

Student Name

Student number

Section

--	--	--

Objectives

- To review command line features, file system access, and permissions

Lab Outcome

- A review of working with the command line
- A review of basic Linux utilities

Additional Notes

- If you are uncertain as to how to proceed or have any problems, refer to class notes from previous for more information, your textbook, and the **man** pages.

In-Lab Demo: Create a new non-empty file with both a hard link and a soft link.

Exercise #1: Working with the find command

Hint: Use the **-mount** option to prevent searching mounted file systems like **/proc** and **/dev**.

Execute the command used to find all files from / with an extension of **cron**

```
[user1 @localhost ~] $ _____
```

```
-----
```

Execute the command used to find all files belonging to **user1**

```
[user1 @localhost ~] $ _____
```

```
-----
```

Execute the command used to find all files belonging to the user of uid 500.

```
[user1 @localhost ~] $ _____
```

```
-----
```

Execute the command used to find all files that have been modified within the last 3 days (there's a trick for past and future):

```
[user1 @localhost ~] $ _____
```

```
-----
```

Exercise #2: Working with links

Create a small non-empty file named **target**

```
[user1 @localhost ~] $ _____
```

Create a directory named **temp**.

```
[user1 @localhost ~] $ _____
```

Go into the **temp** directory.

```
[user1 @localhost ~] $ _____
```

Working with soft links

Create a soft link to the file **target** named **s-link**.

```
[user1 @localhost temp] $ _____
```

Test that you can read the file **target** using the soft link **s-link**.

```
[user1 @localhost temp] $ _____
```

Display both files, **target** and **s-link**, including all file attributes and the inode number.

```
[user1 @localhost temp] $ _____
```

```
[user1 @localhost temp] $ _____
```

What is the first difference between the two entries?

Change permissions of the file **target** to give no access to others.

```
[user1 @localhost temp] $ _____
```

Display both files, **target** and **s-link**, including all file attributes and the inode number.

```
[user1 @localhost temp] $ _____
```

```
[user1 @localhost temp] $ _____
```

What, if anything, has changed in **s-link**?

Rename the file **target** to **new-target**.

```
[user1 @localhost temp] $ _____
```

Test that you can read the file **new-target** using the soft link **s-link**.

```
[user1 @localhost temp] $ _____
```

Briefly explain why this behaves as it does:

Rename the file **new-target** to **target**.

```
[user1 @localhost temp] $ _____
```

Working with hard links

Create a hard link to the file **target** named **h-link**.

```
[user1 @localhost temp] $ _____
```

Test that you can read the file **target** using the hard link **h-link**.

```
[user1 @localhost temp] $ _____
```

Display both files, **target** and **h-link**, including all file attributes and the inode number.

```
[user1 @localhost temp] $ _____
```

```
[user1 @localhost temp] $ _____
```

Are the file attributes, except for the filename, identical? Record any difference

Change permissions on **target** to give no access to the group owners and full access to others.

[user1 @localhost temp] \$ -----

Display both files, **target** and **h-link**, including all file attributes and the inode number.

[user1 @localhost temp] \$ -----

[user1 @localhost temp] \$ -----

Are the file attributes, except for the filename, identical? Record any difference

Rename the file **target** to **n-target**.

[user1 @localhost temp] \$ -----

Test that you can read (**cat**) the file **n-target** using the hard link **h-link**.

[user1 @localhost temp] \$ -----

Remove the file **n-target.**, and both links **s-link** and **h-link**

[user1 @localhost temp] \$ -----

Exercise #3: Working with file permissions

Show the command and circle the practical minimum permissions required to successfully complete the actions listed below.

To produce a detailed directory listing (**ls -l**) within a directory, the user requires for that directory:

[user1 @localhost ~] \$ -----

R W X

To go into a directory, the user requires for that directory:

```
[user1 @localhost ~] $ _____  
  
R   W   X
```

To create a file or subdirectory into a directory, the user requires for that directory:

```
[user1 @localhost ~] $ _____  
  
R   W   X
```

Show the command and circle the practical minimum permissions required to successfully complete the file manipulation actions listed below.

To copy a file the user requires:

for the source directory:

```
[user1 @localhost ~] $ _____  
  
R   W   X
```

for the target directory:

```
[user1 @localhost ~] $ _____  
  
R   W   X
```

for the file:

```
[user1 @localhost ~] $ _____  
  
R   W   X
```

To move a file the user requires (assume the same filesystem for both source and target directory):

for the source directory:

```
[user1 @localhost ~] $ _____  
  
R   W   X
```

for the target directory:

```
[user1 @localhost ~] $ _____  
  
R   W   X
```

for the file:

```
[user1 @localhost ~] $ _____  
  
R   W   X
```

To move a file the user requires (assume different filesystems for source and target directory, perhaps a USB stick and the hard disk):

for the source directory:

```
[user1 @localhost ~] $ _____
```

```
    R    W    X
```

for the target directory:

```
[user1 @localhost ~] $ _____
```

```
    R    W    X
```

for the file:

```
[user1 @localhost ~] $ _____
```

```
    R    W    X
```

Briefly explain the differences between moving on the same and on different filesystems:

To delete a file the user requires:

for the directory:

```
[user1 @localhost ~] $ _____
```

```
    R    W    X
```

for the file:

```
[user1 @localhost ~] $ _____
```

```
    R    W    X
```

Exercise #4: Working with default permissions

Viewing default permissions

Login as **user1**.

Type **umask** and record the output of the command:

```
[user1 @localhost ~] $ _____
```

Based on the user file-creation mask as displayed by **umask**, determine the default permissions for directories and files in octal mode:

File: _____

Directory: _____

Verify it by creating a new file and displaying its long entry.

```
[user1 @localhost ~] $ _____
```

```
[user1 @localhost ~] $ _____
```

Record the default permissions set on the file in symbolic mode:

Record the default permissions set on the file in octal mode:

Verify it by creating and displaying a new directory.

```
[user1 @localhost ~] $ _____
```

```
[user1 @localhost ~] $ _____
```

Record the default permissions set on the directory in symbolic mode:

Record the default permissions set on the directory in octal mode:

Changing default permissions

Set the user file-creation mask to **077**.

```
[user1 @localhost ~] $ _____
```

Type **umask** and record the output of the command:

```
[user1 @localhost ~] $ umask
```

Based on the bitmask, as displayed by **umask**, determine the default permissions for directories and files in octal mode:

File: _____

Directory: _____

Verify it by creating a new file.

```
[user1 @localhost ~] $ _____
```

```
[user1 @localhost ~] $ _____
```

Record the default permissions set on the file in symbolic mode:

Record the default permissions set on the file in octal mode:

Verify it by creating a new directory.

```
[user1 @localhost ~] $ _____
```

```
[user1 @localhost ~] $ _____
```

Record the default permissions set on the directory in symbolic mode:

Record the default permissions set on the directory in octal mode:
