conventions: When instructed to either single click, click, or double click use the left mouse button. When instructed to right click use the right mouse button. A left click or double left click typically selects or executes an icon, while a right click typically opens a menu of options associated with the icon.

Perform the following steps to complete this lab.

Step 1. Log in to CDE
The student logs in with the user name and password assigned by the instructor in the CDE entry box.

Step 2. Access Style Manager (KDE works by accessing the Control Center and displaying the items available in the LOOK&FEEL tab)
If Style Manager is on the Front panel, click on the icon to access it. To access it through the Front panel menus, click on the Tools subpanel from the Front panel and select Desktop Controls. Click on any of the icons with the words Style Manager in the description and the Style Manager window will be displayed.

Step 3. Change Font Settings
Changing the font size using the Font Style Manager will alter the display character size of the next and subsequent windows opened. Change the size of the current font display then open a Terminal Window to verify the change.

Step 4. Change Backdrop Settings
The Backdrop choice enables the user to change the background for the current workspace. Each workspace can have a different backdrop applied to it. Change the backdrop of the current workspace and then change the backdrop of workspace three.

Step 5. Change Mouse Settings
Changing handedness of the mouse will reverse mouse buttons 1 and 3. The maximum time between a double-click can be adjusted. Changes to this will become effective with a new login session. The user can also alter the acceleration and threshold speeds of the mouse. Acceleration will change how fast the mouse pointer moves across the display. Threshold specifies the minimum number of pixels the user must move the mouse at one time before the pointer moves at the accelerated speed. Change the double click speed and acceleration.

Step 6. Change Startup Settings
If the Logout Confirmation Dialog option is set to on, the user will be asked to confirm that they want to log out when exit is chosen. Change the startup controls so that a confirmation is not requested when logging out.

Step 7. Test Changes you have made
Log out, exit, and then log back in to see the effects of the changes.

Step 8. Add an Application Icon to a Subpanel
The Application Manager window contains several Desktop folders that contain icons that can be added to a subpanel. The user can add applications to the subpanel menu list by dragging the appropriate icon from the Application Manager window display and dropping it on the Install Icon area of the subpanel.
   a. Click on the Applications subpanel to open it.
   b. Click on the Applications menu option to open the Applications Manager.
   c. Click on the Desktop_Apps icon and scroll through to find Calculator.
   d. Click on the Tools subpanel on the front panel to open it.
   e. Click and drag the Calculator from the Desktop-Apps window to Install icon in the Tools subpanel. This will add the Calculator as a main option on the Tools subpanel.

Step 9. Remove an Application Icon from a Subpanel
   a. Click on the Tools subpanel to open it.
b. Right click on the Calculator icon to display a menu of options.
c. Select Delete from the menu and click OK to confirm.

Step 10. Add an Application to the Front Panel
The user can place the most frequently used icons on the Front Panel.
   a. Right click on an open space in the Front panel and select Add Icon. This will open an open
      icon space for the new icon.
   b. Click and drag the Calculator icon from the Desktop_Apps window to the open icon area on
      the Front panel.

Step 11. Remove an Application from the Front Panel
   a. Right click on the Calculator icon.
   b. Select the Delete Icon from the menu and click OK to confirm.

Step 12. Add an Application to the Workspace Desktop
The user can also place frequently used icons on the Desktop.
   a. Click and drag the Calculator icon from the Desktop_Apps window to any open location on
      the desktop.

Step 13. Remove an Application to the Workspace Desktop
   a. Right click the Calculator icon.
   b. Select Remove from Workspace from the menu.

Step 14. Close All Open Windows/Applications that are on Your Desktop
Objectives:
• Use Nautilus to find help, set preferences, and navigate files and directories
• Use the launchers and menus on the GNOME panel
• Add, configure, move, and delete objects on the GNOME panel
• Create a floating panel

Background:
In this lab the student will become familiar with the GNOME desktop management environment. The student will use the Nautilus graphical shell to find their way around the system, and to customize GNOME according to the student’s preferences. The student will also learn to use the menus and launchers in the panel, how to modify what is in the panel, and how to add panels of their own.

Tools / Preparation:
To perform this lab the student will need:
 a) A computer with the GNOME desktop environment installed.
 b) A newly created login ID and a password. It is recommended to use a login that has never been used before to assure that the desktop settings have not been changed from the defaults.

Web Site Resources:
• GNOME - Computing made easy - http://www.gnome.org

Notes:

Conventions: When instructed to single click, click, or double click, use the left mouse button. When instructed to right click use the right mouse button. A left click or double left click typically selects or executes an icon while a right click typically opens a menu of options associated with the icon.

Perform the following steps to complete this lab.

Step 1. Login to GNOME and look around.
 a. If the student’s language is not English, choose another language from the Language menu at the top of the login box.
 b. Select GNOME from the Session menu on the left of the login box. The default is GNOME, but choosing GNOME explicitly assures that it is being used.
 c. Type in the student user name and press the Enter key. Type the password and press Enter.

When logging in, three types of objects will be seen. There are three icons on the left, an open window, and a panel across the bottom with several icons on it.

• The icons on the top left are on the Desktop.
• The window with Start Here on the left is a Nautilus window.
• The panel contains menus, launchers, and applets. The one on the left is called the GNOME foot, and is used to bring up the main menu.
There are nine objects on the panel: six on the left, a space, and three more on the right. Position the mouse pointer over each one for one or two seconds to determine what each one does. Write what is seen below.

<table>
<thead>
<tr>
<th>Left Side GNOME Panel Objects (left to right)</th>
<th>Right Side GNOME Panel Objects (left to right)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Step 2. Use Nautilus to find help.**

Elements of the Nautilus window are labeled to make their use intuitive. Across the top is a row of pulldown menus, below that a toolbar, and below that a Location entry field where the student can type in path names and Web URLs.

There is a sidebar to the left that is also used for navigation. Notice the tabs at the bottom left. The remainder of the Nautilus window is used to display the material being viewed. Move the mouse pointer to the File menu, press and hold the left mouse button, and move across the row of menu headings at the top to see what functions are provided from the menus.

a. What is done to find Help regarding the use of Nautilus itself?

b. What is done to find Help on other subjects, such as on topics within the GNOME Users Guide?

**Step 3. Use Nautilus to set preferences.**

Click on the File menu and select Close All Windows. Nautilus will exit. There are three ways to start Nautilus again. Execute one of the following:

a. Click the Start Here icon second from the top on the desktop.
b. Click the Start Here icon second from the left in the panel, next to the GNOME foot.
c. Press the GNOME foot and select Start Here from the menu.

In the main part of the Nautilus window is an icon labeled Preferences. Double click on it. Another screen appears showing what may be customized within GNOME. We will demonstrate by changing the desktop background. As time allows, experiment with other options.

Our objective in setting the background will be to select two colors that merge from one to the other from the top to the bottom of the screen.

a. Double click on the Desktop icon in the Nautilus window. Another screen of options related to the desktop's look appears.
b. Double click on Background. A new window from which to change the desktop appears.
c. Press the mouse on the selector that says Solid and drag the mouse so that it says Vertical Gradient, then let go.
d. Double click the small colored box next to Primary Color.
e. Drag the mouse in the color wheel or use the slide bars to the right to select a primary color to use, and then press OK.
f. Double click the box next to Secondary Color and repeat the process from the previous step.
Click the Apply button to see what the change looks like on the desktop. The change is not permanent yet. Work with the changes that were made until satisfied with the results.

To back out without making changes, press Cancel. If you have previously pressed Apply, your original settings will be restored.

To make your changes permanent, press OK.

**Step 4. Use Nautilus to navigate through files and directories**

Nautilus works much like a Web browser. If you have used graphical browsers such as Netscape or Internet Explorer, much here will be familiar. Here are a couple of simple examples.

Normally you could press the Home button on the toolbar to begin browsing files in your home directory. However, a brand new account has only a few files in its home directory, and these are invisible. Therefore, we will look instead at an important system file.

a. Put your mouse pointer in the entry field following Location and erase any text that is in it. In place of that text type `/etc` and press Enter.

b. Scroll down using your mouse to drag the scroll bar on the right until you see an icon for a file named `passwd`. **Note:** there are also files named `passwd-`, with a trailing minus sign and `passwd.OLD`. These are not what you want.

c. Double click on the icon to view the file. This is the file that lists all your system's login accounts. You should be able to find a line with your login name at the beginning, followed by a colon-delimited list of fields of information.

d. Press the Back button in the toolbar to return to the previous directory listing.

e. Click on the `passwd` icon and drag the mouse to your desktop background. What happened?

f. Try it again pressing the Shift key while you drag the icon. What happened?

g. To close Nautilus from the keyboard, press Shift+Control+W all at the same time.

h. Double click on the icon of the `passwd` file on your desktop. What happens?

i. Click in the Location bar, erase any text in it, and enter `http://www.gnome.org` and press Enter. What do you see?

j. Use your mouse to drag the `passwd` icon on top of the Trash icon on your desktop.

k. With your mouse pointer over the Trash icon, press the right mouse button, and select Empty Trash from the bottom of the menu.

l. Go to Nautilus and click in the small icon furthest to the right in the window frame, the one that looks like an X in a box.

Summarize what you learned in the previous sequence of steps.
Step 5. Use a launcher to start a terminal window.
In the panel at the bottom is an icon that looks like a computer terminal with a GNOME foot. This type of icon is called a launcher, a button that is used to start a program. This icon opens a terminal window running the Linux shell, called bash. The icons on the panel only require a single click to launch the program.

a. Click on the icon to open a terminal window.
b. Select the window by clicking on it with your mouse.
c. Enter the following sequence of commands and note the output. Press the Enter key after typing each command line in order to execute it.

```
hostname
id
pwd
ls -A
ls -A /etc
```

Summarize what you learned in the previous sequence of steps.

Step 6. Remove the time and date applet from the panel.
At the extreme right of the panel you see the time and date displayed. A type of program called an applet, one that is designed to run directly from your GNOME panel, is performing this action. You will replace it with an applet that looks nicer and that doubles as an email notifier.

To remove this or any other object from any panel,

a. Move your mouse cursor over the icon in the panel.
b. Click your right mouse button to bring up a menu.
c. Select the menu choice that says Remove from panel.
d. Did the applet disappear, leaving a blank space in its place?

Step 7. Add the Clock and Mailcheck applet to the panel.
To add the Clock and Mailcheck applet to the panel, follow these steps.

- Click the right mouse button with the pointer somewhere over a blank part of the panel.
- Select the following series of menu choices: Panel, Add to Panel, Applet, Network, and finally Clock and Mailcheck.

Note: In the case of applets you can skip passing through Add to Panel, and go directly to Applet, because an Applet by definition is a program that runs in the Panel.

a. Did the Clock and Mailcheck applet appear in the panel?

Step 8. Move the Clock and Mailcheck applet to the right on the panel.
To move the Clock and Mailcheck applet's icon to the right end:

- Right click with the pointer over the applet's icon.
- Click on Move. A cross-like symbol replaces the mouse cursor.
Move the mouse to the right. Notice that the icon follows it, even if the pointer is not over the icon, and even though you are not pressing the mouse button.
Click either the left or right mouse button to let go of the icon were you would like to place it.

a. What happens as the icon passes over other objects on the panel?

Step 9. Add a drawer to the panel.
A drawer is a panel that is located within another panel. It opens and closes to reveal and hide its contents. Anything that may be put in your main panel may be put in a drawer, including more drawers.

To create a drawer, follow these steps.

- Place your mouse pointer over the background of your main panel, where there are no icons, and right click.
- Select Panel, Add to Panel, and finally Drawer.
- Click the right mouse button with the pointer over the Drawer icon but do not select anything. Just look at the menu choices. Click again to close the menu.
- Do the same with the pointer over the gray box that hovers above the drawer. This object is an empty panel.
- From the menu that appears in step 4, select Applets, Amusements, and finally gEyes.
- Click on the Drawer icon in the panel.
- Click again on the Drawer icon.
- Click on the small arrow at the top of the panel that extends from the open drawer.
- Add a launcher for Emacs from the menus to the drawer's panel.

a. What steps did you take to add the Emacs launcher to the drawer panel?

b. How would you customize the look of the Drawer?

Step 10. Create a floating panel
Users may find it beneficial to park additional panels somewhere on their desktops. For example, a floating panel configured to stay closed unless the mouse pointer is over it might be used to hold system monitor applets.

To create a floating panel, follow these steps.

a. With your mouse pointer in a panel background, right click and select Panel, Create Panel, and finally Floating Panel. Where does the panel appear, and in what direction does it appear to be oriented?

b. Place the mouse pointer over the handle with the arrow on either end of your newly created panel, right click, and select Panel, Properties, then All properties to bring up the Panel properties configuration window.

b. Click the radio button that says Orient panel horizontally. The result should be seen immediately.
d. Enter values 120 and 4 for X and Y respectively to move the panel right of the desktop icons and down from the top just a little. Click the box that says Enable Auto-hide.

e. If you would like to change the appearance of the panel, click on the Background tab, select Color for Background type, click in the colored window next to Color to use, and use the color wheel and slide bars to select a color as you did in Step 3.

f. Click close on the Panel properties window.

g. Move the mouse pointer to the panel, which should be closed, but will pop open when the pointer moves over it. Be sure the cursor is in the small area between the two end handles with the arrows.

h. Right click and select Applets, Monitors, then CPU/MEM usage.

i. Move the mouse pointer in and out of the panel.

j. Add an object of any other sort to this panel. Describe the appearance and behavior of the new panel.

k. What sorts of objects might it make sense to add to a custom panel?

Step 11. Log out.

To log out from your GNOME session and return to the login screen, click on the GNOME foot, and select Log out, the last choice on the menu.

Optional Exercise
If you have time, go back and experiment with setting other desktop preferences. Begin by opening Nautilus as described in Step 3, then click on Preferences.
Fundamentals of UNIX
Lab 3.1.6 – Using CDE Mail Tool
(Estimated time: 30 min.)

Objectives:
- Become familiar with the graphical CDE Mail Tool
- Use Mail Tool Help
- Compose and send e-mail messages and attachments
- Respond to messages
- Delete and undelete messages
- Create Aliases for distribution lists

Background:
In this lab, the student will work with the CDE Mail Tool, also known as Mailer. Mail Tool is a full-featured graphical e-mail management program. The Mail Tool is an e-mail client that is a standard component of the Solaris CDE. With Mail Tool the user can perform all normal functions related to their e-mail. The user can read mail and attachments, create new mail with attachments, delete, and manage their e-mail with mailboxes.

Tools / Preparation:
a) Before starting this lab, the student should review Chapter 3, Section 1 – Using the Mail Tool
b) The student will need the following:
   1. A login user ID, for example user2, and a password assigned by their instructor.
   2. A computer running the UNIX operating system with CDE.
   3. Mail Tool application setup properly.

Web Site Resources:
- Sun Microsystems

Notes:
Perform the following steps to complete this lab.

**Step 1. Log in to CDE**
Login with the user name and password assigned to by the instructor in the CDE entry box.

**Step 2. Access Mail Tool**
Click on the Mail icon on the CDE front panel to activate the Mailer application. If a dialog box is displayed stating, “your e-mail setup does not indicate whether you have a local or IMAP mailbox” choose local.

**Step 3. Access Mail Tool Help**
Click on the Help icon in the upper right corner of the screen and then click on Tasks. Examine the tasks listed.

  a. Which task would be selected to sort the mail in various ways? ____________
  b. What options are available?

**Step 4. Compose and Send an e-mail Message to Yourself**
To compose a new mail message, click on New Message from the Compose menu. Make sure the address the message is being sent to is complete. More than one name, separated by commas, in the To: and CC: fields can be included. When finished creating the message, click on the Send button in the bottom left corner of the pane to send the message.

Compose a message and send it to the student’s mailbox. Be sure to use the format: userX@hostX where userX is the student’s login ID and hostX is the name of the student’s computer.

*Note:* The new message will eventually show up in the student’s inbox. To force the Mail Tool to check for new messages immediately click on Check for New from the Mailbox menu.

**Step 5. Compose and Send an e-mail Message to Another User**
Compose a message and send it to another person in the class. Be sure to use the format: userX@hostX where userX is the user name of the person the student wants to send to and hostX is the name of that person computer.

  a. What user and hostname was the message sent to?

**Step 6. Send a Message With an Attachment**
To attach a file to a message the student is composing, click on Add File from the Attachments menu. A window will be displayed from which the student can choose the file that is to be attached.

Compose another message and attach a file from the home directory. Send this message to another person in the class.

  a. What file was attached? ____________
  b. What user and hostname the message sent to?

**Step 7. Respond to a Mail Message**
To reply to a message that the student received, select the message then choose one of the Reply options from the Compose menu. The student is given the choice of replying to the sender or to all recipients of the message. The student is also given a choice of including the original message or not. If the student wants to reply and include the message received, the student can bypass the Compose menu and click on the Reply, Include Message icon on the toolbar of the Mail Tool. Reply to one of the mail messages that were received.
Step 8. Delete Mail Message
Choosing to delete mail messages does not immediately remove the mail. Rather, the messages are stored until the mail window is closed. This gives the user an opportunity to decide not to delete the message. The user can force the mail program to delete any mail messages that were previously selected for deletion by clicking on Destroy Deleted Messages from the Mailbox menu. Delete one of the mail messages that were received.

Step 9. Undelete Mail Message
While the student continues to work with the mail program, the student can view a list of the deleted mail messages and undelete one or more messages from the list. To restore a message that was just deleted, choose Undelete Last from the Message menu. To restore an earlier message, choose Undelete from List from the Message menu. When a message is undeleted, it will reappear in the list of received mail messages. Restore the message that was deleted in the previous step.

Step 10. Create an Alias
Mail Tool provides a tool that allows the user to set up aliases for either a particular person or a group of people that they frequently send mail to. A Mail Alias is similar to a group or distribution list with other mail systems. Click on Alias from the Options menu to create and manage mail aliases. Choose a name for the alias and enter the e-mail addresses of all users that are to be part of the alias. A comma must separate the e-mail addresses. After an alias has been entered, click on Add to place it in the list of aliases. When finished creating aliases, click on OK.

Create an Alias, including three fellow students, and send a message to the Alias.

Step 11. Create alternate mailboxes
In the CDE Mail Tool, a user can create multiple mailboxes for the purpose of storing mail for later retrieval. These mailboxes can be named to reflect the contents the user intends to store in them. Click on New Mailbox from the Mailbox menu to add another mailbox to the system. Once a new mailbox has been created, the user can then use the Move menu to move received mail into it and save it for future reference. Create an alternate mailbox to save messages in.

Step 12. Exit out of the Mailer application.
Objectives:
- Become familiar with the graphical CDE Calendar Manager
- Use Calendar Manager Help
- View the calendar various ways
- Set calendar options
- Work with appointments
- Work with To Do lists
- Work with other user’s calendars

Background:
In this lab, the student will work with CDE Calendar Manager. Calendar Manager is a full featured graphical schedule and appointment management program. It is a standard component of the Solaris CDE. With Calendar Manager the user can perform all normal scheduling functions and share their calendar with others. The user can set appointments and create To Do lists with Calendar Manager. The user can also view appointments and set meetings on other people’s calendars.

Tools / Preparation:
- a) Before starting this lab, the student should review Chapter 3, Section 2 – Using Calendar Manager
- b) The student will need the following:
  1. A login user ID, for example user2, and a password assigned by their instructor.
  2. A computer running the UNIX operating system with CDE.

Web Site Resources:
- Sun Microsystems – http://www.sun.com/solaris

Notes:
Perform the following steps to complete this lab.

**Step 1. Log in to CDE**
The student should login with the user name and password assigned by the instructor in the CDE entry box.

**Step 2. Access Calendar Manager**
To start Calendar Manager click the Calendar icon on the CDE Front Panel. By default, the Calendar window displays a month view of the student’s calendar.

**Step 3. Access Calendar Manager Help**
Click on the Help icon in the upper right corner of the window and then click on Tasks. Examine the tasks listed.
   a. Which task would be selected to learn how to change the default calendar view?

**Step 4. Change the Calendar View**
The view can be changed by clicking on one of the four view icons at the top right hand corner of the window. Change from the default Month view to the Day, Week and Year views.

**Step 5. Change Calendar Options – Editor Default Settings**
Click on the File, Options menu. The initial category displayed is Editor Defaults. In the Editor Defaults window, the student can set the options to suit any personal preferences.
   a. Change the default appointment Duration of 60 minutes to 30 minutes.
   b. Change the Privacy option to Others See Time Only.

**Step 6. Change Calendar Options – Display Settings**
Click on the Category button and select Display Settings.
   a. Change the time range for the working day to end at 6:00pm.
   b. Change the Default View for the Calendar Manager to Week.

**Step 7. Change Calendar Options – Access List and Permissions**
Click on the Category button and select Access List and Permissions.
   a. Remove the world from the access list so no one else can see that calendar.
   b. Enter the User Name, userx@hostx, of one of the other students and give them permission to View the public calendar entries. Click Add to add that student to the access list.

**Step 8. View Printer Settings**
Click on the Category button and select Printer Settings.
This window enables the student to set personal options for the printing of calendar appointments. What is the name of this default printer? ________________

**Step 9. Change Calendar Options – Date Format Settings**
Click on the Category button and select Date Format. The Date Format window enables the user to set the format for date display and date entry. Change the Date Ordering from MM/DD/YY to YYYY/MM/DD. The student may Change the date ordering back to whatever format preferred.

**Step 10. Add an Appointment Using the Appointment icon**
Change to the Day view on the student’s calendar. Click on the Appointment icon in the top left corner of the window and add an appointment to the calendar. Specify tomorrow’s Date, the Start and End times and What the appointment is. Click Insert to add the appointment to the calendar. Click on tomorrow’s date to verify the appointment that was made is there.

**Step 11. Add an Appointment by Clicking the calendar**
While in the Day view, Double click any time slot to set an appointment. The student should only have to enter What the appointment is and click Insert to add it.
Step 12. Add a Recurring Appointment
While in the Day view, click on next Tuesday’s date. Add an appointment for a regular staff meeting to occur every Tuesday at 9:00 a.m./0900 New York Time or 1400, Universal Time. The student may have to click on the More button to view the frequency and reminders options. Check a future week to verify that the weekly staff meeting is on the calendar.

Step 13. Create a To Do List
Click on the To Do button in the upper left corner of the window and add a To Do for project status due next Tuesday at 9:00 a.m./0900 New York Time or 1400, Universal Time. Click Insert to add it to the To Do list. Add another To Do for pick up laundry on Thursday at 6:00 p.m./1800 New York Time or 2400, Universal Time. To Do items do not show in the calendar. To see the To Do list for the week, click on the Week view for next week and then Click on the View menu option and select To Do List.

Step 14. Find an Appointment
The Find window enables a user to search for text in the appointment entries. The user can specify the time period to search, giving a start and end date. Once appointments have been found, the user can double-click on the details. Create an appointment a month from now and enter the words networker’s conference in the What entry area. Click back to today’s date. To search for the appointment, click on the View menu and then on Find. Enter the text to find as net. This will locate any appointments with the letters net in them.

Step 15. Browse Another User’s Calendar
With the correct permission settings a user can view other people’s calendars. The user can compare appointment times to help arrange meetings that do not conflict with existing appointments. Click on the Browse menu and the Show Other Calendar option. Enter the user name as userx@hostx of a fellow student that has allowed access to their calendar. The student should be able to at least view their calendar.

Step 16. Close All Open Windows/Applications that are on the student's Desktop
Fundamentals of UNIX
Lab 3.3.7 – Other Built-in CDE Applications
(Estimated time: 30 min.)

Objectives:
- Use Voice and Text note applications
- Use Address Manager to store information on contacts
- Use the Calculator application
- Use the Clock application
- Open a terminal window to gain access to the command line

Background:
In this lab, the student will work with several additional user applications that are included with the
Common Desktop Environment (CDE). These applications include: Voice and Text Notes, Address
Manager, Calculator, and Clock. The student will learn how to open a terminal window that will give
access to the UNIX command line. The student will use terminal windows throughout this course to
practice UNIX commands.

Tools / Preparation:
a) Before starting this lab, the student should review Chapter 3, Section 3 – Other Built-in CDE
Applications
b) The student will need will need the following:
   1. A login user ID, for example user2, and a password assigned by their instructor.
   2. A computer running the UNIX operating system with CDE.

Web Site Resources:
- Sun Microsystems – http://www.sun.com/solaris

Notes:
Perform the following steps to complete this lab.

**Step 1. Log in to CDE**
The student should login with the user name and password assigned to by the instructor in the CDE entry box.

**Step 2. Create a Voice Note**
The Voice Notes application allows the student to personalize the mail messages. Select Voice Note from the Applications subpanel to record voice input. Note that selecting Voice Note starts up Audio in record mode. A microphone or voice input device is necessary for this application to work. If the student has voice input capability, record a voice note.

**Step 3. Create a Text Note**
Text Notes allows a user to create Post-It™ style notes on their workspace. Select Text Note from the Applications subpanel to record text input. The note can then be minimized and left on the workspace as a reminder. The user can drag notes into mail messages as mail attachments. Create a short note and minimize the note. It should remain on the desktop.

**Step 4. Create an Address Card**
Address Manager enables a user to organize their contacts in electronic cards. The user can schedule appointments, send Email and dial telephone numbers from the electronic cards. The user can access the Address Manager in one of two ways:
- Right click on the desktop to bring up the Workspace Menu and then select Cards/Address Manager, or
- Access the Address manager from the Front Panel/Cards Subpanel by selecting the Find Card option.

Right click on the desktop, select Cards and then Address Manager. Click on Card from the main menu and the select New. A sample form is displayed for the student to fill in. The student should enter the requested information and then click on Save As when finished and give the card a name.

**Note:** Be sure to enter the student’s information at the tab positions in the body of the card or it may not be saved properly.

Create a couple of additional cards, saving and naming each one. The Cards that were created are stored in the home directory under the .dt/addresses subdirectory. Close Address Manager when finished creating the cards.

**Step 5. Search for an Address Card**
Open Address Manager by clicking on the Cards subpanel and then select Find Card. This will bring up Address Manager. Click on Search and make sure that the Personal Cards option is checked. Enter a name or any string of characters from the body of the card and press enter. The first card that matches will be displayed. All cards that match the student’s search should be listed in the ‘Found’ pull-down menu on the right.

**Step 6. Use the Calculator**
The calculator provides an online tool for quick calculations. It includes basic simple calculator capabilities and more advanced scientific calculator functions. The student can use the calculator in three modes: Financial, Logical, and Scientific. The student can also convert between decimal, binary, and hexadecimal display modes.

The student can place the Calculator on the desktop to have it available whenever it is needed. To do this, click on the Applications subpanel, and then click Applications to open the Application Manager window.

The default mode of the calculator is scientific with a decimal display. Use the calculator to practice adding some numbers. Next, enter the decimal number 255 and then click the decimal button and switch to binary.
a. What is decimal 255 equal to in binary? Ignore the decimal point and the zeros to the right of the
decimal point.

Step 7. Use the Clock
The Clock provides a graphical method of viewing the time, date, and time zone. To activate the clock,
Right click on the desktop and choose Applications, then select OW (Open Windows) Clock. The Clock
only views current time/date settings and cannot be used to change them. Right click on the clock to see
the options available. Change the display from analog to digital and back. Set the stopwatch and then
set an alarm.

Step 8. Open a Terminal Window
A terminal window can be opened in the graphical environment to provide a command line interface to the
system. Multiple terminal windows can be open at the same time. Each terminal window represents a
new shell and displays a shell prompt waiting for user input.

Anything the student can do from the command line can be done from a terminal window. The terminal
window provides a number of advantages over the command line environment. First, the terminal
window is scrollable, which allows the student to view output from previously entered commands.
Second, the backspace key is automatically enabled in a terminal window. With the command line, the
student must enter a series of commands for the backspace key to work properly. The student will be
working in a terminal window throughout much of this course as various UNIX commands and what those
commands do are learned.

To open a terminal window, right click on the desktop, click Tools and then Terminal. The student’s
cursor should be at a dollar sign prompt ($) if the Korn or Bourne shell is being used. The prompt should
be a percent sign (%) if the C shell is being used.

a. Enter the date command: What was the response? ________________
b. Enter the cal command: What was the response? ________________
c. Open another terminal window. Can the two windows be switched back and forth? ________

Step 9. Close All Open Windows/Applications that are on the student’s Desktop
Objectives:
- Become familiar with CDE Help functions
- Use the Help Viewer
- Search the Help Index for specific topics
- Get help on desktop icons with On-Item Help
- Investigate other ways to get help

Background:
In this lab, the student will work with several Help functions built in to the CDE to assist users when performing CDE related tasks. The student will use the main CDE help viewer and search the Help index for help on specific topics. The student will also use the On Item help feature to discover what desktop icons are.

Tools / Preparation:
- Before starting this lab, the student should review Chapter 3, Section 2 – Using Calendar Manager
- The student will need the following:
  1. A login user ID, for example user2, and a password assigned by their instructor.
  2. A computer running the UNIX operating system with CDE.
  3. Help installed.

Web Site Resources:
- Sun Microsystems – http://www.sun.com/solaris

Notes:
Perform the following steps to complete this lab.

**Step 1. Log in to CDE**
The student should login with the user name and password assigned by the instructor in the CDE entry box.

**Step 2. Investigate the CDE Help Subpanel**
The help subpanel is the common starting point for all types of graphical help available. It has a number of help options including CDE help. Click on the Help subpanel and list the options available:

<table>
<thead>
<tr>
<th>Option Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**Step 3. Access and Use the Help Viewer**
Click on the Help icon on the Front panel or click on Help Manager from the Help subpanel to bring up the help viewer window. The Help Viewer is a graphical interface for scanning online CDE-related help. The hypertext links are shown as underlined text or boxed graphics. These links can be used to quickly move to a related help page by clicking on the appropriate area of text in the Help window. The Backtrack button at the top right of the screen allows a user to retrace the path they followed through the help screens. The Print button can be used to print a copy of a particular help topic.

When the student first opens the help viewer, two hyperlinks are available. They are Common Desktop Environment and Overview and Basic Desktop Skills. Click on the 2nd hyperlink, and then click on the Introducing the Desktop link. Under the heading ‘Choose one of the following topics:’ how many options are available and what are they?

<table>
<thead>
<tr>
<th>Option Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**Step 4. Search the Help Index**
The Index Search enables a user to search all volumes of help. To access the Help Index, click on the Index button while in the Help Viewer window. The user can search for specific help items based on a keyword search. The number to the left of the help index item is the number of subtopics available.

Search All Volumes for help with the word printer in them.

a. How many entries were found? __________

b. Which entry has the most subtopics? ___________________
**Step 5. Use On Item Help**  
The On Item Help icon is on the Help subpanel. When this function is selected, the mouse pointer will change from an arrow pointer to a question mark with an arrowhead at its base. The pointer can then be positioned over a Front Panel icon to access the appropriate help page for that item, which will be displayed in a Help Viewer window. The mouse pointer will revert back to an arrow pointer once the help viewer opens. The On Item Help function will only work with icons on the Front Panel or an item on a sub-panel.

Click on the On Item Help icon from the Help subpanel. Move the question mark cursor over the world icon in the Front Panel, located above the Exit button, and click.

a. What did On Item Help say about the world icon?

**Step 6. Other Ways to Access Help**  
Right clicking in any free space on the desktop will display the Workspace menu. From the Workspace menu, the student can select the Help option from the submenu. This is another method for invoking the Help Viewer displayed by clicking on the Front Panel Help icon. Once the Help Viewer has been invoked, the student can use it as described previously. Most windows also have a Help menu option specific to the current application.

a. Right click on the workspace and select Help from the Menu. What options are available?

b. Click on the Calendar icon on the Front panel to start the Calendar Manager. Click on the Help menu option in the upper right corner of the window. What options are available?

**Step 7. Close All Open Windows/Applications that are on the student's Desktop**
Objectives:
- Become familiar with the AnswerBook2 Interface
- Identify some of the user and administrator Book Collections
- Use a browser to search AnswerBook2
- Search for information on an application
- Search for information on UNIX commands, hardware, and concepts

Background:
In this lab, the student will work with Sun Solaris AnswerBook2™ and become familiar with the various collections of available online manuals. The AnswerBook2 system enables the user to use their web browser to view on-line versions of many of the printed Solaris manuals, including the graphics. By default, AnswerBook2 uses the HotJava™ browser to display information. With AnswerBook2, the student can learn about Solaris through personal research and self-study.

Due to the amount of disk space taken up by the AnswerBook2 pages, the student’s system administrator may not have installed AnswerBook2 on the student’s computer hard disk. However, it is possible to access AnswerBook2 files from a compact disc-read only memory (CD-ROM) or a remote server on the student’s computer network.

Tools / Preparation:
a) Before starting this lab, the student should review Chapter 4, Section 2 – Referencing AnswerBook2
b) The student will need the following:
   1. A login user ID, for example user2, and a password assigned by their instructor.
   2. A computer running the UNIX operating system with CDE.
   3. AnswerBook2 installed on a local workstation or a network server.

Web Site Resources:
- Sun Microsystems – http://www.sun.com/solaris

Notes:
Perform the following steps to complete this lab.

**Step 1. Log in to CDE**
The student should login with the user name and password assigned by the instructor in the CDE entry box.

**Step 2. Access AnswerBook2**
If it is installed on the student’s system or a network server, there should be an icon on the Help subpanel, which will provide access to the AnswerBook2™ system. Click on the Help subpanel and select AnswerBook2. The student’s web browser will be activated and will serve as the primary interface to AnswerBook2. The student can move back and forward using the browser buttons.

**Step 3. Investigate the AnswerBook2 Collections**
There are several collections of online books containing a tremendous amount of information, including books in different languages. List some of the book collections that might be of interest as a Solaris user or system administrator. Answers may vary but should include the following:

<table>
<thead>
<tr>
<th>Collection Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Step 4. Search for Instructions on an Application**
To search the student’s Personal Library for the information on Calendar Manager, enter the words calendar and manager in the search window and click on the search button. The search words can be upper or lower case. The student will see several Book Collections listed with references to calendar in them. Some collections may list multiple books under them. The sections of the books that have the student’s topic in them are shown as a hyperlink. The circle to the left of the book section indicates how often the topic appears by how much of the circle is shaded.

a. Scroll down the list of book collections. Is Solaris 8 User Collection listed? ____________
b. Which book under the Solaris 8 User Collection would you look in to know more about how to use the CDE Calendar Manager? ____________
c. Which Section of the book would tell where to click on to learn how to schedule an appointment for someone else? __________________________________________________________________________________________

**Step 5. Search for a information on a UNIX command**
In a previous lab, the student used the **date** command to determine the current system date and time. The student can get help with any UNIX command by searching the manual or ‘man’ pages. A graphical version of the **man** command is available online with AnswerBook2. The next lab will provide practice on accessing ‘man’ pages from the command line.

a. Enter the word date in the search entry area and click on the search button to search for information on the date command. Is the Solaris 8 Reference Manual Collection listed? __________________________________________________________________________________________
b. What section of the man pages is the **date** command in? ____________________________________________________________________________
Step 6. Search for a Hardware Information
The student can also search AnswerBook2 for help with supported hardware and drivers. To see what support Solaris has for Fast Ethernet network interface cards (NICs), search for the words fast and Ethernet and then click search.

a. Look at the list of books under the Solaris 8 Installation Collection. Which Book would contain a list of network interface cards supported by Solaris for Intel?

b. To see network interface cards that are supported by Solaris for Intel which section should be viewed?

Step 7. Search for Information on a Concept or Topic
If the student were interested in learning more about Sun’s Network File System (NFS) the student could search for information on the topic. To see what online manuals contain information on NFS, enter NFS in the search window and click search. NFS is covered briefly later in the course.

a. Look at the list Solaris 8 System Administrator Collection of books. Which Book, when checked, would show an Overview of Managing File Systems?

b. Which Book, when checked, would get a list of NFS Commands?

Step 8. Close the AnswerBook2 Application
Double click on the dash menu button in the upper left corner of the window.
Objectives:
- Become familiar with function of man pages
- Access Command line help using the man command
- Use the man pages to determine the use of various UNIX commands
- Work with man pages and navigate through them

Background:
In this lab, the student will work with command line help in the form of UNIX man pages, which is short for manual pages. The man pages describe what is needed to know about the system’s online commands, system calls, file formats, and system maintenance. The online man pages are part of the UNIX computing environment and are installed by default. Man pages are in the form of simple character based screen displays and are not graphical.

