

Jackson State University
Department of Computer Science
CSC 438-01/539-01 Systems and Software Security, Spring 2014
Instructor: Dr. Natarajan Meghanathan

Project 1: Exploring UNIX Access Control in a Virtual Machine Environment

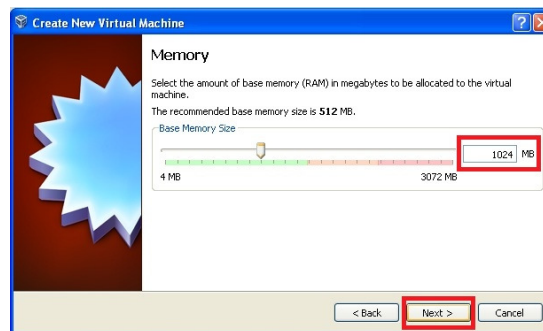
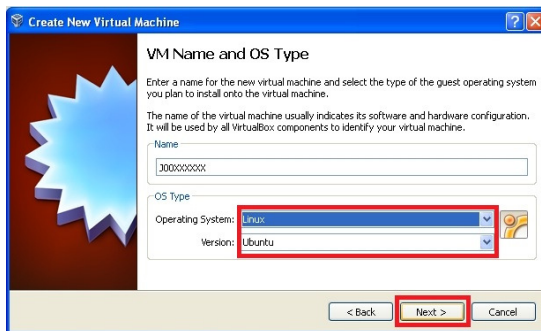
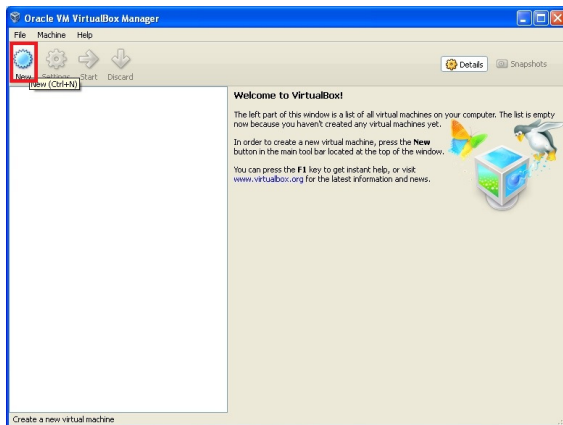
Due: February 26, 2014, 7.30 PM

The objective of this project is to explore the different UNIX access control commands and their features. You will do this project in a virtual machine environment. If you already have a virtual machine installed (either in VM Player or Virtual Box, you can skip the following steps and proceed to Page 4).

Installing VirtualBox 4.2 and Ubuntu OS

Go to <https://www.virtualbox.org/wiki/Downloads> and download VirtualBox for your operating system. If you work on a lab computer, you need to use the Ubuntu VM .iso file that is stored on the local machine. If you work on your personal computer, you need to download the Ubuntu .iso file from the website listed in Step # 1 and continue. You may use the following steps for installing the Ubuntu VM on the virtualbox.

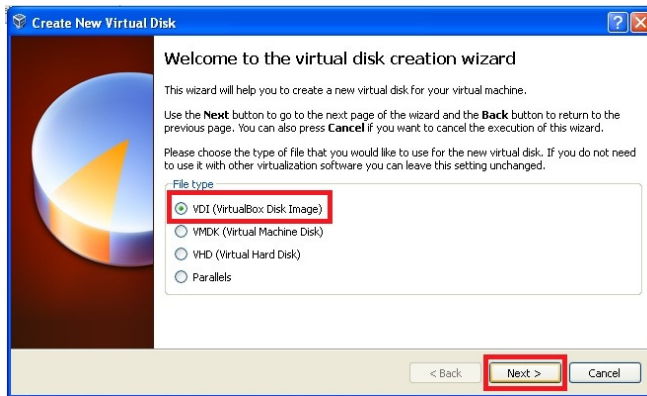
1. The Ubuntu installation file is located on the desktop of your PC (it can be downloaded from <http://www.ubuntu.com/download/ubuntu/download> if the .iso file cannot be located on your desktop).
2. On the VirtualBox Manager screen click on “New”



- When prompted, put your J # for the name of the VM and select “Linux” as OS (when you choose Linux as OS, the program should automatically choose Ubuntu as Version, if not select Ubuntu) and click Next.
- Set the RAM memory at 1024 MB on the following screen and click Next.
- On the following screen make sure that “Start-up disk” is selected. Check “Create new hard disk” if not selected.



