**Kasia Sawicka, August 2018**

**Basic instructions to work on Linux from Windows, deal with Fortran on Linux, and use git from Linux.**

**Version 1.**

This tutorial is aimed at people who have been primarily working on Windows machines and want to start working in Linux using their Windows machines, including working with Fortran code and git.

Contents

[Linux 1](#_Toc522090685)

[1. Connecting to Linux from Windows machine. 1](#_Toc522090686)

[Requirements 1](#_Toc522090687)

[MobaXterm 1](#_Toc522090688)

[Connecting to and navigating in Linux 4](#_Toc522090689)

[2. Fortran on Linux 7](#_Toc522090690)

[Editing files on Linux 7](#_Toc522090691)

[Windows 8](#_Toc522090692)

Linux

1. Connecting to Linux from Windows machine.

**Requirements**

1. A program with an SSH terminal, such as MobaXterm. MobaXterm is portable, so it doesn't require installation. Other programs that can be used include Putty or NX Machine and the workflow is the same/ or very similar for all of them, because they all are usable with Linux commands. MobaXterm is available on the CEH Y: drive.
2. An access to a CEH Linux server. Andrew Everitt set up users Linux accounts for CEH staff. Contact him first if you need your account to be set up.

**MobaXterm**

Navigation in MobaXterm is done by Linux commands, which are similar to the Windows CMD/DOS commands. See the Table 1 comparison below. <http://linuxcommand.org/> is an example of an online tutorial to get a head around navigation and operations using commands.

Table Comparison of Windows CMD/DOS and Linux commands

| **Command's Purpose** | **MS-DOS** | **Linux** | **Basic Linux Example** |
| --- | --- | --- | --- |
| Copies files | copy | cp | cp *thisfile.txt* /home/*thisdirectory* |
| Moves files | move | mv | mv *thisfile.txt* /home/*thisdirectory* |
| Lists files | dir | ls | ls |
| Clears screen | cls | clear | clear |
| Closes shell prompt | exit | exit | exit |
| Displays or sets date | date | date | date |
| Deletes files | del | rm | rm *thisfile.txt* |
| "Echoes" output to the screen | echo | echo | echo *this message* |
| Edits text files | edit | gedit([[a]](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Step_by_Step_Guide/ap-doslinux.html" \l "FTN.AEN8296)) | gedit *thisfile.txt* |
| Compares the contents of files | fc | diff | diff *file1* *file2* |
| Finds a string of text in a file | find | grep | grep *word or phrase* *thisfile.txt* |
| Formats a diskette | format a: (if diskette is in A:) | mke2fs | /sbin/mke2fs /dev/fd0 (/dev/fd0 is the Linux equivalent of A:) |
| Displays command help | *command* /? | man or info | man *command* |
| Creates a directory | mkdir | mkdir | mkdir *directory* |
| Views contents of a file | more | less([[b]](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Step_by_Step_Guide/ap-doslinux.html" \l "FTN.AEN8362)) | less *thisfile.txt* |
| Renames a file | ren | mv([[c]](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Step_by_Step_Guide/ap-doslinux.html" \l "FTN.AEN8374)) | mv *thisfile.txt* *thatfile.txt* |
| Displays your location in the file system | chdir | pwd | pwd |
| Changes directories with a specified path (*absolute path*) | cd *pathname* | cd *pathname* | cd */directory/directory* |
| Changes directories with a *relative path* | cd.. | cd .. | cd .. |
| Displays the time | time | date | date |
| Shows amount of RAM in use | mem | free | free |

Usually, when you open MobaXterm and start a local terminal, the initial location is set to MobaXterm home directory that as a default is in your user folder on C: C:\Users\[USER]\AppData\Local\Temp\Mxt102\tmp\home\_[USER] - see Figure 1. You can test that using following commands:

# view path to current location

pwd

# open directory in Explorator

open /home/mobaxterm

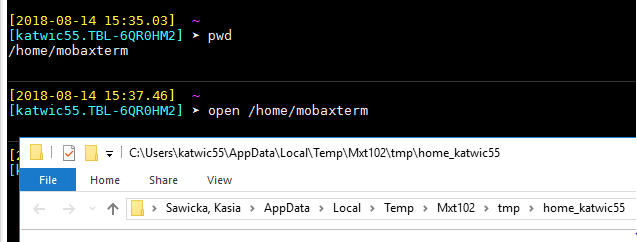


Figure Location of MobaXterm local home directory.

