System Administration (CSNB113) – Lab 4

Topics: Navigation. Searching. Permissions. Deleting. Globbing

Sec:

Student-ID:

Name:

This lab exercise is to be submitted **at the end** of the lab session!

Today we will obtain a number of functions helpful and relevant for the administration of the system. We start with navigation, changing directory, finding files, etc.

Go into directory /usr/share/doc/apt/examples/. Which **single** command do you use?

_____

List ('long list') all files in this directory whose file name starts with an 'a'. Of course, you could just *look* at them, and note those starting with an 'a' here.
Though you could as well use ***globbing*** (see lecture slides), and type:

```
ls -l [a]*
```

_____

_____

From there, **change** into **directory** man8 by writing the **relative path** (see lecture slides). The ***target*** directory is actually /usr/share/man/man8/; so from
/usr/share/doc/apt/examples/
you need to
- go up three times (/examples, /apt, /doc)
using ../ three times, and then - go down to man/ and /man8

Enter the correct command here:

_____

Confirm that you are in the right **target** directory, **after** changing directories, using **`pwd:`**

_____

Note the details of the **last** file that **`ls -l`** shows in this directory:

_____

Go back to your home directory. You use which command?:

_____

Then go into that same directory, /usr/share/man/man8/, using its **absolute path**. The **command** is:

_____

Note the name(s) of the file(s) whose file name starts with an upper-case M: (If it is not visible, try globbing for it!)

_____

You have probably noticed that the files have by default been sorted in **alphabetic** order. It is possible to sort them differently. If you need them sorted in terms of creation / last modification?

Read the manual pages of the command 'ls': **`man ls`**, and find the correct **option** for 'sort by modification time' (**not** 'time of last modification of file status information', please!). You can quit (leave - exit) any man page with '**`q`**'.
Run 'ls -l' another time, and add the option that you found in the man page. Which is the **oldest** file in that directory?

_____

==========

*There is a nice trick, loved by system administrators, who have to type things over and over. It is called **Tab-Completion**. We will experience this one in the next part of this exercise.*

What is the content of the file /usr/share/doc/apt/examples/sources.list? The usual command would be:

```
cat usr/share/doc/apt/examples/sources.list
```

That's ***looong*** and some parts are redundant. Locate the **Tab-key** on your keyboard and use it, whenever I typed **`<tab>`** in the following:
`cat /u <tab> sh <tab> do <tab> /apt/e <tab> s <tab>`
and press 'Enter'
You will notice that the tab key will complete the **path** and the **filename**, as long as they are unambiguous.
Try again:
`cat /u <tab> s <tab>`
This can't complete, because there are a number of possible targets.

However, if you press <tab> twice (or more often), all the possibilities will be displayed, helping you to get further. Try it out:
`cat /u <tab> /s <tab> <tab>` shows the three alternatives: sbin/ share/ src/.

Once you type another 'b', it will autocomplete to /usr/sbin:
`cat /u <tab> s <tab> <tab> b <tab>`

Tab-Completion works well for paths and filenames, but also for (system) commands:
On the prompt, type
`b <tab> <tab>`
and you see all commands starting with 'b'.
Try
`i <tab> <tab>`
for all commands starting with 'i'.

Which commands are available that start with the letters 'time'?

_____

==========

System Administration (CSNB113) – Lab 4

Now we will experience some action of files and directories, like creating and deleting files and directories.

Go back to your home directory.
Create a new directory, called explore: **`mkdir explore`**
Change into that directory:
`cd explore`
As to be expected, there are no files in this directory. Let us create some files. We noticed earlier, that touch is a possibility. Therefore, type
`touch file1`
`touch file2.txt`
`touch file3.conf`
This is okay, but boring, because all files are empty:
`ls -l`
shows that all sizes are '0'.

This is a good moment, to introduce another nice little trick: Redirection. Normally, out is displayed on the monitor. And input is read from mouse/keyboard. Let us show, how output can be redirected to a file, to a printer, or any other device. It is also possible, to read input from a file instead of from a keyboard; though this will be done later only.

When you type
`date`
the date will be displayed on the screen.
Now try
`date > demo`
`ls -l`
`demo cat`
`demo`
What do these do? You might have already guessed it, it **redirects** the output into a **file**, with the name of 'demo'.

Do not confuse it with our earlier 'pipe', or '|'.
*The pipe takes the output and sends it to another **command**; as input for that next command.*

The redirection takes the output and sends it to a device, a file, etc.

Create another two files in the directory 'explore'; one with the name 'host_name' containing the hostname, and one with the name 'who_am_i' containing your user name.
Refer to our earlier exercise to retrieve the commands.
Use the concept of redirection and its operator '>' here, as well.
Lastly, add a file to the directory that contains all the files in the directory:
`ls -l > file_list`
Confirm with **`'cat file_list'`** that it actually does contain all the files.
List all files ('long list') here:

_____

_____

_____

==========

Next we will look at permissions and ownerships.

File 'file1' is noted with the permissions _____

This represents which 3 -digit number?:

_____

Which command is needed to change the permissions to `rwxr-xr--` ? (Fill in the 3 digits)

chmod _____ file1

Change the owner to root:
chown root file1

_____

Why would this not work? Which command would you have to add to make it work?:

_____

Change the group to operator:
chgrp operator file1

Why would this not work? Which command would you have to add to make it work?:

_____

file1 now looks like this here (ls -l, please!):

_____

Now let's start deleting some stuff.
Go back to your home directory, then pwd as usual, and ls -l shows the **d**irectory explore:

_____

You can clearly see the 'd' at the start of the line, at the very left. It shows that it is a **d**irectory.

We are going to delete it:
rmdir explore

_____

Oops, that didn't work. There are still files in that directory. Let's use the recursion to do that:
rm -R explore [answer 'no' when asked]

_____

_____

Still not. Why? There are some files belonging to another user and another group (file1,remember, belonging to root and operator). We need to be more brutal ('f' ==force):

```
rm -Rf explore
```

Use ls -l to prove that the directory and all its files are gone.

Do not forget to shut down your machine; like we did at the end of lab exercise 1, close Virtualbox, and log out.

It is still suggested that you export your installation after the shutdown for reasons of security. Don't forget to submit your lab sheet!