- On the following screen select “VDI (VirtualBox Disk Image)” and click Next

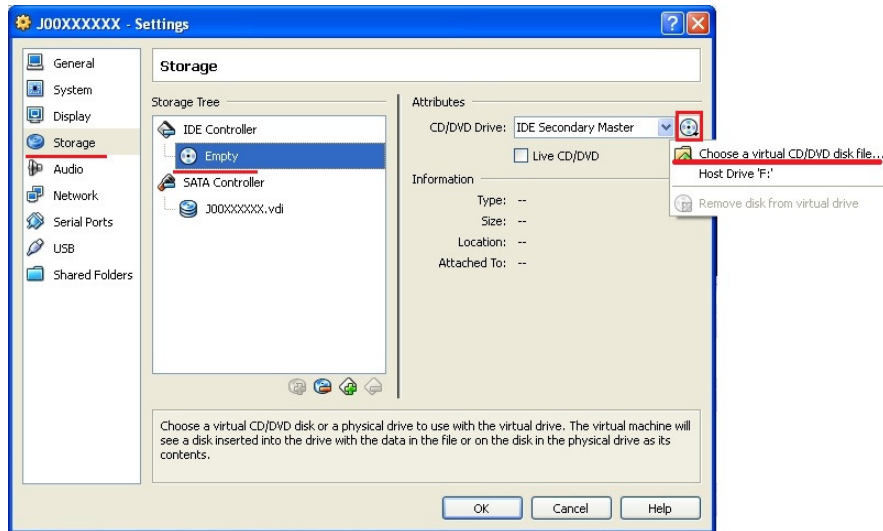


- On the next screen, make sure to select “Dynamically Allocated” and click Next (this option will enlarge the hard disk space as needed).



- Leave everything as it is on the next page and click Next.

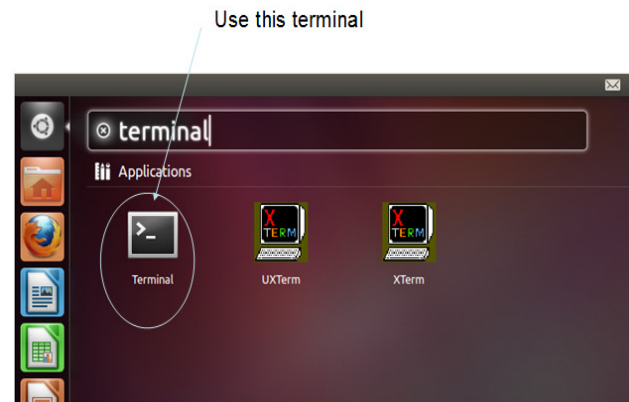
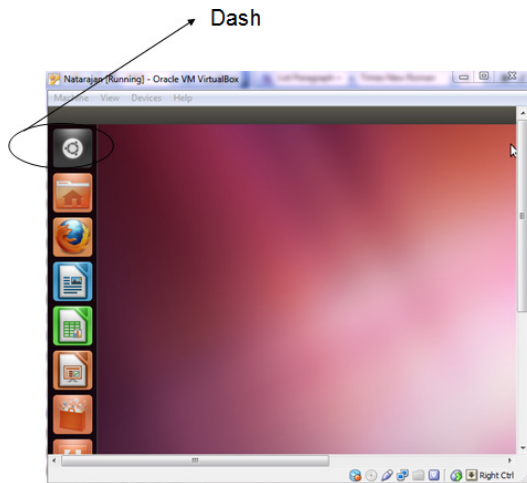
9. The next screen displays the information about the VM you are about to create; click “Create” twice and wait for VirtualBox to create your VM.
10. After your VM appears on VirtualBox Manager’s screen highlight the VM and click on “Settings.”
11. On the popped-up screen select “Storage.” In the Storage menu select “Empty” on Storage Tree under IDE Controller. When “Empty” is selected, click the circular disk object under Attributes and click on “Choose a virtual CD/DVD disk file.”



