Introduction to toolbox: Linux and Git

The objective of this lab is getting to know Linux command line and learn how to backup files using git.

Using command line in Linux

Open terminal window.

ls (list)

When you first login, your current working directory is your home directory. Your home directory has the same name as your user-name, and it is where your personal files and subdirectories are saved.

To find out what is in your home directory, type

```
ls
```

The **ls** command (lowercase L and lowercase S) lists the contents of your current working directory.

ls does not, in fact, cause all the files in your home directory to be listed, but only those ones whose name does not begin with a dot (.) Files beginning with a dot (.) are known as hidden files and usually contain important program configuration information. They are hidden because you should not change them unless you are very familiar with UNIX!!!

To list all files in your home directory including those whose names begin with a dot, type

```
ls -a
```

As you can see, **ls -a** lists files that are normally hidden.

ls is an example of a command which can take options: **-a** is an example of an option. The options change the behaviour of the command. There are online manual pages that tell you which options a particular command can take, and how each option modifies the behaviour of the command.

Can you understand what changes when you type

```
ls –la
```

mkdir (make directory)

We will now make a subdirectory in your home directory to hold the files you will be creating and using in the course of this tutorial. To make a subdirectory called mystuff in your current working directory type

```
mkdir mystuff
```

To see the directory you have just created, type

```
ls
```

or

```
ls –la
```

cd (change directory)

The command cd directory means change the current working directory to 'directory'. The current working directory may be thought of as the directory you are in, i.e. your current position in the file-system tree.

To enter the directory you have just made, type

```
cd mystuff
```

Type **ls** to see the contents (which should be empty)

Make another directory inside the **mystuff** directory called **backups**

Still in the mystuff directory, type

```
ls -a
```

As you can see, in the **mystuff** directory (and in all other directories), there are two special directories called (.) and (..)

The current directory (.)

In UNIX, (**.**) means the current directory, so typing

```
cd .
```

means stay where you are (the **mystuff** directory).

This may not seem very useful at first, but using (.) as the name of the current directory will save a lot of typing, as we shall see later in the tutorial.

The parent directory (..)

(**..**) means the parent of the current directory, so typing

```
cd ..
```

will take you one directory up the hierarchy (back to your home directory). Try it now.

Note: typing **cd** with no argument always returns you to your home directory. This is very useful if you are lost in the file system.

pwd (print working directory)

Pathnames enable you to work out where you are in relation to the whole file-system. For example, to find out the absolute pathname of your current directory, type

```
pwd
```

Understanding pathnames

First type **cd** to get back to your home-directory, then type

```
ls mystuff
```
to list the contents of your **mystuff** directory.

Now type

```
ls backups
```
You will get a message like this -

```
backups: No such file or directory
```

The reason is, backups is not in your current working directory. To use a command on a file (or directory) not in the current working directory (the directory you are currently in), you must either **cd** to the correct directory, or specify its full pathname. To list the contents of your backups directory, you must type

```
ls mystuff/backups
```

~ (your home directory)

Home directories can also be referred to by the tilde **~** character. It can be used to specify paths starting at your home directory. So typing

```
ls ~/mystuff
```
will list the contents of your **mystuff** directory, no matter where you currently are in the file system.

What do you think

```
ls ~
```
would list?

What do you think

```
ls ~/..
```
would list?

Let's recap

Command	Meaning
ls	list files and directories
ls -a	list all files and directories
mkdir	make a directory
cd *directoryname*	change to named directory
cd	change to home-directory
cd ~	change to home-directory
cd ..	change to parent directory
pwd	display the path of the current directory

cp (copy)

`cp file1 file2` is the command which makes a copy of **file1** in the current working directory and calls it **file2**

What we are going to do now, is to take a file stored in your user-directory, and use the **`cp`** command to copy it to your **mystuff** directory.

First, **`cd`** to your **mystuff** directory.

```
cd ~/mystuff
```

Then type

```
cp ~/license.txt .
```

Note: Don't forget the dot . at the end. Remember, in UNIX, the dot means the current directory.

Create a backup of your **license.txt** file by copying it to a file called **license.bak**

mv (move)

`mv file1 file2` moves (or renames) **file1** to **file2**

To move a file from one place to another, use the **`mv`** command. This has the effect of moving rather than copying the file, so you end up with only one file rather than two.

It can also be used to rename a file, by moving the file to the same directory, but giving it a different name.

We are now going to move the file **license.bak** to your backup directory.