From here using the Linux commands you can navigate through your local disk, e.g. go to C: and look up what’s there. Type in:

# change location to Windows c: drive

cd c:

# view details list

ll

ll command is an alternative to ls and shows a list of objects in the disk/directory with additional details – see Figure 2.

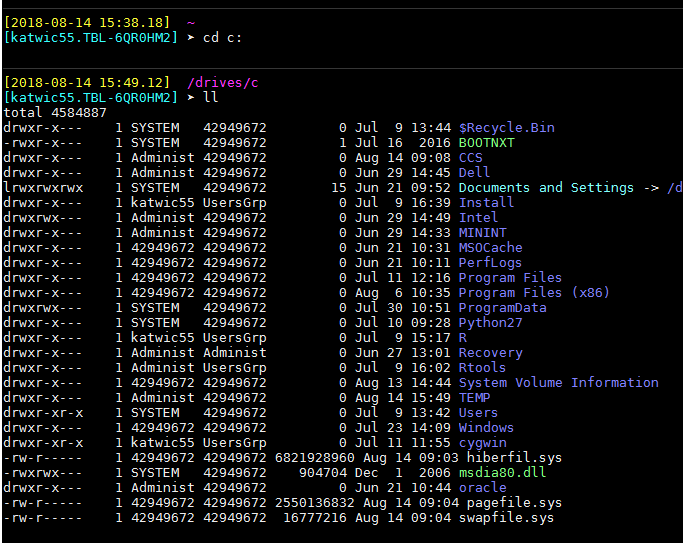


Figure Navigating to and viewing content of a disk/directory.

**Connecting to and navigating in Linux**

At the CEH, Andrew Everitt is a person who can grant access to the CEH Linux servers.

Once you have your account set up, Linux can be accessed via ssh (Secure Shell - a cryptographic network protocol for operating network services securely over an unsecured network). The following systems are seen as general use ones, so you can log into one of these.

* wllf001.nwl.ac.uk
* wllf002.nwl.ac.uk
* wllf003.nwl.ac.uk
* wllf004.nwl.ac.uk
* wllf005.nwl.ac.uk
* wllf013.nwl.ac.uk

To connect to one of the servers above, e.g. the first one type in following command:

# connect to wllf001.nwl.ac.uk server via ssh

ssh wllf001.nwl.ac.uk

You will be prompted to type in your earlier set password.

MobaXterm provides Explorator-like directories tree for the Linux ones, to the left of its window. Here you can see your current location, e.g. Figure 3. However we will be mostly working with commands and so let’s see the same using pwd command – Figure 4. For me it is /users/emmet/katwic55 where users is a high-level directory and emmet and katwic55 are directories named after CEH (old) section and your CEH user name (mine is katwic55).

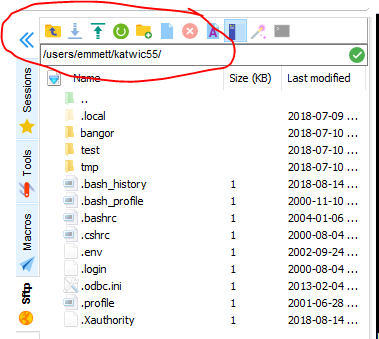


Figure Location of a local user directory on Linux.

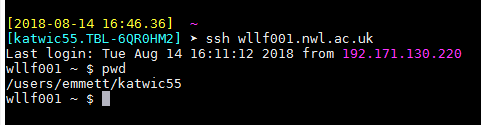


Figure Location of a local user directory on Linux.

Linux file system structure is different to Windows and the high-level directories structure is shown in Figure 5. These directories have always the same names and each Linux system will have them (like each standard installation of Windows creates C: drive). Figure 6 shows these directories on the server we are logged into. We will be only working in /usr/ (or users as looks like CEH changed that name) directory and we don’t have to worry about what is in the other ones unless we are the system administrators.



Figure Linux file system structures and meaning of individual high-level directories, from <https://www.thegeekstuff.com/2010/09/linux-file-system-structure/> (accessed on 14.08.18).

