



School of Computer Science

COMP30640

---

Lab 1  
Linux

---

Teaching Assistant:	Thomas Laurent
Coordinator:	Anthony Ventresque
Date:	Friday 14 <sup>th</sup> September, 2018
Total Number of Pages:	6

Linux is an Operating System that is widely used in industry - in particular the vast majority of server-side systems (web servers, app servers etc.) run on Linux machines. In this lab, you will be connecting to a remote Linux-based Virtual Machine (VM) (i.e. a remote computer with Linux installed on it). You will then play with some basic Linux system commands. You will be connecting to the VM via ssh (a secure protocol for connecting to remote machines). Follow the instructions below to connect to your VM (the way you will do this will depend on the operating system on your machine).

## 1 Connecting to your Linux VM

**Mac OSX and Linux** This is very straightforward, we simply open up the **Terminal** application and type:

```
$ ssh username@hostName
```

That's it!

**Windows** For windows, using ssh is slightly trickier, we need to download a piece of software that will allow us to connect. For a tutorial with visuals take a look at this:

<https://mediatemplate.net/community/products/dv/204404604/using-ssh-in-putty->

- Download Putty from: <http://www.putty.org/>
- When you have downloaded it, find the putty.exe (the executable file) and double-click.
- Once it is installed, run the application.
- Under 'Host Name' enter the host name that was given to you.
- For 'Port', insert '22'
- The connection type will be SSH
- Hit open. (Note: You can actually save configurations with Putty which may be useful for you going forward.)
- Enter the password for the VM when prompted and your connected!

Another option, which might be easier, is to enable a recent Windows 10 update. Have a look at this tutorial:

<https://www.howtogeek.com/336775/how-to-enable-and-use-windows-10s-built-in-ssh-commands/>.

## 2 (Optional) Using Bash on your Laptop

Bash is the name of the command line interpreter (CLI) that you will be using in this course. Today, in fact, you will be using some Bash commands. Bash can be used on different operating systems, however the way you obtain it will depend on your current operating system. See below for details if you would like to have bash on your local machine.

**Use Mac OSX' terminal app**

## Use Linux terminal app

**Install Ubuntu as an app on Windows:** <https://www.microsoft.com/en-ie/store/p/ubuntu/9nblggh4msv6>

**Install Bash for Windows:** <https://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/>

(not recommended)

## 3 Using man to Get Help

Nearly every command and application in Linux will have a man (manual) page, so finding them is as simple as typing `man command` to bring up a longer manual entry for the specified command.

- For example, `man mv` will bring up the mv (move) manual.
- Move up and down the man file with the arrow keys, and get back to the command prompt with "q".
- `man man` will bring up the manual entry for the man command, which is a good place to start!
- `man intro` is especially useful - it displays the "Introduction to user commands" which is a well-written, fairly brief introduction to the Linux command line.
- Some software developers prefer info to man (for instance, GNU developers), so if you find a very widely used command or app that doesn't have a man page, it's worth checking for an info page.
- Virtually all commands understand the `-h` (or `--help`) option which will produce a short usage description of the command and its options, then exit back to the command prompt. Try `man -h` or `man --help` to see this in action.
- man pages can be lengthy, so if you are looking for a specific option etc. it could be useful to look up some word using the syntax `/word` – and then hit the key `n` to move to the next occurrence. For example, try to find the option `-t` in the man page of python.
- If you aren't sure which command or application you need to use, you can try searching the manual pages. Each manual page has a name and a short description.
  - If you know part of the command name, use the following command: `whatis -r <string>`. For example try with the following: `whatis -r cpy`
  - To search the names or descriptions for `<string>` enter: `apropos -r <string>`. For example, `apropos -r "copy files"` will list manual pages whose names or descriptions contain copy files.

## 4 File & Directory Commands

The Linux hierarchy is typical of Unix systems (with some variations depending on the specific distributions). For the moment you just need to know that the file system is a tree that starts at the root (represented with the symbol `/`). Note that if you are familiar with DOS/Windows the path delimiter is the forward slash and not the backward slash... A path then looks like this in Linux: `/var/log/auth.log`. This leads to the file `auth.log` in the repository `log` in the repository `var` which is right after the root of the file system.

- The tilde (`~`) symbol stands for your home directory. If you are anthony, then the tilde (`~`) stands for `/home/anthony`.
- **pwd**: The `pwd` command will allow you to know in which directory you're located (`pwd` stands for "print working directory"). Example: "`pwd`" in the Desktop directory will show `~/Desktop`.
- **ls**: The `ls` command will show you ('list') the files in your current directory. Used with certain options, you can see sizes of files, when files were made, and permissions of files. Example: `ls ~` will show you the files that are in your home directory. Check some of the options of `ls`.
- **cd**: The `cd` command will allow you to change directories. When you open a terminal you will be in your home directory. To move around the file system you will use `cd`. Examples:
  - To navigate into the root directory, use `cd /`
  - To navigate to your home directory, use `cd` or `cd ~`
  - To navigate up one directory level, use `cd ..`
  - To navigate to the previous directory (or back), use `cd -`
  - To navigate through multiple levels of directory at once, specify the full directory path that you want to go to. For example, use, `cd /var/log` to go directly to the `/log` subdirectory of `/var/`. As another example, `cd ~/Desktop` will move you to the Desktop subdirectory inside your home directory.
- **cp**: The `cp` command will make a copy of a file for you. Example: `cp file foo` will make an exact copy of "file" and name it "foo", but the file "file" will still be there. If you are copying a directory, you must use `cp -r directory foo` (copy recursively). (To understand what "recursively" means, think of it this way: to copy the directory and all its files and subdirectories and all their files and subdirectories of the subdirectories and all their files, and on and on, "recursively")
- **mv**: The `mv` command will move a file to a different location or will rename a file. Examples are as follows: `mv file foo` will rename the file "file" to "foo". `mv foo ~/Desktop` will move the file "foo" to your Desktop directory, but it will not rename it. You must specify a new file name to rename a file.
- **rm**: Use this command to remove or delete a file in your directory.
- **rmdir**: The `rmdir` command will delete an empty directory. To delete a directory and all of its contents recursively, use `rm -r` instead.

