

# linux terminal Tutorial

carey pridgeon

October 17, 2016

## 1 Introduction To the Linux Terminal

- In this Worksheet you will cover a range of the commands available in the Linux Terminal and commands useful to you for this module
- Read the **entire** worksheet **before you begin**.
- The format is that the commands you are to use are listed and explained, then the task you are to do with them follows.
- When you see any text in square Brackets, **do not type them, just what they ask you to type in**
- *[filename]* means **choose your own filename**.
- Square bracketed sections means you replace that with your own text/filename/foldername. -For this worksheet we will be using a Server on campus that is not available to the general student population.
- As a result you *cannot* use your university login credentials, you need to be assigned some specifically for this module.
- If you have any problems with the worksheets because things don't work as expected on other machines, the first answer will always be *do it on nostromo*

## 2 Accessing Nostromo from Windows

- On windows we recommend MobaXterm, since that provides everything you need in one application. If you are familiar with X-forwarding, use Putty and an X-client, but mobaXterm is better for most people.
- To connect to Nostromo from your laptop, MobaXterm can be downloaded from the following URL - linky

- If on a university machine, launch MobaXterm by typing **Mobaxterm** into the start menu search then running the application that turns up.
- Mobaxterm provides an X server, so programs with graphical user interfaces will run locally.

## 2.1 Connecting to Nostromo with MobaXterm

- In the prompt window, type:
- **ssh [username]@nostromo**
- At the login prompt, use the password you've been given
- You can allow Mobaxterm to save your password.

## 2.2 Connecting to Nostromo from a Mac, Linux or in Putty.

- **ssh -Y [username]@nostromo**

## 2.3 Editing files

Nostromo has the text editors vim and emacs installed.

When asked to edit files in this tutorial use vim, unless you know how to use emacs.

- **Nano** is not installed, don't ask for it.

## 2.4 Commands reference

- This is **not** the exercise, this is the reference **for** the exercise.
- **ls**
  - List your files.
- **ls -la**
  - List all your files in long format including hidden files (displays permissions)
- **ls -R**
  - List the files in a folder tree recursively
- **cd [directory name]**
  - Change directory
- **cd ~**

- Change to your home directory
- **cd ..**
  - Move up one directory in the tree
- **pwd**
  - Print the working directory
- **rm [file]**
  - Delete the specified file
- **rm -rf [dir]**
  - Delete the specified directory recursively (rm alone will not work)
- **cp [file] [dest]**
  - Copy a file (send a copy to a new folder or filename)
- **mv file [dest]**
  - Move a file (to a new folder or change filename)
- **chmod +x [filename]**
  - Change permissions on a file (in this case making it executable)
- **./[filename]**
  - Run a program in your home folder
- **sh [filename]**
  - Run a bash script (file with an .sh extension)
- **ln -s [file] [/new/location/name]**
  - Create a symbolic link to a file in a specified location.
    - \* A symbolic link allows you to keep a reference to a file somewhere other than where that directory/file exists, so you can access it directly without navigating to that directory/file through the normal path.
    - \* It can also be used to allow typing of a simpler name than the file's real one, which can simplify the process of typing complex filenames.
    - \* This is similar to windows shortcuts some of you will be familiar with, but more more capable.
- **touch [file]**

- Update the timestamp on, or create, a file
- **mkdir [dirname]**
  - Create the specified directory
- **Mkdir -m 700 [dirname]**
  - Create a folder with the specified permissions (owner can execute)
- **less [filename]**
  - Display the contents of a large file one line at a time, advanced by pressing [return], exit by pressing **q**
- **cat**
  - Concatenate specified files to some source file/location, if no destination is set, the terminal is used.
  - **cat [file1] [file2] > [newfile]**
- **wget creative.coventry.ac.uk/~carey/[filename]**
  - Download a file from a webserver
- **grep [string] [file]**
  - Locate a specified string in a given file
- **less [filename]**
  - This utility will allow you to scroll back and forth through a text file one line at a time in the console.
    - \* Many Linux commands accept input from the keyboard (called Standard Input) and produce output to the terminal screen (called Standard Output).
- **> [destination filename]**
  - Redirect standard output to a file, over-writing its contents or creating it if it doesn't exist.
- **»**
  - Append standard output to file at the end of the file, also creating it if it doesn't exist.
- **date;who**
  - List who is logged into the system and the date/time.
- **man [subject]**

- Display the manual (man page) for the specified Linux command or almost any Linux aspect.
- **man -k [string]**
  - search the man system for an entry with a name containing the given string, or containing that string.
- Tar (Tape Archiver)
  - This is a utility to compress a folder into a single file. Originally intended to create single file backups to write to large tapes for long term storage. Google are the worlds largest consumers of tapes of this type. Tar’s native compression is light to the point of nonexistent, so it can be combined with external compression libraries though command options.
- **tar -cvf archive\_name.tar dirname/**
  - Create a single file archive.
- **tar -cvzf archive\_name.tar.gz dirname/**
  - Create a single file archive filtered through gzip for compression
- **\*tar -xvf archive\_name.tar.gz**
  - Extract a file archive
- **tar -zxvf archive\_name.tar.gz -C [optional output dir]**
  - Extract a file archive to a specified directory
- **tar -tvf archive\_name.tar**
  - List the contents of a tar file (compressed or not) without extracting it

### 3 Worksheet tasks

- **Do these tasks one at a time, don’t combine tasks even if it looks like you can.**
- list the files in your home directory so that hidden files are also shown.
- make a directory called 389 in your home directory
- change to this sub-directory
- create a folder called **cli**
- print the full path for this directory and redirect the output to *text.txt*

- Change back to the parent directory
- Create a symbolic link to text.txt in your home directory called txt.
- cat this symlink.
- Create empty file 389/cli/fstab.txt without using a text editor
- Concatenate /etc/fstab to 389/cli/fstab.txt
- Display this file in the terminal
- Search for the word 'reference' in fstab.txt
- Redirect the output you see to the file 389/cli/out.txt and print this to the console.
- Create a file myls in 389/cli and write `ls -la /` in it
- Make this file executable and run it.
- If it works (prints the result of `ls -la /`, run it again and redirect the output, appending it to out.txt
- redirect a list of who is logged in to nostromo to 389/cli/who.txt
- Create a symlink to cli in your home directory.
- Use this symlink to list the contents of cli/
- Search man for entries containing **less**
- Repeat this search and redirect the output to 389/cli/less.txt, then display this file in the terminal.
- Fetch this file <https://www.mozilla.org/media/img/home/voices/mozilla-wordmark-white.20d134cd8af7.png> into 389/cli
- In **man**, find one of the commands you have used in this tutorial.
- tar the 389 folder
- tar the 389 folder with gzip compression
- list content of first tar
- list contents of second tar
- create a folder called tmp in 389 and untar the gzip compressed tar file into it
- recursively list the content of 389

## 4 On completing this worksheet

- Logout, log in again then type the following command
- **cat .bash\_history > cli\_done.txt**
- This will create the file that you will submit as evidence of completing this worksheet.
- In MobaXterm you can download this file to your desktop. Otherwise you can scp it to yourself.