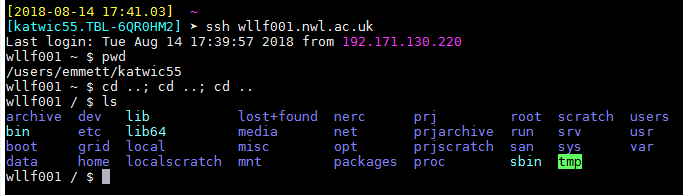


Figure High-level directories on the CEH Linux server.

There are three ways a Windows and Linux directories can communicate (can be accessed from one to another).

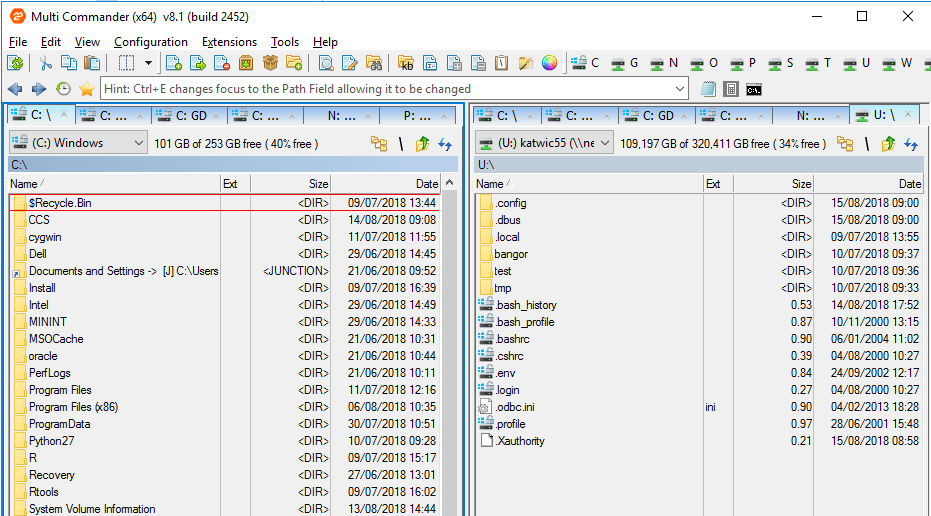
1. Mount your Linux home directory (your user Linux directory) from your Windows machine – this allows to access your Linux files in Windows Explorator, etc – very easy and useful.
2. Use WinSCP program, which is a file manager that allows you to – also very easy and useful, has similar functionality to no 1.
3. Mount your local computer drives from Linux – this allows to access your local computer files from Linux terminal – we will not be using that.

Mounting your Linux home directory from your local Windows computer is done using Windows CMD. The command needed to mount Wallingford's Linux home directory from is:

# mount Linux directory in Windows

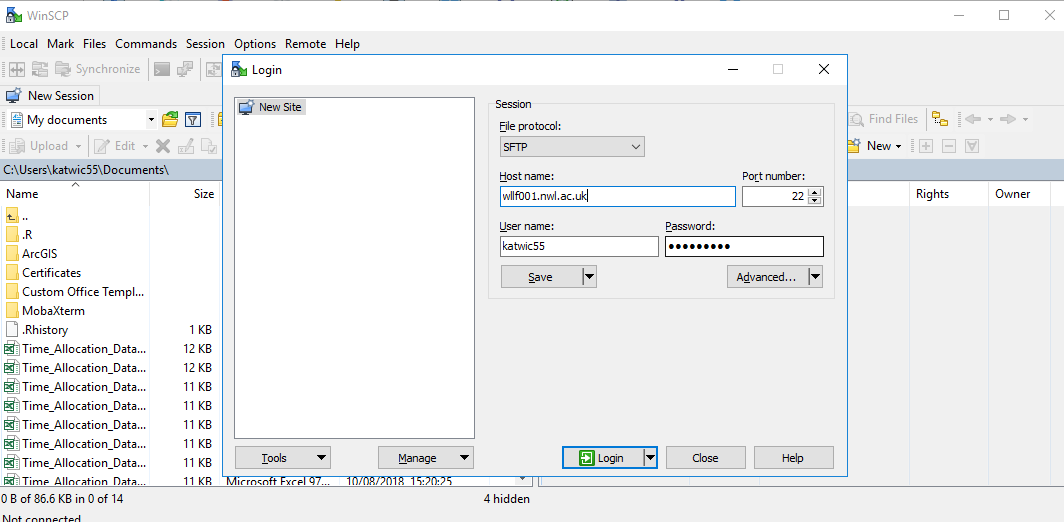
net use u: \\nercwlsmb01.adceh.ceh.ac.uk\[username] /persistent:no

This will result in a new drive U: being visible from a Windows file manager and managing your files from there. See Figure 7 for an example.