12. On the popped-up window show the location of the Ubuntu image file and click OK (by default the Ubuntu is on your desktop, if you have downloaded it from the website, show the path to the downloaded file accordingly).
13. The “Empty” under IDE Controller should have the name of the Ubuntu file, if you have done everything listed above. If it does click OK, otherwise redo steps 2 to 13.
14. Now, it is time to install the Linux OS on the VM. To be able to do that, highlight the VM and click on “Start” or simply double click the VM.
15. Click OK, on series of tips that pop-up. Wait till the Welcome window appears inside the VM (ignore the message in red before the Welcome page).
16. On the Welcome page select “Install Ubuntu”
17. On the next screen click “Forward”
18. On the next screen select “Erase disk and install Ubuntu” and click Forward (do not panic about the warning, the disk mentioned is allocated for VM, so no actual hard drive is erased).
19. Click “Install Now”
20. While installation process is going, you will be prompted to choose your time-zone. Just type the name of the city, and it should give you the choices from which you can select the appropriate one, and click Forward; on the keyboard layout leave everything as it is, and click Forward.
21. On the “Who are you?” page, type your initials and last 4 digits of your Student ID (J#) for “Your Name” box. The system should automatically fill your computer name and a user name. For your password type the last 4 digits of your Student ID (J#) and click “Forward.”
22. Wait until the virtual OS is installed, it may take some time, be patient.
23. After installation is complete, you will be prompted to restart your system. Click on Restart Now and then press any key as prompted (it only restarts VM)
24. In Login page click on your username and type your password (your username should be your initials and last 4 digits of your J#, and the password should be last 4 digits of your J#).
25. Ignore the pop-up and click Close.

Starting the Ubuntu Virtual Machine

Now, double click on the VM on the left side screen that you have got and keep clicking OK for all the messages that come (you can even make them not to show up from next time, if you wish so) and now the Ubuntu VM would have been loaded up (left screenshot below). Now open a terminal window by selecting load up the terminal window, by clicking the Dash icon (the top most icon in the window below, as indicated) and type 'terminal' on the search space that pops up. Click on the left most terminal icon (the pure black one) that shows up and launch the terminal to start working on your project.



Project 1 – Description of Tasks in Detail

The project description clearly describes some of the UNIX access control features and lists the tasks you are supposed to do. You are required to do all the tasks (1 through 22) on your Ubuntu VM.

What to submit:

Submit your project report as a hardcopy with the necessary screenshots for each task. Also, wherever required include the justification or your feedback for the questions asked under each task.

Basic commands

pwd command

After you open the terminal, enter the “pwd” command (pwd stands for present working directory) to find out the path for your current directory, which would also be the home directory, the directory to which you are logged in.

Task # 1: Capture the output of the *pwd* command in a screenshot and include it in your project report.

ls command

The *ls* command lists the files and sub-directories in the current directory. When used along with option –l (long format), we get a lot of useful information about the files and sub-directories.

touch command

The touch command can be used to create a file

chmod command

Task # 2:

Create a directory called *secPrj* using the *mkdir* command.

Use the *cd* command to change your current working directory to *secPrj*

Once you are in *secPrj*, create another directory by name *accessControl*

You need to create two text files in the *accessControl* directory with the following names that will include also the initials of your first name and last name. For example, I would create the files as follows where *n* stands as the initial for my first name and *m* stands as the initial for my last name.

- nm_proj_file1.txt
- nm_proj_file2.txt

Use an editor of your choice (vi, pico, etc) and enter the following text into the text files your created:

Text for nm_proj_file1.txt

```
This is file # 1 for our project
```

Text for nm_proj_file2.txt

```
We are creating another file, which would be our second file for this project
```

Task 3: Use the `ls -l` command to find what are the current permissions for the user, group and others with respect to the two text files `nm_proj_file1.txt` and `nm_proj_file2.txt`? Capture the output of the `ls -l` command in a screenshot and justify your answer based on the values displayed in the output

Altering a file's permissions

You can change the file's permissions using the `chmod` command in two ways. We will see each of them.

Method 1:

Followed by the command name (`chmod`),

- First specify which permissions you are changing: u for user, g for group and o for other.
- Second, specify how they should be changed: + (to add permission) or - (to subtract permission) to read, write or execute.
- Third, specify the file that the changes refer to.