To access the man pages, the student will need to be at a command prompt. The student may login at the command line or open a terminal window and start with a command prompt such as the Korn shell ($). Man pages are very helpful when a user wants to use a command or utility and they have forgotten the syntax or the user needs information about how to use it. The man pages will provide information on how to enter the command, a description of its purpose and what options or arguments are available.

Tools / Preparation:
- a) Before starting this lab, the student should review Chapter 4, Section 3 – Command Line Help
- b) The student will need the following:
  1. A login user ID, for example user2, and a password assigned by their instructor.
  2. A computer running the UNIX operating system with access to the command line.
  4. The `catman` utility must be run by root on each Solaris host.

Web Site Resources:

Notes:

---

---

---

---

---

---

---
Perform the following steps to complete this lab.

**Step 1. Log in to CDE**
The student should log in with the user name and password assigned by the instructor in the CDE entry box.

**Step 2. Access the Command Line**
To access the man pages the student must first be at a command prompt. To access a command prompt, login directly in character mode or open a terminal window under CDE. If a user bypasses CDE or if they telnet or rlogin to a remote computer, the user will have direct access to a command prompt.

**Note:** In this lab we will assume the student is running CDE but the commands will be the same regardless. Telnet and rlogin will be covered later in the course.

Right click on the workspace backdrop and click on tools. Select Terminal from the menu to open a terminal window. If the student is using the Bourne or Korn shells, the student will have a dollar sign ($) prompt. If the student is using the C shell the student will have a percent sign (%) prompt.

**Step 3. Navigate man Pages**
The output from some man pages can be as many as 10 to 20 screens of output. Several keys are helpful in moving around in the man utility once the pages for a specific command have been located. Enter the man intro and use the scrolling commands to move around.

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spacebar</td>
<td>Scroll one screen at a time</td>
</tr>
<tr>
<td>Return</td>
<td>Scroll one line at a time</td>
</tr>
<tr>
<td>b</td>
<td>Back - Move back one screen</td>
</tr>
<tr>
<td>f</td>
<td>Forward - Move forward one screen</td>
</tr>
<tr>
<td>q</td>
<td>Quit – Exit the man command</td>
</tr>
<tr>
<td>/string</td>
<td>Search forward for information</td>
</tr>
<tr>
<td>n</td>
<td>Next - Find the next occurrence of string</td>
</tr>
<tr>
<td>h</td>
<td>Help - Give a description of all scrolling capabilities</td>
</tr>
</tbody>
</table>

**Step 4. Use the Basic man Command**
The man command is used to display on-line man pages for any of the hundreds of UNIX commands that are available. The basic form is `man name` where name is the name of the command for which the user wants information.

Enter the following command: `$ man cal`

a. What kind of calendar does the `cal` command display? ________________

b. What is done to see the calendar for the year? ________________

c. What is displayed if no year is specified? ________________

**Step 5. Use the man command to find out what clear does**

a. Use the `man` command to find out what the UNIX command `clear` does. (man clear) What does it do? ________________

**Step 6. Use the man command to find out what pwd does**

a. Use the `man` command to find out what the UNIX command `pwd` does. (man pwd) What does it do? ________________

---

2 - 4 Fundamentals UNIX 2.0—Lab 4.3.7 Copyright © 2002, Cisco Systems, Inc.
Step 7. Use the man command to see a list of all UNIX commands
The student can get a listing of all of the Solaris UNIX commands with a brief description of what the commands do by entering the $ man intro command. Results may vary with other versions of UNIX.

Enter the following command: $ man intro

a. What does the Description say about the man intro command?

b. Look in the lower left corner of the screen. What is displayed in reversed white on black?

c. What does the percentage shown mean?

d. Press the space bar 5 times. What percentage has been displayed now?

e. Continue pressing the space bar until the cal command is seen. What does it say?

f. Press the letter q (quit) to abort the output of the man command.

Step 8. Use man to see man pages for the man Command
The student can even display a man page on the man command itself.

Enter the following command: $ man man

a. Which option is used to search man pages for a specific keyword?

Step 9. Use man to Search for a keyword
The student can search the man pages for a keyword to get a list of all man pages that have that keyword.

Note: With Solaris, it is necessary to run the catman utility before search the man pages by keyword. This can take considerable time on some systems and is normally completed by the instructor prior to class.

Enter the following command: $ man –k grep

a. List some of the man pages where the grep command was found

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
Step 10. Interpret man Page Headings
There are a number different headings or informational areas in a typical man page. The more common ones are:

<table>
<thead>
<tr>
<th><strong>NAME</strong></th>
<th>Name of the command and other commands that may accomplish the same thing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SYNOPSIS</strong></td>
<td>Shows the syntax of the command with any allowable options and arguments</td>
</tr>
<tr>
<td><strong>DESCRIPTION</strong></td>
<td>Gives an overview of what the command does</td>
</tr>
<tr>
<td><strong>OPERANDS</strong></td>
<td>Target of the command or what the command will take effect on such as a directory or a file</td>
</tr>
<tr>
<td><strong>OPTIONS</strong></td>
<td>Switches that can change the function or effect of the command. They are normally preceded by a dash (-) or minus sign</td>
</tr>
<tr>
<td><strong>SEE ALSO</strong></td>
<td>Refers the user to other related commands and subjects</td>
</tr>
</tbody>
</table>

These headings are displayed in the man page output using all capital letters. Depending on the command and its purpose, the man page may not contain all headings. For instance the `pwd` (Print Working Directory) command does not have Options or Operands information heading since there are no options or operands that can be used with the command. All commands will have at least a Name, Synopsis, and Description.

a. Enter the man command for `clear` and note which headings are shown.

b. Enter the man command for `pwd` and note which headings are shown.

c. Enter the man command for `date` and note which headings are shown.

Step 11. Close the Terminal Window
Double click on the dash button in the upper left corner of the screen.
Objectives:
- Become familiar with UNIX command line syntax
- Use various UNIX commands with options and arguments
- Use the man pages to learn about UNIX commands

Background:
In this lab, the student will work with various UNIX commands to develop an understanding of UNIX commands and syntax. Commands are typed at the shell prompt and they are instructions that tell the system to perform an action. Syntax refers to the structure of the command and specifies allowable options and arguments. The general format for UNIX commands is:

$ command [-option(s)] [argument(s)]. Items in square brackets are optional, meaning they are not always required.

<table>
<thead>
<tr>
<th>Command</th>
<th>Executable program that specifies what the user wants the system to do.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option(s)</td>
<td>Modifies the executable that specifies how the user wants the command to be run.</td>
</tr>
<tr>
<td>Argument</td>
<td>File or directory, including the pathname, or text. If a pathname is not specified for a file or directory argument, then the operating system will use the current directory.</td>
</tr>
</tbody>
</table>

A space must be used as a delimiter between each part of the command entered. UNIX commands are always lower case. Options are usually a single letter preceded by a hyphen (-), also called a dash or minus sign. Multiple options can be combined using only one hyphen. The option might be upper or lower case depending on the command. Many commands do not require all three parts. Multiple commands can be entered on one line by separating the commands with a semicolon (;).

Tools / Preparation:
- Before starting this lab, the student should review Chapter 5, Section 3 – Navigating the File System.
- The student will need the following:
  1. A login user ID, for example user2, and a password assigned by their instructor.
  2. A computer running the UNIX operating system with CDE.

Notes:
Perform the following steps to complete this lab.

Step 1. Log in to CDE
The student should login with the user name and password assigned by the instructor in the CDE entry box.

Step 2. Access the Command Line
To access a command prompt, login directly in character mode or open a terminal window under CDE. If a user bypasses CDE or if they telnet or rlogin to a remote computer, the user will have direct access to a command prompt. In this lab we will assume the student is running a terminal window in CDE, but the commands will be the same regardless. Telnet and rlogin will be covered later in the course.

Right click on the workspace backdrop and click on tools. Select Terminal from the menu to open a terminal window. If the student is using the Bourne or Korn shells, the student will have a dollar sign ($) prompt. If the student is using the C shell the student will have a percent sign (%) prompt.

Step 3. Use the `cal` Command
a. Enter the following command: `$ cal` What was the result? ________________

Step 4. Use the `cal` Command with Arguments
a. Enter the following command: `$ cal 2002` What was the result? ________________

Note: If the student is using a terminal window, the student can scroll up if months are off the screen.

b. Enter the following command: `$ cal 9 1752` What is wrong with this calendar?

__________________________

c. To find out why this is, use the man pages to learn more about the `cal` command. Scroll through the man pages and read the NOTES section to find an explanation of what happened to the calendar in September of 1752.

d. What is the reason the calendar for September of 1752 is missing 11 days?

_____________________________________________________________________

Step 5. Use the `date` command
Enter the following command: `$ date`

a. What was the result?

_____________________________________________________________________

b. What time zone is the student located in? ______________________________

Step 6. Use the `date` Command With an Option
Enter the following command: `$ date -u`

a. What was the result? __________________________________________________________________

b. When using the `date` command, the time is displayed as a 24-hour clock. Compare the results of steps 5 and step 6. How many hours is the student from Greenwich Mean Time (GMT)? __________

c. Use the man pages for the `date` command to find out what the `-u` option does with the date command. What did the man pages indicate?

_____________________________________________________________________

_____________________________________________________________________

_____________________________________________________________________
Step 7. Use the `banner` Command With an Argument
Enter the following command: `$ banner “hi there”`

a. What was the result?

Step 8. Use the `ls` command
The `ls` (list files) command requires at least two parts: the command and an argument. The argument is the file or directory the user wants to list. If the user does not specify an argument, it will default to the current directory. The `ls` command will be covered in greater detail later.

a. Enter the `ls` command by itself: `$ ls`  What was the result?

b. Enter the `ls` command with an option: `$ ls -l` (long listing). What was the result?

c. Enter the `ls` command with an option and an argument: `$ ls -l dir2` (long listing of the dir2 directory). What was the result?

Step 9. Close all open windows and/or applications.
Fundamentals of UNIX  
**Lab 5.3.3 – Navigating the File System**  
*(Estimated time: 20 min.)*

**Objectives:**
- Become familiar with the file system directory tree used in class  
- Determine the current working directory with the `pwd` command  
- Work with absolute and relative pathnames  
- Change directories from one location to another in the file system with the `cd` command

**Background:**
In this lab the student will work with the UNIX file system or directory tree, which has been set up for the class. The student will learn how to determine their current location in the directory tree and how to change from one directory to another.

Remembering which directory a user is currently working in is often difficult. The `pwd` (print working directory) command will display the absolute pathname of the current directory. The `pwd` command is used frequently to check the current location in the directory tree or hierarchy.

A user can move from one directory to another with the `cd` (change directory) command. The directory location a user wishes to change to is specified using an absolute or relative pathname. An absolute pathname specifies a file or directory in relation to the entire file hierarchy. The hierarchy begins at the `/` (root) directory. Absolute pathnames always start at the root (/) directory and list each directory along the path to the destination file or directory.

A relative pathname describes the location of a file or directory as it relates to the current directory or the directory a user is currently in. If a user is in a directory and they want to move down the directory tree, the user can type the path starting with the name of the next directory down in the directory structure. If a pathname does not begin with a slash, it is a relative pathname. Relative pathnames are useful because relative pathnames are usually shorter than absolute pathnames. To use relative path names, the user must know what directory they are currently in since that is the starting point. The `pwd` command will tell the user where they are in the directory tree or hierarchy. Shortcuts such as the tilde (~), dot/dot (..) can also be used with relative pathnames.

**Tools / Preparation:**
- Before starting this lab, the student should review Chapter 5, Section 3 – Navigating the File System
- The student will need the following:
  1. A login user ID, for example user2, and a password assigned by their instructor.
  2. A computer running the UNIX operating system with CDE.

**Notes:**
Use the diagram of the sample class file system directory tree to assist with this lab.

Class File Tree Structure

[Diagram of file system directory tree]

Step 1. Log in to CDE
The student should login with the user name and password assigned by the 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. Use the `pwd` Command
To determine the directory a user is currently in, which is known as the user’s current directory, use the `pwd` command. The `pwd` command does not have any options or arguments. The `pwd` command displays what directory the user is in using the absolute pathname, so there is no doubt. If the student has just logged in, the student’s current directory should be the home directory.

a. Enter the following command: $ `pwd` What is the current directory?
Step 4. Identify Absolute and Relative Pathnames
Use the Class File System Directory Tree diagram to answer the following questions:

a. What is the absolute pathname to the dir2 directory? ________________
b. What is the absolute pathname to the planets directory? ________________
c. What is the absolute pathname of the notes file? ________________
d. From the student’s home directory, what is the relative pathname to the dir4 directory? ________________
e. From the student’s home directory, what is the relative pathname to the flowers directory? ________________
f. If the student is in the dir1 directory, what is the relative pathname to the trees directory? ________________
g. Specify the relative pathnames for all of the dir1 subdirectories and files. ________________

Step 5. Use the cd (Change Directory) Command with Absolute Pathnames
Use ONLY absolute pathnames and no shortcuts.

a. Change to the home directory. What command was used? ________________
b. Change to the dir2 directory. What command was used? ________________
c. Verify what directory the student is currently in. What command was used? ________________
d. Change to the dir4 directory. What command was used? ________________
e. Return to the student’s home directory. What command was used? ________________
f. Change to the fruit directory. What command was used? ________________
g. Use the **pwd** command to verify the current working directory. What was the response? ________________
Step 6. Use the cd (Change Directory) Command with Relative Pathnames
A user can move around in the directory hierarchy using the cd command along with an absolute or relative pathname. Use only relative pathnames and the abbreviations or shortcuts shown in the table.

<table>
<thead>
<tr>
<th>Abbreviation / Shortcut Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>. (dot)</td>
<td>Current (working) directory</td>
</tr>
<tr>
<td>.. (dot/dot)</td>
<td>Parent directory, the directory directly above the current directory</td>
</tr>
<tr>
<td>~ (tilde)</td>
<td>User’s home directory (Korn and C shells)</td>
</tr>
<tr>
<td>cd</td>
<td>cd by itself changes to the user’s home directory</td>
</tr>
</tbody>
</table>

a. Change to the student’s home directory. What command was used? ________________
b. Change to the dir2 directory. What command was used? ________________
c. Verify what directory the student is currently in. What command was used? ________________
d. Change to the dir4 directory. What command was used? ________________
e. Return to the home directory. What command was used? ________________
f. Change to the fruit directory. What command was used? ________________
g. Use the `pwd` command to verify the current working directory. What was the response? ________________
h. Return to the home directory. What command was used? ________________
i. Change to the root directory. What command was used? ________________
j. Return to the home directory and enter the `ls` (list files) command. What directories and files are listed? ________________

Step 7. 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.
Fundamentals of UNIX
Lab 5.4.6 – Listing Directory Information
(Estimated time: 30 min.)

Objectives:
• Learn to display directory and file information
• Use the ls (list files) command with various options
• Display hidden files
• Display files and file types
• Examine and interpret the results of a long file listing
• List individual directories
• List directories recursively

Background:
In this lab, the student will use the ls command, which is used to display the contents of a directory. This command will display a listing of all files and directories within the current directory or specified directory or directories. If no pathname is given as an argument, ls will display the contents of the current directory. The ls command will list any subdirectories and files that are in the current working directory if a pathname is specified. The ls command will also default to a wide listing and display only file and directory names. There are many options that can be used with the ls command, which makes this command one of the more flexible and useful UNIX commands.

Command Format:  \texttt{ls} \ [-option(s)] \ [pathname[s]]

Tools / Preparation:
\textit{a)} Before starting this lab, the student should review Chapter 5, Section 4 – Listing Directory Contents
\textit{b)} The student will need the following:
1. A login user ID, for example user2, and a password assigned by their instructor.
2. A computer running the UNIX operating system with CDE.

Notes:
Step 1. Log in to CDE
The student should login with the user name and password assigned by the 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. Use the Basic `ls` Command
The `ls` (list files) command, when used by itself, will display a listing of all files and directories in the current directory. If the student has just logged in, the student’s current directory should be the home directory.

   a. Enter the command to change to the home directory. What command was used? __________

   b. Enter the command to verify the directory the student is currently in. What command was used?

   c. Enter the following command: `$ ls` What is displayed?

   d. Can it be determined whether the items listed are directories or files? _________________

Step 4. Use the `ls` Command with Arguments
Arguments for the `ls` command can be directory name(s) (relative or absolute) and file name(s).

   a. Enter the command to display the contents of the dir2 directory using a relative pathname from the student’s home directory. What command was used? ________________

   b. What was the response? ______________
c. Enter the command to list the files in the /etc directory, which is a standard UNIX directory under the root, using an absolute pathname. What command was used? __________

d. Enter the command to list the files in the planets directory using an absolute pathname? What command was used? __________

e. Enter the command to list only the dante file in the student’s home directory, to see if it exists and not see all other files and directories. What command was seen? __________

Step 5. Use the ls Command to see Hidden Files
File names that begin with a dot (.) are called hidden files. Hidden files are frequently used to customize a user’s work environment for example .profile, .dtprofile, .kshrc, .cshrc and so on. Hidden files are not shown by default because they are infrequently edited. The current directory link (.) and parent directory link (..) are also hidden and will not be displayed either since these links begin with a dot. Using the ls command with the –a (all) option will list all files in a directory, including hidden (.) files. Note that the –a option is lower case. The student should be in the home directory.

a. Enter the basic ls command without the any options. Are any hidden files seen, those files that begin with a dot? __________

b. Enter the command that will allow all files in the student’s home directory to be seen? __________

c. How many hidden files are there? __________

d. Enter the following to create a new empty file called .hiddenfile using the touch command (be sure to make the first character a dot). $ touch .hiddenfile

e. Enter the ls –a command again. Is .hiddenfile listed? __________

Step 6. Use the ls Command to See File Types
When using the ls command by itself, a user can obtain a listing of directory contents but cannot tell which are files and which are directories. By using the ls command with the -F (File type) a user can display a listing with a symbol to tell what the type of the file is. The symbol, if visible, is found at the end of the file or directory name. Note that the –F option is an upper case F. There are four UNIX file types: Directory, Executable, ASCII text file, and Symbolic link.

- **Directory**: A forward slash (/) after the name indicates this is a directory or subdirectory. A directory is considered a type of file with UNIX.

- **ASCII Text File**: If there is no symbol after the name this indicates a plain ASCII text file with no formatting characters in it. ASCII is the American Standard Code for Information Interchange. An ASCII text file is similar to a DOS text file.

- **Executable**: An asterisk (*) after the name indicates that this is a command, an application, or a script file, which can be run or executed.

- **Symbolic Link**: An at sign (@) after the name indicates a symbolic link which is a way of giving a file an alternate name. Symbolic links are covered later in this course.

a. From the student’s home directory, enter the basic ls command without any options. Could a user tell whether they are looking at files or directories if it were not for the fact that most of the directories have “dir” in their name? __________
b. Enter the command that will allow the student to see the file names in the home directory and their type. List the names of the directories: __________________________

c. What types of files are present? ________________________________________

d. The /usr/bin directory contains many UNIX executable commands, some of which the student has already used. Enter the command to see the files and types in the /usr/bin directory. What command was used? __________________________

e. What types of files are present? ________________________________________

f. The /etc directory contains many different type of UNIX system files. Enter the command to see the files and types in the /etc directory. What command was used? __________________________

g. Which different kinds of file types are seen? __________________________

Step 7. Use the ls Command to Displaying a Long Listing
The previous versions of the ls command displayed the names of directories and files in a wide format (across the screen). The ls command can be used with the –l (long) option to see more detailed information on each file or directory. The ls –l option will also distinguish between files and directories. Note that the –l option is a lower case letter L.

Shown below is an example of a long listing for a file (dante) and a directory (dir1). The listing is interpreted as follows:
• The first position of the display indicates whether this is a file or a directory.
• The lower case letter d indicates a directory.
• The dash (-) indicates a file.
• The group of characters shown as r, w, x and dashes are the permissions for the file or directory.
• The numbers shown as 1 and 5 are the number of links.
• User2 is the file owner.
• Staff is the file or directory access group.
• The 320 and 512 indicate the file size.
• The date and times shown indicate the date and time the file was created or modified.
• Finally the name of the file or directory is shown.

Links, owner, and group are covered later in the course.

```
-rw-r---r--  1   user2   staff   320   Dec 7  11:43  dante
drwxr-xr-x   5   user2   staff   512   Dec 4  13:43  dir1
```

a. From the student’s home directory, enter the basic ls command without any options. What information was displayed on each file or directory listed? __________________________

b. Enter the command that will allow a user to see a long listing for the file names in their home directory. What command was entered? __________________________

c. How many files are over 300 bytes in size? __________________________

d. Who is the owner of the files? __________________________

e. Using -t (time) option will list files with the most recently modified at the top of the list. To get a detailed or long listing of files sorted by time use the ls –lt version of the command. This will show the most recent at the top. Enter the ls –lt command. What is the most recently created or modified file? __________________________
Step 8. Use the `ls` Command to List Individual Directories

Use `ls -ld` to display detailed information about a directory, but not its contents. This is useful when a user wants to see the properties of a directory and not the information about its contents.

a. From the student’s home directory, enter the command that will provide a long listing of just the information for the dir2 directory. ________________

b. From the student’s home directory, enter the command that will provide a long listing for just the information on the fruit directory using a relative pathname. ________________

Step 9. Use the `ls` Command to List Directories Recursively

Use `ls -R` (recursive) to display the contents of a directory and all of its subdirectories. Recursive means to do again and again. This option is useful if a user wants to see all directories, subdirectories and their contents for a particular part of the directory tree. If this is done at a high level in the directory structure the output can be substantial. Notice that the –R option is an upper case R.

a. From the student’s home directory, enter the command that will provide a recursive listing for the dir2 directory. ________________

b. What was the result of the command? ____________________________

Step 10. 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.
Fundamentals of UNIX

Lab 5.5.2 – Directory Listings with Metacharacters

(Estimated time: 30 min.)

Objectives:
- Review some commonly used metacharacters
- Use the `ls` (list files) command with metacharacters
- Use the Asterisk (*) to substitute for zero or more characters
- Use the Question Mark (?) to substitute for a single character
- Use Square Brackets to substitute for a range of characters
- Use the semicolon to execute multiple commands on one command line

Background:
In this lab the student will work with various metacharacters and use them with the `ls` command to refine the student’s directory listings. Metacharacters are keyboard characters with special meaning to the shell. A general definition of a metacharacter is any keyboard character that is not alphanumeric. Metacharacters are used with many UNIX commands to provide greater flexibility. Some of the metacharacters used with UNIX are similar in function to those used with DOS. The asterisk (*) and the question mark (?) are metacharacters which are also known as wildcards. The student will work with the `ls` command and the following common metacharacters with this lab.

<table>
<thead>
<tr>
<th>Metacharacter</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>~</td>
<td>Tilde</td>
<td>Shortcut to home directory</td>
</tr>
<tr>
<td>*</td>
<td>Asterisk</td>
<td>Character substitution (also called splat)</td>
</tr>
<tr>
<td>?</td>
<td>Question Mark</td>
<td>Character substitution</td>
</tr>
<tr>
<td>[ ]</td>
<td>Square Brackets</td>
<td>Range definition</td>
</tr>
</tbody>
</table>

Tools / Preparation:
- a) Before starting this lab, the student should review Chapter 5, Section 5, Identifying and Using Metacharacters.
- b) The student will need the following:
  1. A login user ID, for example user2, and a password assigned by their instructor.
  2. A computer running the UNIX operating system with CDE.

Notes:
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
The student should login with the user name and password assigned by the 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. Use the Basic `ls` Command
The `ls` (list files) command, when used by itself, will display a listing of all files and directories in the current directory. If the student has just logged in, the student’s current directory should be the home directory.

a. Enter the command to change to the student’s home directory. What command was used?

b. Enter the command to verify the directory the student is currently in. What command was used?

c. Enter the following command: `$ ls` What is displayed?

Step 4. Use the `ls` Command With the Asterisk (*) Metacharacter
The asterisk (*) is a substitution symbol that represents zero or more characters, except the leading dot on a hidden file. The asterisk is often referred to as a *wildcard* character. If there were a large number of files in a directory and a user only wanted to see a listing of project files that started with p1 the user could use the asterisk to limit the numbers of files listed. As an example, the command `ls p1*` would list all files and directories starting with p1 and any number of characters after that. The asterisk can be placed anywhere, whether at the beginning, middle, or at the end of the string being tested. The asterisk can also appear multiple times. If a user enters `ls d*` they will see only those files beginning with the letter d and the contents of any subdirectories that start with the letter d.
a. Enter the command to list files and directories in the student’s home directory that start with the letter f. What command was used? = = = = What was listed? 

b. Enter the command to list files and directories in the student’s home directory that start with the letter d. What command was used? = = = = What was listed? 

c. Enter the command to list files and directories in the student’s home directory that end with the number 1. What command was used? = = = = What was listed? 

d. Enter the command to list files and directories in the student’s home directory that have the characters ‘ru’ anywhere in the file name. What command was used? = = = = What was listed? 

e. Enter the command to list files and directories in the coffees directory that start with the letter n using a relative pathname. What command was used? = = = = What was listed? 

Step 5. Use the ls Command With the Question Mark

The question mark (?) is a substitution character that matches any single character, except for the leading dot on a hidden file. The question mark is also referred to as a wildcard character. The example below shows the use of the of the ls command using the question mark in the fourth position. This indicates that the file or directory name must start with dir but any character can be in the fourth position and the file name cannot be more than four characters long.

$ ls  dir?

a. Enter the command to list files and directories in the student’s home directory that start with the letters file in the first four positions with anything in the fifth position but are not longer than five characters. What command was used? = = = = What was listed? 

b. What was listed? = = = =

c. Enter the command to list files and directories in the student’s home directory that start with the letter f in the first position with anything in the second and third positions and the characters e3 in the last two positions that are not longer than five characters. What command was used? = = = =

d. What was listed? = = = =

Step 6. Use the ls Command With Square Brackets

Square brackets ([ ]) can be used to match a set or range of characters for a single character position in the file or directory. The characters inside the brackets do not generally need to be in any order, for example [abc] is the same as [cab]. However, if a user is looking for a range of characters, the characters must be in proper order, for example [a–z] or [3–9]. If a user wants to search for all alphabetic characters, whether lowercase or uppercase, use [A–z] for the pattern to match. A user can use alphabetic or numeric characters for the search pattern.

The examples below uses square brackets along with the asterisk wildcard character. The first example defines a range and will list all files and directories that start with the lower case letters b through f with anything after that. The second example specifies that the first character must be either the letter a or f and anything can be after that.
\$ \textit{ls} \:[b-f]*
\begin{verbatim}
dante    dir1 dir3 file1 file3 fruit  practice
dante_1  dir2 dir4 file2 file4 fruit2
\end{verbatim}
\$ \textit{ls} \:[af]*
\begin{verbatim}
file1   file2   file3   file4   fruit   fruit2
\end{verbatim}

a. Enter the command to list files and directories in the student’s home directory that start with the letters f through p with anything in the remaining positions. What command was used? 
\begin{verbatim}
\text{What was listed?}
\end{verbatim}

b. Enter the command to list files and directories in the student’s home directory that start with any characters but have the numbers 1 through 3 in the last character. What command was used? 
\begin{verbatim}
\text{What was listed?}
\end{verbatim}

c. Enter the command to list files and directories in student’s home directory that start with either d or p and have any characters in the remaining positions. What command was used? 
\begin{verbatim}
\text{What was listed?}
\end{verbatim}

\textbf{Step 7. Use the Semicolon to Separate Commands}

The semicolon (;) enables a user to enter multiple commands on a single command line before pressing enter. The semicolon is also referred to as the \textit{command separator}. The example below shows two examples using the semicolon to separate commands. In the first example, the \texttt{clear} command will clear the screen, the \texttt{cd} command will return the user to their home directory and the \texttt{ls} command will list files in that directory. The second example displays the current date and time, and then the calendar for the current month.

\begin{verbatim}
\$ clear;cd;ls
\text{dante dir1 dir3 file1 file3 fruit practice}
dante_1 dir2 dir4 file2 file4 fruit2
\end{verbatim}

\begin{verbatim}
\$ date;cal
\text{Wed Feb 28 11:05:39 MDT 2001}
\text{February 2001}
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
S & M & Tu & W & Th & F & S \\
\hline
1 & 2 & 3 & & & & \\
4 & 5 & 6 & 7 & 8 & 9 & 10 \\
11 & 12 & 13 & 14 & 15 & 16 & 17 \\
18 & 19 & 20 & 21 & 22 & 23 & 24 \\
25 & 26 & 27 & 28 & & & \\
\hline
\end{tabular}
\end{verbatim}

a. Enter a series of commands on one line to clear the screen, display the current working directory, and then display a long listing of files in the student’s home directory. What series of commands were entered?
\begin{verbatim}
\text{What series of commands were entered?}
\end{verbatim}

\textbf{Step 8. Close the Terminal Window and Logout}

Double click on the dash button in the upper left corner of the screen, and then click the EXIT icon on the front panel.
Objectives:
- Use control characters to perform specific tasks
- Determine file type using the `file` and `strings` commands
- Display the contents of text files with the `cat` and `more` commands
- Display portions of text files with the `head` and `tail` commands
- Determine word, line, and character counts using the `wc` command
- Compare two files using the `diff` command

Background:
In this lab, the student will work with various informational commands. These are important because they allow the user to investigate and discover information about files. The student will use commands to help determine what type a file is and what application created it. The student will also work with several commands that allow the contents of text files to be seen and compare them.

The ability to analyze and manage files and directories using commands is very important in building a solid foundation for further study of the UNIX operating system. UNIX power users and system administrators must have a working knowledge of command line capabilities and syntax. Many operating system management and device configuration tasks require an understanding of UNIX commands and in some cases the command line is the only tool available.

Tools / Preparation:
- a) Before starting this lab, the student should review Chapter 6, Section 1 – Directory and File Management Using the Command Line.
- b) The student will need the following:
  1. A login user ID, for example user2, and a password assigned by their instructor.
  2. A computer running the UNIX operating system with CDE.

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

**Class File Tree Structure**

```
/  
|   
|  home  
|  user2  

dir1  
|   
|  trees  
|  fruit  
|  coffees  
|  beans  

dir2  
|   
|  beans  
|  recipes  
|  nuts  

dir3  
|   
|  planets  
|  flowers  
|  pluto  

practice  
|   
|  dir4  
|  dante  
|  dante_1  

file1  
|   
|  file2  

file4  
|   
|  fruit2  

file3  
```

**Step 1. Log in to CDE**
The student should login with the user name and password assigned by the 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. Use Control Characters to Perform Specific Tasks**
Control characters are used to perform specific tasks such as stopping and starting screen output and others. There are two control keys on most PC keyboards. These keys are normally labeled Ctrl and found in the lower left and right corners of the keyboard. On a Sun workstation, there is one control key in the lower left of the keyboard labeled Control. When displayed on the screen, the Control key is represented by the caret (^) symbol.

To enter a control character sequence, hold down the Control key and press the appropriate character on the keyboard. Control-c is a common control character sequence and is frequently used to interrupt or cancel a process. The actual character in the shell appears as ^C, even though the user presses the Control key and the c key at the same time. Perform the actions indicated to practice using some of the more common control characters.

- **Control-c:** Interrupts the current activity and may be used to abort or terminate processes or long display outputs resulting from the `man`, `cat` or `ls` commands. Control-c is also helpful in restoring the shell prompt if the user types an unrecognized command line, for example `$ ls "`, and receive the secondary prompt (>) in the Korn shell. With Linux, q (quit) is used to terminate the `man` command. Either control-c or q will work with Solaris.

  Display the `man` pages for the `ls` command (`man 1s`) and then abort the output with Control-c (Use q with Linux). What happened? _______________

- **Control-d:** Indicates end-of-file or exit. Control-d is used to exit some UNIX utilities such as bc, write, and several others. Control-d is used to exit a terminal window, and to logout of a terminal session or
command line login session. As a general rule, when stuck, or if Control-c does not work, try Control-d.

a. Start the basic calculator utility by typing bc at the shell prompt.
b. Multiply two numbers together by typing 458*594 and then press enter.
c. Exit the calculator by pressing Control-d.
d. What was the prompt while using the calculator?

**Control-u:** Erases the entire command line. The most common uses for Control-u are:
e. A quick way to erase a command line that the user decided not to execute.
f. If a user is logged into a remote system and the backspace key does not work.
g. To be used to ensure that the user is starting with a fresh user id and password entry when logging in.
h. Because passwords are not seen when they are typed, use Control-u to erase the password and start over when it is known that an incorrect character or characters have been typed.

If a user enters a command such as `ls -R /` by accident, the user would want to erase the command line before they pressed enter. Enter a command and Press Control-u before pressing enter to execute the command. What would the `ls -R /` command have done?

---

**Step 4. Determine File Type with The File Command**

There are many types of files found on a UNIX system. The file type can be determined by using the `file` command. This information can be important when a user is attempting to open or read a file. Determining the file type can help a user decide which program or command to use to open the file. The output from this command will most often be one of the following: Text, Executable or Data.

a. **Text Files:** Examples include ASCII or English text, commands text, and executable shell scripts. This type of file can be read using the `cat` or `more` commands and can be edited using vi or another text editor. Use the `file` command to determine the file type for the dante file in the home directory. What kind of file is it?

b. **Executable or Binary Files:** Examples include 32-bit executable and extensible linking format (ELF) code files and other dynamically linked executables. This file type indicates that the file is a command or program. Use the `file` command to determine the file type for the cal file in the `/usr/bin` directory. What kind of file is it?

c. **Data Files:** Data files are created by applications running on the system. In some cases the type of file is indicated, for example the FrameMaker (Desktop Publishing software) document. Use the `file` command to determine the file type for the beans file in the `dir1/coffees` subdirectory. What kind of file is it?
Step 5. Use the strings Command
The strings command can be used to print out readable characters in an executable or binary file. Someone with a programming background can interpret the output produced by strings. The command is introduced here solely as a method for demonstrating the printable characters of an executable file. The strings command must be used to read an executable file such as /usr/bin/cal. The strings command also shows the usage syntax of the command in most cases.

a. Use the strings command to see the readable characters in the /usr/bin/cal file. List some of the output from the strings command.


Step 6. Display the Contents of a File with the cat Command
The cat, short for concatenate, command displays the contents of a text file on the screen. The cat command is often used to display short text files such as script files which are similar to batch files. If the file fills more than one screen, the data scrolls off the screen. –This will happen unless the user is using a scrolling window, such as a terminal window, within the CDE environment.

a. Use the cat command to display the contents of the dante file in the home directory. What happened to the display of the text?

Step 7. Display the Contents of a File with the more Command
The more command is the preferred method of displaying a text file since this command automatically displays the file contents one screen at a time. If the information in a file is longer than one screen, the following message appears at the bottom of the screen where n is the percentage of the file already displayed: --More--(n%). Pressing the Enter key will continue the display one line at a time. The Space bar will continue one screen at a time.

a. Use the more command to display the contents of the dante file in the home directory. What happened to the display of the text?

Step 8. Display Portions of a File with the head Command
The head command is used to display the first n lines of one or more text files. The first 10 lines are displayed by default if the -n option is omitted. The head command is useful when the user only wants to check the first few lines of a file regardless of its length.

a. Use the head command by itself to display the first portion of the dante file in the home directory. How many lines were displayed?

b. Use the head command with the -n option to display the first 20 lines of the dante file in the home directory. What command was entered?

Step 9. Display Portions of a File with the tail Command
Use the tail command to display the last n lines of a file. The last 10 lines are displayed by default if the -n option is omitted. The tail command is useful for checking the most recent entries in large log files. Backup utility programs frequently write their results to a log file showing which files were backed up and when. The final entries in a backup log file are usually the total number of files backed up and messages indicating whether the backup finished successfully. The -n option displays the last n lines of the file.

a. Use the tail command by itself to display end of the dante file in the home directory. How many lines were displayed?
Step 10. Determine Line, Word and Character Counts Using the \texttt{wc} Command
The \texttt{wc} (word count) command can be used to display line, word, byte, or character counts for a text file. This command is useful when trying to determine characteristics of a file or when comparing two files. Using \texttt{wc} without options will give a line, word, and byte count of the contents of the file. Using it with individual options allows the user to determine which of these they would like to see.

\begin{enumerate}
\item Use the \texttt{wc} command to determine the number of lines, words, and characters in the \texttt{dante} file in the home directory. How many lines, words, and characters are there?
\end{enumerate}

Step 11. Count the Number of Directory Entries using \texttt{wc}.
Use the \texttt{wc} with the \texttt{ls} command to determine the number of entries, files and directories, in the student's home directory. To do this the user must pipe the output of the \texttt{ls} command to the \texttt{wc} command. The pipe symbol is the vertical bar that is on the same key as the backslash (\). At the command prompt, enter the command: \texttt{ls | wc -w}.

\begin{enumerate}
\item How many file and directory names, or words, are there?
\end{enumerate}

Step 12. Determine the Differences Between Files with the \texttt{diff} Command
The \texttt{diff} (difference) command is used to compare two text files and find differences between them. The \texttt{wc} command can be used to compare files since it counts lines, words, and characters. It is possible for two files to have the same line, word, and character counts but have different characters and words. The \texttt{diff} command can actually find the differences between the files.

The output of the \texttt{diff} command will display line-by-line differences between two text files. There are two options with the \texttt{diff} command, \texttt{-i} and \texttt{-c}. The \texttt{-i} option ignores the case of the letters, for example A is equal to a. The \texttt{-c} option performs a detailed comparison and produces a listing of differences with three lines of context. With this option, the output begins with identification of the files involved and their creation dates.

\begin{enumerate}
\item Use the \texttt{diff} command to perform a detailed comparison and determine the differences between the \texttt{fruit} and \texttt{fruit2} files. What lines, or fruits, are in the fruit file that are not in the fruit2 file?
\end{enumerate}

Step 13. 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.
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, the student 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. The student will create a simple directory structure and then create some files in those directories. The student 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, the student should review Chapter 6, Section 1 – Directory and File Management Using the Command Line.

b) The student will need the following:
   1. A login user ID, for example user2, and a password assigned by their instructor.
   2. A computer running the UNIX operating system with CDE.

Notes:
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
The student should login with the user name and password assigned by the 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 the student 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 (.), can be used multiple times in a file or directory name. For example, Feb.Reports.Sales is a valid file or directory name. 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.

- **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 (.). The student 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`. A user 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.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Is this a UNIX file or directory name? (yes/no)</th>
<th>Why or Why Not?</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. 12345abcde678</td>
<td>No</td>
<td>Lower case only</td>
</tr>
<tr>
<td>b. Hobbies:2</td>
<td>Yes</td>
<td>Contains numbers</td>
</tr>
<tr>
<td>c. Adcd-123</td>
<td>Yes</td>
<td>Contains numbers</td>
</tr>
<tr>
<td>d. Sales<em>repts</em>2001</td>
<td>No</td>
<td>Contains special characters</td>
</tr>
<tr>
<td>e. D.projects.bj.2001</td>
<td>Yes</td>
<td>Contains special characters</td>
</tr>
<tr>
<td>f. Projects&gt;1.bj-2001</td>
<td>No</td>
<td>Contains special characters</td>
</tr>
</tbody>
</table>

**Step 4. Create Files with the `touch` Command**

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

Using the `touch` command, a user 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. The student 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)**

a. From the home directory, change to the practice directory using a relative pathname. What command was used? ________________

b. Enter the `pwd` command to verify what directory the student is currently in. What was the response? ________________

c. Use the `touch` command to create a file in this directory called newfile. What command was used? ________________

d. Use the `touch` command to create another new file in this directory called filenew. What command was used? ________________

e. Enter the command to display a long listing of the files in the practice directory. Are the files the student created listed? ________________, Who is the owner of files? ________________

f. What is the group associated with the files? ________________

g. What is the date and time created? ________________

h. What is the size of the files? ________________

i. Use the `file` command to determine the file type for newfile. What kind of file is it? ________________
j. Create 3 files at the same time with the touch command: new1, new2, and new3. What command was used? 

k. Enter the command to display a long listing of the files in the practice directory. Are the three new files that were 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 the subdirectories can contain files. Directories can be created using either an absolute or a relative pathname. A user can specify more than one directory name on the same line to create more than one new directory. The user 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. A user can create multiple levels of directories including all the directories in a pathname simultaneously. If the student uses 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 the home directory, change to the practice directory using a relative pathname. What command was used? 

b. Use the mkdir command to create a subdirectory in this directory called newdir. What command was used? 

c. Enter the command to display a long listing of the files and directories in the practice directory. Is the subdirectory created by the student 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? 

g. How else could a user 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 was entered? 

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. A user can remove several files at once by specifying their names after the rm command or the user 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 the user is 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 the student created earlier from the practice directory. What command was used? ____________________________

b. Enter the command to display a long listing of the files in the practice directory. Is the file the student created gone? ____________________________

c. Use the `rm` command with the `-i` (interactive) option to remove the file new the student created earlier from the practice directory. What did the interactive option do? ____________________________

d. Remove the 3 files the student created earlier named new1, new2, and new3. Use the question mark (?) wildcard to remove all three with one command. What command was used? ____________________________

e. Enter the command to display a long listing of the files in the practice directory. Are the three files gone? ______________

f. What is another way a user could have removed the new1, new2, and new3 files? ____________________________

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. When the `rm` command is used with the `-r` option it can remove a single directory, whether it is 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 the student created earlier. What command was used? ____________________________

b. Enter the command to display a long listing of the files and directories in the practice directory. Is the subdirectory the student created gone? ____________________________

c. Change to the medium subdirectory the student created earlier. What command was entered? ____________________________

d. Remove the low subdirectory the student created earlier. What command was used? ____________________________

e. Change back to the practice subdirectory using a relative pathname and shortcuts. What command was entered? ____________________________

f. Remove the high and medium subdirectories with one command. What command was used? ____________________________

Step 8. Practice What Has Been 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 a user can create an entire directory structure with one command. Create multiple files in each of the directories. Remember a user can create multiple files with one command. Remove the files and then remove the directories so the student has no files or directories in the student’s 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.
Fundamentals of UNIX

Lab 6.2.6 – Basic CDE File Manager
(Estimated time: 30 min.)

Objectives:
• Learn to access CDE File Manager
• Identify graphical file and directory icons
• Compare File Manager options to command line file management
• Change between folders
• Create new files and folders also known as directories
• Remove files by putting them in the trash
• Recover files from the trash
• Permanently remove files by shredding them

Background:
In this lab the student will work with Common Desktop Environment (CDE) File Manager. The CDE method of file and directory management allows the user to do many of the same tasks that were performed earlier at the command line. The CDE provides a graphical interface to file management and executes most of the same commands the student used from the command line behind the scene. The File Manager enables the user to graphically organize files into a hierarchical structure of folders also known as directories and subfolders also known as subdirectories. The student will work with File Manager to become familiar with the graphical views of files and directories. The student will create new files and folders and practice deleting and recovering files.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 6, Section 1 – Directory and File Management Using the Command Line.
b) The student will need the following:
   1. A login user ID, for example user2, and a password assigned by the instructor.
   2. A computer running the UNIX operating system with CDE.

Notes:
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
The student should login with the user name and password assigned by the instructor in the CDE entry box.

Step 2. Access File Manager
File Manager can be accessed by clicking the File Drawer icon on the Front Panel or by right clicking on the workspace desktop and then on the Files menu. The File Manager, by default, opens a view of a folder that is the student’s home directory.

Note: the term folder is used interchangeably with the terms directory and subdirectory.

From that folder, a user can change to other folders, both up and down the hierarchy, to view each directory’s contents. The path to the current folder is always displayed in the upper area of the File Manager window.

   a. Click on the File Drawer icon on the Front Panel to start File manager. What is the directory path displayed in the upper area of the panel? ______________

Step 3. Identify File and Folder Icons
Directories are displayed as folder icons. Files are displayed as appropriate icons based on the type of file. File types are based on their function and the applications that created them. Examples of file types include: Audio, Binary, Core, Graphic, Postscript, and Standard. File Manager displays different icons depending on the content of the file to help the user distinguish and identify file types. The most common file icon a user will see will be the standard file icon. If a file is associated with a particular application, that application will automatically start when the icon is double-clicked.

   a. Scroll up and down through the window using the vertical scroll bar on the right side of the window. What types of icons does the student see in the home directory? ______________

   b. Right click on the dante file icon and click Properties from the menu. Click on the Information category button at the top. What information about the file is displayed? ______________

   c. What UNIX command would give the student similar information about the dante file if the student were at the command line? ______________
d. With the Properties Information window open for the dante file, open a terminal window by clicking on the File menu in File Manager and then click Open Terminal. The student may need to move the windows around to see them both at the same time. In the Terminal window, enter the command: `ls -l dante`. What information is provided in the Properties window that is not shown with the `ls -l` command? 

Step 4. Change Between Folders
If a user double-clicks on a folder icon, the File Manager moves the user into that directory and displays its contents. The user can only move down through the file system hierarchy in this way. There is always a special ‘go up’ icon displayed in the upper left corner of the files and folders window. Double-clicking on this icon will move the user up to the next level up in the directory hierarchy. Work through the following exercises, comparing the results to the class file directory tree diagram.

a. Double click on the dir1 folder. What is in this folder? 

b. Double click on the coffees folder. What is in this folder? 

c. What does the path name in the upper area of the window indicate as the current working directory? 

d. Right click on the beans file icon and select Properties and then click the information button. What kind of file is beans? Close the properties window. 

e. Double click the 'go up' icon until back at the home folder (/home/userX). How many times were required to double click to get to the home folder? 

f. Double click on the dir3 folder and then double click again on the planets folder. What is in the planets folder? 

g. Click on the File option from the menu and click Go Home. What is the current directory now? 

Step 5. Use File Menu Options
The File menu options enable a user to perform a number of tasks common to file and directory management tasks. These same tasks can be accomplished using the command line. The following is a list of File menu options with a brief description of each.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New folder</td>
<td>Create a new directory or subdirectory, with adequate permission</td>
</tr>
<tr>
<td>New file</td>
<td>Create a new file in any directory where the user has permissions</td>
</tr>
<tr>
<td>Go Home</td>
<td>Change to the user’s home directory if the user’s current folder is different from their home folder</td>
</tr>
<tr>
<td>Go up</td>
<td>Go up one level of folder in the directory tree or hierarchy.</td>
</tr>
<tr>
<td>Go To</td>
<td>Change to a specified directory</td>
</tr>
<tr>
<td>Find</td>
<td>Locate files based on their name or content</td>
</tr>
<tr>
<td>Open Terminal</td>
<td>Open a Terminal window with a shell prompt where the user can enter UNIX commands</td>
</tr>
<tr>
<td>Removable Media Manager</td>
<td>Provides access to removable media such as CD ROMs and floppies</td>
</tr>
<tr>
<td>Open Floppy</td>
<td>Access a floppy disk in the drive (DOS or UNIX)</td>
</tr>
<tr>
<td>Close</td>
<td>Close the File Manager window</td>
</tr>
</tbody>
</table>
a. Click on the File menu from within File Manager to access the options available. Which option would be used to create a directory in the current directory?  

b. Click on the menu option to go to the home directory. What option was used?

c. Click on the menu option to change to another directory and change to the dir2/beans directory. What option was used?  

d. Fill in the following table with the UNIX command that a user would use to accomplish the same task as the menu option listed.

<table>
<thead>
<tr>
<th>File Menu Option</th>
<th>Comparable UNIX Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Folder</td>
<td></td>
</tr>
<tr>
<td>New File</td>
<td></td>
</tr>
<tr>
<td>Go Home</td>
<td></td>
</tr>
<tr>
<td>Go Up</td>
<td></td>
</tr>
<tr>
<td>Go To</td>
<td></td>
</tr>
</tbody>
</table>

Step 6. Create New Folders
When the New Folder option is chosen from the File menu, a separate window is displayed where the user can enter the new folder name. Once the name has been entered, the user has the choice of left clicking either OK or Apply. Clicking OK adds the new folder and closes the New Folder window. Clicking Apply adds the new folder, and keeps the New Folder window open so the user can add another new folder.

a. While in the home folder, click on the New Folder option from the File menu and add a new folder called newfolder. Double click on the new folder created to make a change to it. Is there anything in the folder?  

b. What does the path at the top of the window show the current folder to be?

c. While in newfolder, create a directory called subfolder. Double click on subfolder. What is the path at the top of the window now?  

d. Click on the File menu and then click Go Home to return to the home directory.

e. Note the home and userX folder icons at the top of the window, which graphically show the path to the current directory. The home folder icon has a pencil with line through it and the userX folder does not. What does the pencil with a line through it on the home folder icon mean?

Step 7. Create New Files
As with the New Folder option, a window will be displayed in which the user type the name of the file to be created. If the user clicks OK, the file will be created and the window will close. If the user clicks Apply, the file will be created and the window will stay open so the user can create another new file.

If a user attempts to create a folder or file with the same name as an existing folder or file, the File Manipulation Error window will notify the user. Click OK, then type an alternative name for the folder or file that is to be created.

a. From the home folder change to newfolder created earlier. What is in the folder?
b. Click on the New File option from the File menu and create a new file called newfile1. Click apply and create another new file called newfile2. Close the New File window and note the contents of the newfolder directory. Who is the owner of the files?

Step 8. Remove and Recover Files
A benefit of File Manager over the command line environment is the ability to recover deleted files. This is also known as the “undelete” function. If a file is deleted using File Manager, it can undelete it if it has not been overwritten. Within the CDE, any file or directory that is deleted is placed within the trashcan. The files within the trash can be undeleted by selecting the put back option from the file menu. The trashcan also allows the user to shred the files or directories within the trashcan. Shredding will permanently deleting the files.

   a. Change to the newfolder directory. Click on newfile1 to select it and then click on the Selected menu and select Put in Trash or right click the file and select from the menu. Click on newfile2 to select it and then put it in the trash also. Are these files permanently deleted?

   b. Click on the Trash icon on the Front Panel and click on the Trash option from the menu. Are the files just deleted there?

   c. Click on newfile1 to select it and then click the File menu and select Put Back from the menu. The file should be restored. Can the file be seen in the newfolder in File Manager?

   d. Click on newfile2 to select it and then click the File menu and select Shred from the menu to permanently delete it. What was the result?

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.
Fundamentals of UNIX

Lab 6.3.6 - Basic GNOME File Management
(Estimated time: 40 min.)

Objectives:
Use GNOME's Nautilus file manager to:
  • Navigate folders
  • View files
  • Duplicate files
  • Link files
  • Create folders
  • Move files between folders
  • Change permissions on files

Background:
An earlier lab introduced the student to features of the GNOME desktop environment's user interface, including the use of Nautilus. In this lab we will concentrate on ways that Nautilus can be used to perform fundamental file management operations such as navigation, viewing, duplicating, renaming, moving, linking, changing permissions, and deleting files and folders.

Tools / Preparation:
To perform this lab the student will need
  a) A computer running Linux with the GNOME desktop environment installed.
  b) A login ID and password.

Notes:


Step 1. Launch Nautilus from Start Here.
   a. Login and start Nautilus from the Start Here icon on the desktop. What are two other ways to start Nautilus?

Step 2. Open the student’s home directory.
   a. Press the Home button in the toolbar. What icons are seen in Nautilus' main panel?
   b. What is an even quicker way to get to the student's home directory on starting Nautilus?

Step 3. Navigate using Tree in the Nautilus sidebar.
   a. Click the Tree tab in the Nautilus sidebar. What is seen here?
      • Click the arrow to the left of the first folder (/) to close and open it alternately. Leave it open.
        Note: do not click the folder icon.
      • Open /etc in the same way as above. /etc is located about a third of the way down the list.
      • Click the folder icon labeled init.d.
   b. What is the difference between clicking the arrows and the folders in the Tree sidebar panel?

Step 4. Switch to View as List.
   The icons in the main panel represent plain text files. The plain text files content may have little meaning to a user until more is learned about shell scripting. However, this directory contains a collection of files that the user can read and copy, which makes these files useful for this lab. This directory contains over 50 files. It may be more convenient to view its contents as a list.
   a. Click the pull-down menu on the right of the Location bar and select View as List. What difference did it make?
   b. What would cause this display to list files by size? Try to do this.
   c. What would cause this display to list by name, in reverse alphabetical order? Try to do this.

Step 5. View a file, then return to the folder.
   a. Locate the file in this folder named crond and double-click on it. If the computer asks whether the student wants to execute or display the file, select display. Briefly describe what is seen.
b. What would return the view to the directory /etc/init.d? List other ways that would cause this to happen.

Step 6. Put a file on the desktop.

a. Return to view /etc/init.d if necessary. Drag the icon for crond to the desktop and drop it somewhere below the Trash icon. What happened? Why did this happen?

b. Try again to put crond on the desktop using Ctrl+Drag. What difference did it make? Why did this make a difference?

Step 7. Learn to select single or multiple files.

a. In the window open on /etc/init.d, click the first file in the list. Ctrl+Click the third file in the list. What happens?

b. Click any other icon or filename to unselect the two files. Click again the first file in the list. Shift+Click the third file in the list. What is the difference between Ctrl+Click and Shift+Click in list view?

Optional Step: Experiment with selection in icon view

a. Switch to View as Icons and experiment with selection. What differences are there?

b. Return to View as List

Step 8. Copy files to the home directory.

- Choose New Window from the File menu to open a new Nautilus window.
- Open the student's home directory in the new window.
- Position the two windows so both windows can be seen in the main panel.
- Select the first five files in the window viewing /etc/init.d.
- Copy them to the home directory using Ctrl+Drag.

Step 9. Open a file with gedit.

a. Click one of the files copied to the home directory. From the File menu choose Open With followed by gedit. What happens?
b. In the gedit window, type a few words or letters of text. Do not worry about messing the file up. This file was copied as a junk file in which to scribble. Could the file be added to and saved?

c. Why was the file able to be changed?


a. Select one of the files in the home directory.
b. Duplicate the file by using the File menu or by using right-click to bring up the popup menu. Notice the name of the copy.
c. Rename the file to something that reflects good UNIX naming practice.
d. What characters are recommended for file names?

Step 11. Create and rename a new folder.

a. Make a new folder using New Folder from the File menu.
b. Rename it to my_folder.

Step 12. Move one file to the subdirectory.

a. Dragging files in the window that displays them may move files to a subdirectory one level down.
b. Select a file.
c. Drag and drop its icon on top of the my_folder icon.


a. Click on my_folder and choose Open in New Window. What is in the folder?

Step 14. Drag files to the new window.

a. Select two files in the home directory and drag them to the window viewing my_folder. What happened?

Step 15. Link a file within the home directory.

a. Shift+Click a file in the home directory and drag its icon to the desktop. What is different about the icon? Why is there a difference?
Step 16. Link a file within the home directory.

a. Right-click a file icon in the home directory and select Make Link. Note that using right-click directly is easier than clicking and going to the File menu. What is the link named?

b. Can links be made to folders as well? Try to do this.

Step 17. Cut, Copy, and Paste Files.

a. Highlight one file in the home directory and choose Copy File from the Edit or popup menu. Go to the window on my_folder and choose Paste Files from the Edit or popup menu. What happens?

b. Try the same sequence using Cut File in the home directory rather than Copy File. What happens?


a. Right-click a file in the home directory and choose Show Properties.
b. Click the Permissions tab in the Properties window.
c. Remove execute permission for everyone from the file.
d. Add write permission for group. How was this done?

Step 19. Drag items to the trash.

- Highlight unnecessary files and folders in the home directory and drag them to the Trash.
- Empty the trash from the file menu or from the popup menu with the mouse pointer over the Trash icon.
Objectives:
- Practice the use of the \texttt{cp} command to copy files and directories
- Copy files within the same directory to create backups
- Copy files to other directories
- Copy directories within the same directory
- Copy directories to other directories

Background:
In this lab, the student will perform more advanced file and directory management tasks using the command line interface and the \texttt{cp} (copy) command. Copying files is a normal occurrence when working with the file system. Files may be copied between local drives such as the floppy disk, hard disk, and CD-ROM. Files can also be copied between local drives and networks drives on servers. A common use of the \texttt{cp} command is to make a backup of an existing file for safe keeping so the original can be modified. The \texttt{cp} command can also be used to create a local backup of a directory or group of directories in a tree structure.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 7, Section 1 – Advanced Directory and File Management using the Command Line.

b) The student will need the following:
   1. A login user ID, for example user2, and a password assigned by the instructor.
   2. A computer running the UNIX operating system with CDE.

Notes:
Use the diagram of the sample Class File System directory tree to assist with this lab.

**Class File Tree Structure**

```
/  
 \   
 |   home
|    user2
|    dir1
|     users
|     fruit
|     beans
doctors
|     nuts
|     recipes
|     notes
|     planets
|     dir3
|     dir2
|     flowers
|     pluto
|     mars
|     practice
dir4
|     dante
|     dante_1
|     file1
|     file2
|     file3
|     file4
|     fruit2
```

**Step 1. Log in to CDE**
Login in the CDE entry box with the user name and password assigned by the instructor.

**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.

**Copying Files - Overview**
Files can be copied in several ways:

- A new file can be created with a different name in the same directory.
- Files can be copied to a different location in the directory hierarchy with the same or different name.
- Files can also be copied to a different disk such as a floppy or to a centralized server under the same or different name.

**Step 3. Copy Files Within A Directory**
If a user wishes to copy a file to create a new file with a different name within the same directory use the format below. The user can specify relative and absolute pathnames when using the `cp` command.

**Command Format:** `cp source_file destination_file`

This will copy from an existing filename (old source_file) to a new file name (new destination_file) in the same directory. **Note:** A user cannot have two files of the same name in the same directory.

a. Check to see if the working directory is currently the home directory. What command was used? ____________ What command would be used to change to the home directory if not there already? ____________

b. Copy the dante file and create a new file called dante.bak in the home directory to serve as a backup for the dante file. What command was used? _______________

c. Display a long listing for all files that start with the letters "da" and any other characters in the remaining positions. How many files were listed? ________________
d. Create a new file named proj-template in the home directory using the `touch` command. Copy this file to create another new file named proj-b. What command was used?

```
e. Create a backup of the proj-b file in the home directory named proj-b.bak by copying the proj-b file. What command was used?
```

```
f. Remove the proj-b file from the home directory to simulate the deletion or corruption of the file. Since the student has a backup file, what command can be used to restore the lost proj-b file?
```

```
g. Copy the fruit file to create another new file called fruit;new. Note: Place a semicolon between fruit and new. Could a new file be created?
```

```
h. What error message was received? Why was this error message given?
```

Step 4. Copy Files to Another Directory
To copy one or more files to another directory use the format:

```
Command Format: cp [-i] source_file(s) destination_directory
```

This format copies the existing file or files to another directory in the directory structure. When copying a file this way, it will normally have the same name in the destination directory. A user can add a slash and a file name after the destination directory to give the file a different name if desired.

If a file is copied and the name of the destination directory does not exist a new file is created with that name, otherwise the file is copied to the directory specified. When copying more than one file to a directory, the `cp` command assumes the last entry is a directory name and the prior entries are files.

a. From the home directory, list the contents of the practice directory and note the number of files in it.

```
b. Copy the dante.bak file from the current home directory into the practice directory for safekeeping. What command was used?
```

```
c. If the dante-bak file already existed in the practice directory, what version of the `cp` command would help prevent overwriting the dante.bak file?
```

```
d. If the practice directory did not exist or the directory name was mistyped, what would be the result of the previous command?
```

```
e. Copy the all of the files beginning with the lower case letter “f” and anything in the remaining characters from the home directory into the practice directory using a metacharacter (wildcard) and only one command. What command was used? How many files were copied?
```

```
f. Copy the beans file from the coffees subdirectory to the practice directory using a relative pathname. What command was used?
```

```
g. While in the home directory, create a new subdirectory called play under the practice directory. What command was used?
```
Step 5. Prevent Overwriting of Files
If a user copies a file and the target name already exists the user will overwrite or "clobber" the file and will not receive a warning. To prevent overwriting an existing file when copying, use `cp -i` (interactive) option as a security measure. The `-i` option only prompts the user if they are about to overwrite an existing file and gives the user a choice. Answering "y" will overwrite the file; answering "n" will return the shell prompt without copying. It is a good idea to get into the habit of using `cp -i` since this can prevent accidental mistakes.

a. If changes are made to the proj-b file and then the command: `cp proj-b proj-template`
   is issued, what will happen to proj-template if it already exists? ________________

b. What command could be used to be notified when the proj-template file was about to be overwritten? ________________

c. Type the command line from the above question and answer n (no) to override

Step 6. Copy a Directory within the Same Directory
To copy a directory and its contents to another directory a user must use the `cp -r` (recursive) command. If the destination directory does not exist, it is created. Without the `-r` option, files and subdirectories contained within a directory will not be copied and the user will receive an error. When used with the `-i` option, `cp` prompts for verification before overwriting an existing file.

Command Format: `cp [-r-] source_directory(s) destination_directory`

a. Verify that the working directory is the home directory. What command was used to do this?
   ________________

b. Copy the contents of the dir2 directory to create a new directory in the home directory called dir2.bak. What command was used? ________________

c. Use the `ls` command to verify that the dir2 directory has been copied. Are dir2 and dir2.bak seen? ________________

Step 7. Copy a Directory to Another Directory

a. Copy the contents of the dir2 directory to the practice subdirectory using a relative pathname. What command was used? ________________

b. Use the `ls` command to verify that the dir2 directory has been copied to the practice subdirectory. What command was used? ________________

c. Copy, using an absolute pathname, the contents of the planets directory to the play subdirectory, which was created earlier under the practice directory. What command was used?
   ________________

d. Use the `ls` command to verify that the planets directory has been copied to the play subdirectory. What command was used? ________________
Step 8. Remove Files and Directories Created in this Lab
Refer to the Class file system tree structure and remove all files and directories create in the home directory during this lab, including those creating under the practice directory. Use the `rm` and `rm -r` commands to accomplish this. Care should be taken during this process so use the `-i` option when removing the files and directories to ensure that these are the ones to be removed.

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.
**Fundamentals of UNIX**

*Lab 7.1.5—Renaming and Moving Files and Directories*

*(Estimated time: 45 min.)*

**Objectives:**
- Become familiar with the `mv` (move) command to rename and move files and directories.
- Rename a file in the current directory
- Rename a file in a non-current directory
- Move a file to another directory in the directory structure
- Rename a directory within the current directory
- Move a directory and its contents to another location in the directory structure

**Background:**
In this lab, the student will work with the versatile `mv` (move) command to rename and move files as well as directories. Files and directories can be renamed and moved to other locations in the directory structure using the same multipurpose command. There is no rename command in UNIX. The `mv` command changes the name of the original file, whereas the `cp` command copies a file and gives it a new name leaving the original file intact.

**Tools / Preparation:**

a) Before starting this lab, the student should review Chapter 7, Section 1 – Advanced Directory and File Management Using the Command Line.

b) The student will need the following:
1. A login user ID, for example user2 and a password assigned by the instructor.
2. A computer running the UNIX operating system with CDE.

**Notes:**
Use the diagram of the sample Class File System directory tree to assist with this lab.

**Class File Tree Structure**

```
/  
  ↓  
home  
  ↓  
user2  
  ↓  
dir1  
  ↓  
users  
  ↓  
fruit  
  ↓  
beans  
  ↓  
coffeers  
  ↓  
  
dir2  
  ↓  
  
  
dir3  
  ↓  
  
  
practice  
  ↓  
dir4  
  ↓  
dante_1  
  ↓  
file1  
  ↓  
file2  
  ↓  
file3  
  ↓  
fruitz2  
  ↓  
file4  
  ↓  
dante  
  ↓  
pluto  
  ↓  
mars  
  ↓  
notes  
  ↓  
recipes  
  ↓  
beans  
  ↓  
nuts  
  ↓  
```

**Step 1. Log in to CDE**

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

**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.

**Renaming and Moving Files - Overview**

Files can be renamed or moved in several ways:

1. A file name can be changed or renamed to a different name in the same directory.
2. Files can be moved to a different location in the directory hierarchy with the same or different name.
3. Files can also be moved to a different disk such as a floppy or to a centralized server under the same or different name.

**Step 3. Rename a File in the Current Directory**

The command format below shows the syntax to RENAME a file in the same directory. This format changes the name of the source, old File Name, to a target file name, new File Name, in the same directory. Note that the `-i` (interactive) option is available with the `mv` command. The `mv` `-i` option prompts for confirmation whenever the move would overwrite an existing target file.

**Command Format:** `mv  [i] source_file  target-name`

a. Check to see if the working directory is currently the home directory. What command was used? ___________. What command would the student use to change to the home directory if the student were not there already? ___________.

b. Copy the all files starting with the letters “fi” from the home directory to the practice directory using a relative pathname. What command was used? ___________. Verify that the files were copied using the `ls` command. How many files were copied? ___________
c. Change to the practice directory using a wildcard instead of typing the complete directory name of "practice". What command was used? 

d. Rename file1 to newname and verify. What commands were used? 

e. Create a new file using the touch command called proj-may-2001. However, after creating the file, suppose that the project begins in June, not May. Change the name to proj-june-2001. What command was used? 

Step 4. Rename a File in a Non-Current Directory
Files can be renamed in other directories without leaving the current directory by specifying the path to those files.

a. From the home directory, rename file2 in the practice directory to newname2. What command was used? Use the ls command to verify that the file was renamed. 

b. From the home directory, rename the beans file in the coffees subdirectory to the name java using absolute pathnames. What command was used? Verify that the file was renamed. 

c. From the home directory, change the java file name back to its original name beans using relative pathnames. What command was used? 

Step 5. Move a File to Another Directory
To move a file to a different directory, use the format shown below. This format moves the source file or files to a new target directory. A slash and a file name can be added after the destination directory to give the file a different name if desired. This version moves the source to a target directory name. This version tells the source what to move and where to move it.

Command Format: mv [i] source_file(s) target-directory

a. Create a new subdirectory under the practice directory called projects. What command was used? 

b. Change to the projects subdirectory and list the contents. Are there any files or directories in it? 

c. Create four new files in the projects subdirectory called June-1, June-2, July-1 and July-2 using the touch command. Create them all at once with one command. What command was used? 

d. To have a separate directory just for the July project files, create another new directory called proj-07. What command was used? 

e. Move the two project files for July, July-1 and July-2, over to the new directory that was just created using a wildcard and a relative pathname. What command was used? 

f. Would the command mv ju* proj-07 have moved only the July project files to the proj-07 subdirectory? Why not? 

g. Create a new directory named proj-06 and move the two project files for June, June-1 and June-2, into the new directory.
Step 6. Rename a Directory within the Current Directory

The `mv` command can also be used to rename a directory or move it to a different location. As with files, the `mv` command has two basic formats when used to rename and move directories. The first format renames a directory within the current directory. This is the Move Old Name to New Name format.

**Command Format:** `mv [i] source_directory target_name`

- Old Directory Name
- New Directory Name

a. Change to the projects directory and list the contents. Are the proj-06 and proj-07 directories both listed? List the contents of each subdirectory to verify that the two files exist.

b. From now on a new folder will be created for the projects for each month, and the month name will be used instead of the number. For example you want to use proj-june instead of proj-06. Rename the existing proj-06 directory to proj-june and the proj-07 to proj-july. What two command lines were used?

c. List the contents of the projects folder. Are the two projects folders named by month now?

d. Since a new directory will be created each month for projects when the year changes, the project directories will start to have the same name. Rename the existing proj-june to proj-01-june and rename the proj-july to proj-01-july so that the year is part of the directory name. Then the directories will be unique each year. What commands were used?

---

Step 7. Move a Directory and its Contents

This is the Move What to Where format of the `mv` command for use in moving a directory from one location in the directory tree to another location. When moving a directory this way, if the target directory location exists, the source directory will be copied into the target location. If the location does not exist, the source directory will be renamed.

**Command Format:** `mv [i] source_directory target_directory`

- What to Move
- Where to Move it

a. Change to the practice subdirectory and list the contents. Is the projects directory listed?

b. List the contents of the projects directory. Are the proj-01-june and proj-01-july directories both listed?

c. Enter the command to move the project subdirectory into the home directory. What command was used?
Step 8. Remove Files and Directories Created in this Lab
Refer to the Class file system tree structure and remove all files and directories created in the home directory during this lab. Include those files and directories created under the practice directory. The student will use the `rm` and `rm -r` commands to accomplish this. Care should be taken during this process so use the `–i` option when removing the files and directories to ensure that these are the ones to be removed.

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.
Fundamentals of UNIX
Lab 7.1.7—Redirection and Piping
(Estimated time: 45 min.)

Objectives:

- Become familiar with input / output (I/O) redirection
- Redirect standard output to create a File
- Prevent overwriting files with redirection
- Append output to an existing file
- Pipe the output from one command to another

Background:
In this lab, the student will use advanced UNIX commands to accomplish redirection and piping. Every UNIX command has a source for standard input and a destination for standard output. The input to a command is normally from the keyboard, although it can come from a file. The output from a command normally goes to the monitor or screen. The UNIX computing environment enables command I/O to be controlled using redirection. This is useful when attempting to save the output of a command to a file for later viewing. By piping, the user can take the output from one command and use it as input to another command for further processing.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 7, Section 1 – Advanced Directory and File Management Using the Command Line.

b) The student will need the following:
   1. A login user ID, for example user2, and a password assigned by the instructor.
   2. A computer running the UNIX operating system with CDE.
   3. Networked computers in classroom with class file system installed.

Notes:
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 in the CDE entry box with the user name and password assigned by the instructor.

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.

Input / Output Redirection - Overview
There are several metacharacters used as redirection symbols. Output redirection uses the right angle bracket (>), which is also referred to as the greater-than symbol. Input redirection uses the left angle bracket (<) or the less-than symbol. Error output redirection uses the right angle bracket preceded by the number two (2>). This lab will focus on output redirection.

General format: Command Redirection-Symbol File (text file or device file)

Step 3. Redirect Standard Output to Create a File
Standard output is redirected much more frequently than standard input or standard error. Many commands, such as ls, cat, head, and tail generate standard output to the screen and it is frequently desirable to redirect this output to a file for future viewing, manipulation or printing. By substituting a file name, the user can capture the output of a command rather than letting it go to the default monitor. This is a good way to create a sizable test file for practice.

The right angle bracket (>) or greater-than symbol allows the command to send output to a file. Using the single right angle bracket will create a new file if the file name specified does not exist. If the file name exists it will be overwritten. Note: the spaces between the command, the redirection symbol and the file name are optional.

Command Format: command > file

a. Verify that the working directory is the home directory. What command was use? 
What command would be used to change to the home directory, if not there already?
b. To keep track of what is in the home directory capture a listing of files and directories to a file in the home directory. What command would be used to redirect the output of a long file listing and create a new output file called homedir.list?

c. Where was this new file homedir.list placed? Use the ls command to verify that the new file is present.

d. What command could be used to view the contents of the file just created one page at a time?

e. Capture the first 10 lines of the homedir.list file using the head command and create a new file called dhomedir.list-top-10 using redirection. What command was used? View the contents of the file using the more command.

f. Capture the last 10 lines of the homedir.list file using the tail command and create a new file called dhomedir.list-bot-10 using redirection. What command was used? View the contents of the file using the more command.

g. Capture the output from the cal 2002 command to the file named calendar. View the contents of the file. What was captured?

h. Capture the output from the cal 2010 command to the file named calendar. View the contents of the file. What is in the file? What happened to the 2002 calendar?

Step 4. Prevent Overwriting Files with Redirection
In the Korn shell, an option called noclobber can be set to prevent overwriting of files during redirection. This can be done on the command line by using $ set -o noclobber. The ‘o’ stands for options. To reenable clobbering, use $ set +o noclobber. To enable/disable clobbering with the C Shell: % set noclobber and % unset noclobber.

a. Enter the command to turn on the noclobber on with the Korn shell. What command was entered?

b. Enter the command: ls -l > homedir.list What was the result?

c. Enter the command: ls -l > homedir.list2 What was the result?

Step 5. Append Output to an Existing File
The double right angle bracket (>>) can be used if the user wishes to append, add to the end, to an existing file instead of overwriting it. This option creates a new file if one does not exist or appends to an existing one.

