

ITS 332 Networking Lab

Introduction to Ubuntu Linux

Dr Steven Gordon

Revision 927

6 November 2009

1 What is Ubuntu Linux?

Linux is an operating system based on [Unix](#), one of the earlier multi-user operating systems developed in the 1970's and 1980's. Unix was originally a single operating system, but over time several commercial variants were developed. These Unix operating systems were particularly popular in the 1980's and 1990's, especially within academic and technology organisations. Some of the Internet applications and protocols were first developed on Unix, and hence Unix-based computer systems have a strong link with computer networking.

Today Unix operating systems are still used, mainly in servers and high-end workstations. In the 1990's Linux appeared, a free operating system with Unix-like functionality (or at least a kernel for an operating system). In the 2000's, Linux also became popular in typical Unix domains of servers and workstations, and also has been growing in the desktop field (however, in quantity of installs, Linux still does not compare with Microsoft Windows). As with the original Unix, there are many variants, or [distributions](#) of Linux, differing in the applications and graphical environments they provide (e.g. [RedHat](#), [Debian](#), [Fedora](#), [Ubuntu](#), [Xandros](#)). We will be using the [Ubuntu](#) Linux distribution.

Ubuntu Linux is a free, open-source Unix-based operating system, that has been developed mainly for desktop (and laptop) installations. The aim is make a user-friendly Linux distribution. It is now one of the more popular Linux distributions. Ubuntu is installed on the Network Lab computers, and will be used extensively to demonstrate computer network operations in ITS 332. This document aims to give a quick introduction to some of the most common operations that you will need during the course.

1.1 Why Not Microsoft Windows?

Why use Ubuntu Linux, and not Microsoft Windows, especially since Windows is by far the most popular desktop operating system, and hence very popular with server systems? There are several reasons we will use be using Linux instead of Windows:

1. Linux is well-suited for learning of networking concepts:
 - (a) Linux has simple, yet powerful, operations for many networking tasks such as: changing an IP address, creating routing tables, testing network connectivity, inspecting traffic received/sent, and so on.

- (b) Implementing and compiling simple client/server applications is straightforward on Linux.
 - (c) A Linux PC can easily be configured as a router (all the PCs in the Network Lab have two interface cards).
2. Experience in Unix-based operating systems is important: Although Windows is the most commonly used operating system for desktops, Unix-based operating systems (including Linux) are common for network servers, network devices and embedded systems. For example, many routers, switches and specialised computer devices use Linux.
 3. Ubuntu Linux is free, as are all the applications we use (and none of them are pirated!)

2 Common Operations

2.1 Starting Ubuntu Linux

When the computer boots, within the first several seconds a program gives you the option to start Windows or Ubuntu. You should select Ubuntu, which will boot Ubuntu Linux.

2.2 User Accounts and Login

Once Ubuntu has started you are presented with a login screen. You should login with the username/password provided in the class.

Different users in Ubuntu have different privileges (e.g. ability to view or edit system files, view or edit other people's files, change important operating system parameters). The user with the most privileges (that is, can do everything!) is called *root* (sometimes also called *super-user*). The problem with logging in as *root* is that a simple typing mistake may delete the entire hard drive!

The user you login as is just a normal user – let's refer to them as *student*. The user *student* has the ability to view and edit their own files in the directory `/home/student`, view most system files (that is almost all files on the hard disk, except those of other users) and view configuration options (such as IP address). You must always login as this normal user, and perform most operations as this normal user.

However, sometimes during the lab classes it will be necessary to perform tasks that require more privileges than the *student* user. For example, you require *root* privileges to install new software, change IP addresses and modify system files (such as configuration parameters for the web server). The *student* user has been configured to allow them to temporarily gain the privileges of the *root* user for these tasks. You do this using the `sudo` command.

Let's assume there is a command you need to execute in the command line shell (see Section 2.4). The command is:

```
command parameter1 parameter2
```

However, you must execute this as *root* user (since as the normal user, you are not allowed). So you would actually run the command by preceding it with `sudo`:

`sudo command parameter1 parameter2`

On the first use of `sudo` you will be prompted for a password - it is the password you logged in as *student* with. Then the command will execute. If you do not use `sudo` (and the command is privileged), the the command will not execute (usually returning an error like *Permission denied*).

A final note on the *root* user. As we said before, you can potentially delete the entire hard drive. As we give you the access to perform operations as *root* user, you must act responsibly. Anyone caught using these privileges incorrectly will be punished. This includes deleting system or other users files, copying other users files, changing parameters of the operating system and installing software which is not needed for the class. Punishment may range from loss of marks for the lab class, to more severe punishment in line with that for cheating (e.g. zero for the course).

2.3 Window System

Ubuntu has a graphical windows system like most other operating systems. It is quite intuitive. The main functions can be obtained from the *Applications*, *Places* and *System* menus from the top left corner of the screen.

Although many of the networking operations can be performed using the graphical tools, almost all have a command line interface.

2.4 Command Line Shell

Like almost all Unix-based systems, operations can be performed via a command line shell or terminal. In Ubuntu, to start a new terminal select *Accessories* and then *Terminal* from the *Applications* menu.