Example:

To set the executable permissions for the user on file `nm_proj_file1.txt`, run the `chmod` command as follows:

```
natarajan@natarajan-VirtualBox:~/secPrj/accessControl$ chmod u+x nm_proj_file1.txt
natarajan@natarajan-VirtualBox:~/secPrj/accessControl$ ls -l
total 8
-rwxrw-r-- 1 natarajan natarajan 33 2012-09-20 09:22 nm_proj_file1.txt
-rw-rw-r-- 1 natarajan natarajan 78 2012-09-20 09:23 nm_proj_file2.txt
natarajan@natarajan-VirtualBox:~/secPrj/accessControl$
```

Now, to remove the write permission for the group from file `nm_proj_file1.txt`, run the `chmod` command as follows:

```
natarajan@natarajan-VirtualBox:~/secPrj/accessControl$ chmod g-w nm_proj_file1.txt
natarajan@natarajan-VirtualBox:~/secPrj/accessControl$ ls -l
total 8
-rwxr--r-- 1 natarajan natarajan 33 2012-09-20 09:22 nm_proj_file1.txt
-rw-rw-r-- 1 natarajan natarajan 78 2012-09-20 09:23 nm_proj_file2.txt
natarajan@natarajan-VirtualBox:~/secPrj/accessControl$
```

Now, to set write and execute permissions for both the group and others, run the `chmod` command as follows:

```
natarajan@natarajan-VirtualBox:~/secPrj/accessControl$ chmod go+wx nm_proj_file1.txt
natarajan@natarajan-VirtualBox:~/secPrj/accessControl$ ls -l
total 8
-rwxrwxrwx 1 natarajan natarajan 33 2012-09-20 09:22 nm_proj_file1.txt
-rw-rw-r-- 1 natarajan natarajan 78 2012-09-20 09:23 nm_proj_file2.txt
natarajan@natarajan-VirtualBox:~/secPrj/accessControl$
```

For the two text files you created in Task 2, do the following tasks sequentially. Use the chmod command according to the method described in this section. Capture the running of the chmod command for each task and then run ls -l to display the latest file permissions (as shown above in the examples) after running the chmod command:

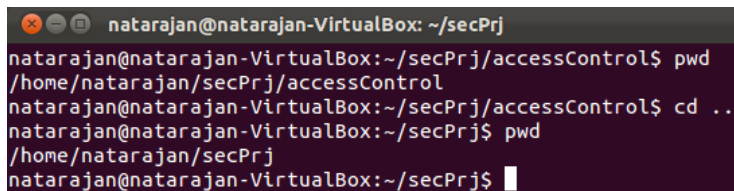
Task 4: Set the executable permission for all the three classes (user, group and others) on file nm_proj_file2.txt. You should do this using only one run of the chmod command.

Task 5: Set the write permission for the class others on both the files nm_proj_file1.txt and nm_proj_file2.txt. You should do this using only one run of the chmod command. Explore how to do this.

Task 6: Remove the write permission given for the group and others on file nm_proj_file2.txt. You should do this using only one run of the chmod command.

Task 7: Remove the read permission for others and write permission for the group on nm_proj_file1.txt. Can you do this using one run of the chmod command? Include screenshots of the outputs/error messages if you get any. If it is not possible to change using only one single run of the command, try to do the change using two runs of the chmod command.

Now, go one directory up, by using the cd .. command as shown below. You can always crosscheck where you are currently by using the pwd command.



```
natarajan@natarajan-VirtualBox: ~/secPrj
natarajan@natarajan-VirtualBox:~/secPrj/accessControl$ pwd
/home/natarajan/secPrj/accessControl
natarajan@natarajan-VirtualBox:~/secPrj/accessControl$ cd ..
natarajan@natarajan-VirtualBox:~/secPrj$ pwd
/home/natarajan/secPrj
natarajan@natarajan-VirtualBox:~/secPrj$
```

Similar to changing permissions for files, you can change permissions for directory using the chmod command. Do the following tasks sequentially. Use the chmod command described in this section. Capture the running of the chmod command for each task and then run ls -l to display the latest file permissions (as shown above in the examples) after running the chmod command:

Task 8: Using the ls -l command, find out what are the “read, write, permissions” for the user, group and others on the accessControl directory you had created earlier?