Command Format: command >> file

a. Enter the command to display a banner that says: Happy Bday and use the redirection symbol to capture the output to a file called bday4me. What command was used?

b. Enter the command to display a banner that says: YOURNAME!, some name, and use the double redirection symbols to append the output to the bday4me file. View the contents of the bday4me file. What is in the file?

c. Enter the command to display the calendar for a specific birth month and year. For example, if someone were born in June 1965, enter: cal 6 1965. Use the double redirection symbols to
append the output to the bday4me file. What command was used?

d. View the contents of the bday4me file. Note that the output from three commands has been combined in the bday4me file. What day of the week was this person born on?

**Step 6. Pipe from One Command to Another**

One of the most powerful metacharacters is the pipe (|). The pipe takes the standard output of one command and passes it as standard input into the following command, usually the `more` command or the `lp` (line printer) command. The pipe can also pass the standard output into a file processing command like `grep`, or `sort` which is covered in Chapter 8. The pipe symbol is sometimes referred to as a double vertical bar and is found below the backspace key. A user must always have a command on each side of a pipe. Spaces between the commands and the pipe are optional.

**Command Format: command | command**

a. Use the pipe metacharacter to send the output of the `ls -l` command as input to the `more` command. What happened as a result of piping the output to the `more` command?

b. Look at the files listed with the `ls -l | more` command and note some of the dates created or modified. To see a listing of files or directories that were created or modified in the same month, the `grep` command can be used to search for that month. Specify the month exactly as it is displayed in the listing. (e.g. Oct). Enter this command: `ls -l | grep Oct` or enter the desired month. What was the result?

c. Directories always have a size of 512 bytes. Enter the command: `ls -l | grep 512`. What was in the resulting listing?

d. In KDE, the default size of a directory is 4096, so this command will not return any values.

e. Multiple pipes can be used to connect multiple commands. Enter a command that will take the output of the long file listing and pipe it to the `tail` command and then to the `sort` command. What command was entered?

f. The `ps` (process status) command is used to see what processes are running a UNIX system. Pipe output of the `ps -e` command to the `more` command. The `-e` option will show every process running on the system. What happened as a result of piping the output to the `more` command?

**Step 7. Remove Files and Directories Created in this Lab**

Refer to the Class file system tree structure and remove all files and directories created in the home directory during this lab. Include those creating under the practice directory.

**Step 8. 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.
Objectives:
- Learn to use more advanced features of CDE file managers
- Move files using drag and drop
- Copy files using drag and drop
- Move files using the selected options menu
- Copy files using the selected options menu
- Rename files using the selected options menu
- Put files in workspace using the selected options menu
- Set view menu options

Background:
In this lab, the student will work with some of the more advanced features and functions of Common Desktop Environment (CDE) File Manager. The student will perform more advanced file and directory management tasks such as those that were performed earlier at the command line. The student will move and copy files using the drag and drop as well as the menu options methods. The student will also rename files and copy them to the desktop. The File Manager methods of viewing will also be covered.

Tools / Preparation:
   a) Before starting this lab, the student should review Chapter 7, Section 2 – Advanced Directory and File Management Using File Manager.
   b) The student will need the following:
      1. A login user ID, for example user2, and a password assigned by the instructor.
      2. A computer running the UNIX operating system with CDE.
      3. Networked computers in classroom with the class file system installed.

Notes:
Use the diagram of the sample Class File System directory tree to assist with this lab.

Class File Tree Structure

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

Step 1. Log in to CDE
Login in the CDE entry box with the user name and password assigned by the instructor.

Step 2. Access File Manager
Note: In KDE, Start Applications -> System -> File Manager
File Manager can be accessed by clicking the File Drawer icon on the Front Panel or by right clicking on
the workspace desktop and then on the Files menu. The File Manager, by default, opens a view of a
folder that is the home directory. The term folder is used interchangeably with the terms directory and
subdirectory. From that folder, the user can change to other folders, both up and down in the hierarchy, to
view each directory's contents. The path to the current folder is always displayed in the upper area of the
File Manager window. Note: In KDE, Edit -> Create New -> Text File to create a new file.

a. Click on the File Drawer icon on the Front Panel to start File manager. What is the directory path
displayed in the upper area of the panel? ________________

Step 3. Move Files Using Drag and Drop
Moving files can be accomplished using the Select + Drag + Drop technique. To move a file from one
folder to another, position the mouse pointer over the file icon. Hold down the left mouse button and drag
the icon to the appropriate folder icon. Once the file icon is positioned over the folder icon, release the
mouse button and the file will be moved to that folder. The user must have write permissions to the target
folder in order to move a file into it. KDE prompts the user to Copy Here, Move Here, and Link Here when
file is dragged to a new location.

a. Create two new files in the home folder called myfile1 and myfile2 using the File Manager File
menu. Which option was used? ________________

b. Create two new folders in the home folder called mydir1 and mydir2 using the File Manager File
menu. What option was used? ________________

c. Move myfile1 file into mydir1 using the select, drag and drop technique. How can the user see if
the file is in mydir1? ________________, Return to the home folder. Note: Double click on the
Go Up icon if necessary.

d. Move myfile2 file into mydir2 using the select, drag and drop technique. Was the file moved?
______________
Step 4. Copy Files Using Drag and Drop
Copying files can be accomplished using the Control + Select + Drag + Drop technique. Press the Control key before the file icon has been selected and keep the key held down while the drag-and-drop process takes place. This will cause the file to be copied rather than just moved to the other folder.

**Note:** File icons cannot be moved or copied to the path icon display in the top part of the File Manager window display. If the user wants to move or copy files to the parent directory of the current folder, the file icon can be dragged and dropped to the Go Up icon.

a. Create two new files in the home folder called myfile3 and myfile4 using the File Manager File menu options. Which option was used? ____________________

b. Create two new folders in the home folder called mydir3 and mydir4 using the File Manager File menu options. What option was used? ____________________

c. Copy myfile3 file into mydir3 using the control + select, drag, and drop technique. Is the file in mydir3? __________ Is it still in the home folder? __________

d. Copy myfile4 file into mydir4 using the control + select, drag and drop technique. Is the file in mydir4? __________

Step 5. Move Files Using the Selected Options Menu
Moving files can also be accomplished using the Selected menu in File Manager. To move a file from one folder to another, click on the File icon to select it. Click on the Selected menu and then on the Move to option. **Note:** In KDE, select the file using the cursor, then click Edit -> Move Files or press F8.

a. Move myfile1 from mydir1 back to the home folder using the Selected menu option Move to. What was entered for a Destination folder? ____________________

b. Move myfile2 file from mydir2 to the mydir3 folder using Selected menu option Move to. What was entered as the Destination folder? ____________________

Step 6. Copy Files Using the Selected Options Menu
Files can also be copied using the Selected menu in File Manager. To copy a file from one folder to another, click on the file icon to select it. Click on the Selected menu and then on the Copy to option. **Note:** In KDE, select the file using the cursor, then click Edit -> Copy Files or press F7.

a. Copy myfile1 from the home folder to mydir1 folder using the Selected menu option Copy to. What was entered for a Destination folder? ____________________

b. From within the mydir3 folder, copy myfile3 to the mydir4 folder using Selected menu option Copy to. What was entered for a Destination folder? ____________________
Step 7. Rename Files Using the Selected Options Menu
Files can also be renamed using the **Selected menu** in File Manager.

a. Rename myfile3 to newmyfile3. What happened to the icon when Rename was chosen from the Selected menu?

b. Right click on myfile3 to bring up a small menu of options. Can the file be renamed from here? What can be done with the file using this menu?

c. What is the command used to rename a file when at the command line in a terminal window?

Step 8. Put Files in Workspace Using the Selected Options Menu

a. From within the mydir3 folder, copy myfile3 to the workspace, or desktop, using Selected menu option Put in Workspace. Can the file be seen on desktop? Is the file still in the mydir3 folder?

b. Double click on myfile3 on the desktop. What application was activated to open this file?

c. Right click on myfile3 on the desktop and select Remove from Workspace. Can the file be seen on desktop? Is the original file still in the mydir3 folder?

Step 9. Set View Menu Options
The View menu allows the user to customize the File Manager windows. There are a number of options available on the View menu. The Set View Options menu allows the user to change the way File Manager displays the files and folders in its windows. The user can also change the ordering of the display by name, file type, date, or size. Using Set View Options, the file system can be displayed by single folder or as a hierarchical tree. By using the Tree display for the folders, it becomes easier to move around the file system hierarchy.

a. Use Set View Options to change the representation of files and folders from Large icons to Name, Date and Size listing. How does this listing differ from the listing the user would get in a terminal window with the `ls -l` command?

   **Note:** In KDE, View -> View Mode -> Detailed List View. Change the view of the files and folder to whatever preference wanted.

b. Use Set View Options to change the view of files and folders to a Tree view showing folders first and then files. What is the default representation of files and folders when in the tree view? **Note:** In KDE, View -> View Mode -> Tree View.

Step 10. Remove Files and Directories Created in this Lab
Refer to the Class file system tree structure and remove all files and directories created in the home directory during this lab. Include those files created under the practice directory.

Step 11. Close CDE File Manager 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.
Fundamentals of UNIX

Lab 8.5.2—Finding, Searching and Sorting Files

(Estimated time: 45 min.)

Objectives:

- Become familiar the grep and sort file processing commands
- Find files by name using the find command
- Find files by type using the find command
- Find files by date last modified using the find command
- Search for strings in files using the grep command
- Editing files using the sed command
- Sort files with the basic sort command
- Sort files using options with the sort command
- Use CDE File Manager to find files

Background:
In this lab, advanced UNIX commands are used to find files and specific strings contained in files. CDE File manager will be used to locate files based on file name or file contents. The find command can be used to find files anywhere in the directory structure. The grep command is used to search for specific string or characters in file and list the files and lines where they are found. This lab requires the user to use the sort command to sort the contents of file and pipe the results of other command to the sort command and to grep.

Tools / Preparation:

a) Before starting this lab, review Chapter 8, Section 2 – File Processing Commands
b) The following materials are required:
   1. A login user ID (e.g. user2) and password.
   2. A computer running the UNIX operating system with CDE
   3. Networked computers in classroom with class file system installed

Notes:
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 by the 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.

Finding Files - Overview
The `find` command can be used to find files based on specific criteria. Once a file or group of files that match a search criterion is found, another command can be executed on the matching files. The `find` command can be used for many purposes including deleting, backing up, or printing files once they are found. The `find` command can be used to locate files on a local hard drive or on remote servers.

The `find` command starts at the point in the directory hierarchy specified and searches all directories and subdirectories below that point. Searching the hard drive by starting at the root can take a long time. Once a file is found, it is listed with the starting directory and any subdirectories below it. To learn more about the `find` command, refer to the man pages.

There are a number of options and variations with the `find` command. The format of the `find` command is shown below:

Command format:  

```
find  path  expression  action
```

Starting Directory  Search Criteria  Optional Commands
Path Options
The path names the directory where the search begins. Path can be a tilde (~) representing the home directory, a dot (.) representing the current directory, an absolute or relative pathname, or even the root directory.

Search Expression Options
Expression is one or more search criteria options that indicate what to look for and is specified by one or more values. Basic find options include file name, type, and size. Options must be preceded by a dash.

Note: When searching by filename the asterisk (*) and question mark (?) wildcards can be used but the string and the wildcard character must be in quotes (single or double).

Action Options
The Action option at the end of the command is optional and can be used to execute various commands after the desired file(s) have been found.

Step 3. Find Files by Name
In the following example, the search begins in the /usr/bin directory and finds all files whose name starts with the letter c.

Example: find /usr/bin -name 'c*' (or "c*"")

a. Enter the command to change to the home directory. What command was entered? 

b. Use the find command to locate a file named mars starting in the home directory. What command was used? What directory was it located in?


c. Starting in the dir1 directory, use the find command to locate a file named beans. What command was used? What directory was it located in?

d. Starting in the home directory, use the find command to locate all files and directories in the class file system tree that start with the letter p. What command was used?


e. How many directories and how many files were located?

f. What were their path and file/directory names?

Step 4. Find Files by Type
In the following example, the search starts in the /etc directory and finds all files whose type is d (directory)

Example: find /etc -type d

a. Starting in the home directory, use the find command to identify all files with a type of d (directories and subdirectories). What command was used?

b. How many directories were identified?
Step 5. Find Files by Date Last Modified

In the following example, the search starts in the users home directory and finds all files, which have not been modified for more than 90 days.

Example: `find ~ -mtime +90`

a. Use the `find` command to identify all files that have not been modified within the last 90 days. What command was used? ____________________________

b. How many files were identified? ____________________________

Step 6. Search for Strings in files

The `grep` (Global Regular Expression Print) command is used to search a file or the output of a command for a specified text string. A string is one or more characters; it can be a character, a word, or a sentence. A string can include white space or punctuation if enclosed in quotations. The `grep` command searches a file for a character string and prints all lines that contain that pattern to the screen. The `grep` command is frequently used in a pipeline with other commands. For instance, one can issue the `ps` (process status) command and look for all occurrences of a specific process. The `grep` command is case sensitive. The pattern must match with respect to uppercase and lowercase letters, unless the `-i` option is used, which ignores the case. The `-v` option searches for all lines that do not match the string specified.

Command format: `grep [option(s)] string path/filename`

In the following example, the `grep` command is used to search all files (indicated by the asterisk) in the current directory (indicated by the ./) to locate files that have the character string xyz in them.

Example: `grep xy z ./`

a. Use the `grep` command to identify all files in the home directory that have the word mango in them. What command was used? ____________________________

b. How many files were listed? _________ What are the names of the files? ____________________________

c. Change to the parent directory of the user home directory (/home). What command was used? ____________________________

d. Use the `grep` command to search all files in the home directory and list the filename and lines that have the word week in them. What command was used? ____________________________

e. How many file/lines were listed? ____________________________ What are the names of the files? ____________________________

f. Pipe the output of the `ls –l` command to the `grep` command and search for all files owned by current user ID. What command was used? ____________________________ How many files were listed? ____________________________
Step 7. File Editing with sed

The **sed** (stream editor) command is another pattern matching utility with effective processing capabilities. The sed utility reads lines of a text file, one by one. The sed utility applies a set of editing commands to the lines without opening the file in a text editor like vi. Like **grep**, **sed** makes no changes to the original file and sends the results to standard output. In order to make the changes permanent, the user must redirect the output to a file. Similar to the **grep** command, **sed** uses a number of special metacharacters to control pattern searching. **sed** is a very powerful and flexible command when used from the command line and in shell scripts.

**Command format:**
```
   sed [option(s)] address path/filename [>newfile] OR Command | sed [option(s)] address
```

<table>
<thead>
<tr>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sed -n '20,25p' file</code></td>
<td>Displays only lines 20 through 25</td>
</tr>
<tr>
<td><code>sed '5d' file</code></td>
<td>Deletes line 5</td>
</tr>
<tr>
<td>`ls –l</td>
<td>sed '/[Tt]est/d' &gt; newfile`</td>
</tr>
<tr>
<td><code>sed ‘s/….//’ file</code></td>
<td>Deletes the first four (…..) characters from each line.</td>
</tr>
<tr>
<td><code>sed ‘s/….$/’ file</code></td>
<td>Deletes the last four (…..) characters from each line.</td>
</tr>
<tr>
<td>`ls –l</td>
<td>sed ‘5,$d’ &gt; newfile`</td>
</tr>
<tr>
<td><code>sed –n ‘/^$/d’ file &gt; newfile</code></td>
<td>Deletes blank lines from file placing the results in newfile.</td>
</tr>
<tr>
<td>`ls –l</td>
<td>sed ‘s/ */:/g’`</td>
</tr>
<tr>
<td><code>sed ‘1,10s/Windows/UNIX/g’ file</code></td>
<td>Search (s) for Windows and globally (g) replace all occurrences of Windows on every line wherever it appears in the first 10 (1,10) lines.</td>
</tr>
<tr>
<td>`ls –l</td>
<td>sed ‘s/$/EOL/’`</td>
</tr>
<tr>
<td><code>sed ‘s/\^/ ‘/g’ file</code></td>
<td>Searches for the beginning of each line (^) of the file and adds spaces.</td>
</tr>
<tr>
<td><code>sed –e ‘s/Dante/DANTE/g’ –e ‘s/poet/POET/g’ dante &gt;newdante</code></td>
<td>Performs two edits on the same command line and places the results in the newdante file.</td>
</tr>
</tbody>
</table>

a. Change directories to the user home directory then **cat** the fruit file.
b. **Use sed** on the fruit file and delete all lines containing the word ‘orange’. By default the output will display on the screen. What command was used?
c. Generate a long listing of the files in the current directory. Use the output of a long listing again with `sed` to delete all lines containing the number 0 (zero). What command was used?

d. Redirect the output generated in step c) to a file named ls1.file. What command was used?

e. Using `sed`, delete the last line of the fruit file. What command was used?

f. Using `sed`, delete line number 2 to line number 5 from the fruit file. What command was used?

g. Pipe the output of the `ls -l` command to `sed` and delete lines 4 to the last line in the output. Place the output in a new file named ls2.file. What command was used?

h. Using `sed`, search for and delete the first three characters of each line in the file3 file. What command was used?

i. Using `sed`, append a pound sign (#) character to the end of each line of the fruit2 file. What command was used?

j. Using `sed`, substitute all occurrences of the string 'the' with 'SUN' in the dante file. What command was used?

Step 8. Sort Files with the Basic `sort` Command

The `sort` command provides a quick and easy way to organize data in either numerical or alphabetical order. The `sort` command works only with ASCII text files and will produce unpredictable results with executables or files created by applications such as FrameMaker. This command uses the ASCII character set as its sorting order, working from left to right on a character-by-character basis. By default, `sort` relies on white space to delimit (separate) the various fields within the data of a file. There are a number of options available with the `sort` command. These enable the operator to define the type of `sort` to perform as well as the field on which to begin sorting

Command format: `sort [options] [input_filename]`

In the following example, the `sort` command is used to produce an ASCII type of `sort`, beginning with the first character of each line for file2.

Example: `sort file2`

a. Display the contents of the fruit file in the home directory. What command was used?

b. Use the `sort` command to `sort` the contents of the fruit file. What command did was used? Are the names of the fruits in the file sorted in alphabetical order now?
Step 9. Sorting Files with sort Options
There are many options that can be used with the sort command. The example below is a numerical (n) sort on the second field of a file (sort skips one separator with the +1 syntax).

Example: sort +1n fileX

a. Create a file called pslist by redirecting the output of the ps -e (process status - list every process. Remember, the redirection symbol is a greater-than symbol) command. What command was used? ____________________________

b. Display the contents of the pslist file just created using the more command. How many column of information are there? ____________ Is the file sorted by any of these columns now? ______

c. The process ID or number is the first column. Sort the file on the first column (no options are required). Was the output from the sort command sorted by process ID? ______

d. The process name is the last or 4th column. To sort this file on the 4th column and redirect the output to a file named pslist2, what command would be entered?

Step 10. Find Files with File Manager
The Find option on the File manager File menu provides tools to locate files and directories based on various search criteria. File manager Find can perform many of the functions of the find command as well as the grep command. The criteria can be either the name of a folder or file or, in the case of a file, the contents.

a. Start File Manager by clicking on the file folder on the front panel and click on the File menu. Which option will search for files by name or contents? ____________

b. Select the following from the Find window: Find items in: My home directory, Whose name: Contains, the characters "file". Then Click on find. How many files were found? ____________

c. Start another search but this time click on the More Criteria button. Click on the name option to deselect it and then click on content. Click OK. Enter the word work in the field for what to search for and click on find. What was the name of the file that contained the word work?

________________________

d. Start another search. Click on the More Criteria button. Click on the content option to deselect it and then click on size. Click OK. Select Greater Than and enter 10 (10 Kilobytes or appx. 10,000 bytes) in the field for what to search for and click on find. Was the beans file listed?

________________________

e. Navigate to the beans file and right click on it. Click Properties and then Information. What is the exact size of the beans file? ____________

Step 11. Remove Files and Directories Created in this Lab
Refer to the Class file system tree structure and remove all files and directories created in the home directory during this lab.
Step 12. Close the File Manager Window
Click on the dash button in the upper corner of the window.

Step 13. 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.
Objectives:

- Become familiar with the UNIX vi text editor
- Review the three vi Modes
- Review keystrokes to move between vi modes
- Create a new file with vi
- Invoke vi with showmode
- Review the save and quit commands
- Open an existing file with vedit
- Use editing commands
- Customize the student’s session
- Use search commands

Background:
In this lab, the student will use a UNIX text-editing tool: the vi editor. The vi is pronounced “vee eye”. This text editor is primarily used for creating and modifying files that customize the user’s work environment and for writing script files to automate tasks. System administrators use text editors to create and modify system files used for networking, security, application sharing, and so on. The vi editor became a part of the UNIX operating system shortly after UNIX’s inception and is universally available with UNIX systems. The vi editor is a very flexible and powerful editor with many options. These will be reviewed here with examples of their use.

For users learning to become system administrators, it is important to know how to use vi. It is sometimes the only full screen editor available to edit crucial system files. Examples of these include scripts and environment control files. Skill in using vi is also needed if the windowing system is not available. The vi editor is a useful tool when working remotely on other UNIX workstations or servers. Administrators routinely remote login or telnet to another UNIX computer to perform maintenance and troubleshooting tasks using vi. The availability and operation of vi is consistent across all UNIX platforms.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 9, Section 1 – The vi Editor
b) The student will need the following:
   1. A login user ID, for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system.
   3. Networked computers in classroom with class file system installed.

Notes:
Use the diagram of the sample Class File System directory tree to assist with this lab.

Class File Tree Structure

```
/   
  |   
  home
  |   
  user2
  |   
  dir1
    |   
    users
    |   
    fruit
    |   
    coffe
    |   
    beans
  |   
  dir2
    |   
    beans
    |   
    recipes
    |   
    notes
  |   
  dir3
    |   
    planets
    |   
    flowers
    |   
    pluto
  |   
  dir4
    |   
    practice
    |   
    dir4
    |   
    dante_1
    |   
    dante
    |   
    file2
  |   
  file1
  |   
  file2
  |   
  file3
  |   
  file4
```

Step 1. Log in to CDE
The student should login with the user name and password assigned by the 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 the Three vi Modes
There are three modes of operation in vi. Understanding the function of these three modes is the key to working with vi. All commands available with vi can be classified in one of the three modes. The table below lists the modes and a brief description of each. Review the three modes and answer the following questions.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Function / Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command mode</td>
<td>Initial default mode for creating and editing files, cursor positioning, and modification of existing text. All commands are initiated from this mode.</td>
</tr>
<tr>
<td>Entry mode</td>
<td>Used for entry of new text. Entering an insert command such i (insert), a (append), and o (open new line) will take the user from command mode to entry mode. Entry commands are stand-alone and are entered without pressing the Enter key.</td>
</tr>
<tr>
<td>Last-line mode</td>
<td>Used for saving the user’s work and quitting vi. Type a colon (;) to get to this mode. Pressing the Enter key or Esc key returns to command mode.</td>
</tr>
</tbody>
</table>

a. Which vi mode is primarily used to enter new text? __________
b. Which vi mode is used to save the student’s work and quit vi? __________
c. When starting the vi editor, which mode is the default? __________
Step 4. Review Keystrokes to Move Between vi Modes

The table below shows how to switch modes and get from one mode to another. Review the keystrokes required to move between modes and answer the following questions.

<table>
<thead>
<tr>
<th>From Mode</th>
<th>To Mode</th>
<th>Commands / Keystrokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>Entry</td>
<td>i (input), o (open new line), a (append to existing line)</td>
</tr>
<tr>
<td>Entry</td>
<td>Command</td>
<td>Press Esc (Escape)</td>
</tr>
<tr>
<td>Command</td>
<td>Last-line</td>
<td>Colon (:)</td>
</tr>
<tr>
<td>Last-line</td>
<td>Command</td>
<td>Press Esc or Enter</td>
</tr>
<tr>
<td>Entry</td>
<td>Last-line</td>
<td>Press Esc to return to Command mode, then enter a colon</td>
</tr>
<tr>
<td>Last-line</td>
<td>Entry</td>
<td>Press Enter or Esc to return to Command mode, then enter an insert command</td>
</tr>
</tbody>
</table>

a. Which single character alphabetic commands will put vi in Entry mode? ________________

b. Which key will return vi to Command mode from either Last-line or Entry mode? __________

c. Which command will put vi into Last-line mode from Command mode? ________________

Step 5. Create a new File with vi

The vi editor is started from the command line. Whenever a user initiates vi, the user is opening a file. The user can specify the name of the file they want to create or edit when the user starts vi or they can open a new file to be named later. It is common to start vi and specify a file name. If the file exists, it is opened for editing. If the file does not exist, it is created.

Command Format: vi [option(s)] [filename]

a. Change from the student's home directory to the practice directory. Verify that the practiced directory is being used. What command was used? __________

b. Open a new file called myvifile using the command: vi myvifile. What does the vi document screen look like? __________________________

c. Press the lower case letter I to begin inserting text at the first line. Is there any indication on the screen that this is the Insert Entry mode? __________

d. Type in the student's name. If a mistake was made do not try to correct it at this time. Was text able to be entered in the Insert mode? __________

e. Press the Esc key to leave Insert Entry mode and return to Command mode. Is there any indication on the screen that the student is back in Command mode? __________

f. Type a colon (shift + :) to go from Command mode to Last-line mode. Is the student now at the bottom left corner of the screen at a colon (:) prompt? __________ If not, press Esc again, and enter another colon.

g. Type a lower case w (write), to save the new file. Follow this by a lower case q (quit), to exit the vi editor. What was the result of the wq commands? __________________________ What is the prompt now? __________________________

h. The new file should be saved in the student's practice directory on the hard disk. Display a long listing of this directory to verify that that file is there. How many bytes are in the file? __________________________
Step 6. Invoke vi with Showmode
With the basic vi editor there is no indication as to what mode the user is in. The showmode option displays the current mode in the lower right corner of the screen when the user is in any of the Entry modes. Entry modes are Insert, Append, or Open. A user can start vi and enter the Last-line command set showmode to turn showmode on or, with Solaris, the user can use vedit instead of vi. Vedit starts the vi editor with showmode turned on. If the user is in command mode, nothing is displayed. If the user is in last-line mode the cursor will usually be in the lower left corner with a colon displayed. Use vedit when possible so the student will know what mode the student is in. The remaining exercises will use vedit instead of vi. With most Linux distributions, running vi actually runs vim (vi improved) that displays ‘INSERT’ when in entry mode. Last-line mode is indicated by a colon (:) and command mode is not indicated. With Solaris, run vedit and with Linux run vi, which runs vim.

a. Open another new file called myvifile2 using the command: vedit myvifile2. What does the vi document screen look like?

b. Press the lower case letter ‘i’ to begin inserting text at the first line. What is displayed in the lower right corner of the screen?

c. Press the Esc key to leave Insert Entry mode and return to Command mode. Is there any indication on the screen that the student is back in Command mode?

d. Press the lower case letter ‘o’ to open a new line. What is displayed in the lower right corner of the screen?

e. Press the Esc key to leave Open Entry mode and return to Command mode. Is there any indication on the screen that the student is back in Command mode?

f. Press the lower case letter ‘a’ to Append to the current cursor position. What is displayed in the lower right corner of the screen?

g. Press the Esc key to leave Append Entry mode and return to Command mode. Is there any indication on the screen that the student is back in Command mode?

h. Type a colon (shift + :) to go from Command mode to Last-line mode. Is the student now at the bottom left corner of the screen at a colon (:) prompt? If not press Esc again and enter another colon.

i. Type a lower case ‘q’ (quit) to exit the vi editor and an exclamation mark (!) to quit immediately and ignore any entries that may have made. What is the prompt now?
Step 7. Review the Save and Quit Commands
In the previous steps, the student created a file and saved it with the \texttt{w} (write) command and exited vi with the \texttt{q} (quit) command. The student also exited vi without saving any changes using the \texttt{q!} command. There are several Save and Quit commands available with vi. The following table lists some of the more common ones.

\textbf{Note:} Save and Quit commands, except for ZZ, are entered only when in Last-line mode.

<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>:w</td>
<td>Write buffer. Save changes and continue working in vi.</td>
</tr>
<tr>
<td>:w new_filename</td>
<td>Write buffer to new_filename and continue working in vi.</td>
</tr>
<tr>
<td>:wq</td>
<td>Write buffer, save changes, and quit vi</td>
</tr>
<tr>
<td>ZZ (upper case)</td>
<td>Save changes and quit vi. Alternative to :wq</td>
</tr>
<tr>
<td>:q!</td>
<td>Quit without saving changes</td>
</tr>
<tr>
<td>:wq!</td>
<td>Write buffer, save changes, and quit vi. The ! will override read only permission if the user is the owner of the file.</td>
</tr>
</tbody>
</table>

a. Most save and quit commands are entered in which mode? 

b. Which command will allow the student to exit vi and not save any of the changes the student made?

c. Which two commands listed in the table perform the same function?

d. Which command allows the student to save the current file the student is editing under another name and remain in vi to continue editing the current file?

Step 8. Open an Existing File with vedit
If the student starts vi or vedit and specifies an existing file name, it is opened for editing. Here the student will add some new text in Entry mode and try a few cursor positioning commands.

a. Open myvifile that the student created earlier using the command: \texttt{vedit myvifile}. What does the vi document screen look like?

b. Position the cursor at the end of the student's name and press the lower case letter 'a' to Append to the line the student typed earlier. Type some text, and then press Enter for a hard return and type some more text. Enter about three lines of text this way. What mode is the student currently in?

c. Press the Esc key to leave Append mode and return to Command mode. In Command mode, can the student position the cursor for additional editing?

d. The student can move the cursor with the arrow keys while in the various entry modes and the student will remain in that mode. The table below contains some of the more common cursor positioning keys and commands. If the student uses the alphabetic commands while in an entry mode the commands will be entered as text. The student will need to press Esc to get to command mode to use them. The student should practice using these while editing this file. Which character moves the student back one word at a time?
Cursor Positioning Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>j or Down arrow</td>
<td>Move down one line</td>
</tr>
<tr>
<td>k or Up arrow</td>
<td>Move up one line</td>
</tr>
<tr>
<td>Space bar</td>
<td>Move right or forward one character</td>
</tr>
<tr>
<td>w</td>
<td>Move forward one word, including punctuation</td>
</tr>
<tr>
<td>Back Space</td>
<td>Move left one character</td>
</tr>
<tr>
<td>$</td>
<td>Move to end of line</td>
</tr>
<tr>
<td>0 (zero) or ^</td>
<td>Move to beginning of line</td>
</tr>
<tr>
<td>Return</td>
<td>Move down to beginning of next line</td>
</tr>
</tbody>
</table>

Step 9. Use Editing Commands
There are many editing commands that can be used to modify existing text in a file. These include commands for deleting and changing text. The majority of these commands are entered while in command mode.

a. Open a new document with vedit. What name was given to the file? __________________

b. Insert some text. Add five or more lines of text and press enter at the end of each line. The student should make some mistakes while typing.

c. Delete some of the text. While in command mode, position the cursor to the desired location and use the options shown in table below to delete some of the mistakes. Note: These commands are all lower case and are entered without the Enter key.

<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Delete character at the cursor</td>
</tr>
<tr>
<td>dw</td>
<td>Delete word or part of word to right of cursor.</td>
</tr>
<tr>
<td>3dw</td>
<td>Delete three words</td>
</tr>
<tr>
<td>dd</td>
<td>Delete line containing the cursor</td>
</tr>
<tr>
<td>3dd</td>
<td>Delete three lines</td>
</tr>
</tbody>
</table>
d. Undo and change some text. To change text and undo changes, use the commands shown in the table below. **Note:** Many of these commands change the user to Insert mode until Escape is pressed.

**Note:** These commands are all lower case.

<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>cw</td>
<td>Change word, or part of word, at the cursor location to the end of the word.</td>
</tr>
<tr>
<td>3cw</td>
<td>Change three words</td>
</tr>
<tr>
<td>r</td>
<td>Replace character at cursor with one other character</td>
</tr>
<tr>
<td>u</td>
<td>Undo previous command</td>
</tr>
</tbody>
</table>

e. Copy and paste text. To copy and paste text, use the following options

<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>yy</td>
<td>Yank a copy of line and place in the clipboard</td>
</tr>
<tr>
<td>P (upper case)</td>
<td>Put (paste) yanked or deleted line above current line</td>
</tr>
<tr>
<td>p</td>
<td>Put (paste) yanked or deleted line below current line</td>
</tr>
</tbody>
</table>

f. Save the file and quit vi. What command was used? ____________

**Step 10. Customize the Student’s Session**

The vi editor includes options for customizing the student’s edit sessions. These are summarized in the table below. The `set` command is used from last-line mode to control these options. The `set nu` option shows line numbers and is used frequently when writing programs and script files. The line numbers are not part of the file.

a. Open the document that was just created with vedit. Use the commands in the table to customize this vi session.

<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>:set nu</td>
<td>Show line numbers</td>
</tr>
<tr>
<td>:set nonu</td>
<td>Hide line numbers</td>
</tr>
<tr>
<td>:set showmode</td>
<td>Display current mode of operation</td>
</tr>
<tr>
<td>:set noshowmode</td>
<td>Turn off mode display</td>
</tr>
<tr>
<td>:set</td>
<td>Display all vi variables set</td>
</tr>
<tr>
<td>:set all</td>
<td>Display all possible vi variables and their current settings</td>
</tr>
</tbody>
</table>
Step 11. Use Search Commands
The commands in the table below allow the student to perform advanced editing such as finding lines or conducting searches for text. Note that the forward slash (/) and the question mark (?) search options are also last-line commands but they do not require a colon first. The n (next) and N (next previous) commands can be used to find the next occurrence after the /string or ?string commands found what the student was looking for.

<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>G (upper case)</td>
<td>Go to last line of file</td>
</tr>
<tr>
<td>:21</td>
<td>Go to line 21</td>
</tr>
<tr>
<td>/string</td>
<td>Search forward for string</td>
</tr>
<tr>
<td>?string</td>
<td>Search backward for string</td>
</tr>
<tr>
<td>n</td>
<td>Find next occurrence of string</td>
</tr>
</tbody>
</table>

a. Use the commands in the table to practice searching for a particular line or string of text.

b. Quit vi without saving any changes. What command was used? _______________

Step 12. Remove Files and Directories Created in this Lab

Step 13. 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.
Objectives:
Use Emacs to:
• Access the Emacs TUTORIAL
• Find Emacs help
• Visit and edit files
• Save files and edit

Background:
The instructions in this lab are written with GNU Emacs in mind, which is installed on most Linux desktop systems by default. XEmacs is also available in some distributions of Linux. If the student happens to have XEmacs on the student’s computer system, feel free to use it instead of GNU Emacs. The student may want to use both and compare them. All the procedures in the lab should work identically in both versions of Emacs.

For the purposes of this lab, the term Emacs means both versions of the editing system, unless specifically cited as GNU Emacs or XEmacs.

Tools / Preparation:
To perform this lab the student will need:
• A UNIX computer with Emacs installed.
• A login ID and password.

Notes:
**Step 1. Start Emacs.**

The executable command that starts GNU Emacs is called `emacs`. The command to start XEmacs is `xemacs`. In the exercises that follow, wherever it says `emacs`, shown in lower case, `xemacs` will also work, if the student has it on the system.

There are several ways to start Emacs.

- Login and open a terminal window.
- To start Emacs, run this command in a terminal window: `emacs -q &`

The `-q` option causes Emacs to start without startup customizations. The student should do it this way for the lab to be sure to get the default startup. The ampersand (`&`) causes the command to run in background.

Another way to start Emacs:

    emacs &

The normal way to start Emacs:
Emacs loads customizations unless the user tells it not to. The potential for customization in Emacs is vast. Virtually no two users modify it the same way.

    emacs -nw

The `-nw` means *no windows*, this means to start Emacs running in a terminal window rather than with its graphical interface. The student may wish to try starting that way and compare the operation. A few users prefer the character interface and use it exclusively.

If the student is running GNOME, Emacs is available as a menu item. Press the GNOME foot, then Programs, Applications, and Emacs.