- **mkdir**: The `mkdir` command will allow you to create directories. Example: `mkdir music` will create a directory called "music".

Let's start with some exercises!

### Exercise

- Explore the filesystem tree using `cd`, `ls` and `pwd`. Look in `/bin`, `/usr/bin`, `/sbin`, `/tmp` and `/boot`. What do you see?
- Make a directory called `OS` (`mkdir`) in your home directory and then rename it to `COMP30640` (`mv`)
- Make a sub-directory `lab1` and in it start a text file `lab1_commands.txt` using `vi` (or `nano` if you are not familiar with `vi`) where you'll write all the examples and exercises you'll do today.
- Copy the file `/etc/passwd` into your home directory (`cp`).
- Move it into the subdirectory `COMP30640` and create a symbolic link to `/proc/mem-info` called `meminfo.link` (look at `man ln`).
- use the following command `ls /var/log/ | wc -w` to get the number of files in `/var/log/`. How does `wc` work? (check this in the `man` page corresponding to `wc`). We will spend some time on the `|` symbol next week – but do you have an idea of what it does? try `ls /var/log/` alone to give you an idea of what happens.

## 5 Some System Information Commands

- **df**: The `df` command displays filesystem disk space usage for all mounted partitions. `df -h` is probably the most useful - it uses megabytes (M) and gigabytes (G) instead of blocks to report. (`-h` means "human-readable")
- **du**: The `du` command displays the disk usage for a directory. It can either display the space used for all subdirectories or the total for the directory you run it on. Two interesting options are `-s` ("Summary") and `-h` ("Human Readable")
- **free**: The `free` command displays the amount of free and used memory in the system. `free -m` will give the information using megabytes, which is probably most useful for current computers.
- **top**: The `top` ('table of processes') command displays information on your Linux system, running processes and system resources, including CPU, RAM & swap usage and total number of tasks being run. To exit `top`, press "q".
- **htop**: has my preference over `top`. You are unlikely to have it already installed so run the following command to install it:

```
sudo apt-get install htop
```

Play with `htop` a little to understand how it works.

- `uname -a`: The `uname` command with the `-a` option prints all system information, including machine name, kernel name & version, and a few other details. Most useful for checking which kernel you're using.
- `lsb_release -a`: The `lsb_release` command with the `-a` option prints version information for the Linux release you're running, for example:

## Examining files. *Why use more if you have less*

"cat" stands for conCATenate. You can use this command to dump the entire text file to the screen.

```
$> cat /etc/passwd
```

If the text file is too long, you might find that it scrolls past too quickly and you cannot see the beginning of the file anymore. In which case, you can use either the "more" or "less" command.

```
$> more /etc/passwd
```

Or

```
$> less /etc/passwd
```

Both these commands perform similar functions as they allow to see the text file one page at a time. You use the spacebar to continue paging, `<enter>` key will move down one line, and "q" to quit. "less" has actually more features than "more" :) The most useful feature is that it can scroll backwards (or up) whereas "more" cannot. Press "h" (while in the program) to see more options. Another interesting option is the search option, which is similar to the one you've seen when we presented `man`. If you are looking for a specific string in a text file use the syntax `/string` - and then hit the key n to move to the next occurrence. For example, try to find your login name in `/etc/passwd`.

## Show part of files. *Can you make heads or tails of it?*

Note: `/etc/passwd` is a file storing some login info for a user: user number, group number, shell at login, home directory etc. It also contains OS processes considered as "users". Use commands `whoami`, `id`, `groups` to try to get some of the information contained in `/etc/passwd`. `head` and `tail` are two opposite commands, showing the beginning or the end of a file respectively. Try the following commands – what do they give you?

```
$> head /etc/passwd
```

```
$> tail /etc/passwd
```

Both commands have various options that can be powerful – and a little complicated:

- `$> head -n 3 /etc/passwd` shows the first 3 lines
- `$> tail -n 5 /etc/passwd` shows the last 5 lines
- `$> tail -n +4 /etc/passwd` shows from the fourth line to the end
- `$> head -n -1 /etc/passwd` shows from the second line to the tenth line

A classical use of `tail` consists in showing the content of a dynamic file, that is evolving over time. `/var/log/` contains a lot of files that store log messages, i.e., information about what's happening in the system. Try `tail -f /var/log/syslog`.

### Exercise

- Combine `head` and `tail` using a pipe to print 15th line to 20th line in `/etc/passwd`.
- Combine `head` and `tail` using a pipe to print the 7th line in `/etc/passwd`