Task 9: Remove the “read” permission for the “user” on the accessControl directory. Justify your answer for each of the following questions, using screenshots:

- Can you use the ls -l command to list the contents of the secProj directory?
- Can you enter the accessControl directory using the cd command?
- Can you list the contents of the accessControl directory if you are able to successfully enter into the directory?

Task 10: Go back to the secProj folder. Add the “read” permission to “user”. Then, remove the “write” permission from “user”. Justify your answer for each of the following questions, using screenshots:

- Can you use the ls -l command to list the contents of the secProj directory?
- Can you enter the accessControl directory using the cd command?
- Can you list the contents of the accessControl directory if you are able to successfully enter into the directory?

- Can you open the file nm_proj_file1.txt (using an UNIX editor of your choice) and add the second line “This is another line of this file”, save it and return back to command prompt?
- Can you create a third file named nm_proj_file3.txt using the touch command?

Task 11: Go back to the secProj folder. Add the “write” permission to “user”. Then, remove the “execute” permission from “user”. Justify your answer for each of the following questions, using screenshots:

- Can you use the ls -l command to list the contents of the secProj directory?
- Can you enter the accessControl directory using the cd command?
- Create a file named nm_proj_file3.txt using the touch command in the secProj folder. Can you move this file from this folder to the accessControl folder using the mv command? If you are not sure of how to use the mv command, research on it and find out.

Setting permissions using a numeric (octal) code

Another form of the chmod commands lets the user to directly set the permissions, by using a numeric (octal) code to specify them. This code represents a file’s permissions by three octal digits: one for owner permissions, one for group permissions and one for other permissions.

The Access rights are: Read (r – 4), Write (w – 2) and Execute (x – 1), nothing (0).

Example: The following command sets all the three permissions to the user (4+2+1), read and execute permissions (4+1) to the group and read-only permission (4) for others.

```
chmod 754 A.txt
```

Task 12: Go back to the secPrj directory. Turn on the “write” permission for the accessControl directory using the chmod command with octal code explained in this section. You encounter a problem of what octal numbers should I put for the group and others. This is the drawback of this method of setting access permissions. We have to kind of know the current permissions for all the three classes and include them as part of the chmod command. So, use the ls -l command to find the current access permissions for group and others on the accessControl directory and then use an appropriate chmod command

Task 13: Change the access permissions to the file nm_proj_file3.txt as follows: Set read and write only permissions for the user, read and execute permission for the group and execute only permission for others

umask command

The chmod command allows you to alter the permissions on a file-by-file basis. The umask command allows you to do this automatically when you create any file or directory. Everyone has a default umask setting set up either by the system administrator or in their .profile file.

The umask command allows you to specify the permissions of all files created in the current login session after you issue the umask command. You basically specify the permissions that you want for your files by telling umask what you want to subtract from the full permissions value 777 (rwxrwxrwx).

For example, the following umask command ensures that all the files created after the execution of this command will have by default “read-write” access for users and “read-only” access for group and others. You can test whether the access permissions have been set appropriately by creating a file using the touch command and then run ls -l to list the contents of the folder.


```
natarajan@natarajan-VirtualBox:~/secPrj$ umask 133
natarajan@natarajan-VirtualBox:~/secPrj$ touch nm_proj_file3.txt
natarajan@natarajan-VirtualBox:~/secPrj$ ls -l
total 4
drwxrwxr-x 2 natarajan natarajan 4096 2012-09-20 09:22 accessControl
-rw-r--r-- 1 natarajan natarajan 0 2012-09-20 09:27 nm_proj_file3.txt
natarajan@natarajan-VirtualBox:~/secPrj$ █
```

Task 14: Go back to the SecPrj folder. Type umask and observe what value is displayed. Show a screenshot of your output. Explain what the umask value displayed means?

Note: You will actually get a 4-number output when you type the umask command. The first number would be always 0. So discard that. The second number is associated with permissions for user, the third number is associated with permissions for the group and the fourth number is associated with setting the permissions for others.

Task 15: Set the umask command in such a way that for files created in that session will have only the following permissions: “read-write” access for users, “write-only” access for the group and “read-only” access for others. Create a file named nm_proj_file4.txt using the touch command and find out the access permissions of this file by running the ls -l command.