Read the splash screen. The student will see this if the student has no initialization file or if the student started with the `-q` option. Emacs assumes the user may be new and in need of help.

   a. What information does the splash screen show?

   b. If the student also tried starting with `-nw`, was the splash screen the same?

**NOTE:** If the student waits long enough, the splash screen disappears, but the buffer named `*scratch*`, including the asterisks, remains.

**Step 2. Bring up the Emacs TUTORIAL.**

Graphical Emacs may be run using the menus and mouse in much the same way as any other GUI editor. The purpose of this lab is to familiarize the student with the Emacs way of doing things, which, once learned, is much faster. We suggest the student save exploration of the menus for another time.

To start the TUTORIAL, execute the key combination Control+h t.
Step 3. Work through the first part of the TUTORIAL.

There is no better way to learn about Emacs than to start with the Emacs TUTORIAL, which has been standardized for many years. It is designed for the user to read and do exactly as it says as the user progresses, using the TUTORIAL text itself as a scratch file to play with. By the time the student gets to the end the student will know all the essentials of Emacs, enough to be comfortable using it to accomplish real work.

For now, go through the first four sections of the TUTORIAL. Begin with the section at the beginning on notation, followed by the sections titled:

- SUMMARY
- BASIC CURSOR CONTROL
- WHEN EMACS IS HUNG

NOTE: The XEmacs TUTORIAL inserts a section “Cursor Control with an X Terminal” before “When XEmacs is hung”.

This TUTORIAL should take the student about 15 minutes.

a. What key allows the student to back out of a command when Emacs seems to be tangled up?

Step 4. Learn to get help.

Help is available at every keystroke in Emacs. The amount of information available is vast, because Emacs was designed to be self-documenting. The student should never need to buy a book learn about.

- To see all general categories of help available, type \texttt{c-\textasciitilde h \textasciitilde ?}. If the student is using XEmacs, type \texttt{c-\textasciitilde h \textasciitilde \textasciitilde ?}.

  a. How would the student produce a list of all key bindings? ____________
  
  b. What key is bound to the command \texttt{M-x isearch-forward}? ____________
  
  c. What key is bound to \texttt{M-x kill-line}? ____________
d. How would the student find out what command the key `C-x C-f` runs? What command is it?

```
```

e. How would the student find information about commands whose names include the string 'buffer'? Are there any such commands?

```
```

f. How would the student find documentation about the command `M-x find-file`? Hint: commands and functions are synonymous in Emacs. What does it do?

```
```

Step 4. Edit files.

- Type `C-x C-f` and respond `junkfile` to the prompt. Note: Always press Enter following prompts. Look at the modeline, this is the black band in reverse colors one line above the bottom.

  a. What happened?

  b. Did the file `junkfile` have to exist before the student visited it? ____________

  c. Watch the modeline and type one letter. What happens in the modeline?

Type in a few words or characters and move the cursor around using Emacs commands learned in the TUTORIAL. Note: Do not use the mouse or menus.

  d. What is the primary difference between vi and Emacs in going from typing text to moving the cursor or executing commands?

  e. Type `C-x C-f` and respond `/etc/passwd` to the prompt. Did Emacs warn the student that the `junkfile` has not been saved junkfile before going to another file?

  f. How is the modeline different from what the student saw before?

  g. Try to type some characters. Could this be done?

  h. What does the key `C-x k` do? Try using it.

  i. Type `C-h k C-x k` to see complete documentation on `M-x kill-buffer`.

  j. Type `C-x 1` to return to a single window.

Step 5. Learn to kill file buffers without saving, and to save files.

- First type `C-x C-f /etc/passwd` to visit the password file again.
• Open a second new file for edit. Type `C-x C-f ~/xyz` The tilde (~) character is a shortcut for the student's home directory. Type a few letters in the new file's buffer.

• Type `C-x C-b (M-x list-buffers)` to see a list of open buffers in another window.
  
a. Are there the same number of buffers as open files?

b. Notice the column MR. What does this column indicate about the state of the file buffers for password, xyz, and `junkfile`? Type `C-x 1` to return to a single window.

c. Type `C-x k (M-x kill-buffer)` to attempt to delete or kill the current file buffer. Enter to accept the default. What happens?

d. Type `y` and press Enter. Did Emacs let the student kill it?

e. Type `y` again, and this time type `yes`. Did it work?

f. The student should now be back to the password file. Kill the buffer to return to `junkfile`. Watch the modeline and type `C-x C-s (M-x save-buffer)`. How does the appearance of the modeline change?

g. What message appears in the `status` line or the bottom line?


a. First, type a few more characters into `junkfile` to return its status to modified. Type `C-h k C-x C-c` to learn about `M-x save-buffers-kill-emacs`. How does this function allow quitting Emacs easily?

b. Type `C-x C-c` to begin exiting from Emacs. Did this prompt the student regarding any unsaved buffers?

c. Restart Emacs with the command: `emacs -q junkfile` Notice that Emacs may be started with one or more file names as arguments.

d. Type `C-x C-c`. Did Emacs prompt for verification?
Fundamentals of UNIX

Lab 9.3.6 – Using the CDE Text Editor

(Estimated time: 30 min.)

Objectives:
- Access the CDE Text Editor
- Open a new file and enter some text
- Save the file
- Use Help with File menu options
- Use the Edit Menu
- Use the Sun Workstation Editing Keys (Optional)
- Use Find / Change
- Select and replace text
- Use the Format menu to change options
- Use the Text Editor Options menu

Background:
In this lab, the student will work with the Common Desktop Environment (CDE) Text Editor. Text Editor is a full-screen graphical text editor that supports a mouse and can be used to edit files. Text Editor is similar to the Windows Notepad. As with vi, this editor does not put any special formatting characters into the file and is suitable for creating system environment and script files.

Tools / Preparation:
- Before starting this lab, the student should review Chapter 9, Section 2 – Using the CDE Editor
- The student will need the following:
  1. A login user ID, for example user2 and a password assigned by the instructor.
  2. A computer running the UNIX operating system with CDE.
  3. Networked computers in classroom with the class file system installed.

Notes:
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
The student should login with the user name and password assigned by the instructor in the CDE entry box.

Step 2. Access The CDE Text Editor
To start the CDE Text Editor, click on the Text Note sub panel from the Front Panel and open a Text Editor window.

  a. Click on the Text Note menu sub panel above the Text Note icon on the Front Panel to start the CDE Text Editor. What is the default name for the file that opens? _________________

Step 3. Enter Some Text
The student can type any text into the Text Editor window. If the student wants to insert characters into an area where the student has already typed, position the pointer where text is to be inserted, then click the left mouse button. By default, Text Editor is in Insert mode. That is, when a user types, the characters will be automatically inserted into the text rather than overwriting any existing characters to the right of the pointer. The student can change from insert to overwrite mode at any time while working with Text Editor by clicking on the Options menu and selecting Overstrike.

  a. Enter a small paragraph of text and press Enter at the end of each line. What happens if the student does not press enter at the end of each line? _________________

Step 4. Save the File

  a. Click on the File menu and select Save from the menu. Name the file mycdefile. What is the default directory where this file was saved? _________________

  b. Click on the File menu and select Close to exit the CDE Editor.

  c. Open a Terminal window and list the contents of the student’s home folder. What is the size of the file the student just created? _________________
Step 5. Use Text Editor Help with File Menu Options
a. Click on the File menu and review the options available. List the options here.

b. Click on the Help menu, select Reference and then Text Editor File Menu to see what each menu item does. What does Help say about the Include option?

Step 6. Use the Edit Menu
The Edit menu contains the standard options to manipulate text, such as Cut, Copy, and Paste. However, if no text is selected, the cut and copy options will not be available. Many of the Edit menu functions are available by selecting the text the user wants to work with and right clicking with the mouse. Select Edit from the menu displayed, and the student will be able to cut, copy, and paste.

a. Open mycdefile, which the student created earlier. Select some text with the mouse, click on the Edit menu, and select Copy.

b. Move the mouse to another area in the text and click to select a place to insert what was copied. Click on the Edit menu and select Paste. Was the text the student copied pasted into the new location?

c. Select some more text with the mouse, click on the Edit menu and select Cut. If the student has accidentally removed this text and wants to put it back, what Edit menu option would the student use?

d. Select some text with the mouse, right click, and select Edit from the menu. Click on Copy and then reposition the cursor. Right click, select Edit, and then Paste.

e. Click on the Edit menu again. Select the Check Spelling option and check the document for spelling errors.

f. Close the Text editor. Was the student prompted to save the changes?

Step 7. This part is Optional. Use the Sun Workstation Editing Keys
The Sun workstation keyboard has a set of editing keys on the left side that can be used instead of the editing menu options. By selecting text with the mouse, these keys can be used to copy, cut, and paste, among other things.

a. If the student has access to a Sun Workstation, practice the cut, copy, and paste actions that were previously performed using the keyboard editing keys.

Step 8. Use Find / Change
The Find option enables the student to find text or to find and change text. The Find option is case sensitive, so anything the user is searching for will only be found if it is an exact match. Similarly, any text can be changed to be exactly the same as the text the user typed in the change box.

a. Open mycdefile, which was created earlier. Select Find/Change from the Edit menu. Enter a word or string that is in the text and use the find function to search for it.

b. Enter a word or string to change what was just found. Click on Change. Did the old text change to the new text?
Step 9. Select and Replace Text
Text can be selected using any of the following four methods:

1. Dragging over the text while pressing the left mouse button.
2. Double-clicking with the left mouse button on a word to highlight that word.
3. Triple-clicking with the left mouse button to highlight the paragraph
4. Quadruple-clicking with the left mouse button to highlight all of the text in file.

Once text has been highlighted using any of these four methods, it will be replaced by whatever characters are next typed at the keyboard.

a. Practice selecting text using all four methods listed above. Then replace the text that was selected by typing new text. The student can also use the Edit menu and the right click mouse option after text has been selected.

Step 10. Use the Format Menu to Change Options
The Settings option under the Format menu will enable the student to change margins and alignment of the text, either for a paragraph or for the entire document. Once settings have been defined, clicking again on the Format menu gives a shortcut to changing settings for the current paragraph (Paragraph) or the entire document (All).

a. Click on the Format menu and select Settings. Change the right margin to 60 and Select the Justify button. Click All to apply the settings to the entire document. Click Close. Did the settings take effect? 

Step 11. Use the Text Editor Options Menu
The Options menu enables the student to switch between insert and overstrike modes. The chosen mode will affect any new text typed into an existing text area. The wrap-to-fit mode formats the text in the current Text Editor window by automatically wrapping or moving text to the next line when it reaches the edge of the window.

a. Practice switching between insert and overstrike modes.

b. Turn on the wrap-to-fit option and type and type two or 3 additional sentences.

c. Click on the Help menu and select Reference from the menu. Click on Text Editor Options Menu and review what the Backup on Save option does. What does the Text editor name the file that is backed up? 

Step 12. Remove Files and Directories Created in this Lab

Step 13. Logout
Exit the CDE Text Editor by clicking the File menu and click Close. Close any terminal windows. Click the EXIT icon on the front panel.
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, the student 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. The student can determine the file type, permissions, owner, and group with the `ls -l` command. The student 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, the student should review Chapter 10, Section 2 – File System Permissions
b) The student will need the following:
   1. A login user ID, for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system.
   3. Networked computers in classroom with class file system installed.

Notes:

------------------------------------------------------------------------

------------------------------------------------------------------------

------------------------------------------------------------------------

------------------------------------------------------------------------

------------------------------------------------------------------------

------------------------------------------------------------------------

------------------------------------------------------------------------

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

### Class File Tree Structure

```
/  
|   
|__home
|   |__user2
  
|__dir1
|   |__trees
|   |___fruits
|   |___beans
|   |___coffees
|__dir2
|   |__dir3
|   |   |__dir4
|   |   |__planets
|   |   |__flowers
|   |___notes
|   |___mars
|   |___pluto
|__dir3
|___practice
|   |__dir4
|   |__dante
|___file1
|___file2
|___file3
|___file4
|___fruit2
```

### Step 1. Log in to CDE
The student should login with the user name and password assigned by the 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 will be displayed. Hidden files and directories are those that begin with a dot.

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.

<table>
<thead>
<tr>
<th>File Type</th>
<th>A dash (–) in the first position indicates a regular file. A d indicates directory.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permissions</td>
<td>3 Sets of permissions for User, Group, and Others.</td>
</tr>
<tr>
<td>Links</td>
<td>Links to other files and directories.</td>
</tr>
<tr>
<td>Owner</td>
<td>User (login) ID of user who created the file or directory unless ownership was assigned.</td>
</tr>
<tr>
<td>Group</td>
<td>Group name that owner belongs to as established by the system administrator.</td>
</tr>
<tr>
<td>Size</td>
<td>File size in Bytes</td>
</tr>
<tr>
<td>Modification Date / Time</td>
<td>Month, Day, Year, and Time the file was created or last modified.</td>
</tr>
<tr>
<td>File Name</td>
<td>File or Directory Name</td>
</tr>
</tbody>
</table>
Step 4. Interpret Permissions

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

<table>
<thead>
<tr>
<th>Permission</th>
<th>Symbol</th>
<th>Plain File</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>r</td>
<td>File can be displayed or copied. A copied file takes on new owner.</td>
<td>Contents can be listed with the <code>ls</code> command. Must also have execute permission to use <code>ls</code> command options.</td>
</tr>
<tr>
<td>Write</td>
<td>w</td>
<td>File can be modified, moved, and removed, but only if the directory the file resides in has write permission.</td>
<td>Files can be added or deleted. Directory must also have execute permission.</td>
</tr>
<tr>
<td>Execute</td>
<td>x</td>
<td>File can be executed with either shell script or executables.</td>
<td>Controls access to the directory. A user can <code>cd</code> to the directory and list the contents. Files can be moved or copied to the directory.</td>
</tr>
<tr>
<td>No Permission</td>
<td>-</td>
<td>A dash (-) indicates permission is denied.</td>
<td>A dash (-) indicates permission is denied.</td>
</tr>
</tbody>
</table>

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 permission? ________________

Step 5. Determine User Permissions for a File

The nine permissions are divided into 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 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 or 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?

Step 7. Determine Other (Public) Permissions for a File
The last set of characters, called others permissions, are the permissions everyone else has. Others refer 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 the student’s home directory display the long directory listing for the pwd command file in the /usr/bin directory. What command is seen? 

b. What are the permissions for the file? 

c. What are the User permissions? 

d. What are the Group permissions? 

e. What is 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 the student’s practice directory.

a. From the student’s home directory create a new file called newfileperms in the practice directory. What command and pathname was used?
b. From the student’s home directory list the contents of the practice directory to see the permissions of newfileperms. What command and pathname were used? __________________

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 the practice directory.

a. From the student’s home directory create a new directory called newdirperms in the practice directory. What command and pathname were used? __________________

b. From the student’s home directory list the contents of the practice directory to see the permissions of newdirperms. What command and pathname were used? __________________

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 11. Remove Files and Directories Created in this Lab**
Remove all files and directories created in the student’s home directory during this lab.

**Step 12. 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.
Fundamentals of UNIX

Lab 10.3.3 – Changing Permissions from the Command Line
(Estimated time: 45 min.)

Objectives:

- Work with file system to control security access
- Review `chmod` command modes
- Change file permissions using symbolic mode
- Change directory permissions using symbolic mode
- Determine octal mode permissions
- Change file permissions using octal mode
- Change directory permissions using octal mode
- Create a script file using the vi editor and make it executable

Background:
In this lab, the student will analyze and change UNIX file system security permissions using command line utilities. File and directory permissions can be changed using the `chmod` (change mode) command. Normally the default permissions for a file or directory will be adequate for most security needs. There are times when the student will want to change the permissions on a file or directory. By default, all files are created with permissions that allow the user category of others to read the file. This means anyone with a login id can see the contents of the file and copy it. For classified files and private information, the user can modify the permission of the file to prevent others from accessing it.

Shell scripts are another example where the student would want to change permissions. When a user creates a shell script file, or any file for that matter, the default permissions do not include execute. This is the case even for the owner/creator of the file. To run the shell script, the user must change the permissions by adding the execute permission for the user or owner category.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 10, Section 3 – Changing Permissions from the Command Line
b) The student will need the following:
   1. A login user ID, for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system.
   3. Networked computers in classroom with class file system installed.

Notes:
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
The student should login with the user name and password assigned by the 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 chmod Command Modes
The `chmod` (change mode) command is used by a file's owner, or superuser, to change file permissions. The two modes of operation with the `chmod` command are symbolic, or relative, and octal, or absolute. The general format of the `chmod` command is shown below. The mode portion will change depending on whether the user is using symbolic or octal mode.

Command format: `chmod mode filename`

**Symbolic mode** uses combinations of letters and symbols to add or remove permissions from various categories of users. Symbolic mode is also referred to as relative mode.

**Octal mode** uses numbers to represent file permissions. Octal mode is also referred to as absolute or numeric mode.

a. Which `chmod` mode uses numbers to represent file permissions? ______________

b. Which `chmod` mode uses letters or symbols to represent permissions? ______________

c. What is another term for octal mode? ______________

d. What is another term for symbolic mode? ______________
Step 4. Change File Permissions Using Symbolic Mode

When using symbolic mode to set permission, the user typically works with one category of users, although the user can give all categories the same permissions simultaneously. The mode is referred to as relative since the user is assigning or removing permissions relative to the ones that are already there. The user can add one or more permissions to a specific category of users or take them away. The command format for symbolic mode uses letters and symbols.

The mode portion of the command format is made up of three parts:

- **Who** – is the category of users the student is working with shown as u = user, g = group, o = other or a = all.
- **Op** – is the Operator or what the student is going to do as in set (=), remove (-), or give (+).
- **Permissions** – is the Permission or Permissions to be assigned for the users as either r = read, w = write, or x = execute.

The following example removes (-) the read permission (r) from the file dante for the other (o) category of users. **Note**: There should be no spaces between the o, dash (-), and r.

```
chmod o - r dante
```

The next example gives (+) the write permission (w) to the file dante for the group (g) and other (o) categories of users.

```
chmod g o + w dante
```

a. From the student’s home directory, create a new directory under the practice directory called chmoddir using a relative pathname. What command was used to create the directory?

b. Change to the chmoddir directory and create a new file called symfile. What command was used to create the file?

c. Use the `ls -l` command to determine the permissions for the new symfile file. These are the default permission for a file. What are the permissions for User, Group, and Other?

d. The student decides other users, other than the student and members of the student’s group, are not to be able to see the contents of symfile or copy it. Use the `chmod` command, in symbolic mode, to remove the r (read) permission for other users for the file symfile. What command was used?

e. List the permission of the file again. What is the permission for the others user category now?

f. What command would the student use if the student wanted to remove the read permission for both the group and others with a single command?

Step 5. Change Directory Permissions Using Symbolic Mode

a. Change back to the practice directory. What command was used?
b. From the practice directory, list the permissions for the new chmoddir directory that was created earlier. These are the default permissions for a directory. What are the permissions for User, Group, and Other?

c. Can users other than the student or members of student’s group copy files from the student’s chmoddir directory? Why or why not?

d. The student does not want other users to be able to copy files from the chmoddir directory.
   Change to the practice directory and use the `chmod` command in symbolic mode to remove the read permission and the execute permission for the others category of users from the directory chmoddir. What command was used?

e. List the permissions of the directory again. What are the permission for the others user category now?

f. Can the members of the student’s primary group or staff create new files in or copy files into the chmoddir directory? Why or why not?

g. Change to the practice directory and use the `chmod` command in symbolic mode to add the write permission for the student’s primary group for the directory chmoddir. What command was used?

h. Change the permission back to the default permissions using symbolic mode. What command or commands were used? Note: groups and permissions can be combined with one command or the student can use two separate commands.

**Step 6. Determine Octal Mode Permissions**

Octal mode provides a quick numeric means of changing permissions for all categories of users simultaneously while still allowing each set of permissions to be different. There are three possible permissions for each set, r, w, and x. There are three possible permissions for each type of user category as user, group, or other. Each set of permissions can be assigned a numeric value, from 0 to 7, depending on which permissions are allowed.

The r (read) permission is assigned a value of 4, the w (write) permission a value of 2, and the x (execute) permission a value of 1. By adding up the numbers we can get a total of all three permissions for that category of user either User, Group, or Other. For instance if the Owner permission for a file is r w x, we add 4 (read) + 2 (write) + 1 (execute) which equals 7. If the group had r w – permissions, they would have 4 + 2 + 0 (no execute) for a total of 6. If other had only r they would have 4 + 0 + 0 (no write or execute) for a total of 4. The octal_mode for this file or directory is 764.

<table>
<thead>
<tr>
<th>User Permissions</th>
<th>Octal Sum</th>
<th>Group Permissions</th>
<th>Octal Sum</th>
<th>Other Permissions</th>
<th>Octal Sum</th>
<th>Octal Mode Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>r w x</td>
<td>4+2+1</td>
<td>r w -</td>
<td>r w -</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r w -</td>
<td>4+2+0</td>
<td>r w -</td>
<td>r - -</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r - -</td>
<td>4+0+0</td>
<td>r - -</td>
<td>r - -</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r w x</td>
<td>4+2+1</td>
<td>r - x</td>
<td>r - x</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a. Fill in the following table by converting the character permissions (r,w,x, -) to their octal equivalents. Convert each set of permissions first for User, Group, or Other. Then enter the octal_mode, a three digit number, under Octal Mode permissions.
Step 7. Change File Permissions Using Octal Mode
With octal mode, it is not necessary to specify the category of users since the position of each number represents one of the three user categories. The octal_mode is made up of three numbers, each of which is the sum for one of the user categories for User, Group, and Other. Octal values are combined to identify the octal_mode that is used with the chmod command.

**Command Format:** `chmod octal_mode filename`

a. Change to the chmoddir directory and create a new file called octfile. What command was used to create the file? ________________________________

b. Use the `ls -l` command to determine the permissions for the new octfile file. These are the default permission for a file. What are the alphanumeric permissions for User, Group, and Other? ________________________________

c. What is the octal mode equivalent of the user, group, and other permission for this file? ________________________________

d. The student decides that other users are not to be able to see the contents of or copy octfile. Use the chmod command in octal mode to remove the r (read) permission for other users for the file octfile. What command was used? ________________________________

e. List the permission of the file again. What are the permission for the others user category now? ________________________________

f. What command would the student use if the student wanted to remove all permissions for both the group and others with a single command? ________________________________

Step 8. Change Directory Permissions Using Octal Mode
The format below is used to change the permissions on a directory. The –R (recursive) option changes the permissions on the specified directory and on all subdirectories and files within it.

**Command Format:** `chmod [-R] octal_mode directoryname`

a. Change to the practice directory. What command was used? ________________________________

b. From the practice directory, list the permissions for the chmoddir directory. These are the default permissions for a directory. What is the alphanumeric permission for User, Group, and Other? ________________________________

c. What is the octal mode equivalent of the user, group, and other permission for this directory? ________________________________

d. Use the chmod command in octal mode to remove the read and the execute permission for the others category of users from the directory chmoddir. What command was used? Remember, the user must always specify all three sets of permissions with octal mode even if the permissions are not to be changed. ________________________________

e. List the permissions of the directory again. What are the permissions for the others user category now? __________ Did the permissions remain the same for the user and group? __________

f. Can the members of the student’s primary group create new files in or copy files into the student’s chmoddir directory? __________ Why or why not? __________
g. The student decides that members of the student's group are to be able to copy files to the student's directory. Change to the practice directory and use the `chmod` command in octal mode to add the write permission for the student's primary group for the directory `chmoddir`. The user should have rwx, the group should have rw, and other should have no permissions to the directory. What command was used?

h. Change the permissions back to the default permissions (rw-r-xr-x) using octal mode.

**Step 9. Create a Script File and Make it Executable**

In this step, the student will create a simple text script file using the vi editor. The student will then need to make it executable in order to run or execute the script file. Script files can be very useful to help automate repetitive tasks.

a. Change to the `chmoddir` directory and start the vi editor. With Solaris use `vedit` and with Linux run `vi`. As the student starts the editor, specify or open a new file called `myscript`. Press i to go into Insert Entry mode and type the following commands as lower case text. Press Enter after each one.

```
clear
pwd
ls -l
banner "my script"
```

b. Press Esc to return to command mode and then type a colon to get to last-line mode. Press wq to write or save the file and quit vi.

c. List the file to determine its permissions. What are they?

```
--------------
```

d. Type `myscript` as though it were a command and press Enter. What was the response?

```
--------------Why did it not execute?
```

e. Change the permissions for the `myscript` file so that the user permissions include x (execute) so that the student as the owner can execute or run the file. The student can use either symbolic or octal mode. What command was used to change the permissions?

```
--------------
```

f. List the file to verify that the permissions changed. What are the permissions for the user (owner) now?

```
--------------
```

g. Type `myscript` as a command again and press Enter. What was the response?

```
--------------
```

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

Remove all files and directories created in the student's 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.
Lab 10.3.4 Implement the Model for the ATMGUId class

Estimated Time: 10 minutes

Learning Objective:
- In this lab activity, the student will use adapter classes for implementing listener interfaces and event handling.

Description / Scenario:
- Create an Adapter class and an Interface class, and use them with the ATMGUId.
- Using the Interface class will allow the implementation of the Model View Controller pattern for the ATMGUId.
- With the Adapter class the programmer need only implement the method(s) that are used.
- What are Adapter classes and how are they used?
The Java platform provides adapter classes. These classes are abstract classes that implement the Window Listener interface methods. The programmer can create a handler class that extends from one of these adapter classes. The adapter classes have the name of the Listener interface and the word ‘Adapter’. Since only one class can be extended from, in some cases the student will not be able to extend from the adapter class. The Applet is such a case.
- The application of Interface classes.
Java does not allow for multiple inheritances. So a class could not extend from two classes. Interfaces are abstract classes that define abstract methods and constants. Interfaces provide a mechanism for a subclass to define behaviors from sources other than the direct and indirect super classes.

File Management:
Retrieve the previous lab “Lab10.3.2.2”. This lab will be used here and in the remainder of the chapter 10 labs. Continue saving the work in the same chap10 folder. Use Save As and title it “Lab10.3.4”. This file will be used again in the next lab.

Tasks:

Step 1 Modify ATMWindowHandler
- Create an Adapter class by modifying the ATMWindowHandler class to extend WindowAdapter.
- The additional methods in the ATMWindowHandler class will no longer be needed. Only the windowClosing Event is implemented.

Step 2 Create a Controller class
- Create the ATMBUTTONHandler which implements the ActionListener. Copy the code used in the ATMController for the actionPerformeEd events. In the ATMBUTTONHandler class, add an attribute that will hold a reference to an object of the ATMGUId and an attribute that will hold reference to an object of the Teller. Add a constructor to this class that accepts a reference to an object of the type ATMGUId. Assign this reference to the attribute. In the ATMGUId class define a method getTeller() which returns teller. In the ATMGUId class register the Buttons with the ATMBUTTONHandler reference. In the ATMBUTTONHandler Constructor call the getTeller() method of the ATMGUId class and assign the teller object to the teller attribute.
- In the Teller class remove the references to ATMController which exists when imported from previous lab. Add the window Listener to the frame as shown below:

```java
frame.addWindowListener(new ATMWindowHandler());
```
c. In the main method of the Teller class remove the code which was imported from previous lab and instantiate a Teller class. Compile and run the Teller class.
Step 3 Review Questions:
   a. What is a reason to use the Adapter class?

   b. How is a marker interface different from an interface class?

   c. What is the function of an interface class?

Step 4 Documentation:
   a. Using the Document “How to use UMLTestTool”, follow the instructions to verify that the JBANK classes match the JBANK UML diagram shown below.
   b. Write all needed javadoc comments and document elements for the lab. Then, using BlueJ, select Tools and create the javadocs by selecting Project Documentation.
Fundamentals of UNIX

Lab 10.4.1 – Changing Permissions with File Manager
(Estimated time: 30 min.)

Objectives:

- Access File Manager to work with file and directory permissions
- Determine file permissions with File Manager
- Change file permissions with File Manager
- Determine folder permissions with File Manager
- Change folder permissions with File Manager

Background:
In this lab the student will work with the Common Desktop Environment (CDE) File Manager to analyze and make changes to file system permissions. The CDE File Manager utility provides a graphical interface to the file system and can be used to view or change file and folder permissions.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 10, Section 4 – Changing Permissions with File Manager
b) The student will need the following:
   1. A login user, for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system with CDE.
   3. Networked computers in classroom with the class file system installed.

Notes:
Use the diagram of the sample Class File System folder tree to assist with this lab.

Class File Tree Structure

Step 1. Log in to CDE
The student should login with the user name and password assigned by the instructor in the CDE entry box.

Step 2. Access File Manager
File Manager can be accessed by clicking the File Drawer icon on the Front Panel or by right clicking on the workspace desktop and then on the Files menu. The File Manager, by default, opens a view of a folder that is the student’s home folder. The term folder is used interchangeably with the terms folder and subfolder. From that folder, the student can change to other folders, both up and down in the hierarchy, to view each folder’s contents. The path to the current folder is always displayed in the upper area of the File Manager window.

a. Click on the File Drawer icon on the Front Panel to start File manager. What is the folder path displayed in the upper area of the panel? ________________

Step 3. Check and Change File Permissions with File Manager
The student can view and change the Properties for a file or folder by selecting it from the File Manager window and then choosing Properties from the Selected menu. An alternate method is to right click the file or folder and choose Properties from the menu displayed. After choosing Properties, a Properties window appears. Clicking on the Information category button, allows a user to view information about the file, such as file size and last modification date. Clicking on the Permissions button, allows viewing and changing of permissions. To change permissions for Owner (User), Group, or Other, select the desired permission by clicking on the checkbox.

a. Click on the dante file and click on the Selected menu and then the Properties option. Click on the Permissions button. What are the permissions of the dante file?

b. Change to the practice folder and create a new file called fmfile using the File menu. Right click on this file and select properties from the menu. Click on the permission button. What are the permissions for this file? ________________
c. Click on the checkbox for read access by other users to deselect it by removing the check mark. Will users other than the student or members of the student’s group be able to see the contents of this file now? Close the properties window for the file.

d. Using File Manager, navigate up thru the file system structure to the /usr/bin folder. Locate the cat command file. What is the symbol on the file icon?

e. Click on the cat file and click on the Selected menu and then the Properties option and right click the file icon and select properties. Click on the permissions button at the top of the window. What are the permissions of the cat file?

f. Now that the student has seen the permission for this file, what is the meaning of the lightning bolt?

g. Click on the checkbox for read access by other users to deselect it by removing the check mark. Could this be done? Why or why not?

h. Close the properties window for the cat file.

Step 4. Check and Change Folder Permissions with File Manager

a. Using File Manager, navigate to the student’s home folder and click on the practice folder to select it. Click on the Selected menu and then the Properties option.

b. Click on the permissions button. What are the permissions of the practice folder?

c. Change to the practice folder and create a new folder called fmfolder. Right click on this folder and select properties from the menu. Click on the Permissions button. What are the permissions for this folder?

d. Click on the checkbox for read access and execute access by other users to deselect it by removing the check mark. Will users other than the student or members of the student’s group be able to see the contents of this folder or copy files from it now?

e. Click on the checkbox for write access for the student’s group to select it by adding the check mark. Will members of the student’s group be able to change files in the student’s fmfolder?

f. Note the “Apply changes to” option. What is the purpose of this option and what are the choices?

g. Close the properties window for the fmfolder.

h. Using File Manager, navigate up thru the file system structure to the /usr/bin folder. What is the symbol on this folder icon? What does this mean?

Step 5. Remove Files and Directories Created in this Lab

Remove all files and directories created in the student’s home directory during this lab.
**Step 6. Logout**
Exit the CDE File Manager by clicking the File menu and click Close. Close any terminal windows. Click the EXIT icon on the front panel.
Objectives:

- Determine who is logged on
- Switch to a different user’s account
- Determine the student’s real user identity
- Determine our effective user identity

Background:
In this lab, the student will use advanced UNIX commands to determine the student’s identity and the identity of other users logged on to a system. When the student logs on to a UNIX system, the student will use a Real User ID, for example user4. It is possible to switch temporarily to another users account for testing or access to files. When the student has switched to the other users account it becomes the student’s Effective User ID. In this lab, the student will work with commands that allow the student to determine the Real User ID, the Effective User ID and be able to switch between the two. The student will also be able to find out who is logged on remotely to a UNIX system.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 8, Section 3 – Identifying Users
b) The student will need the following:
   1. A login user ID, for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system with CDE.
   3. Networked computers in classroom with class file system installed.

Notes:
Step 1. Log in to CDE
The student should login with the user name and password assigned by the 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. Determine Who is Logged on
The who command displays information about all users currently logged on the local system. This command lists the user's name, terminal, login time, elapsed time since the last activity on the terminal line, and the machine (host) name the user logged on from. A UNIX workstation will typically have only one user logged in, the primary user, but a UNIX server can have many users logged in simultaneously.

If the student is using CDE at the workstation with a terminal window the student’s user ID may appear multiple times, one for the console, one for the CDE session, and once for each terminal window the student has open.

a. Use the who command to determine who is currently logged on to the student’s system. Normally the student’s user ID will be the only one listed. The student may have other users listed if those other users have an account on the student’s system and are currently logged in remotely. What was the result of the command?

b. Use the who -H (headings) command to see who is currently logged on to the student’s system with headings displayed. What are the headings displayed?

c. Use the who -q (quantity) command to see the user IDs of those logged on and a count of users. Note: the student’s user ID may be counted twice if the student is the only user logged in. What was the result of the command?
d. **Note:** This portion is optional. If there is a central UNIX server in the classroom with student accounts, obtain the IP address of the server from the instructor and `telnet` (or `rlogin`) to the server. The student will need to provide the login ID and a password to login remotely. Issue the `who` command on the remote server. Who else is logged in?

**Command format:** `telnet ip-address`

The ip address is the 32-bit address of the server.

e. **Note:** This portion is optional. If the student has other user accounts defined on the workstation, have one of the student’s lab partners `telnet` to the IP address of that workstation and login remotely. Check with the instructor if the student does not know the student’s IP address. Use the `who` command to see the user IDs of those logged on. Who else is logged in?

---

**Step 4. Switch to a Different User’s Account**

The student can temporarily switch to another user account to access files and directories that belong to that user by using the `su` (switch user) command. When switched to another user’s account, the student will have access to all of the same files that the other users have. To switch user IDs, the student must supply the password of the user ID the student is switching to. This is not necessary if the student is currently logged in as root. When the student switched to another user account, the student became that user and has all access and privileges that the other user has. To switch back to the student’s previous user ID, type `exit`.

The format of the `su` command is shown below. If the optional dash (-) is used, the student will switch to another UID and have the system read the new user's initialization files. This is as if the student logged out of the student's own account and logged in to the new users. This will allow the student to have the same environment, that is the same custom prompt and aliases that the other user has. If the student does not use the dash, the student will have access to the other user's files but the student's own environment will remain.

**Command format:** `su [-] username`

a. Use the `su` command without the dash (-) option and switch to another user account that is setup on the student’s workstation as guest or userZ. What command was used? ___________. What was the prompt? ___________.

b. Enter the command to verify the student’s current directory. What is the student’s current directory? ___________. Switch back to the student’s own user account by typing `exit` at the command line.

c. Try to touch a file named `guestfile` and remove the `file1` file. Why could this not be done? ___________. Switch back to the student’s own user account by typing `exit` at the command line.

d. Use the `su` command with the dash (-) option and switch to another user account that is setup on the student’s workstation as guest or userZ. What command was used? ___________

e. Enter the command to verify the student’s current directory. What is the student’s current directory? ___________. Switch back to the student’s own user account by typing `exit` at the command line.
Step 5. Determine Your Real User ID
The login id that the student used to initially login to a UNIX system is the Real User ID (RUID). The who am i command can be used to help determine the Real User Identity (RUID) when working with different user accounts:

who am i - Displays login ID, Terminal, Date/Time logged on, and Machine name

a. Use the su command and switch to another user account that is setup on the student's workstation as guest or userZ. What command was used? _________________
b. Use the who am i command to determine the Real User ID. The who am i command is the only three word command in UNIX. This should be the student's original login ID. What was the result of the command? _________________
c. Switch back to the student's own user account by typing exit at the command line. Type who am i again. What was the result of the command? _________________

Step 6. Determine The Student's Effective User ID
Switching to another user's ID will give the student the characteristics and permissions of the account the student switched to. This now becomes the student's Effective User ID (EUID). The who am i command shows the Real User ID (RUID). The id command displays the student's Effective User ID (EUID) and the primary group the student is a member of. The id command can also be used with the -a (all) option to show all file access groups the effective user is a member of. File access is covered in Chapter 10.

a. Use the su command and switch to another user account that is setup on the student's workstation as guest or userZ. What command was used? _________________
b. Use the id command to determine the Effective User ID. This should be the user ID the student switched to. What was the result of the command? _________________
c. Use the id -a command to determine the Effective User ID. This should be the user ID the student switched to. It will also show the student the primary group and any other groups this Effective User ID is a member of, if any. What was the result of the command? _________________
d. Switch back to the student's own user account by typing exit at the command line. Type id again. What was the result of the command? _________________

Step 7. 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.
Fundamentals of UNIX

Lab 11.3.2 – Command Line Printing
(Estimated time: 30 min.)