Some of the more common operations you will use include:

cd change directory

ls list the files in the directory

man view the manual (help) for a command

cp copy a file

mv move/rename a file

rm remove/delete a file

mkdir make/create a directory

rmdir remove/delete a directory

less display a file

cat display a file

echo print text to the screen (standard output)

pwd display the name of the present/current working directory

wc display the number of lines, words and bytes in a file

> redirect output to file

< redirect file to input

ps list the current processes running

& place process to be started into the background

Ctrl-c stop (kill) the currently active process

Ctrl-z suspend the currently active process

bg place the the just suspended process into the background

fg bring the background process to the foreground

An example of using some of these commands is shown below. For the computer that this example was executed on, the prompt is: *username@hostname:pwd\$* (e.g. *sgordon@ginger:~\$*)

```
sgordon@ginger:~$ pwd
/home/sgordon
sgordon@ginger:~$ mkdir test
sgordon@ginger:~$ cd test
sgordon@ginger:~/test$ pwd
/home/sgordon/test
sgordon@ginger:~/test$ nano example.txt
sgordon@ginger:~/test$ cat example.txt
Hello, my name is Steve.
sgordon@ginger:~/test$ ls
example.txt
sgordon@ginger:~/test$ ls -l
total 4
-rw-r--r-- 1 sgordon sgordon 25 2009-11-06 16:34 example.txt
sgordon@ginger:~/test$ wc example.txt
  1  5 25 example.txt
sgordon@ginger:~/test$ cp example.txt copy-of-example.txt
sgordon@ginger:~/test$ ls
copy-of-example.txt  example.txt
sgordon@ginger:~/test$ rm example.txt
sgordon@ginger:~/test$ ls
copy-of-example.txt
sgordon@ginger:~/test$ mv copy-of-example.txt example.txt
sgordon@ginger:~/test$ ls
example.txt
sgordon@ginger:~/test$ rm example.txt
```

```
sgordon@ginger:~/test$ ls
sgordon@ginger:~/test$ ls -al
total 12
drwxr-xr-x  2 sgordon sgordon 4096 2009-11-06 16:36 .
drwxr-xr-x 75 sgordon sgordon 8192 2009-11-06 16:33 ..
sgordon@ginger:~/test$ echo 'Hello'
Hello
sgordon@ginger:~/test$ echo 'Hello' > another-example.txt
sgordon@ginger:~/test$ cat another-example.txt
Hello
sgordon@ginger:~/test$ wc another-example.txt
1 1 6 another-example.txt
sgordon@ginger:~/test$ rm another-example.txt
sgordon@ginger:~/test$ ls
sgordon@ginger:~/test$ cd ..
sgordon@ginger:~$ rmdir test
sgordon@ginger:~$
```

We will introduce network-specific operations during the labs. For reference, some networking commands are listed in Section 4.

2.5 Text and Source Code Editing

Although everyone has their own preferences about text and source code editors, two standard editors in Ubuntu that are recommended are:

gedit A GUI based editor, with syntax highlighting. Can be opened from *Accessories* then *Text Editor* from the *Applications* directory, otehrwise executing **gedit** from the command line.

nano A command line based editor. Provides a quick and simple way to edit a file

2.6 Applications

Some of the applications that we may use during the labs include:

Wireshark Capture and view traffic on a network interface. Command: **wireshark**. Also available via the GUI menus.

Apache Web Server A common web server.

3 Advanced Operations

3.1 Installing Software

Although it should not be required during the labs, (and you must not install any software unless asked to by the instructor!), Ubuntu has a simple command line interface to installing software, using **apt-get**:

```
apt-get install NAME
```

where `NAME` is the name of the software package you want to install. Of course, you need administrator privileges to install software (hint: `sudo`).

3.2 Compiling C Code

You can use the GNU C Compiler to compile C code:

```
gcc -o EXECUTABLE FILE.c
```

while compile `FILE.c` and create the executable program named `EXECUTABLE`.

4 Command Reference

Use `man` to see a detailed description of commands on Ubuntu. Table 1 lists some of the important commands, as well as their version on Windows.

Description	Ubuntu	Windows
Network interface configuration	<code>ifconfig</code>	<code>ipconfig</code>
Test network connectivity	<code>ping</code>	<code>ping</code>
Test network route	<code>tracert</code>	<code>tracert</code>
Routing table configuration	<code>route</code>	<code>route</code>
Network statistics	<code>netstat</code>	<code>netstat</code>
ARP tables	<code>arp</code>	<code>arp</code>
DNS (simple)	<code>host</code>	<code>nslookup</code>
DNS (medium)	<code>nslookup</code>	<code>nslookup</code>
DNS (detailed)	<code>dig</code>	<code>nslookup</code>
Capture and view traffic	<code>wireshark</code>	Wireshark
Enable network interface	<code>ifup</code>	-
Disable network interface	<code>ifdown</code>	-

Table 1: Important networking commands

5 File Reference

For some of this files, you can view detailed description and format specification via the `man` pages, e.g. `man hosts`, `man interfaces`.

File	Description
<code>/etc/hosts</code>	Local domain names
<code>/etc/resolv.conf</code>	Local DNS server
<code>/etc/network/interfaces</code>	Network interface information
<code>/proc/sys/net/ipv4/ip_forward</code>	IP forwarding is on (1) or not (0)
<code>/etc/apache2/sites-available/default</code>	Configuration for Apache web server
<code>/var/lib/dhcp3/dhclient.X.leases</code>	DHCP IP address leases
<code>/etc/dhcp3/dhclient.conf</code>	DHCP configuration

Table 2: Important networking files

A Notes

Record any additional notes about Ubuntu here (e.g. important points made by the instructor, summary of things you learned, mistakes you made). You should use this in future labs, as well as in preparation for assessment items like exams.