Week 10 unassessed exercise

The current lab exercise is meant to be done on a UNIX or Linux machine. If you don’t have access to a UNIX or Linux machine yourself, you have the following options:

1. If you’re on a Windows 10 or Windows 11 machine, you may want to install (for free) the Windows Subsystem for Linux (WSL2) which gives you access to a Linux shell.
2. If you’re on a Mac, you can use the (pre-installed) MacOS terminal, which is in essence a UNIX shell that is broadly compatible with Linux.
3. Go to the COMSC Linux lab (Abacws room 5.44) and use one of the Linux PCs over there.
4. Do a remote login to one of the machines in the Linux lab, using the Apache Guacamole server. Detailed instructions can be found at https://wiki.cs.cf.ac.uk/index.php?title=Remote_Desktop_access_to_the_Linux_Lab Please select SSH (Secure Shell) instead of RDP (Remote Desktop Protocol) as you only need a shell, not an entire desktop. Also, please make sure you’re on the VPN.

The idea of the current lab exercise is to be able to understand what the various UNIX/Linux shell commands are doing. This means that at some points, you will be asked to predict what will be the output of the commands. If you can correctly predict this, then you're starting to understand how UNIX/Linux works from a user perspective (so don't rush through this too fast).

Once you have managed to get access to a UNIX/Linux shell (using one of the four methods mentioned above) enter the following commands (don't enter the starting '$' as this represents your shell prompt)

```
$ mkdir dir1
$ mkdir dir2
$ cd dir1
```

Try to predict the output of the following command before entering it

```
$ pwd
```

Your current working directory is empty. Try to predict the output of the following command

```
$ ls -al
```

Now create a new textfile

```
$ touch myfile.txt
```

See if the new file shows up in the directory

```
$ ls -al
```

The new file is empty, so when you enter the following command (to display its contents) should not display anything

```
$ cat myfile.txt
```

Let's write some text into the file

```
$ echo first line >> myfile.txt
$ echo second line >> myfile.txt
```

Let's go to dir2 and create a copy, hard link and symbolic link of myfile.txt

```
$ cd ../dir2
$ cp ../dir1/myfile.txt copy.txt
$ ln ../dir1/myfile.txt hardlink.txt
$ ln -s ../dir1/myfile.txt softlink.txt
```

Check that hardlink.txt has indeed the same inode as myfile.txt

```
$ ls -il
$ ls -il ../dir1
```

Write an additional line of text in the original file

```
$ echo third line >> ../dir1/myfile.txt
```

Can you already predict the output of the following commands?

```
$ cat ../dir1/myfile.txt
$ cat copy.txt
$ cat hardlink.txt
$ cat softlink.txt
```

Let's go back to dir1

```
$ cd ../dir1
```

Let's display the file permissions of myfile.txt and then change them

```
$ ls -l myfile.txt
$ chmod 666 myfile.txt
```

Can you already predict what the following command will display when it comes to the file permissions?

```
$ ls -l myfile.txt
```

Change the file permissions again. What does the following command do?

```
$ chmod o-r,o-w myfile.txt
```

Can you already predict what the following command will display when it comes to the file permissions?

```
$ ls -l myfile.txt
```

Go up to the parent directory

```
$ cd ..
```

Find all files in the directory tree below that end with ".txt"

```
$ find . -name '*.txt'
```

Go to the dir2 directory

```
$ cd dir2
```

Which of the text files contains the word "line"

```
$ grep line *.txt
```

Still remember the difference between copy.txt and link.txt?
Then what will be the output of the following command?

```
$ diff copy.txt hardlink.txt
```

How many lines, words and characters do copy.txt and hardlink.txt contain?

```
$ wc copy.txt
$ wc hardlink.txt
```

Let's remove the file that was used to create both hardlink.txt and softlink.txt

```
$ rm ../dir1/myfile.txt
```

One of the following commands will create an error. Can you predict which one?

```
$ cat hardlink.txt
$ cat softlink.txt
```

We're now at the end of the practical, so let's clean up by recursively removing the two directories and all of its contents

```
$ cd ..
$ rm -r dir1 dir2
```

To finish the practical, logout from the shell

```
$ exit
```