Objectives:

• Develop an understanding of UNIX command line printing
• Review UNIX printing environment components
• Print files with the `lp` command
• Monitor print jobs and queue status
• Cancel print requests

Background:
In this lab the student will work with UNIX printing commands to send jobs to printers and manage print queues. Printing services are an essential component of any network operating system. UNIX provides local and remote printing capabilities. The student will review the major component of the UNIX printing environment and will work with the command line method of printing using the `lp` command. The `lpstat` command is used to determine the status and availability of network printers the student can print to. The student will also monitor the print queue status, determine print request IDs and remove print requests using the `cancel` command.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 11, Section 1 – The UNIX Printing Environment, Section 2 – Command Line Printing and Section 3 Managing Print Queues.

b) The student will need the following:
   1. A login user ID, for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system.
   3. Networked computers in classroom with class file system installed.
   4. Network printer available and print server running.

Notes:
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
The student should login with the user name and password assigned by the 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 Printing Environment Components
With the UNIX printing environment, users can have local printers attached to their workstations and can also print to remote network printers. The system administrator sets up the printing environment by installing printers and defining print queues and servers to support them. There are three main components in the UNIX printing environment:

- **Printer**: The printer is a physical printing device. The printer may be attached to a workstation or a network server or the printer can be attached directly to the network using a hub or switch.

- **Printer Name (queue)**: The printer name is the name of a print queue associated with the physical printer. It is a logical name for the printer, which is assigned by the system administrator. This is the name the users print to. A print queue is a directory on the hard disk of a computer where print requests from users are stored.

- **Print Server**: The print server is the computer that manages incoming print requests and releases them when the printer is ready. Print servers run the printer daemon lpsched, which manages print requests. The print queue is located in the hard disks of the print server. Print requests or print jobs are stored on the hard disk until they are printed and then the requests are deleted or purged. A print server can be a workstation or a network server. The local computer can act as the print server for a local printer. Network print servers are usually centralized and can handle multiple printers and queues.
Use the terms printer, printer name, and print server to fill in the blanks for the following questions.

a. A device that runs the lpsched daemon and holds the print in print queues on its hard disk is a: ______________

b. A ______________ is a physical device that outputs the printed material.

c. The logical name that the user prints to is a ______________

Step 4. Print Files with the lp Command
Printing of ASCII text or PostScript™ files can be done from the command line using the lp (Line Printer) command. Do not use this method to print data files that are created in applications like FrameMaker or Sun’s Star Office, or binary files. The function of the lp command is to queue data for printing. The format of the lp command with options available is shown below.

Command Format:    lp [-options ] [ filename(s) ]

Below are some examples using the lp command with various options to print files. If the student does not specify a printer, lp will print to the student’s default printer. An administrator must set up the default printer in advance. Although these examples print only one file, it is possible to send multiple files simultaneously to the printer and wildcard metacharacters such as * and ? can also be used.

1. Display the name of the student’s default printer.
   $ lpstat -d

2. Print file2 in the student’s home directory on the default printer.
   $ lp ~/file2

3. Use the -d option to specify another printer, if one is available.
   $ lp -d staffp ~/file2

4. Use the -o nobanner option to suppress banner page.
   $ lp –o nobanner ~/file2

5. Use the -n option to specify the number of copies.
   $ lp –n 3 ~/file2

a. Change to the student’s home directory. What command was used? ______________

b. What is the name of the student’s default printer? ______________

c. Send a print request to print the fruit file to the default printer. What command was used? ______________

d. What message was displayed on the student’s screen after submitting the print request?
   ______________

e. Did the fruit file print on the default printer with a banner page showing the student’s user ID?
   ______________

f. Send a print request to print the mars file, which is in the planets directory file to the default printer and suppress the banner page. What command was used? ______________

g. If the student had another printer available called laser5, what command would be used to send a print request for the fruit2 file to this printer? ______________
h. List the files and subdirectories in the student’s home directory. Use the long listing option and redirect, using the > symbol, the output to a file called userXhomedir.lst, where userX is the student’s user ID. What command was used?

i. Send a print request to print the file the student just created to the default printer. What command was used?

j. Print the calendar for the current month to the default printer. What command was used?

Step 5. Monitor Print Jobs and Queue Status
When the student uses the `lp` command to send a print request to a printer, the student is actually sending it to a print queue. The print queue is a special directory that is stored on the hard disk of the student’s workstation or of a remote network server. Since printers do not have hard disks to store documents, all requests or print jobs must be spooled or go to the queue first. If the printer is available, the request is serviced immediately. If the printer is busy, the request is queued until the printer is available. An administrator can monitor and manage the print queues for multiple printers.

The `lpstat` (line printer status) command is used to display the status of the printer queue. To see the print requests for a specific printer, use the basic form of the command, which specifies the printer name or queue to display. If no printer name is specified, the student will see a list of requests on the default printer.

a. If possible, turn off the printer or have the instructor stop the lpsched daemon on the print server temporarily so that the student can see what is in the queue before it is printed.

b. Send individual print requests for the files in the student’s home directory that start with ‘file’ to the default printer. Send the requests in sequence starting with file1, then file2, and so on. What command was used?

c. Use the following table to record the results of using the `lpstat` command with various options listed after having sent some print jobs to the queue.

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
<th>Purpose</th>
<th>Results of Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Printer</td>
<td>Name of printer (queue)</td>
<td>Displays requests for a specific printer’s queue</td>
<td></td>
</tr>
<tr>
<td>-p</td>
<td>Printers</td>
<td>Displays status of all printers</td>
<td></td>
</tr>
<tr>
<td>-o</td>
<td>Output or Outstanding</td>
<td>Displays status of all output or outstanding print requests</td>
<td></td>
</tr>
<tr>
<td>-d</td>
<td>Default</td>
<td>Displays which printer is the system default</td>
<td></td>
</tr>
<tr>
<td>-t</td>
<td>Tell All</td>
<td>Displays complete status information for all printers</td>
<td></td>
</tr>
<tr>
<td>-s</td>
<td>Summary</td>
<td>Displays a status summary for all printers configured on the user’s system</td>
<td></td>
</tr>
<tr>
<td>-a</td>
<td>Accepting</td>
<td>Displays which printers are accepting requests</td>
<td></td>
</tr>
</tbody>
</table>

Step 6. Cancel Print Requests
There are times when the user will want to remove a print request from the print queue. If the print job has not already printed and is still in the print queue the user can cancel it. The user may only cancel those print requests that they have initiated. The user cannot remove another user's print requests. Only the system administrator can cancel print jobs for other users.
The `cancel` command enables the student to cancel print requests previously sent with the `lp` command. To do this, the student must first use the `lpstat` command to identify the request-ID. If the student cancels a print job, it does not affect the request ID numbers of the other jobs still in the queue.

a. Use the `lpstat` command to determine the request-ID of the print jobs in the queue that belong to the student. How can the student tell if these are the student’s print jobs?

b. Note the request-ID for one of the student’s print requests and use the `cancel` command to remove it from the print queue. What command was used?

c. Use the `lpstat` command again to determine the request-ID of the print job the student removed is still there. Is it gone?

d. Use the `lpstat` and `cancel` commands to remove all remaining print jobs belonging to the student from the print queue.

**Step 7. Remove Files and Directories Created in this Lab**
Remove all files and directories created in the student’s home directory during this lab.

**Step 8. 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.
Fundamentals of UNIX

Lab 11.4.3 – Using CDE Print Manager

(Estimated time: 30 min.)

Objectives:

- Access CDE Print Manager
- Check Printer Properties
- Monitor the Printer Queue
- Use File Manager to Print Files
- Cancel a Print Job from the Queue

Background:
In this lab, the student will work with the Common Desktop Environment (CDE) Print Manager and File Manager to control the student’s printing environment. CDE Print Manager is a graphical tool for managing print queues and it performs many of the same functions as the `lpstat` command. The CDE Print Manager utility provides a graphical interface to the printing system and can be used to view or change print queue jobs. File Manager can be used in conjunction with Print Manager to print files and performs similar functions to the `lp` command.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 11, Section 4 – Using CDE Print Manager
b) The student will need the following:
   1. A login user ID, for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system with CDE.
   3. Networked computers in classroom with the class file system installed.
   4. Network printer available and print server running.

Notes:
Use the diagram of the sample Class File System folder tree to assist with this lab.

Class File Tree Structure

Step 1. Log in to CDE
The student should login with the user name and password assigned by the instructor in the CDE entry box provided.

Step 2. Access CDE Print Manager
To activate Print Manager, Click the Printer icon on the Front Panel. The Printer Jobs window displays a list of the current printers. Double clicking the printer icon will show any outstanding print requests in the print queue. Only one printer icon will be displayed in the Printer Jobs window. This represents the printer chosen by the user. This may or may not be the same as the system default printer.

a. Click on the Printer icon on the Front Panel to start Print Manager. What is the title of the window that opens? ________________________________

b. What is the name of the printer shown in the window? ________________

c. What is the status of the print queue as shown in the lower left corner? ________________

Step 3. Check Printer Properties

a. Click the printer to select it. Click on the Selected menu and click Properties. What is the name of the Printer Queue? ________________ What is the Device name? ________________

b. Right click on the printer and select properties from the menu. Does this accomplish the same thing as using the Selected menu? ________________

c. What is the status of the Printer Queue? ________________ What is the status of the Device Name? ________________
Step 4. Monitor the Printer Queue

a. If possible, have the instructor login as root and disable the printer on the print server temporarily so that the student can see what is in the queue before it is printed.

b. Open a Terminal Window and send individual print requests for the files in the student’s home directory that start with file to the default printer. Send the files in sequence starting with file1, then file2, and so on. What command was used? __________________________

c. Double click on the printer icon in the Printer Jobs window to see the print jobs in the queue.

d. Select one of the print jobs and click on the Selected menu. Select Properties from the menu. What is the size of the file to be printed? __________________________

e. Right click on one of the other jobs in the print queue and select properties from the menu. What is the size of the file to be printed? __________________________ What information is displayed in the Printer Jobs Properties window that is not contained in the lpstat display?

Step 5. Use File Manager to Print Files

The student can print a file from the File Manager window by dragging and dropping it on the Print Manager icon or dropping the object directly into the Print Manager Printer Jobs window. Once the file object has been dropped into the Print Jobs window, another window will be displayed in which the student can specify several options for printing. Once the Print button is selected, the file will be sent to the appropriate printer queue.

a. Access File Manager by clicking the File Drawer icon on the Front Panel or by right clicking on the workspace desktop and then on the Files menu. Keep the Printer Jobs window open.

b. Navigate to the planets directory, select the mars file, and drag the file to the Printer jobs window. Are the options in the Print window similar to those available with the lp command?

________________________

c. Click on the Print button. What window was displayed next? __________________________

d. Drag the pluto file to the Printer icon on the Front Panel. Were the results the same as dragging the file to the Printer Jobs window? __________________________

Step 6. Cancel a Print Job from the Queue

a. Select one of the print jobs and click on the Selected menu. Select cancel from the menu. What was the response? __________________________

b. Select one of the print jobs and click on the Selected menu. Select cancel from the menu. What was the response? __________________________

c. Click on the Yes button. Was the print job removed from the queue? ____________

d. Cancel all remaining print jobs in the queue.

Step 7. Remove Files and Directories Created in this Lab

Remove any files and directories created in the student’s home directory during this lab.

Step 8. Logout

Exit the CDE File Manager by clicking the File menu and click Close. Close any terminal windows. Click the EXIT icon on the front panel.
Objectives:

- Develop an understanding of backup and compression utilities
- Review `tar` archiving options
- Back up selected files with `tar`
- Back up a directory with `tar`
- Compress files
- Uncompress files
- Back up and compress the student’s home directory
- Restore the student’s home directory
- Use `jar` to archive and compress

Background:
In this lab, the student will work with the built-in multipurpose UNIX utilities to back up, compress, and restore data. This is the purpose of having a backup of important files or transferring multiple files as one file to and from another user.

Backing up data for safekeeping is also known as archiving. Archiving is one of the most important aspects of network security and support. Backups are a key component in a comprehensive security plan.

Transferring files to and from other users is done frequently using either email or file transfer commands (`ftp` and `rcp`) that will be covered in Chapter 16.

In this lab, the student will work with the Tape Archive (`tar`), `compress`, and Java Archive (`jar`) utilities to create a file used as a backup or to transfer to other users. We will also cover how to restore files from a `tar` or `jar` file.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 12, Section 3 – Backing Up, Compressing, and Restoring Files, and Section 4 – Combining Backup and Compression

b) The student will need the following:
   1. A login user ID, for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system.
   3. Networked computers in a classroom with the class file system installed.

Notes:
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
The student should login with the user name and password assigned by the 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 tar Archiving Options
The UNIX operating system has several integrated utilities that allow multiple files to be backed up and compressed. The tar (tape archive) command enables a user to back up single or multiple files in a directory hierarchy. The tar command is standard with all versions of the UNIX operating system. Although the tar command was originally developed for use with tape drives, tar can copy files to other locations on the hard disk, floppy disk, or other removable media. The tar command can create an archive from a single file. However, tar is primarily used to combine multiple files, such as the contents of a directory, into a single file and then extract the files later if they are needed. The newer jar command covered next compresses automatically. By itself, tar does not compress the files as it bundles them. The command syntax is shown below.

Command Format: `tar function [modifier] [output file] filename(s) / directory(s)`

The most frequently used options available with the tar command, shown all in lower case, are `c`, `t`, and `x`. It is not necessary, but accepted to precede these options with a dash (-) as with other UNIX command options.
**tar Command Function Options**

<table>
<thead>
<tr>
<th>Function</th>
<th>Meaning</th>
<th>Function Performed</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>Create (combine)</td>
<td>Create a new tar file</td>
</tr>
<tr>
<td>t</td>
<td>Table of Contents</td>
<td>List the table of contents of the tar file</td>
</tr>
<tr>
<td>x</td>
<td>Extract files</td>
<td>Extract the specified files from the tar file</td>
</tr>
</tbody>
</table>

**tar Command Function Modifiers**

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Meaning</th>
<th>Function Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>File name</td>
<td>Specify the tar file to be created as either a file on the hard disk, for example /tmp/file.tar, or a device file for an output device like a floppy disk, optical drive, or tape drive. In Solaris the floppy disk is shown as /vol/dev/aliases/floppy0 and the optical drive or tape drive as /dev/rmt0.</td>
</tr>
<tr>
<td>v</td>
<td>Verbose (view)</td>
<td>Execute in verbose mode. This mode allows the user to view what the tar command is doing as it is copying, displaying the table of contents, or extracting to or from the backup file or device. This option is normally used with the c, t and x tar options.</td>
</tr>
</tbody>
</table>

a. What option would the student use to archive one or more files? _________

b. What modifier would the student use to specify the name of a new tar file as the archive? _________

c. What modifier would the student use with the c, t, or x option to see the results of the tar command? _________

**Step 4. Back up Selected Files with tar**

To back up a group of files using the tar command and create a single tar file, use the syntax shown below. The cvf option is used to create (c) in verbose (v) mode a file (f). In this case, the files to be archived, such as tree1 and tree2, are on the hard drive in the current directory. The tar file to be created, such as trees.tar, will also be placed on the hard drive in the current directory, since no other directory or device is specified. The device can be a floppy or tape drive. The student can specify several files, including wildcards, or the name of a directory to archive all of the files in the directory. Whenever specifying the name of the tar file to create, add .tar to the end of the file so the student and others know it was created using the tar command. tar does not automatically add .tar.

```
tar cvf trees.tar tree1 tree2
```

a. Change to the student’s home directory. What command was used? _________

b. Create a new directory called tardir to put the student’s tar backup archives in. What command was used? _________

c. Backup the dante files, dante and dante_1, and create an archive tar file called dantefiles.tar in the tardir directory using a relative pathname. Use verbose mode. What command was used?

```
tar cvf dantefiles.tar dante dante_1
```

d. What was the response? __________________________
Step 5. Back up a Directory with tar
The student can also use the tar command to backup an entire directory including all the files and subdirectories contained within the directory.

a. From the student’s home directory, backup the contents of the dir2 directory. Use tar to create a new tar file archive called dir2.tar in the tardir directory using a relative pathname. Use verbose mode. What command was used? ________________

b. What was the response? ________________________________

c. Change to the tardir subdirectory. What command was used? __________

d. Use the tar command with the table of contents (t) option, verbose (v), and file (f) modifiers to see the contents of the dir2.tar file. What command was used? ________________

e. How many directories and files were archived in the dir2.tar file? ________________

Step 6. Compress Files
It is useful to archive files that have not been used for a while and then compress those files so they take up less disk space. It is also a good idea to compress files before transferring them to another UNIX user, that has the uncompress command, to save time. Any file, including those archives created with tar, can be compressed. Compression is a valuable tool since it reduces the amount of disk space files occupy, while still keeping the files readily available. The compress command is used to compress files and is included with all versions of the UNIX operating system.

When files are compressed with the compress command, the original file is replaced using the same name but with a .Z suffix appended to the end of the file name. Notice that this is an uppercase letter Z. When the compress command is used with the verbose (-v) option, it will show the name of the input (file.lst) and output file (file.lst.Z) and the amount of compression achieved. Use the ls -l (list long) command before compressing a file to see its original size, in bytes, and then again afterward to see the compressed size. Multiple files can be compressed simultaneously and wildcard metacharacters are supported. Compressed files are considered binary and cannot be viewed with the cat or more commands. The example below will compress all files beginning with file and display the result in verbose mode.
Command Format: `compress  option  file1  file2`

Example: `compress  -v  file*`

a. Change to the student’s home directory and create a new subdirectory called `compressdir`. What command was used? ________________

b. Display a long listing of the files in the `dir1/coffees` subdirectory in order to see their current size. What command was used? ________________

c. What is the current size of the `beans` file? __________

d. Change to the `coffees` subdirectory using a relative pathname. What command was used? ________________

e. What command could be used to find out what type of file `beans` was? ________________

   What type of file is the `beans` file? ________________

f. Copy the `beans` file from the current directory (`coffees`) to the `compressdir` subdirectory the student created earlier using the tilde (`~`) to represent the home directory path. What command was used? ________________

g. Change to the `compressdir` subdirectory using a relative pathname. What command was used? ________________

h. Compress the `beans` file using the verbose (`-v`) option. What command was used? ________________

i. What was the result of the `compress` command? ________________

j. Display a long listing of the files in the student’s current directory (`compressdir`) in order to see the files current size. What is the size of the `beans.Z` file now? __________. Does the original `beans` file still exist? __________

k. Check the size of the `files.tar` file in the `tardir` directory and then compress it. How much was it compressed? __________ What is the file’s name after it has been compressed? __________

Step 7. Uncompress Files
The corresponding command used to reverse the effects of the `compress` command is `uncompress`. Files cannot be used in their compressed form so it is necessary to use the `uncompress` command to restore the files to their original size. The `uncompress` command is a UNIX utility and can only be used to uncompress files compressed with the UNIX `compress` command.

Command Format: `uncompress  option  file1  file2`

Example: `uncompress  -v  file`

Note: It is not necessary to specify the `.Z` extension with the `uncompress` command. This command can uncompress multiple files and supports the use of wildcard metacharacters such as `?` and `*`.

a. Change to the `compressdir` subdirectory. What command was used? ________________

b. Display a long listing of the files in the student’s current directory (`compressdir`) in order to see the files current size. What is the size of the `beans.Z` file now? __________
c. Uncompress the beans file using the verbose (-v) option. What command was used?
   ________________

d. What was the result of the compress command? ________________

e. Display a long listing of the files in the student’s current directory (compressdir) in order to see the
   files current size. What is the size of the beans file now? _________ Does the compressed
   beans.Z file still exist? _________

Step 8. Back Up and Compress The Student’s Home Directory
It is good idea to perform a regular nightly backup of the student’s home directory or the important files as
a minimum. This section describes the process used to archive the student’s home directory to a tar file
and then compress the file. This is done in case the student would need to restore one of the files at a
later date. Restoring files from a tar file is covered in the next step.

a. Change to the student’s home directory and create a new subdirectory called backup. What
   command was used? ________________

b. Backup the entire home directory using the tar command and create an archive file called
   home.tar in the backup directory. Use the command tar cvf ~/backup/home.tar *

c. Change to the backup directory and display a long listing to verify that the home.tar is present.
   What is the size of the file? ________________

d. View the table of contents of the home.tar file. What command was used? __________

e. Compress the home.tar file using the verbose (-v) option. What command was used?
   ________________

f. What was the result of the compress command?
   ________________

g. Display a long listing and verify that home.tar.Z is there. What is the size of the file now?
   ________________

Step 9. Restoring Files from a tar File
In this section, the student will restore a file from the compressed tar file of the student’s home directory
that was previously created to simulate the recovery of important files from a backup. Just as tar can
combine files to a single archive file, tar can also be used to restore them. After the student has
uncompressed the tar file and extracted the original files, the student can move the files to the real home
directory as needed.

Example 1: tar xvf trees.tar tree1 tree2 would extract tree1 and tree2
files from the trees.tar file.

Example 2: tar xvf trees.tar would extract all files from the trees.tar file.

a. Change to the student’s home directory and rename the file1 and file2 files to file1.xyz and
   file2.xyz in preparation for restoring the original files from the home.tar file.

b. Change to the backup directory and uncompress the home.tar.Z file containing a backup of
   all the student’s files in preparation to restore the file1 and file2 files. What command was used?
   ______________________

   c. View the table of contents of the home.tar file only listing the files beginning with “file”. What
      command was used? ______________________
d. Record the pathname of the file1, file2, and file3 files exactly as it appears from the table of contents output.

e. Extract the file1 and file2 files from the home.tar file using the Extract (x) option with verbose (v) and file (f) modifiers. What command was used?

f. Which directory were the extracted files placed?

g. Move the file1 and file2 files into the student's home directory, to complete the restore process.

h. Now recompress the home.tar file, since the file was successfully recovered the needed files. The home.tar will not be needed again soon.

Step 10. Use jar to Archive and Compress
The `jar` (java archive) command is similar to the `tar` command, but compresses the resulting file in the same step. It is a Java™ application that combines multiple files into a single `jar` (Java archive) file. It is also a general-purpose archiving and compression tool, based on ZIP and the ZLIB compression format. The `jar` command is standard with the Solaris operating system, but is available on any system that has Java virtual machine (JVM) installed. The syntax and options for the `jar` tool are almost identical to the `tar` command. The following is an example of `jar`:

```
jar cvf trees.jar tree1 tree2
```

a. Change to the student's home directory and use the `jar` command to create a compressed archive file called fruit.jar in the backup directory using the two fruit files, fruit and fruit2. What command was used?

b. What was the result of the `jar` command?

c. Display a long listing of the backup directory. Is the `jar` archive file listed?

d. What is the size of the fruit.jar file? Were the files compressed as they were archived?

e. Are the original fruit files still in the student's home directory?

f. Change to the backup directory and view the table of contents of the jar file. What command did you use?

g. Use the `jar` command with the Extract (x) option and the verbose (v) and file (f) modifiers to extract the files in the fruit.jar file into the backup directory. What command was used?

h. What was the result from the `jar` extract?

Step 11. Remove Files and Directories Created in this Lab
Remove all files and directories created in the student’s home directory during this lab.

Step 12. 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.
Objectives:

- Become Familiar with CDE graphical backup, compression, and restore capabilities
- Access the Files subpanel Archive Option
- Archive a File with the Archive Option
- Archive a Folder with the Archive Option
- Restore an Archived File and Folder with File Manager
- Compress a file with the Files subpanel Compress Option
- Uncompress a File with File Manager

Background:
In this lab, the student will work with the Common Desktop Environment (CDE) Utilities to backup, compress, and restore files and folders. The Files subpanel on the Front panel in conjunction with File Manager can be used to archive, compress, and restore files creating the same results as using the `tar`, `compress`, and `uncompress` commands.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 12, Section 5 – GUI Backup Tools
b) The student will need the following:
   1. A login user ID, for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system with CDE.
   3. Networked computers in classroom with the class file system installed.
   4. Network printer available and print server running.

Notes:
Use the diagram of the sample Class File System folder tree to assist with this lab.

**Class File Tree Structure**

```
/  
   +-- home
       +-- user2

   +-- dir1
       +-- trees
       +-- fruit
def2
   +-- dir2
       +-- beans
       +-- coffee
   +-- dir3
       +-- planets
       +-- flowers
       +-- pluto
   +-- practice
       +-- dir4
dante1
   +-- file1
   +-- file2
   +-- file3
   +-- fruit2
```

**Step 1. Log in to CDE**
The student should login with the user name and password assigned by the instructor in the CDE entry box.

**Step 2. Access Files Subpanel Archive Option**
Click on the Files subpanel from the CDE Front Panel and then select the Archive menu option. An Archive window will open, to allow the student to enter the following information:

- **Folder for archive:** This is the folder where the archive (tar) file will be placed. The default is the student’s home folder. For example, /home/userX.
- **Name for Archive:** The name the student wishes to give the archive file.
- **File or folder to Archive:** The name of a folder, file, or group of files to be combined into an archive.

**Step 3. Archive a File**

a. Click on the Archive option from the Files subpanel above the file drawer icon. Leave the “Folder for Archive” entry blank. What is the default folder where the archive will be placed?

b. Tab to “Name for Archive” and enter dante.tar. Then, tab to “File or Folder to Archive” and enter dante.

c. Click OK to archive the dante file. What window was displayed when OK was clicked?

d. Close the Archive status window.

e. Open File Manager and check the student’s home folder for the new archive file. What is the icon on the Archived file?
Step 4. Archive a Folder

a. Click on Archive option from the Files subpanel. Leave the “Folder for Archive” entry blank. What is the default folder where the archive will be placed? 

b. Tab to “Name for Archive” and enter practice/dir2.tar. Where will this archive file be placed?

c. Tab to “File or Folder to Archive” and enter dir2.

d. Click OK to Archive the dir2 folder. What window was display when OK was clicked?

e. Close the Archive status window.

f. Open File Manager and navigate to the practice folder? Is the archived folder there?

Step 5. Restore an Archived File and Folder

The student can restore an archive quickly using File Manager.

Note: If the original file exists in the folder where the Archive is restored, it will be overwritten by the restored version.

a. Open File Manager and locate the tar file that was created previously from the dante file. The file should be in the student’s home folder.

b. Select the dante archive file and click on the Selected menu. What option from the menu would allow the student to extract the files from the tar file?

c. Select the Archive Unpack option from the menu. What was the response?

d. Navigate to the practice folder where the archive file for dir2 was placed.

e. Double click the archived dir2 file icon to see a table of contents of the tar file. Close the Archive window. This is the same as using the t option with the tar command.

f. Select the dir2 archive file and either use the Selected menu or right click the mouse and click on the Archive Unpack option. Were the dir2 directory contents unpacked?

Step 6. Compress and Uncompress a File

The student can also compress and uncompress files with File Manager. The compressed file will be renamed with a .Z extension. The compressed file will reside in the folder it was in before the compress. The default is the home folder. The student should compress a file before ftp-ing or emailing the files to another UNIX user or to save disk space if it is a file that is not used often.

a. Click on the Files subpanel from the CDE Front Panel and then select the Compress File menu option. A Compress window will open to allow the student to enter the name of the File to compress.

b. Enter the relative path to the beans file. What path was entered?

c. Open File Manager and navigate to the coffees folder. What is the compressed name of the beans file?
d. What is the icon for a compressed file? ________________________________

e. Use the Selected menu or right click on the file to see its properties. Click on information. Can the student tell what type of file it was? ________________________________

f. Close the Properties windows and right click on the compressed file icon again. Click on Uncompress to restore the beans file to its original form.

g. Use the Selected menu or right click on the file to see its properties. Click on information. Can the student tell what type of file it is? ________________________________

h. Close the uncompress status window.

**Step 7. Remove Files and Directories Created in this Lab**

Remove any files and directories created in the student’s home folder during this lab.

**Step 8. Logout**

Exit the CDE File Manager by clicking the File menu and click Close. Close any terminal windows. Click the EXIT icon on the front panel.
Fundamentals of UNIX

Lab 13.3.2 – Managing System Processes
(Estimated time: 50 min.)

Objectives:

- Develop an understanding of UNIX process management
- Review system process concepts
- Review the ps command and options
- List processes in the current shell
- List all processes running on the system
- Search for a specific process by command name
- Identify a process to terminate
- Use the kill command to terminate a process
- Find and terminate a process by user
- Terminate a process by command name

Background:
In this lab, the student will work with UNIX commands to identify system processes and control them. The UNIX network operating system manages tasks using processes. Processes can be initiated by either the operating system or by users. The majority of tasks the student will perform in the UNIX environment start a process. A process can start or spawn a child or subprocess, thus creating a process hierarchy or tree similar to the file system structure with parent / child relationships. The student will work with the ps (process status) command to monitor system processes and the kill command to terminate unwanted process. The student will also work with the Solaris commands pgrep (process grep) and pkill (process kill).

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 13, Section 1 – UNIX Systems Processes, Section 2 – Displaying Processes and Section 3 – Terminating Processes

b) The student will need the following:
   1. A login user ID for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system.
   3. Networked computers in classroom with class file system installed.

Notes:
Use the diagram of the sample Class File System directory tree to assist with this lab.

### Class File Tree Structure

```
/ → home → user2
  ↓ dir1 → dir2 → dir3
  |   |   |   
  |   |   |   
  trees → beans → dir4 → planets
  |   |   |   |
  |   |   |   |
  fruit → coffee → flowers → pluto
  |       |       |
  |       |       |
  nuts → beans → mars → dante
  |       |       |
  |       |       |
  file1 → file2 → file3 → fruit2
```

**Step 1. Log in to CDE**
The student should login with the user name and password assigned by the 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 System Process Concepts**
Each program the student runs creates a process, which is assigned a unique process identification number (PID). The PID is used by the system to identify and track the process until it has completed. The operating system (OS) kernel manages the initiation and termination of all processes. Every process requires system resources such as central processing unit (CPU) time and random access memory (RAM) space to work in. The OS allocates these system resources to each process when it starts and deallocates them when the process ends. The first two processes started when a UNIX system is booted are the sched (scheduler) and init (initialization), which manage other processes. There are several different types of processes on a UNIX system. These are summarized below:

- **Daemon:** Daemons are processes that are started by the UNIX kernel and exist for a specific purpose. For instance, the lpsched daemon exists for the sole purpose of handling print jobs.

- **Parent:** A process that spawns another process is referred to as its parent. A process called init daemon is the first one started. Every process except init has a parent process.

- **Child:** A process that is spawned by another process is referred to as a child process.

- **Orphan:** A process whose parent process terminates before it can return its output.

- **Zombie:** A child process that does not return to the parent process with its output. This process becomes lost in the system.
Using the information above, fill in the blanks in the following sentences.

a. Nearly every process that starts on a UNIX system gets assigned a unique __________________ by the kernel that is used to track the process from start to finish.

b. When new processes start the kernel also assigns system resources such as ___________ and__________

c. A process that never returns to the parent with its output is called a __________ process.

d. A process that is spawned by a parent process is called a __________ process.

e. A __________ process is one that spawns another process.

f. A UNIX system process that runs to provide services is a __________

g. If a parent process ends before the child can finish, it creates an __________ process.

Step 4. Review the ps Command and Options
The ps (process status) command is used to list the processes currently running on the system. This is normally done if a process is taking too long or appears to have stopped as indicated by a terminal window not responding or hanging. By listing the processes, the student can see the name of the command or program that initiated the process plus any child processes it may have spawned. By executing the ps command more than once, the student can see if a process is still running by looking at the time for the process, which is the amount of CPU time the process is using. If the amount of time does not increase, then the process may have stopped. The student can use the ps command to check the process ID (PID) of the process and then 'kill' the process if it is taking too long or has stopped.

The output of the ps command will display the PID number and the command or program associated with it. The PID number is normally used to terminate a process. There are three main options with the ps command as shown in the table.

<table>
<thead>
<tr>
<th>Command Format:</th>
<th>ps [-options]</th>
</tr>
</thead>
</table>

### ps Command Options

<table>
<thead>
<tr>
<th>ps Option</th>
<th>Meaning</th>
<th>Function or Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ps</td>
<td>No Options</td>
<td>Display information for current user processes in current shell or terminal window.</td>
</tr>
<tr>
<td>ps -e</td>
<td>Every</td>
<td>Display information about every process on the system.</td>
</tr>
<tr>
<td>ps -f</td>
<td>Full</td>
<td>Generate a full listing with all available information on each process.</td>
</tr>
<tr>
<td>ps -u userid</td>
<td>User</td>
<td>Display all processes for a particular user.</td>
</tr>
</tbody>
</table>

The basic ps command displays the information about process in the student’s current shell only. The student will only see processes that have been initiated with this terminal window.

<table>
<thead>
<tr>
<th>PID</th>
<th>TTY</th>
<th>TIME</th>
<th>CMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>785</td>
<td>pts/6</td>
<td>0:45</td>
<td>dbprog</td>
</tr>
<tr>
<td>742</td>
<td>pts/6</td>
<td>0:00</td>
<td>csh</td>
</tr>
<tr>
<td>689</td>
<td>pts/6</td>
<td>0:00</td>
<td>/bin/ksh</td>
</tr>
</tbody>
</table>

a. From the student’s current terminal window, practice using the ps command with each of the options shown.

3 - 8 Fundamentals UNIX 2.0—Lab 13.3.2 Copyright © 2002, Cisco Systems, Inc.
The `ps -ef` command displays all information about every process running on the system.

<table>
<thead>
<tr>
<th>UID</th>
<th>PID</th>
<th>PPID</th>
<th>C</th>
<th>STIME</th>
<th>TTY</th>
<th>TIME</th>
<th>CMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>0</td>
<td>0</td>
<td>80</td>
<td>16:46:41</td>
<td>?</td>
<td>0:01</td>
<td>sched</td>
</tr>
</tbody>
</table>

The following table defines the Column Headings for the `ps -ef` Command:

<table>
<thead>
<tr>
<th>Column Headings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UID</td>
<td>The user ID of the user that initiated the process.</td>
</tr>
<tr>
<td>PID</td>
<td>The process identification number of the process. The PID is used to kill a process.</td>
</tr>
<tr>
<td>PPID</td>
<td>The parent process identification number of the process.</td>
</tr>
<tr>
<td>C</td>
<td>The priority of the process.</td>
</tr>
<tr>
<td>STIME</td>
<td>Start time for the process.</td>
</tr>
<tr>
<td>TTY</td>
<td>Terminal type as the controlling terminal for the process.</td>
</tr>
<tr>
<td>TIME</td>
<td>The amount of CPU time used by the process.</td>
</tr>
<tr>
<td>CMD</td>
<td>The command name or daemon as name of the program executed</td>
</tr>
</tbody>
</table>

**Step 5. List Processes in the Current Shell**

a. In the student’s current terminal window issue the `ps` command with no options. What information is displayed? ________________

b. How many processes were displayed? ________________

c. What was the process ID (PID)? ________________

d. What was the command (CMD) that started the process? ________________

**Step 6. List All Processes Running on the System**

The `ps -ef` command will list all processes and can produce a fairly long listing.

a. In the student’s current terminal window issue the `ps -ef` command. What headings are displayed? **Note**: The student may want to pipe the `ps -ef` command to the `more` command to view the headings. ________________

b. How many processes were displayed? ________________

c. Count the number of processes by running the `ps -ef` command again and then pipe the output to the `wc` (word count) command. The first number is the number of lines displayed, which is also the number of processes. What command was used? ________________

How many processes were running? ________________

d. Display the output of all system process sorted by PID number and viewable one page at a time. What command was used? ________________

e. What is the command that has process ID number 1? ________________
Step 7. Search for a specific Process by Command Name

In order to stop a process the student must find the Process ID. On most systems there are hundreds of processes running and the `ps -ef` listing can be quite long. If the user knows the name of the executable program that started the process, the user can find the PID faster. By piping the output of the `ps` command through `grep`, a user can search for the specific process they want to terminate and determine the correct PID. As the student will recall, the `grep` command can search for any type of character string in the output of another command. Specific to Solaris, is the `pgrep` (process grep) command used to search for a specific process. The `-l` (long output) option will display the names of the processes associated with the PID found. The `-e` option displays the PID and the name of the initiating command, which allows `grep` to search on this information.

a. In the student’s current terminal window issue the `ps –e | grep lp` command to look for all processes that are related to the line printer scheduler daemon.

b. How many processes were displayed? ________________

c. What is the lowest process ID number of the processes displayed? ________________

d. In the student’s current terminal window issue the `pgrep -l lp` command to look for all processes that are related to the line printer scheduler daemon. What is the difference in output between `ps` and `pgrep`? ________________

Step 8. Identify a Process to Terminate.