Note: After finishing the above task, logout of your session and try to login again. After logging in again, check the umask value by typing the umask command. You would notice the default value that appeared initially before you started Task 14.

Links

When UNIX creates a file, it does two things. First, it sets aside space on the storage device to store data. Second, it creates a structure called an INDEX NODE or “I-node” to hold the basic information about the file. Typically, an I-node for a file stores the following information:

- Length of the file in bytes
- Name of device that contains the file
- Userid of owner
- Groupid
- File permissions
- Last modification time
- Last access time
- Last time inode was changed
- Number of links pointing to the file
- Type of file (ordinary, directory, special, symbolic link...)
- Number of blocks allocated to the file

The contents of an i-node can be looked at using the stat command.

Task 16: In the SecPrj folder, create a file called nm_proj_file5.txt using the touch command. Use an editor of your choice and enter the following message: “This is a test file for the stat command”. Then, run the stat command as follows:

```
stat nm_proj_file5.txt
```

Capture a screenshot of the output and include in your report. Answer the following questions:

- How many blocks have been allotted to the file?
- What is the size of each block?

Task 17: In the SecPrj folder create two folders AC1 and AC2. Create four text files in the order specified below, each using the touch command:

- In the AC1 folder, create a file by name “test-file1.txt”.
- In the AC2 folder, create a file by name “test-file2.txt”
- In the AC1 folder, create a file by name “test-file3.txt”
- In the AC2 folder, create a file by name “test-file4.txt”

The i-numbers of all the files and directories in the current directory can be seen by running the `ls -li` command.

Answer the following questions:

- (a) Do files test-file1.txt and test-file2.txt have consecutive I-numbers? Explain your observation.
- (b) Do files test2-file2.txt and test-file4.txt have consecutive I-numbers? Explain your observation.

Multiple links to the same file:

One of the most elegant features of the UNIX file system is that it allows multiple links to the same file. In other words, a file can be known by more than one name. The unique identifier for a file is its i-number and not its name.

Sometimes it is useful to have a file that is accessible from several directories, but is still only one file. This can reduce the amount of disk space used to store redundant information and can make it easier to maintain consistency in files used by several people.

For example, suppose you are working with someone else, and you need to share information contained in a single data file that each of you can update. Each of you needs to have easy access to the file, under a name that depends on the context in which each of you want to use the file. Also, you want any additions or changes one of you makes to be immediately available to the other.

Task 18: Create two directories AC3 and AC4 in the SecPrj directory. Enter to the AC3 directory. Create a text file by name test-file1.txt using the touch command and fill the file with a message of your choice. Then, enter the AC4 directory run the following link command

```
natarajan@natarajan-VirtualBox:~/secPrj/AC4$ ln ../AC3/test-file1.txt test-file2.txt
```

You are basically assigning another name, test-file2.txt, to the original file test-file1.txt. test-file2.txt in directory AC4 and test-file1.txt in directory AC3 are one and the same.

Run the `ls -li` command in the directories AC3 and AC4. Observe the i-numbers of these two files. What are they? Are they the same or different? Show the screenshots.

Task 19: Open the test-file2.txt created in task 18 using an editor of your choice and modify the contents of the file. Save it. Now open the file test-file1.txt. Are they the same? Show the screenshots.

Task 20: Do the following gone after the other:

- Go to the AC3 directory and find the access permissions assigned for the file test-file1.txt.
- Go to the AC4 directory and change the access permissions, as stated below, for the file test-file2.txt in AC4 by running the chmod command:
 - The user has “read-write-execute” access, the group has “read-write” access and others have “read-only” access.
- Go to the AC3 directory and find the access permissions assigned for the file test-file1.txt. Are the access permissions for test-file1.txt different or the same? Show the screenshots

Symbolic links:

What we saw so far are called “hard-links”. Hard links can be created only within a file system. To create a link to a file in a different file system, you need to create what is called a symbolic link. A symbolic link contains the pathname of the original file. Whenever you access a symbolic link, UNIX uses that pathname to find the file. We will now continue our work with the AC3 and AC4 folders created for tasks 18 through 20.

When you type the `ls -l` command, you can see the number of hard links to a file. This will appear next to the access permissions for the file.