First, change directories to your **mystuff** directory. Then, inside the **mystuff** directory, type

```
mv license.bak backups/.
```

Type **`ls`** and **`ls backups`** to see if it has worked.

rm (remove), rmdir (remove directory)

To delete (remove) a file, use the **`rm`** command. As an example, we are going to create a copy of the **license.txt** file then delete it.

Inside your **mystuff** directory, type

```
cp license.txt tempfile.txt
ls
rm tempfile.txt
ls
```

You can use the **`rmdir`** command to remove a directory (make sure it is empty first). Try to remove the backups directory. You will not be able to since UNIX will not let you remove a non-empty directory.

Create a directory called **tempstuff** using **`mkdir`**, then remove it using the **`rmdir`** command.

clear (clear screen)

Before you start the next section, you may like to clear the terminal window of the previous commands so the output of the following commands can be clearly understood.

At the prompt, type

```
clear
```

This will clear all text and leave you with the prompt at the top of the window.

cat (concatenate)

The command cat can be used to display the contents of a file on the screen. Type:

```
cat license.txt
```

As you can see, the file is longer than the size of the window, so it scrolls past making it unreadable.

less

The command less writes the contents of a file onto the screen a page at a time. Type

```
less license.txt
```

Press the [space-bar] if you want to see another page, and type [q] if you want to quit reading. As you can see, **`less`** is used in preference to **`cat`** for long files.

Simple searching using less

Using **`less`**, you can search though a text file for a keyword (pattern). For example, to search through **license.txt** for the word 'source', type

```
less license.txt
```

then, still in **`less`**, type a forward slash [**/**] followed by the word to search

```
/source
```

As you can see, less finds and highlights the keyword. Type [**n**] to search for the next occurrence of the word.

grep (don't ask why it is called grep)

`grep` is one of many standard UNIX utilities. It searches files for specified words or patterns. First clear the screen, then type

```
grep source license.txt
```

As you can see, **`grep`** has printed out each line containg the word source.

Or has it ????

Try typing

```
grep Source license.txt
```

The **grep** command is case sensitive; it distinguishes between Source and source.

To ignore upper/lower case distinctions, use the **-i** option, i.e. type

```
grep –i source license.txt
```

To search for a phrase or pattern, you must enclose it in single quotes (the apostrophe symbol). For example to search for terms and conditions, type

```
grep –i 'terms and conditions' license.txt
```

Some of the other options of **grep** are:

-v display those lines that do NOT match
-n precede each matching line with the line number
-c print only the total count of matched lines

Try some of them and see the different results. Don't forget, you can use more than one option at a time. For example, the number of lines without the words source or Source is

```
grep –ivc source license.txt
```

The following adds line numbers to the printout of the search results for 'terms and conditions'

```
grep –in 'terms and conditions' license.txt
```

Let's recap

Command	Meaning
cp *filename1 filename2*	copy file1 and call it file2
mv *filename1 filename2*	move or rename file1 to file2
rm *filename*	remove a file
rmdir *directoryname*	remove a directory
cat *filename*	display a file
less *filename*	display a file a page at a time
grep 'keyword' *filename*	search a file for keywords

Reference

The above tutorial is based on the "UNIX Tutorial for Beginners", available at:
http://www.ee.surrey.ac.uk/Teaching/Unix/ and licensed under a Creative Commons License.

USING GIT AND CLOUD STORAGE

Create a user account with either **github.com** or **bitbucket.org** (if you do not already have one).

Using the web page, create a new repository and name it ***<your_first_name>-rtech*** and initialize it online. **This repository will be the main backup for your work throughout this course.**

Find out what the web address for your new repository is.

Open Linux terminal and go to your home directory.

Then type

```
git clone URL_for_your_repository
```

Use **`ls`** to confirm that ***<your_first_name>-rtech*** has been downloaded to your home directory**.**

Type **`gedit`** to open a text editor. Write something in the file and save it in the ***<your_first_name>-rtech*** directory.

Close **`gedit`**.

In terminal, **`cd`** to ***<your_first_name>-rtech***

Type

```
git config user.email "youremail@example.com"
```

Type

```
git status
```

You should now see the newly created text file in red.

Using that file name type

```
git add file_name_in_red
```

When you now type

```
git status
```

you should see the same file name but in green. You are now ready to commit this file.

Type

```
git commit –m "Learning git"
```

Your file has now been committed but not yet uploaded to cloud. To upload your files, type

```
git push
```

In your web browser, **verify that the file** has been uploaded to the ***<your_first_name>-rtech*** repo.

>>> Show the result to your lab instructor! <<<

Delete the **<your_first_name>-rtech** and **mystuff** directories from the lab computer.