The `ps -ef` command displays a full listing of every process, including the Process ID (PID) and its Parent Process ID (PPID). When trying to terminate a program or release a hung terminal window, it may not be enough to kill the process ID that is associated with the unresponsive application. It may be necessary to kill the Parent of that process. On rare occasions it may be necessary to kill the Parent of the Parent. It is important to be able to look at a PID and PPID to be able to trace from the child up the hierarchy to the parent processes that spawned them.

To do this, the student must first identify the PID of the lowest level unresponsive process. Normally the student would try to kill that processes PID. If this does not stop the process, the student may need to kill its parent. Killing a parent process will kill all child processes spawned by it. It is also much quicker to kill a parent process rather than killing several child processes.

a. From the current terminal window in CDE, enter the `ps` command.

b. How many processes were running? Why are there so few processes?

________________________________________

c. What is the name of the process running and what does it represent? ________________

d. What is the Process ID (PID) of this process? ________________

e. Enter the command `csh` to open a C Shell session under the Korn Shell. What does the student’s prompt look like now? ________________

f. Enter the command to display full information on processes running in the current shell. What command was used? _________ What processes are running now? _________

g. Is the Process ID of the Korn Shell (/bin/ksh) the Parent Process ID (PPID) of the C Shell (csh)?

________

h. Enter the command `sleep 1000 &` to create a process that suspends execution for 1000 seconds or approximately fifteen minutes. The ampersand (&) runs the command in the background and returns the shell prompt so the student can continue working.
Step 9. Use the kill Command to Terminate a Process.

Signals are used to terminate, suspend, and continue processes. Using Ctrl-c can sometimes terminate a process that is not responding. This sends an interrupt (INT) signal to the process, terminating it and any child processes it might have spawned.

The kill command provides a direct way to terminate unwanted command processes. It is useful when a user wants to stop a command that takes a long time to run, or when the user needs to terminate a process that they cannot quit in the normal way. Specifying the process id normally kills processes.

**Command Format:**  

```
kill [-signal] process-id
```

To terminate a process with the kill command, the student would first type `ps` to find out the PID or PIDs for the process or processes and then type `kill` followed by the PID or PIDs. If the student uses the kill command without specifying a signal, signal 15 (SIGTERM) is sent to the process with the specified PID number. This is referred to as a soft kill and usually causes the process to terminate. It is best to soft kill a process, if possible, since it closes files properly and terminates the process or processes smoothly.

If the student needs to forcibly terminate a process, the student can use the -9 option to the kill command. This option is referred to as a sure kill and is necessary for killing shells that will not respond to any other signal to terminate.

**Command Format:**  

```
kill -9 Process-id
```

**Note:** For processes other than shells, use the `kill -9 (SIGKILL)` command as a last option because it is an abrupt method and does not allow for proper process termination.

a. Enter the command to display full information on processes running in the current shell. What command was used? __________ What processes are running now? __________

b. Since the sleep process is now an orphan and has been adopted by the init process (PID #1), enter the command to perform a soft kill on the PID for sleep. If sleep has ended, with more than 15 minutes passing, repeat steps 7h through 7j again and then soft kill the sleep program. Enter the `ps -f` command again. What processes are running now? __________

c. Enter the command `csh` to open another C Shell session under the Korn Shell.

d. Enter the command to display full information on processes running in the shell. What processes are running now? __________

e. Is the Process ID (PID) of the Korn Shell (/bin/ksh) the Parent Process ID (PPID) of the C Shell (csh)? __________

f. Enter the command `sleep 1000 &` again.
g. Enter the `ps -f` command again. Is the Process ID of the C Shell (csh) the Parent Process ID (PPID) of the `sleep` command? __________

h. Use the `soft kill` command to kill the C shell process ID (PID). Use `ps -f` again to find out if the shell was killed. Did the shell die? __________ Why or why not? __________

i. Use the `sure kill` command to kill the C Shell PID. What was the response from the `kill` command? __________

**Step 10. Find and Terminate a Process by User**

The `ps` command can be used with the `-u` (user) option to find processes for the student or another specific user. This command is used more often than `ps –ef` since the student typically manages processes only owned by the student. The student may find processes for users by their login name or UID number. A user can only terminate their processes, but the superuser can terminate any process running on the system

**Command Format:** `ps -u login-ID or UID`

a. Start the Clock (OW Clock), the Calculator, and Calendar applications.

b. Use `ps` command with the `-u` option to find all processes running for the student’s login ID. For example userX or the student's numeric UID, for example 1004. What is the process ID for the Clock application? __________

c. Use a `soft kill` to terminate the Clock application. Is the clock still open and running on the student’s desktop? __________

d. Use the `ps –u` command combined with `grep` to find the PID number of the Calculator application instead of using `ps –u` and looking at the entire list. What command was used? **Note:** Guess at what letters the Calculator application command name might have in it and `grep` for that. __________

e. Terminate the Calculator application. Is the application still open and running on the desktop? __________

f. Find the PID number for the Calendar application then terminate it. What is the command name for the Calendar? __________ __________ __________ **Note:** sometimes the program name to look for or `grep` for is not intuitive. In this case, the Calendar application was one of the last processes that were started therefore having a higher PID number.

**Step 11. Terminate a Process by Command Name**

The `pkill` command is specific to Solaris and works exactly like the `pgrep` command, except that it terminates the process by matching process or processes command name (CMD) and sending a kill signal.

**Command Format:** `pkill CMD name`

a. Start a C Shell (csh) program and run the `sleep 500 &` command in a terminal window.

b. Use the `pkill` command to terminate the sleep process by its command name. Use the `ps –f` command again. Is it Dead? __________

c. Exit the C Shell.
Step 12. Remove Files and Directories Created in this Lab
Remove all files and directories created in the student’s home directory during this lab.

Step 13. 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.
Fundamentals of UNIX
Lab 14.3.4—Korn and Bash Shell Features
(Estimated time: 45 min.)

Objectives:

- Develop an understanding of Korn and Bash shell features
- Review the shell
- Review aliases
- Create aliases
- Display aliases
- Remove and bypass an alias
- Display command history
- Re-execute commands
- Edit the command line
- Complete a filename
- Customize the student’s shell prompt

Background:
In this lab, the student will work with either or both the Korn and Bash shell to understand its features and capabilities. The shell is the primary user interface to a UNIX system. The concept of the shell was introduced in Chapter 1 along with other key UNIX operating system components such as the kernel and the file system. The UNIX environment provides support for many built-in, for example Bourne, Korn, Bash, and C, and third party shells. This lab provides a brief review of the function of the shell and focuses on the most popular shells used with UNIX systems today, the Korn and Bash shell. The lab goes into greater detail about the unique features of each shell and provides opportunities to become a more efficient user. The student will work with aliases, history, re-execution of commands, and custom prompts in this lab.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 14, Section 1 – Review of the Shell, Section 2 – Additional Shell Features, and Section 3 – Shell Variables and Custom Prompts

b) The student will need the following:
   1. A login user ID for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system and the Korn or Bash shell.
   3. Networked computers in classroom with class file system installed.

Notes:
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
The student should login with the user name and password assigned by the 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 the Shell
A shell is an interface between the user and the kernel. The shell acts as an interpreter or translator. In other words, the shell accepts commands issued by the user, interprets these commands, and executes the appropriate programs. Shells can be command-line driven or Graphical. The system administrator decides which shell will be the default for a user when they create the user account. This lab will focus on the Korn (ksh) and Bash shells, which are the most widely used shells for Solaris and Linux. The student can use the ps (process status) command to see which shell is being used or type: echo $SHELL.

**Bourne shell ($):** The Bourne shell was the original shell program for the UNIX environment. Stephen Bourne developed the Bourne shell for the AT&T System V.2 UNIX environment. This shell does not support the alias or history commands or command line editing capabilities and is used primarily by system administrators. The Bourne shell prompt is a dollar sign ($).

**Korn shell ($):** The Korn shell is a superset of the Bourne shell and was developed by David Korn at Bell Labs. The Korn shell has many of the Bourne shell features plus added features such as aliasing, history, and command line editing. This is the most widely used shell and is the industry standard for system users. The Korn shell prompt is also a dollar sign ($).

**C shell (%):** A shell based on the C programming language. Like the Korn shell, C shell has additional features such as aliasing and history. The C shell was developed by Bill Joy of Sun Microsystems and is still widely used today. The C shell prompt is a percent sign (%).
The Bourne-Again Shell, or bash ($), has the feel of the Bourne and Korn shells and incorporates features from the C and Korn shells. Bash is the most popular shell with Linux and is the default for most distributions. The Bash shell prompt is also a dollar sign ($). Using the information above fill in the blanks in the following sentences.

a. A shell acts as an interpreter or ________ between the user and the kernel.

b. The default shell for a typical user is decided by the ____________

c. The student can determine which type of shell being used by issuing the ________ command.

d. The ________ is based on the C programming language and was developed by Bill Joy of Sun Microsystems. This shell uses the percent sign (%) as a prompt.

e. The ________ shell was the original shell and does not support aliasing or history.

f. The ________ shells use the dollar sign ($) as a prompt and supports aliasing, history, and command line editing.

Step 4. Review Aliases

An alias is a way to give a command a different name for use in the shell. Aliases provide an excellent way to improve efficiency and productivity when using shell commands. When set from the command line, aliases are only activated for the shell in which they are created. Adding aliases to the student's .kshrc or .bashrc file, which will be covered in the next lab, will activate them upon login or whenever a new window or shell is opened.

Command Format: alias aliasname=value

There are no spaces between the alias command, the equal sign (=), and the command or commands being assigned to the alias. Use single quotes for commands with options, spaces, or other special characters. See the examples shown below.

The reasons to use aliases are summarized below, along with some examples of how to create them. The syntax shown here is for the Korn and Bash shells only.

Substitute a short command for a long one. The student can reduce the number of keystrokes for commonly used long commands by creating an alias for the command.

Example: $ alias c=clear

Create a single command for a series of commands. The student can string several commands together and assign the commands one short alias name to reduce keystrokes.

Example: $ alias home='cd;ls'

Create alternate forms of existing commands. Some commands such as rm (remove files and directories) and cp (copy files) can be dangerous. An alias will allow the user to change the meaning of these commands to include the -i option so the user is prompted before accidentally overwriting a file or directory.

Example: $ alias copy='cp -i'

Using the information above, fill in the blanks in the following sentences.

a. An alias is a way to give a _________ a different name for use with the shell and can result in improved productivity.

b. When set from the command line, aliases are only activated for the _________ in which they are created.
c. Use __________ for commands with options, spaces, or other special characters.

d. There are no __________ between the alias command, the equal sign (=), and the command or commands being assigned to the alias.

Step 5. Create Aliases

a. Create an alias called h, with the lower case letter h, to substitute for the history command. What command was used? __________

b. Try the new h alias. What does it do? __________________________

c. Create an alias called p, with the lower case letter p, that will display every process running on the system with a full listing one screen at a time. What command was used?

____________________

d. Try the new p alias. What does it do? __________________________

e. Create an alias called mv that will substitute the command mv -i command for the mv command to prevent accidentally overwriting files when moving. What command was used?

____________________

f. Try the new mv alias. First copy the dante file into the practice directory, then attempt to move dante to the practice directory using the mv alias. What does it do?

____________________

Step 6. Display Aliases
To display aliases, use the alias command with no arguments/options. Using the alias command by itself will display all aliases set for the current session. Some aliases are pre-defined with the Korn shell.

a. Display all aliases for the student's current session. Can the aliases created previously be seen now? __________________________

b. Use the alias command and pipe the output to the wc –l command. This will count the lines of output from the alias command. Subtract the ones the student created from the total. How many other aliases were predefined as part of the student's Korn shell?

____________________

Step 7. Remove and Bypass an Alias
The student can unset a previously defined alias with the unalias command. This will remove the alias the student does not want to use anymore and the alias will no longer appear in the alias listing.

**Command Format:** unalias aliasname

a. Remove or unset the h alias the student defined previously. What command was used? __________

b. Display all aliases for the student's current session. Can the h alias be seen? __________

c. The student can also temporarily bypass an Alias. To bypass the alias and use the original version of a command, use a backslash before the command (\rm file1 file2). For example, the student has an alias rm that runs the rm -i command and the student has to remove many files and does not want to be prompted for each. The student can bypass the rm alias so when the student uses the rm command it does not execute with the interactive -i option.
d. Use the bypass option to bypass the mv alias the student created previously and use the original version of the mv command to move the dante file from the practice directory to the student’s home directory. Was there a prompt before overwriting? 

```

```

e. Type exit at the command line to close the student’s current terminal window and then start another new terminal window. Enter the alias command to list currently defined aliases for this session. Are any of the aliases the student defined previously there now? Why or why not? 

```

```

Step 8. Display Command history

The history feature records commands typed in the shell. In the Korn shell, history is automatically set up when the user first enters the shell. Whenever the user types a command, this function records it in a history file as an event. The Korn shell keeps a record of the last 128 commands entered but only displays the last 16 commands by default. The number of last commands entered that the Bash shell records is determined by the HISTSIZE variable and may be as high as 1000. Type echo $HISTSIZE to determine the current number.

```

```

Command Format: command

```

Step 9. Repeating Commands

The r (re-execute or repeat) command is one of many predefined aliases in the Korn shell. This enables the user to repeat commands from the history list. Bash uses the !! command to repeat previous commands.

<table>
<thead>
<tr>
<th>Bash Shell</th>
<th>Korn shell</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>! !</td>
<td>r</td>
<td>Repeats previous command</td>
</tr>
<tr>
<td>! 5</td>
<td>r 5</td>
<td>Repeats event number 5</td>
</tr>
<tr>
<td>! -2</td>
<td>r –2</td>
<td>Repeats command before last</td>
</tr>
<tr>
<td>! ls</td>
<td>r ls</td>
<td>Repeats last command beginning with ls</td>
</tr>
<tr>
<td>cd !$</td>
<td>No equivalent</td>
<td>Changes directory to the last argument of the previous command line</td>
</tr>
<tr>
<td>rm !*</td>
<td>No equivalent</td>
<td>Removes files used as arguments of previous command line</td>
</tr>
</tbody>
</table>

a. As mentioned previously, the r command is a predefined Korn shell alias. Enter the alias command. Examine the alias listing, what command does the r alias actually execute? If the student wanted to know more about the fc command what could be done?

```

b. Execute the following commands one at a time on a separate command line: ls –l, date, echo HI, clear, ls, cal. Now enter the history command. Identify the command number of one of the previous commands. What was the command number and command?
c. Repeat that command using the repeat command for the shell the student is using and the history line number. Did the earlier command used re-execute? 


d. Repeat a previous command that started with the letter ‘a’. This will repeat alias command in this case. What command was used? 


e. Experiment with some of the other repeat commands in the table above.

Step 10. Edit the Command Line
The in-line edit mode enables the student to edit a previous command on the current command line using keystrokes from the student’s preferred editor. This is helpful when the user wants to alter a previously executed, and sometimes lengthy, command line instead of retying it. When enabled, this feature is only activated for the current shell. This feature is set and used the same way in both the Korn and Bash shells.

Command Format: set [-+]o vi or set [-+]o emacs

Using set -o vi turns command-line editing on and specifies vi as the user’s preferred editor while set +o vi turns it off. Once the editing has been turned on, pressing the Esc key activates the in-line editor. The user then has access to vi commands to navigate through the history list and modify previously executed commands. The following table shows some of the most commonly used vi line editing commands.

<table>
<thead>
<tr>
<th>Command Line Edit Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>vi Command</strong></td>
</tr>
<tr>
<td>k</td>
</tr>
<tr>
<td>j</td>
</tr>
<tr>
<td>l</td>
</tr>
<tr>
<td>h</td>
</tr>
<tr>
<td>i</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>cw</td>
</tr>
<tr>
<td>x</td>
</tr>
</tbody>
</table>

a. Enter a command such as: more dir1/coffees/bean where the name of the file beans is misspelled. Press Enter to receive an error.

b. Turn command line editing on so the student can edit the previous command line. Press Esc to enter vi command mode. Press k to ‘kick up’ through the command history until the student gets to the command with the error in it.

c. Use the vi commands from the table above to move to the misspelled character in the command and press r to replace the d with the letter s. Press Enter to execute the command. Did the command execute correctly this time? 

d. Practice using the command line editing feature by recalling and editing previous command lines.

Step 11. Filename Completion
Both the Korn and Bash shells contain another feature that completes the name of a file or command. The set –o vi command turns on both the command line editing and filename completion features. Type the first few characters of a file name or command followed by a specific sequence of keys to instruct the shell to complete the remainder of the file name or command.

To use the completion feature to complete a filename:
1. Type a command, such as `ls`, `cat`, `rm` and so on, followed by one or more characters of a file name.

2. Press the Esc and backslash keys (`Esc \`) in sequential order, for the Korn shell, or the `Tab` key, for the Bash shell.

If the shell finds more than one file in the current directory that begins with the letters that were entered, the student can enter the next one or more characters in the desired file name. Then press `Esc \` for the Korn shell or `Tab` for the Bash shell.

Another way to use the completion feature is to request the shell to display a list of files that matches the entered filename. To display this list use `Esc=` for the Korn shell or `TabTab`, entered twice, in the Bash shell. When the desired file is determined from the list, reenter the command using an additional one or more characters then presses the `Esc \` or `Tab` again.

   a. Change directory to the student’s home directory.
   b. Copy the file1 file to file1.january. Make another copy naming it file1.febuary and again naming the new file file1.march.
   c. Type `ls -l file1` followed by either the `[Esc]=` keys (ksh) or `Tab` twice (bash). Did the shell display a list of files starting with file1?
   d. What keys would the student type next to list the file1.febuary file without typing the entire filename?

Step 12. Customize The Student’s Shell Prompt

The Korn and Bash shells use the Bourne shell as a basis for its features. These shared features include a login initialization file and variables. This is a placeholder for information used for customizing the shell. In addition, the Korn and Bash shells provide aliasing, command tracking, and command-line editing which the student has used so far in this lab. These features allow the user to be more productive. The initialization file used to store this information is discussed in the next lab.

The default prompt for the Korn and Bash shell is the dollar sign ($). The user can customize their own shell prompt to the user’s own choice using the `PS1` (Prompt String 1) predefined shell prompt variable `(PS1=$)`. `PS1` can include a wide range of expressions such as character strings, commands, or other variables. The variable `PS1` is a shell variable. Any change in the variable setting will remain until the shell is exited or until a subshell is opened. To make the customized prompt available from one session to the next, place the `PS1` variable in the `.kshrc` or `.bashrc` file. This will be covered in the next lab.

**Command Format: PS1=value**

a. The following are several different ways the standard `PS1` prompt ($) can be customized. Try each of these and then experiment with some of the student’s own choices.

   `$ PS1="Good morning$ "`  
   Assigns the prompt to a character string. Replace “Good morning “ with one of the student’s own choices.

   `$ PS1="`uname -n` `!$ "`  
   Uses a command (`uname -n`) with the history line number (!) for a unique prompt. This command causes the prompt to show the name of the host on which the user is working. An example would be `saturn41`. The back quotes ( ``) are used to substitute the output of the `uname` command instead of interpreting it literally. Double quotes surround the entire string.
$ PS1="$ "
Sets the prompt back to the original Korn shell dollar sign ($) prompt with a space after it.

$ PS1='${PWD} $'
The prompt will contain the current working directory. Single quotes surrounding the string containing
the PWD variable ($PWD) tells the shell to evaluate the current value of the PWD variable every time
the working directory changes.

For simple prompt settings in the Bash shell the PS1 variable can be set to:

- \u for username
- \d for the date
- \h for the hostname
- \$ for the dollar sign
- \W for the working directory

```
PS1=\u@\h:\W
```
example, would result in the prompt similar to jamie@colorado$.

**Step 13. Remove Files and Directories Created in this Lab**
Remove all files and directories created in the student’s home directory during this lab.

**Step 14. 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.
Objectives:

- Develop an understanding of Korn and Bash shell customization
- Review initialization files, variables and commands
- Review system wide initialization files
- Review user specific initialization files
- Modify the .profile or .bash_profile file
- Create the .kshrc or .bashrc file
- Test the student's customized initialization files

Background:
In this lab, the student will work with UNIX initialization files to customize the student's Korn or Bash shell login environment and adapt various options to the student's needs. In the previous lab, the student worked with aliases and custom prompts that were only active for the current session. When the student closed the terminal window or logged out, the custom settings were lost. These and other features can be automatically made available each time the student logs in by modifying specific initialization files. In this lab the student will work with various initialization files to make these changes take effect each time student logs in. The student will also review system wide and user specific initialization files and variables.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 14, Section 3 – Shell Variables and Custom Prompts and Section 4 – Shell Initialization Files

b) The student will need the following:
   1. A login user ID, for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system.
   3. Networked computers in classroom with class file system installed.

Notes:
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
The student should login with the user name and password assigned by the instructor in the CDE entry box provided.

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 Initialization Files, Variables and Commands
Initialization files: Initialization files contain a series of commands and variable settings that are executed when a shell is started. These files are not executable but are read by the shell when the user logs in to customize their environment. There are two levels of initialization files. The first level is system wide. System initialization files are maintained by a system administrator and reside in the /etc directory. The second level is user specific initialization files that reside in a user's home directory. The Korn shell user has two files, .profile and .kshrc (Korn shell run control) to customize their environment. The Bash shell user has two files, .bash_profile and .bashrc (Bash shell run control) to customize their environment. Notice that all of these shells have a dot (.) as the first character which means these shells are hidden.

Variables: In the previous lab, the student worked with variables like PS1 (prompt string 1) to customize the student’s Korn or Bash shell prompt. Since these variables were defined from the command line and were not put in an initialization file, they were only active for the current shell and when that shell was closed, the variables were lost. With this lab, the student will edit the initialization files and add variables and commands so that the variables and commands will be available every time the student logs in. Variables are either predefined and set automatically when logging in, or set by the user. The system administrator can also set the variables for the user. The variables can apply to the current, or local, shell only or to all shells and subshells, as global shells, depending on which initialization file they are in. Users can customize many of these variables for their own environment by placing the variables in the initialization files in the user’s home directory. To make the customizations available to all subsequent shells, the variable must be exported.

Commands: Unlike variables, commands, similar to aliases) cannot be exported. Commands cannot be made available to all subshells. If commands are to be available in a subshell, they must be placed in a
secondary initialization file that is read each time a subshell is opened such as the .kshrc (Korn shell) or .bashrc (Bash shell).

Using the information above fill in the blanks in the following sentences.

a. The two levels of initialization files are ________________, and ________________

b. For the Korn shell user, the ____________ file and the ____________ file reside in the user's home directory and can be used to customize their login environment.

c. For the Bash shell user, the ____________ file and the ____________ file reside in the user's home directory and can be used to customize their login environment.

d. ____________ are used to customize the user's shell and are either global or local.

e. In order for a variable to be available to subshells, it must be ____________

f. If commands are to be available in a subshell, the commands must be placed in a ________________ file that is read each time a subshell is opened such as the .kshrc or .bashrc.

Step 4. Review System wide Initialization Files

The primary system wide initialization file is the profile file which is kept in the /etc directory. The /etc/profile file is created by default when the operating system is installed and can be edited and customized by a system administrator. The /etc/profile applies to all Bourne, Korn, and Bash users. When a Bourne, Korn, or Bash shell user logs in, the system reads the /etc/profile file first, and then the system reads the user's .profile file for Bourne and Korn users or .bash_profile for Bash users. The third file that is read during login is the .kshrc file for Korn shell users or .bashrc for Bash shell users. This means that the user's preferences for variable settings can override the default settings set and maintained by a system administrator in the /etc/profile file. The /etc/profile file performs several functions, some of which are listed below:

**Exports Environment Variables:** Export Environment Variables makes environment variables available to subshells such as LOGNAME for login name.

**Exports the PATH variable for Default Command Path:** The default path is a list of directories where the shell will look when a command is executed. Exporting it makes the list of directories available to all shells and subshells.

**Sets the TERM Variable Default Terminal Type:** This defines the screen and keyboard characteristics of the user's workstation.

**Displays Contents of /etc/motd File:** The 'message of the day' file can be customized to display greetings or provide system information.

**Sets Default File Creation Permissions:** Sets umask values that determine the default permissions when a new file or directory is created.

Using the information above fill in the blanks in the following sentences.

a. The primary system wide initialization file is the ____________ file, which is kept in the ____________ directory.

b. The /etc/profile file is created by default when the operating system is installed and can be edited and customized by a _____________.

c. When a Korn shell user logs in, the system reads the ____________ file first, and then it reads the user's ____________ file, and then the ____________ file.
Step 5. Review User specific Initialization Files
After the system wide initialization file is read that sets system wide defaults, the user initialization files for the shell are read. The user initialization files provide great flexibility to the user for customizing their environment. Login initialization and customization can be accomplished using the system wide file /etc/profile by itself or in combination with the user specific initialization files.

The user initialization files can be set up as templates by the system administrator, and then modified by the user. The user specific initialization file or files such as .profile, .kshrc, and .bashrc are stored in the home directory of the user. The .kshrc file is the primary file that is used to customize the Korn shell user's environment and the .bashrc is used to customize the Bash shell user's environment. Depending on network administration policy, user specific initialization files can perform all or part of the following:

- Set Default Prompt as in the previous lab.
- Define Default Printer provides access to a printer.
- Set Default Permissions set umask for new files and directories.
- Set Default Terminal Type –is used by vi and other tools.
- Set noclobber –to prevent overwriting of files during redirection.
- Set Command Path –defines the directories to look in for executable files.
- Define Custom Commands –defines the aliases as described as in previous lab.

Using the information above fill in the blanks in the following sentences.

a. After the __________________________is read for system wide defaults, the __________________________ for the shell are read.

b. Login initialization and customization can be accomplished using the system wide file /etc/profile by itself or in combination with the user specific initialization files of __________ and __________ for Korn shell users or __________ and __________ for Bash shell users.

c. The __________ file is the primary file that is used to customize the Korn shell user’s environment and the __________ file is the primary file that is used to customize the Bash shell user’s environment.

d. Among other things, the .kshrc and .bashrc user initialization file can be used to define __________ as described as in previous lab as substitutes for commands.

Step 6. Modify The Student's .profile or .bash_profile File
The system administrator, as part of creating a user account, assigns the login shell. The login shell determines which initialization files are read during login. The Korn and Bash shells use two user specific environment files to set the user's environment the ~/.profile and ~/.kshrc (Korn shell) or .bash_profile
When a new Korn or Bash shell user is defined, a basic version of the .profile or .bash_profile file is created and placed in the user's home directory. Korn shell users need to modify the .profile file and define and export the ENV variable. This tells the Korn shell the location and name of the run control file to read each time a new shell is started, like a terminal window, as shown below. The Korn shell run control file does not have to be named .kshrc but it usually is. Likewise, Bash shell users need to modify the .bash_profile and define and export the BASH_ENV variable that points to the .bashrc file. These commands are usually added to the end of the .profile or .bash_profile file. This command sets the environment variable to point to the .kshrc or .bashrc file in the $HOME directory. HOME is a variable that is defined by the system to be the absolute path to the user's login directory.

**Command Format (Korn shell):**
```
ENV=$HOME/.kshrc; export ENV
```

**Command Format (Bash shell):**
```
BASH_ENV=$HOME/.bashrc; export BASH_ENV
```

a. Verify that the student is in the home directory and then use the `ls -la` (long list of all files) command to determine if the .profile or .bash_profile file exists. The student should include hidden files and may need to pipe the `ls -la` command to `more` to find the hidden files. In most cases the .profile or .bash_profile file is created automatically when a new user account is setup. If not the student will need to create it. Who is the owner of the .profile file? __________ What are the student's permissions for this file? __________ Should the student be able to make changes to it? __________

b. Use the `more` command to view the contents of the .profile or .bash_profile file. Note the PATH variable at the beginning, which defines directories where the system will look when the student issues a command. What is the PATH variable set to? __________

c. Is the PATH variable exported so it can be used in subshells? __________

d. The directory paths in the PATH variable definition are separated by colons (:). Is /usr/bin one of the paths listed? __________ List the contents of the /usr/bin directory and pipe it to the `more` command. Write down at least five of the commands listed that the student is familiar with.

```
```

e. Copy the .profile or .bash_profile file to create a backup of the file called .profile.bak or .bash_profile.bak for safekeeping. If the student makes mistakes and needs the backup file the student can remove .profile or .bash_profile and make a copy of the backup file using the original filename.

f. Use vi to modify the .profile or .bash_profile file. Add the ENV or BASH_ENV and export commands shown above, in the command format line, to the end of the file and then save and quit vi. View the contents of the file again. Are the commands the student entered present? __________ Check the commands to verify they were type in correctly.

**Step 7. Create The Student's .kshrc or .bashrc File**

Korn and Bash shell commands and features should be placed in the .kshrc (Korn shell run control) or .bashrc (Bash shell run control) file. The contents of the .kshrc and .bashrc files typically include: A customized prompt, aliases, activation of shell features, and custom variables

a. There is not usually a basic version of the .kshrc or .bashrc in the users home directory unless an administrator places one there as a template for a starting point. The student must create one or in some cases the student can copy an existing one from another user.

b. Create a new .kshrc or .bashrc file using vi. What command was entered? __________
c. If the student wants to save the changes and exit the vi editor what Last-line command would be used? __________

d. If the student wants to exit the editor without saving any changes what Last-line command would be used? __________

e. Set the student’s prompt to the name of the student’s machine with a dollar sign and a space after it. What was entered? Hint: see the Shell Variables and Custom Prompts section in Chapter 14. ______________

f. Set the vi editor to be activated for command line editing. What did was entered? __________

g. Create an alias called h as a short cut to represent the history command and another alias called c as a short cut for the clear command.

h. Create an alias called p as a short cut to represent the ps -ef | sort | more command. What command was entered? ______________

i. Create an alias called cp to replace the regular cp command that would prompt the student interactively if the student were about to overwrite a file. What command was entered? __________

j. Save the changes to the .kshrc or .bashrc file and exit vi. The student can exit without saving changes if the student wants to start again.

k. Use vi or the more command to view the student’s .kshrc file and verify that the commands were entered correctly.

**Step 8. Test The Student’s Customized Initialization Files**

Now that the student has modified the .profile or bash_profile file in the home directory and also created a new .kshrc or .bashrc file, it is time to test the results.

a. Close the terminal window and Logout (EXIT) of the system.

b. Login again as the same user and open a terminal window.

c. Has the student’s prompt changed as expected? ______________

d. Try the aliases that were defined in previous steps. Do all the aliases work? ______________

---

e. If none of the Korn shell customization features worked, the student may have a problem with the ENV variable in the .profile file. If some things worked and some did not the student probably has some mistakes in the .kshrc or .bashrc file.

f. Edit the .profile and .kshrc or .bash_profile and .bashrc files as necessary to get all of the customization changes to work.
Step 9. Modify the .kshrc or .bashrc and reread the file
   a. Open the .kshrc or .bashrc file in vi
   b. Modify the PS1 variable (the prompt) to be PS1="$PWD> ".
   c. Add a new alias named hm that changes back to the student's home directory, clears the screen, and displays a long listing one page at a time. What line did the student put in the .kshrc or .bashrc file to define the alias? Hint: what symbol, or metacharacter, is used when executing multiple commands on one command line? 
   d. Save the file and quit vi.
   e. Enter the command to reread the .kshrc or .bashrc file. What command line was entered?
   f. What does the student's prompt now display?
   g. Does the hm alias work?

Step 10. Remove Files and Directories Created in this Lab
Remove all files and directories created in the student’s 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.
Objectives:

- Write and execute a simple shell script
- Write, execute, and debug a more advanced shell script

Background:
In this lab, the student will write, execute, and debug simple and more complex shell scripts. Basic shell scripts are small files consisting of several UNIX commands that display basic system information or manipulate files. More complex shell scripts consist of variables and built-in programming commands which include the `read` command for interactive input, the conditional commands of `if`, `test`, and `case`, and the looping commands of `for`, `while`, and `until`.

It is not a requirement that a user writes shell scripts to use or manage a UNIX system. However, the more UNIX is used, the more a user will find instances when a repeatedly performed task has no single UNIX command. Writing a shell script is a way to create a custom command that can perform a single task or series of tasks and can be used repeatedly.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 15—Introduction to Shell Scripts.

b) The student will need the following:
1. A login user ID, for example user2 and a password assigned by the instructor.
2. A computer running the UNIX operating system.
3. Networked computers in classroom with class file system installed.

Notes:
Use the diagram of the sample Class File System directory tree to assist with this lab.

**Class File Tree Structure**

```
/    
|-- home
|-- user2

-- dir1
    |-- users
    |-- fruit
    |    |-- nuts

-- dir2
    |-- beans
    |-- recipes

-- dir3
    |-- planets
    |    |-- mars
    |    |-- pluto

-- practice
    |-- dir4
    |    |-- dante

-- file1
    |-- file2
    |-- file3
    |-- fruit
```

### Step 1. Log in to CDE
The student should login with the user name and password assigned by the 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. Simple shell scripts
An alias is typically used to execute one or more simple commands separated by a semicolon (;). A shell script is a text file that contains a sequence of commands for the shell to execute one line at a time. Complex tasks involving variables and other programming functions are accomplished using shell scripts.

The following are the general steps for creating a shell script:

- Decide what the script will do
- Make a list of commands
- Create a new file for the script
- Identify the shell the script will use
- Add commands and comments
- Save the script file
- Make the script file executable
- Type the name of the script to execute it
- Debug and modify the script if errors occur

a. When naming a script file the student should avoid using the name of an _________ or a _________________.

b. Typically, script files end with ______________________

c. Typically, user created shell scripts are located in what directory? ______________________
d. Which character is used before a comment in a script file? 

e. What is added to the script that tells the system which shell to use when interpreting the lines in the script? 

f. What is the output of `echo "date"`? 

g. What is the output of `echo "date"`? 

h. What is the output of `echo "$LOGNAME"`? 

i. What is the output of `echo "$LOGNAME"`? 

Step 4. Executing the script

Shell scripts always run in a separate shell. If a shell script is executed from a Terminal window shell, a sub shell is started to run the script.

There are two ways to execute a shell script. A user can use the `ksh` or `bash` command followed by the script file name. A user can also make the shell script file an executable file by typing the script name to execute.

a. What would the student type to execute the script named whoison.sh. The whoison.sh file does not have execute permission. 

b. The whoison.sh file has the permissions `r w -r - -r -`. Type the command to give the file `r w x r - x r - -` permissions. 

c. The executable whoison.sh script file is in the ~/bin directory. When the student types the whoison.sh file to execute it, the shell displays the message “command not found”. What is the problem?

Step 5. Debugging the script

A common problem when writing shell scripts is that the shell does not interpret the command the way the user expects.