Task 21: Do the following, one after another. Attach screenshots to justify your answers

- Run the `ls -l` command in directories AC3 and AC4. How many hard links do the files test-file1.txt and test-file2.txt have?
- Create a symbolic link by name test-file3.txt in AC4 for the test-file1.txt in AC3 as follows:
`ln -s test-file1.txt test-file3.txt`
- Can you display the contents of test-file3.txt? If so, what is that you get?
- Run the `ls -l` command in AC4 and show the output. Compare the sizes of the files test-file3.txt and test-file2.txt? Are they the same or different why?
- Delete the file test-file1.txt in AC3

Research on the difference between symbolic links and hard links, and justify what you observe below.

- Can you display the contents of test-file3.txt in AC3? If so, what is that you get?
- Run the `ls -l` command in AC3 and show the output.
- Run the `ls -l` command in AC4 and show the output.
- Can you display the contents of test-file2.txt in AC4? If so, what is that you get?

Shredding a file

When you delete a file, you may think there is no way to get it back. However, the actual disk space used by the file is not wiped clean. Rather, it is marked as being available for reuse by the file system. Eventually, the disk space will be reused and the old data will be overwritten by new data. On a large, busy UNIX system, this can happen within seconds. However, there is no guarantee when this will happen, and sometimes old data can stay within the unused part of a disk space for sometime. Indeed, there are special “undelete” tools that are able to look at the unused portion of a disk and recover the old data.

Moreover, even if data is overwritten, in extreme cases it is possible for the data to be recovered, as long as the data has not been overwritten more than once. If you can take a hard disk to a lab with very expensive data recovery equipment, it may be possible to sense traces of the old data on the magnetic surface of the disk. So, in general, if simple file deletion is not enough, we can use a program called `shred`. The syntax is

```
shred -fvuz [filename]
```

Note that you need not use any of the above f, v, u and z options and can simply shred a file. You can also use one or more of the above options.

The goal of shred is to overwrite existing data so many times that even the most expensive data recovery equipment in the world will feel foolish trying to read the magnetic traces. All you need to do is specify the names of one or more files and the *shred* command does the work automatically.

The `-v` option (verbose option) will display the messages as the *shred* command progresses.

By default, the shred command will overwrite the data many times and will leave the file with random data. Random data, of course, is a tipoff that the file has been “shredded”. To hide this, you can use the `-z` option, which tells shred to finish the job by filling the file with all zeros. Going further, if you want to delete the file after processing, use the `-u` option. Finally, to override restrictive file permissions, you can use the `-f` (force) option.

In the `secPrj` directory, make a directory called `test`. Inside the `test` directory, create four test files `test-file1.txt`, `test-file2.txt`, `test-file3.txt` and `test-file4.txt` using the `touch` command and fill each of the text files with at least 100 characters using an editor of your choice. Do the following one after the other:

Task 22:

In each step/case, justify why you see the appropriate file sizes.

- Run `shred -v test-file1.txt`. What is the output?
 - Run the `ls -l` command. What is the size of the file `test-file1.txt`? Why do you think the file has that size?
 - Open the `test-file1.txt` using an editor of your choice and just close the file. Save the file before closing. What is now the size of the file `test-file1.txt`?
 - Open the `test-file1.txt` again using an editor of your choice and just close the file. Again, save the file before closing. What is now the size of the file `test-file1.txt`?
 - Open the `test-file1.txt` again using an editor of your choice and just close the file. Now do not save the file before closing. What is now the size of the file `test-file1.txt`?
- Run `shred -z test-file2.txt`. Do you get any output?
 - Run the `ls -l` command. What is the size of the file `test-file2.txt`? Why do you think the file has that size?
 - Open the `test-file2.txt` using an editor of your choice and just close the file. What are the contents of the file `test-file2.txt`? Do not save the file before closing. What is now the size of the file `test-file1.txt`?
- Run `shred -u test-file3.txt`. Do you get any output?
 - Run the `ls -l` command. Do you see `test-file3.txt` listed in your test folder? Show the output.
- Use `chmod` command to make the `test-file4.txt` write protected for user, groups and others.
 - Run the `ls -l` command and show the output.
 - Run `shred -f test-file4.txt`.
 - Run the `ls -l` command and show the output.