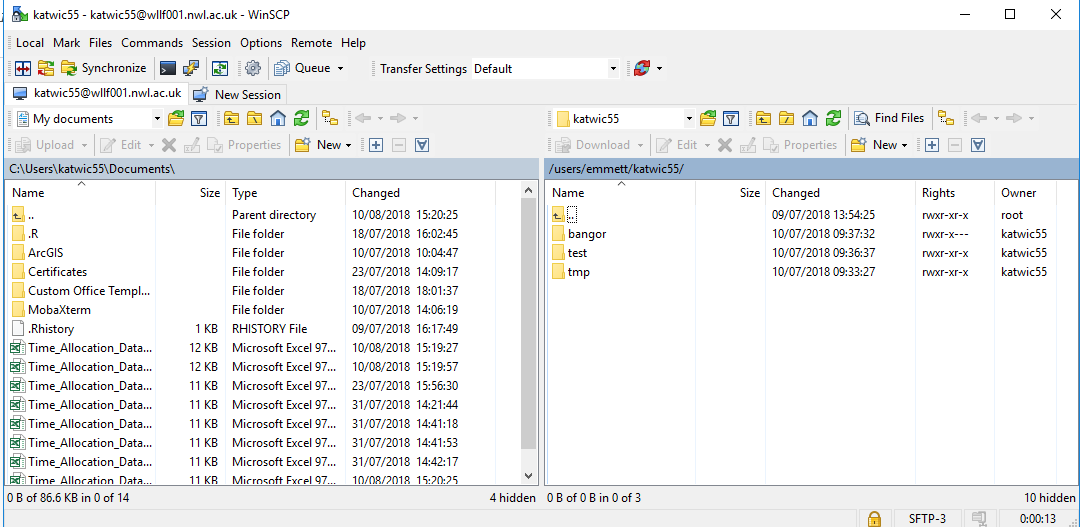


Figure

To access your Linux files from WinSCP, install the program, launch it (on a first launch choose the style you prefer – commander or explorator) and a new window will prompt you to enter your credential for a connection to a Linux server of your choice (Figure 8). After logging in you will see your local Windows computer directories on one side and your Linux user directory on the other side of the commander (if you chose this option) (Figure 9).



Figure



Figure

1. Fortran on Linux

**Editing files on Linux**

Now that your Linux home directory is mounted in Windows, you can edit the files from an editor of your choice from Windows or you may choose to do it from Linux. Typically in Windows you use IDE of some sort to code, alternatively editors such as Notepad++. In Linux, while there are some IDEs out there, most people tend to use a general purpose Text Editor instead (given that syntax highlighting is supported (:

* Gedit (For people used to GNU)
* Emacs (But real programmers don’t use this)
* Vim (Real programmers use this)
* Nano (Baby programmers use this)
* There is also Text Editor that MobaXterm comes with and is similar to Gedit.

CEH Linux servers have choice of these editors and it’s up to you which one to use. If the editor of your preference is not installed, contact the Linux system administrator.

CEH Linux server has also Eclipse IDE that can be used to develop and compile Fortran (and other languages) code. The benefit of using Eclipse is that it provides interactive debugging.

**Writing and compiling a Fortran program on Linux**

Fortran syntax on Linux is the same as on Windows. Some online treads report difference in results (e.g. precision difference) after comparing running Fortran programs on Windows and Linux, but these are confirmed to be either down to compilation options or different system versions.

Let’s write a simple Fortran program using Gedit editor. To keep things tidy let’s create a ‘test\_fortran’ directory in your home directory in Linux. Note that Linux is case sensitive, and as a general good practice spaces in directories names and file names should be avoided. Note & sign at the end of the command. The & lets you keep using the same terminal :

# create a directory

mkdir test\_fortran

# open a new file with a specified file name in Gedit and option to keep using the same terminal for other commands

gedit HellowWord.f90 &

**Makefiles**

**Debugging a Fortran program on Linux**

**Eclipse on Linux**

1. Git on Linux

**Windows**

This tutorial is

1. Git on Windows