A user can use the `ksh` or `bash` command with the `-x` (echo) and `-v` (verbose) options to help identify where the problems are in script. The `-x` option displays each line after it has been interpreted by the shell and places a plus sign (+) after each line is executed. The `-v` option is similar to the `-x` option except it displays the commands before the substitution of variables and commands are done.

a. What command line was entered to generate the following output from the whoison.sh script file?

```
echo "Hello $LOGNAME!"
+ echo Hello user10!
Hello user10!
echo "The number of users logged on is: `who | wc -l`"
who | wc -l
+ echo The number of users logged on is: 3
The number of users logged on is: 3
echo "Today’s date is: `date`"
date
+ echo Today’s date is: Fri Jun 21 2002
Today’s date is: Fri Jun 21 2002
```
Step 6. Creating, executing, and debugging a simple shell script

a. Create a script named firstscript.sh in the student’s home directory that does the following:
   1. Specifies which shell the script uses to interpret the script commands.
   2. Describes the purpose of the script that is to display users and print the current date and calendar.
   3. Displays to the screen “The number of users logged on is:” followed by the correct output.
   4. Displays to the screen “Today’s date is:” followed by the correct output.
   5. Displays to the screen “Below is the current months calendar” followed by the correct output.

b. Create a bin directory in the student’s home directory in preparation for the student’s next script file.

c. Modify the PATH variable in either the .profile or .bash_profile so that the bin directory just created in the student’s home directory is in the search path.

d. Create another script named info.sh in ~/bin directory. The script should contain comments and do the following when executed:
   1. Declares the script is to run in either the Korn or Bash shell.
   2. Clears the screen.
   3. Greets the person logged on.
   4. Display the hostname (uname –n).
   5. Displays the current date.
   6. Displays the calendar for the current month.
   7. Displays the current working directory.
   8. List the contents of the current directory.
   9. Displays a list of processes running in the current shell.
   10. Print in large letters “Have a Nice Day”.

e. Execute the info.sh script by starting a Korn shell first. Then, make it executable for the owner of the file and anybody in the same group.

f. Execute the script using ksh –x, ksh –v, and ksh –xv to debug.

Step 7. Complex script using shell programming
Built into every UNIX shell is a complete programming language consisting of commands and constructs that can be used to create more complex scripts beyond simply listing a series of commands.

In addition to containing built in shell programming commands such as if, else, read, case, while, and exit, complex scripts use predefined and user defined variables.

Variables
A variable is a placeholder for information required by processes so that they can function properly. A variable has a name and holds a value. Changing the value of a variable is called setting the variable. There are two types of variables used in shell scripts, shell variables and environment variables.
Shell variables are maintained by the shell and are only known to the current shell. These variables are local variables and can be viewed with the `set` command. Environment variables are variables known to the current and all child or subshells.

Environment variables can be displayed with the `env` command.

Local variables are set using the format `VARIABLE=value`. For example, `BACKUPDIR = /home/user2/myfiles` sets the variable named `BACKUPDIR` (backup directory) equal to the value of `/home/user2/myfiles`. Variable names are capitalized by convention but can be lowercase. Use the `echo` command to display the value of a variable.

### Positional Parameters

Positional parameters are special built in shell variables that can provide the name of the script file or arguments to the script as it executes. Their values are taken from arguments on the command line.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose and example</th>
</tr>
</thead>
</table>
| $0        | Command or name of script  
   `echo "The name of the script file is $0"` |
| $1 - $9   | Command line argument number  
   `echo "The first argument is: $1"` |
| $*        | All arguments entered on the command line  
   `echo "The args specified on the command line = $*"` |
| $#        | Number of arguments entered on the command line  
   `echo "The number of args on the command line = $#"` |

### Interactive Input

To make a shell script more flexible and user friendly the student can pause the script and ask for a single line of input. This is accomplished with the `echo` command to prompt the student for input and the `read` command to pause and take input from the keyboard. Whatever the student enters is stored in the variable name specified after the `read` command. Interactive input is a very powerful feature of shell scripts.

```
  echo "Enter your name: \c"    #Prompt the user for name  
  read name                     #Read from keyboard and save input in the "name" variable
```

**a.** Create a script named `whoison2.sh` in the `~/bin` directory that will accept a user name as an argument then displays all lines from the output of the `who` command containing information about the specified user. Specifically, the script will do the following:

1. Specifies which shell the script uses to interpret the script commands.
2. Reads the first argument specified when the script is run and assigns it to the variable named “user”.
3. Display all lines from the output of the `who` command containing information about the specified user.
b. Create then execute the grepfor.sh script shown below. Use 'root' for the pattern and '/etc/passwd' as the file to search the first time the student runs the script.

```bash
#!/bin/ksh
# grepfor.sh prompts for a pattern and a file then generates output
echo 'Enter the pattern to search for: \c' or echo -e 'Enter the pattern to search for: \c' (bash)
read pattern
echo 'Enter the filename to search: \c' or echo -e 'Enter the filename to search: \c' (bash)
read filename
echo
echo Results:
grep $pattern $filename
```

c. Modify the script so that when the student runs it and specifies “user” for the pattern and “/etc/passwd” for the file the result is the number of user accounts in the password file that have “user” in the name. What was changed?

---

**Step 8. Complex scripts that use the if, test, and case commands**

To be useful, a program must be able to test for conditions and make decisions. The program must be able to examine the result of a command and choose between two or more courses of action. The simplest test is to determine whether a condition is true or false. If the condition is true, execute any number of subsequent commands. If the condition is not true, continue with the script.

Commands that perform some tasks based on whether a condition succeeds or fails are called conditional commands. The three most frequently used conditional commands are `if`, `test`, and `case`.

### The if-then Command Format:

- `if` command is successful
  - `then`
  - `execute command or commands`
  - `fi`

### The if-then-elif (else if) Command Format:

- `if` command is successful
  - `then`
  - `execute command or commands`
  - `elif` command is successful
  - `then`
  - `execute command or commands`
  - `else`
  - `execute command or commands`
  - `fi`

### The if-then-else Command Format:

- `if` command is successful
  - `then`
  - `execute command or commands`
- `else`

The `test` built-in shell conditional command is often used for testing one or more conditions following the `if` command.

### The if-test Command Format:

- `if test expression` or `if [ expression ]`
  - `then`

---
execute command or commands

fi

execute command or commands

fi

The **case** built in shell command is used when there are many conditions to test.

**Command Format:**

```bash
  case value in
  value1 )
      Command
      Command
      ;;
  value2 )
      Command
      Command
      ;;
  * )
      Command
      ;;
  esac
```

The value of a case variable is matched against `value1`, `value2`, and so on, until a match is found.

---

**a.** Modify the `whoison2.sh` script in the `~/bin` directory so that it will accept a user name as an argument and if the user is logged on, display "<specified user> is currently logged on".

**b.** Create a `test1.sh` script in `~/bin` that contains the following lines:

```bash
#!/bin/ksh
# test1.sh script to demonstrate the test command
echo "Do you want a long listing of the files in the current directory?"
  echo " enter y for Yes or n for No:"
```

**c.** Modify the `test1.sh` script to:

- Read the answer given by the user running the script into a variable named 'answer'.
- Test the answer entered by the user running the script. If the answer is "y", clear the screen and execute a long listing. Otherwise, if the answer is "n", execute `ls` without any options. What remaining lines were added to the script file to satisfy requirements a and b?
Step 9. Complex scripts that use Flow control (loops)

The student can use loops to control the flow of execution in a script. A loop repeats a sequence of instructions repeatedly until a predetermined condition is satisfied.

Often a script is concerned with performing the same operation or set of operations on each file in a directory or list, each line in a file, or each word in a line. The shell provides three looping constructs to accomplish this type of action: the `for` loop, `while` loop, and `until` loop.

<table>
<thead>
<tr>
<th>The for Loop</th>
<th>The while Loop</th>
</tr>
</thead>
<tbody>
<tr>
<td>The for loop executes a list of commands one time for each value of a loop variable.</td>
<td>The while loop repeatedly executes a group of commands within the body of the loop until the test condition in the expression becomes false. In other words, while the expression is true, execute these commands.</td>
</tr>
<tr>
<td><strong>Command Format:</strong></td>
<td><strong>Command Format:</strong></td>
</tr>
<tr>
<td><code>for</code> variable in list</td>
<td><code>while</code> command</td>
</tr>
<tr>
<td><code>do</code></td>
<td><code>do</code></td>
</tr>
<tr>
<td>Commands</td>
<td>Commands</td>
</tr>
<tr>
<td><code>done</code></td>
<td><code>done</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The until Loop</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>The until loop is very similar to the while loop but it executes a series of commands while a command continues to fail, non-zero exit status. When the command finally executes successfully, the loop is terminated and execution passes to the first statement following <code>done</code> construct. The until condition is checked at the top of the loop, not at the bottom.</td>
<td></td>
</tr>
<tr>
<td><strong>Command Format:</strong></td>
<td></td>
</tr>
<tr>
<td><code>until</code> command</td>
<td></td>
</tr>
<tr>
<td><code>do</code></td>
<td></td>
</tr>
<tr>
<td>Commands</td>
<td></td>
</tr>
<tr>
<td><code>done</code></td>
<td></td>
</tr>
</tbody>
</table>

a. What does the following For Loop script do when executed?

```bash
#!/bin/bash
for FILE in file*
do
    ls -l $FILE
done
echo “all done!”
```
b. Create then execute the shell script above and name it loop1.sh. Place it in ~/bin.

c. Modify the loop1.sh script so it displays a long listing of all files entered on the command line. For example $ loop1.sh file1 file2 or $ loop1.sh file*, instead of the ones specified in the script like above.

d. Create a script named loop2.sh and place it in ~/bin. The script will use the Until Loop. When run the script will check to see if a user, specified as an argument to the script, is logged on. If the specified user is not currently logged on, the script will display the message "<specified user> is not logged on" and sleep for five seconds before trying again. Once the specified user logs on, the script will display the message "<specified user> is logged on".

e. Test the loop2.sh script in a terminal window using the student’s user name as the argument to the script. For example, $ loop2.sh user1. What did the script display?

f. Run the script again using a different user name. The user name must be listed in the /etc/password file. For example, $ loop2.sh user5. What did the script display?

g. Leave the script running and open another terminal window. Using the telnet utility, the student should login to the student’s own system. Use $ telnet <student's hostname or ip address> to login as the user name specified as the argument to the student’s script. In this example it would be user5. What did the script running in the first terminal window display?

h. Create a new script named counter1.sh and place it in the ~/bin directory. Enter the following lines of this While Loop script:

```bash
#!/bin/ksh
# counts from 1 to 5 then displays “I can count!”
#
num=0 #initialize the count variable
while [ $num -lt 6 ] #loop while $num is less than 6
do
    echo "number: $num"
    num=`expr $num + 1` #increment count by one
done
echo "I can count!"
```

i. Execute the script, debug if necessary. Did the script count from one to five? _____ What must be changed in the script so it counts from one to five?

Step 10. 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.
Fundamentals of UNIX

Lab 16.2.4 – Networking Commands
(Estimated time: 45 min.)

Objectives:

- Develop an understanding of UNIX and TCP/IP networking commands
- Ping another TCP/IP host
- Use traceroute to check internet path
- Telnet to another host
- Remote login to another host
- Terminate a local process from a remote host
- Use rlogin and pkill to recover from a “hung” CDE session
- Retrieve a file from another host with ftp

Background:
In this lab, the student will work with UNIX and Transmission Control Protocol / Internet Protocol (TCP/IP) Networking Commands. TCP/IP provides network connectivity support for the UNIX operating system and many other network operating systems. All hosts attached to the Internet run the TCP/IP protocols. Several network commands are available to test connectivity and to connect to remote host machines for administration and troubleshooting. In this lab, the student will work with the TCP/IP ping troubleshooting utility, telnet remote access utility, the FTP File transfer protocol, the UNIX rlogin remote access command, and the traceroute command.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 16, Section 4 – Network Concepts and Utilities.
b) The student will need the following:
   1. A login user ID, for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system.
   3. Networked computers in classroom with class file system installed.

Notes:

---------------------------------------------------------------------

---------------------------------------------------------------------

---------------------------------------------------------------------

---------------------------------------------------------------------

---------------------------------------------------------------------

---------------------------------------------------------------------

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

### Class File Tree Structure

```
/  
|   
|   home
|   user2
|   
|   dir1
|   |  
|   |  users
|   |  fruit
|   |  coffee
|   |  beans
|   |  nuts
|   
|   dir2
|   |  beans
|   |  recipes
|   |  notes
|   |  flowers
|   |  mars
|   |  pluto
|   |  dante
|   |  dante_1
|   |  file1
|   |  file2
|   |  file3
|   |  file4
|   |  fruit
```

### Step 1. Log in to CDE
The student should login with the user name and password assigned by the 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. Ping another TCP/IP Host
**Ping** (Packet Internet Groper) is a very useful utility that is part of the basic TCP/IP protocols package and is standard with every UNIX system. A computer that is running the TCP/IP protocol stack can make use of the **ping** command.

Ping is a good low-level troubleshooting tool, testing layers 1 through 3 of the OSI model. The **ping** command tests basic connectivity between TCP/IP hosts by sending an ICMP (Internet Control Message Protocol) echo request to another computer or 'host' on a TCP/IP network. If there is a reply from the destination host, then there is a good connection between them. If a user is unable to run an application on a remote host, the user can **ping** it as a basic connectivity test. If a user does not get a response, the problem may not be with the application, but instead, the host may be down or the network link may be down. **Note**: In order to use the **ping** command in Solaris, the student may need to specify the full pathname of: `/usr/sbin/ping`

Any network operating system that is running the TCP/IP protocol can send and respond to **ping**. A user can **ping** the name of a host computer if the user has a naming service running such as Domain Name Service (DNS) or if the user has entered the host name and IP address in their `/etc/hosts` file. If not, the user can directly **ping** the IP address of the host, which is preferred since it is a more basic test of connectivity. IP Addresses use a **dotted decimal** format such as 172.16.238.15. IP Addresses are a 32-bit address divided into four parts known as **octets**. Each octet is made up of eight bits. Instead of showing the eight binary bit values (0 or 1), which would be difficult to remember, they are converted to decimal for readability. Each octet can have a value of from 0 to 255 decimal or 256 possible combinations, $2^{8}$.
Command Syntax:  
\texttt{ping \ host\_ip\_address}

\textbf{OR}  \texttt{ping \ host\_name}  \textbf{(if name resolution is available)}

a. If the student has a server in the classroom, obtain the IP address from the instructor. Ping the IP address of the server. What command was used? \underline{\text{______________}} What was the response? \underline{\text{______________}}

b. Obtain the IP Address of one of the student’s lab partners and use the \texttt{ping} command to test connectivity between both machines. What devices did the \texttt{ping} packet have to go through to complete its path? \underline{\text{__________________________________________}}

c. Name resolution converts host names to IP address through the use of a name server or a hosts file on the student’s computer. If name resolution is available, \texttt{ping} the name of the student’s lab partner’s machine by its host name. What command was used? \underline{\text{______________}} What was the response? \underline{\text{______________}}

\textbf{Step 4. Use \texttt{traceroute} to check internet path}

\texttt{traceroute} is another useful utility that is part of the basic TCP/IP protocol suite. Any computer running TCP/IP can use the \texttt{traceroute} command.

\texttt{traceroute \ hostname \ or \ traceroute \ [ip \ address \ ]}

\texttt{traceroute} is a good troubleshooting tool for checking the connection between computers that are interconnected with routers. Routers provide for connection of private networks and the public Internet. The \texttt{traceroute} command checks the time it takes for a packet to get from one router to the next and can help isolated slow links. Each router the packet goes through to reach its destination is listed in the command output and is referred to as a hop. As with ping, \texttt{traceroute} tests physical connections and IP addressing from one host to another on a TCP/IP network.

Each 'hop' is a physical machine that reads the packet address and forwards it until the address reaches its final destination.

a) Run the \texttt{traceroute} command on cisco.com. How many hops did it take to reach cisco.com from the student’s computer? \underline{\text{______________}}

\textbf{Step 5. Telnet to another Host}

Telnet is a client-server \textbf{terminal emulation} program that allows a user to connect to another system. The telnet server simulates a terminal in order to authorize a user or \texttt{telnet} client to connect to a remote system and work in that environment. When a user \texttt{telnets} to another host, the user is prompted for a name of an account on the remote system and password. Once a session is established from the user's \texttt{telnet} client, the user can enter commands as if they were entering the commands directly on the server console. The user can telnet to several different hosts and have multiple telnet sessions opened simultaneously.

Telnetting to another host is helpful under the following circumstances:

- To access information on another workstation.
- To access the user’s workstation remotely to read mail.
- To kill a process that has caused the student’s or another user’s workstation to hang.

Telnet is also used as a network troubleshooting tool. It is a nongraphical communications utility that can be used to check the upper layers of the OSI model. Telnet runs at layer 7, the Application layer. If the student is having trouble executing another, perhaps graphical, client-server application, the student can try to telnet to the host or server to verify that the TCP/IP protocol stack is functioning correctly. Remember, not all network operating systems support the telnet server function. For instance, the student
can telnet from a Windows 9x or NT/2000 workstation to a UNIX server, but the student cannot telnet to the Windows workstation.

The user can telnet to the name of a host computer if they have a naming service running such as Domain Name Service (DNS) or if the user has entered the host name and IP address in their /etc/hosts file. If not, the user can telnet the IP address of the host. When using telnet, the user can:

- Open a session on a remote machine
- Alternate between the remote session and the local session
- Access machines that do not run under the UNIX environment

**Command Syntax:** `telnet host_ip_address` (or hostname)

a. If the student has a UNIX server in the classroom, telnet to the IP address of the server. **Note:** if the student does not have a UNIX server in the classroom, go to exercise ‘d’. What command was used? _______________ What was the response? _______________

d. Telnet to the student’s lab partner’s system using the student user ID or a guest account, if one was created. What is the student’s prompt now that the student is a remote terminal connected to the other system. _______________

e. What command displays the name of the system the student is currently logged on to remotely? _______________

f. Enter some UNIX commands such as `ls`, `id` and `pwd` to see what directory the student is in on the remote machine. What is the directory now? _______________

g. Type a command and try the Backspace key. Does it work? _______________ What can the student do to make the Backspace key work? **Hint:** see the Tip in the Telnet Utility section of Chapter 16. _______________

h. Terminate the student’s telnet session to the remote host. What was the response? _______________

**Step 6. Remote Login to Another Host**

Use the `rlogin` command to establish a remote login session on another UNIX host, which can be a server or a workstation. Remotely logging in to another host is helpful under the following circumstances:

- To access information on another workstation.
- To access the student’s workstation remotely to read mail.
- To kill a process that has caused the student or another users workstation to hang.

The student can rlogin to any UNIX system provided the student knows the username and password. It is possible that the student has the same named account on multiple systems that would allow the student to rlogin to another system using the same account name.

Use the `-l` option to specify a different login ID for the remote login session. The system administrator can set up a guest account so users can remotely log on to a server. Before attempting to remotely login to another system as a different user, be sure the student has an account on the desired remote machine.
Command Format: `rlogin hostname -l username`

a. If the student has a UNIX server in the classroom, `rlogin` to the hostname or IP address of the server. If the student does not have a classroom server, `rlogin` to the student’s partners system. What was the response?

b. Issue the `id` command. What is the user id shown?

c. Issue the `uname –n` command. What was the result?

d. Issue the `pwd` command. What was the result?

d. Terminate the student’s `rlogin` session to the remote host by typing `exit` or `Ctrl-d` at the command prompt. What was the response?

Step 7. Terminate a Local Process From a Remote Host
Sometimes the user may start a process, either an application or command line, that causes the user’s system to ‘lock up’. The keyboard and mouse do not work and the user cannot open a terminal window to kill the process. In situations like this, it is best to first try to `rlogin` or `telnet` to the user’s computer from another system and attempt to kill the problem process before rebooting the user’s system. After successfully killing the process that caused the student’s system to not respond, type `exit` to end the student’s remote session then return to the student’s system.

If the student cannot identify the problem process that is locking up the system then use the `pkill -9 ksh`, or `csh` if the student’s login shell is the C shell program, command. This will terminate the login session on the student’s system, including CDE, and return back to the CDE Login Manager.

For the next exercises the student will work with a partner. The first person will do steps a-d, then the second person will complete the same steps.

a. Start the Clock application so it is visible on the student’s screen.

b. The student should leave the machine and work with a partner to `rlogin` as the student to the student’s machine from the partner’s machine. Verify that the student is in the student’s home directory by issuing the `pwd` command.

c. Terminate the process for the Clock application. What was the result?

d. Remain remotely logged on to the student’s system and terminate the login session as if the student’s system was not responding. Did the student’s system return to the CDE Login Manager?

e. Return to the student’s system and log back in.

Step 8. Retrieve a file from Another Host with `ftp`
The `ftp` (File Transfer Protocol) application is part of the TCP/IP protocol suite and is standard with the UNIX operating system. It can be used to transfer files using ASCII, for text files, or binary, for all other file types, mode between systems using similar or dissimilar operating systems. This provides a basic means of transferring files from one file system to another. Remember, if the student transfers a binary file using the default ASCII (text) mode the copied file will be useless.

Most computers running Servers with web sites set up for downloading files sometimes provide an *anonymous* ftp account so users can pull files off the server. For this kind of an account, at the Name
prompt, the word *anonymous* is entered instead of accepting the default displayed. If a password is required for the anonymous account, it will usually be the user’s full email address.

**Command Syntax:**

```
ftp   host_ip_address
```

Or

```
ftp   host_name
```

(if name resolution is available)

Once the student has successfully used `ftp` to access a remote host, some familiar file and directory access commands like `cd` and `ls` are available.

a. Use the `ftp` command to connect to the classroom server or the student’s lab partner’s system using the IP address. The student can use the machine name if name resolution is available. What was the response?

b. If the student has an account on the machine the student is going to ftp to, the student's login name will automatically be entered as the account **Name** and the student will be prompted for a password. If the student does not have an account on the remote system, the student will be prompted for a login name. Enter either another known account name on the remote system or **anonymous** for the login name and a password.

c. What is the student’s prompt now? ________________

d. To view a list of FTP available commands, enter the ? at the `ftp>` prompt. What are some of the `ftp` commands available? ________________

e. Enter the `pwd` command to determine what the current default directory is on the remote system. What is the current directory? ________________

f. If permissions are set by the site’s system administrator for a user to see the contents of a directory the `ls` command will display files in that directory. Enter the `ls` command to get a listing of files. Identify a file to be copied to the student's machine and the file type. Is it a text file or another file type (binary)? ________________ What command was used to determine the file type? ________________

g. When the student copies a file from ('get') a remote system it is placed in the current directory of the local system. When the student copies a file to ('put') a remote system the file is placed in the current directory of the remote system. Use the `cd` command to change directories on the remote system and the `lcd` (local change directory) command for the student’s local system. Change to the practice directory on the student's local system so when the student copies the file from the remote system it will place the file in the practice directory. What command was used?

______________

h. Set the student’s file transfer mode depending on the type of file the student chose to copy. If the student file is a text file what needs to be done? ________________ If the student’s file is not a text file what needs to be done? ________________

i. Retrieve (copy) the identified file from the ftp host using the `ftp get filename` command. The file will be paced in the student's current working directory (~/practice) using the same name unless the student specifies otherwise. What was the response? ________________

j. If the student has permissions, use the `ftp put filename` command to copy a file to the remote host.

k. If the student has set the transfer mode to binary and now wants to copy a text file, what ftp command would the student use to set the transfer mode to ASCII? ________________
l. End the student’s ftp session and return to the student’s shell prompt. What ftp command was used? 

m. Verify the file the student copied from the remote system is in the student’s practice directory.

**Step 9. Remove Files and Directories Created in this Lab**
Remove all files and directories created in the student’s home directory during this lab.

**Step 10. 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.
Objectives:

- Develop an understanding of UNIX and TCP/IP networking commands
- Ping another TCP/IP host
- Use traceroute to check network path
- Telnet to another host
- Remote login to another host
- Terminate a local process from a remote host
- Use rlogin and pkill to recover from a 'hung' CDE session
- Retrieve a file from another host with ftp

Background:
In this lab, the student will work with UNIX and Transmission Control Protocol / Internet Protocol (TCP/IP) Networking Commands. TCP/IP provides network connectivity support for the UNIX operating system and many other network operating systems. All hosts attached to the Internet run the TCP/IP protocols. Several network commands are available to test connectivity and to connect to remote host machines for administration and troubleshooting. In this lab, the student will work with the TCP/IP ping troubleshooting utility, telnet remote access utility, the FTP File transfer protocol, the UNIX rlogin remote access command, and the traceroute command.

Tools / Preparation:

a) Before starting this lab, the student should review Chapter 16, Section 4 – Network Concepts and Utilities.

b) The student will need the following:
   1. A login user ID, for example user2 and a password assigned by the instructor.
   2. A computer running the UNIX operating system.
   3. Networked computers in classroom with class file system installed.

Notes:
Use the diagram of the sample Class File System directory tree to assist with this lab.

Class File Tree Structure

```
/  
│  
│  home  
│  
│  user2  
├── dir1  
│   └── users  
│        └── fruit  
│               └── coffee  
│                   ├── beans  
│                   │     └── nuts  
│                   └── trees  
│                           └── flowers  
│                                   └── notes  
│                                           └── pluto  
│                                               └── mars  
├── dir2  
│   └── recipes  
│         └── coffee  
│                 └── beans  
│                         └── nuts  
├── dir3  
│   └── practice  
│       └── pluto  
├── dir4  
│   └── dante  
│       └── file1  
└── file2  
    └── fruitz  
```

**Step 1. Log in to CDE**
The student should login with the user name and password assigned by the 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. Ping another TCP/IP Host**
**Ping** (Packet Internet Groper) is a useful utility that is part of the basic TCP/IP protocols package and is standard with every UNIX system. A computer that is running the TCP/IP protocol stack can make use of the **ping** command.

Ping is a good low level troubleshooting tool, testing layers one through three of the OSI model. The **ping** command tests basic connectivity between TCP/IP hosts by sending an ICMP (Internet Control Message Protocol) echo request to another computer or 'host' on a TCP/IP network. If there is a reply from the destination host, then there is a good connection between them. If the user is unable to run an application on a remote host, the user can **ping** it as a basic connectivity test. If the user does not get a response, the problem may not be with the application. Instead, the host may be down or the network link may be down. **Note:** In order to use the **ping** command in Solaris, the student may need to specify the full pathname of: /usr/sbin/ping

Any network operating system that is running the TCP/IP protocol can send and respond to **ping**. A user can **ping** the name of a host computer if they have a naming service running such as Domain Name Service (DNS) or if the user has entered the host name and IP address in the `/etc/hosts` file. If not, a user can directly **ping** the IP address of the host, which is preferred since it is a basic test of connectivity. IP Addresses use a **dotted decimal** format such as 172.16.238.15. IP Addresses are a 32-bit address divided into four parts known as **octets**. Each octet is made up of eight bits. Instead of showing the eight binary bit values (0 or 1), which would be difficult to remember, they are converted to decimal for readability. Each octet can have a value of from 0 to 255 decimal or 256 possible combinations, 2 to the 8th power.
Command Syntax: `ping host_ip_address`

OR `ping host_name` (if name resolution is available)

a. If the student has a server in the classroom, obtain the IP address from the instructor. Ping the IP address of the server. What command was used? __________________ What was the response? __________________

b. Obtain the IP Address of one of the student’s lab partners and use the `ping` command to test connectivity between both machines. What devices did the `ping` packet have to go through to complete its path? __________________

c. Name resolution converts host names to IP address through the use of a name server or a hosts file on the student’s computer. If name resolution is available, `ping` the name of the student’s lab partner’s machine by its host name. What command was used? __________________ What was the response? __________________

Step 4. Use traceroute to check Internet path
Traceroute is another useful utility that is part of the basic TCP/IP protocol suite. Any computer running TCP/IP can use the `traceroute` command.

`traceroute hostname` or `traceroute [ip address]`

Traceroute is a good troubleshooting tool for checking the connection between computers that are interconnected with routers. Routers provide for connection of private networks and the public Internet. The `traceroute` command checks the time it takes for a packet to get from one router to the next and can help isolated slow links. Each router the packet goes through to reach its destination is listed in the command output and is referred to as a hop. As with `ping`, `traceroute` tests physical connections and IP addressing from one host to another on a TCP/IP network.

Each 'hop' is a physical machine that reads the packet address and forwards the address until it reaches its final destination.

a) Run the `traceroute` command on www.cisco.com. How many hops did it take to reach www.cisco.com from the student’s computer? __________________

Step 5. Telnet to another Host
Telnet is a client-server terminal emulation program that allows a user to connect to another system. The telnet server simulates a terminal in order to authorize a user or `telnet` client to connect to a remote system and work in that environment. When a user `telnets` to another host, the user is prompted for a name of an account on the remote system and password. Once a session is established from the user’s `telnet` client, the user can enter commands as if they were entering the commands directly on the server console. The user can telnet to several different hosts and have multiple telnet sessions opened simultaneously.

Telnetting to another host is helpful under the following circumstances:

- To access information on another workstation.
- To access the user’s workstation remotely to read mail.
- To kill a process that has caused the student or another user's workstation to hang.

Telnet is also used as a network-troubleshooting tool. It is a nongraphical communications utility that can be used to check the upper layers of the OSI model. It runs at layer 7, the Application layer. If a user is having trouble executing another, perhaps graphical, client-server application, the user can try to telnet to the host or server to verify that the TCP/IP protocol stack is functioning correctly. Remember, not all network operating systems support the telnet server function. For instance, the user can telnet from a
Users can telnet to the name of a host computer if the user has a naming service running such as Domain Name Service (DNS). A user can telnet if the user has entered the host name and IP address in their `/etc/hosts` file. If not, the users can `telnet` the IP address of the host. When using telnet, a user can:

- Open a session on a remote machine
- Alternate between the remote session and the local session
- Access machines that do not run under the UNIX environment

**Command Syntax:** `telnet host_ip_address` (or `hostname`)

a. If the student has a UNIX server in the classroom telnet to the IP address of the server. If the student does not have a UNIX server, go to exercise ‘d’. What command was used? __________________________ What was the response? __________________________

b. What command displays the name of the system the student is currently logged on to remotely? __________________________

c. Type `exit` or **Ctrl-d** to terminate the telnet session and return to the student’s local system.

d. Telnet to the student’s lab partner’s system using the student’s user ID or a guest account, if one was created. What is the student’s prompt now that the student is a remote terminal connected to the other system? __________________________

e. What command displays the name of the system the student is currently logged on to remotely? __________________________

f. Enter some UNIX commands such as `ls`, `id`, and `pwd` to see what directory the student is in on the remote machine. What directory is the student in? __________________________

g. Type a command and try the Backspace key. Does it work? __________________________ What can the student do to make the Backspace key work? **Hint:** see the Tip in the Telnet Utility section of Chapter 16. __________________________

h. Terminate the telnet session to the remote host. What was the response? __________________________

**Step 6. Remote Login to Another Host**

Use the `rlogin` command to establish a remote login session on another UNIX host, which can be a server or a workstation. Remotely logging in to another host is helpful under the following circumstances:

- To access information on another workstation.
- To access the user’s workstation remotely to read mail.
- To kill a process that has caused the student or another users workstation to hang.

The user can rlogin to any UNIX system provided the user knows the username and password. It is possible that a user can have the same named account on multiple systems that would allow the user to rlogin to another system using the same account name.
Use the -l option to specify a different login ID for the remote login session. The system administrator can set up a guest account so users can remotely log on to a server. Before attempting to remotely login to another system as a different user, be sure the student has an account on the desired remote machine.

**Command Format: rlogin hostname -l username**

a. If the student has a UNIX server in the classroom, rlogin to the hostname or IP address of the server. If the student does not have a classroom server, rlogin to the student’s partners system. What was the response? ________________________________

b. Issue the **id** command. What is the user id shown? ________________

c. Issue the **uname –n** command. What was the result? ________________

d. Issue the **pwd** command. What was the result? ________________

e. Terminate the student’s rlogin session to the remote host by typing exit or **Ctrl-d** at the command prompt. What was the response? ________________

**Step 7. Terminate a Local Process From a Remote Host**

Sometimes a user may start a process, either an application or command line, that causes the user’s system to ‘lock up’. The keyboard and mouse do not work and the user cannot open a terminal window to kill the process. In situations like this, it is best to first try to rlogin or telnet to the user’s computer from another system and attempt to kill the problem process before rebooting the system. After successfully killing the process that caused the student’s system to not respond, type exit to end the remote session then return to the student’s system.

If the student cannot identify the problem process that is locking up the student’s system then use the **pkill -9 ksh** (or **csh** if the student’s login shell is the C shell program) command. This will terminate the login session on the student’s system, including CDE, and return back to the CDE Login Manager.

For the next exercises the student will work with a partner. The first person will do steps a-d, then the second person will complete the same steps.

a. Start the Clock application so it is visible on the student’s screen.

b. The student should leave the machine and work with a partner to rlogin as the student to the student’s machine from the partner’s machine. Verify that the student is in the home directory by issuing the **pwd** command.

c. Terminate the process for the Clock application. What was the result? ________________________________

d. Remain remotely logged on to the student’s system and terminate the student’s login session as if the system was not responding. Did the student’s system return to the CDE Login Manager? ________________

e. Return to the student’s system and log back in.

**Step 8. Retrieve a file from Another Host with ftp**

The **ftp** (File Transfer Protocol) application is part of the TCP/IP protocol suite and is standard with the UNIX operating system. The **ftp** application can be used to transfer files using ASCII (for text files) or binary (for all other file types) mode between systems using similar or dissimilar operating systems. This provides a basic means of transferring files from one file system to another. Remember, if a user transfers a binary file using the default ASCII (text) mode the copied file will be useless.
Most computers running Servers with websites set up for downloading files sometimes provide an anonymous ftp account so users can pull files off the server. For this kind of an account, at the Name prompt, the word anonymous is entered instead of accepting the default displayed. If a password is required for the anonymous account, it will usually be the user’s full email address.

Command Syntax:   ftp host_ip_address

Or   ftp host_name   (if name resolution is available)

Once the student has successfully used ftp to access a remote host, some familiar file and directory access commands like cd and ls are available.

a. Use the ftp command to connect to the classroom server or the student's lab partner's system using the IP address. The student can use the machine name if name resolution is available. What was the response?

b. If the student has an account on the machine the student is going to ftp to, the student's login name will automatically be entered as the account Name and the student will be prompted for a password. If the student does not have an account on the remote system, the student will be prompted for a login name. Enter either another known account name on the remote system or anonymous for the login name and a password.

c. What is the prompt now? ________________

d. To view a list of FTP available commands, enter the ? at the ftp> prompt. What are some of the ftp commands available? ________________

e. Enter the command to determine what the current default directory is on the remote system. What is the current directory? ________________

f. If permissions are set by the site's system administrator for a user to see the contents of a directory the ls command will display files in that directory. Enter the ls command to get a listing of files. Identify a file to be copied to the student's machine and the file type. Is it a text file or another file type (binary)? ________________ What command was used to determine the file type? ________________

g. When the student copies a file from ('get') a remote system it is placed in the current directory of the local system. When the student copies a file to ('put') a remote system it is placed in the current directory of the remote system. Use the cd command to change directories on the remote system and the lcd (local change directory) command for the student's local system. Change to the practice directory on the student's local system so when the student copies the file from the remote system it will place the file in the practice directory. What command was used?

h. Set the student's file transfer mode depending on the type of file the student chose to copy. If the student's file is a text file what needs to be done? ________________ If the student's file is not a text file what needs to be done? ________________

i. Retrieve (copy) the identified file from the ftp host using the ftp get filename command. The file will be placed in the student's current working directory (~/practice) using the same name unless the student specify otherwise. What was the response? ________________

j. If the student has permissions, use the ftp put filename command to copy a file to the remote host.
k. If the student had set the transfer mode to binary and now wanted to copy a text file, what ftp command would the student use to set the transfer mode to ASCII? 

l. End the student ftp session and return to the student’s shell prompt. What ftp command was used? 

m. Verify the file the student copied from the remote system is in the student’s practice directory.

**Step 9. Remove Files and Directories Created in this Lab**
Remove all files and directories created in the student’s home directory during this lab.

**Step 10. 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.