UNIX FOR THE TERRIFIED

Richard Evans

THE UNIVERSITY of York

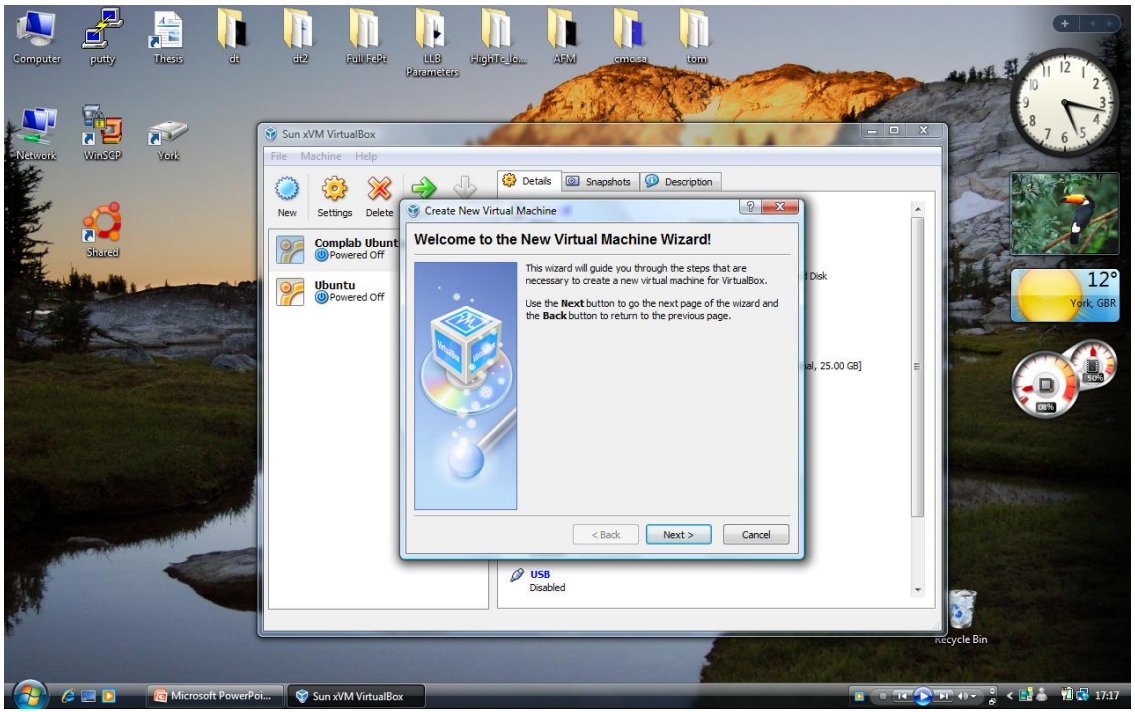
rfle500@york.ac.uk

What is Unix?

- Unix is a computer operating system like Windows™
- However, it was designed with programming in mind
- Linux is a freely available (open source) Unix-like operating system and is used in devices ranging from mobile phones to super computers
- Why use Linux?
  - Linux can use the command line to perform almost all tasks
    - Computer and file management
    - Compiling and running programs
    - Connecting to other Linux machines and the internet

Starting Ubuntu

- For CompLAB you will be using Ubuntu (a version of Linux) to create, compile, run and analyse your programs
- To use Ubuntu, we need to configure the virtual machine
- Start by opening the Sun Virtual Box software on the start menu



Setting Up the Virtual Machine

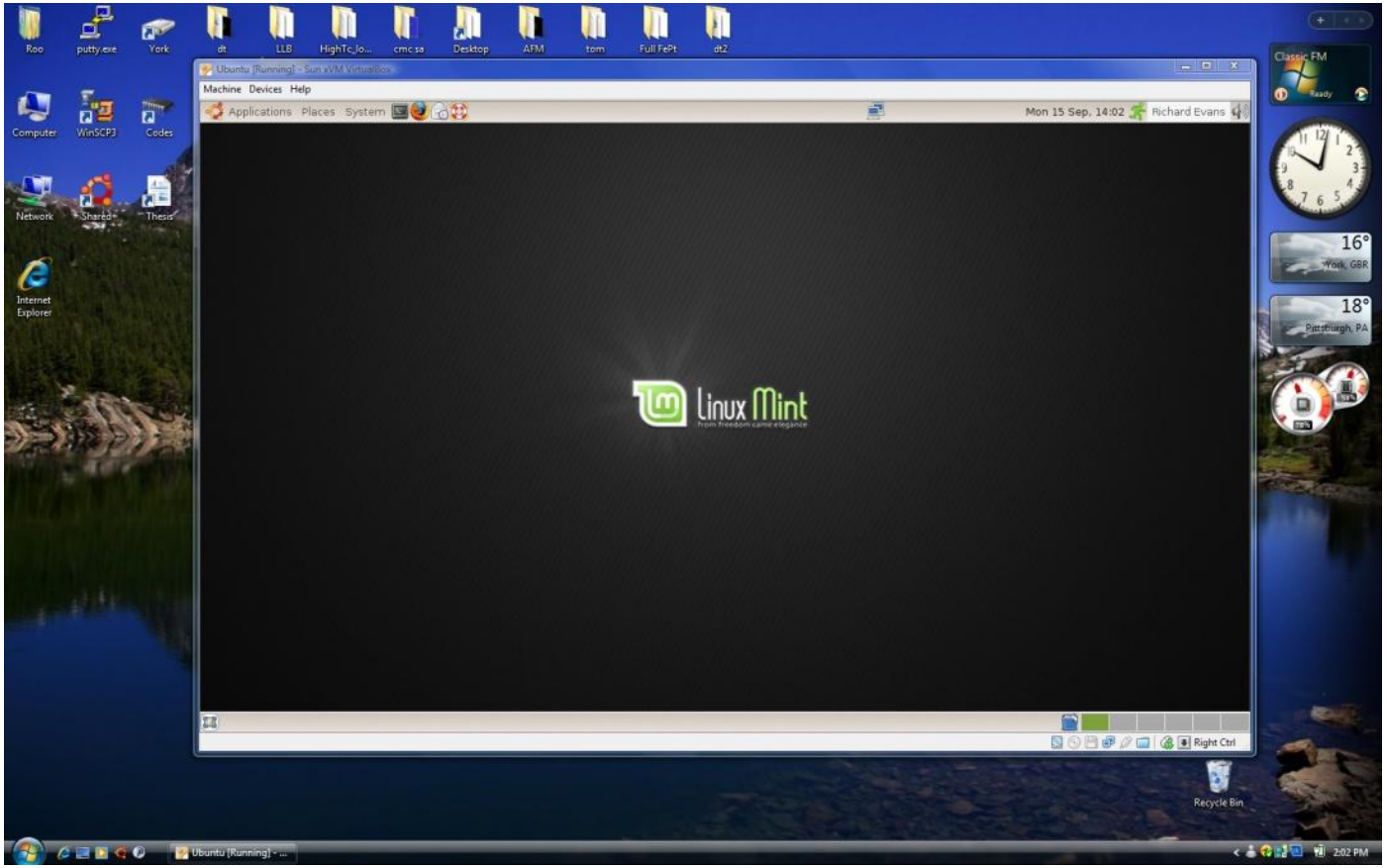
- To set up the virual machine for the first time, you need to type a name and an OS type (Ubuntu) in the new virtual machine wizard
- Then allocate 1GB of main memory for the virtual machine
- Finally you need to add a virtual machine file, located in

C:\Program Files\VDI\

- Click Finish, and then "Start" the virtual machine
- This is also how to start the virtual machine in future

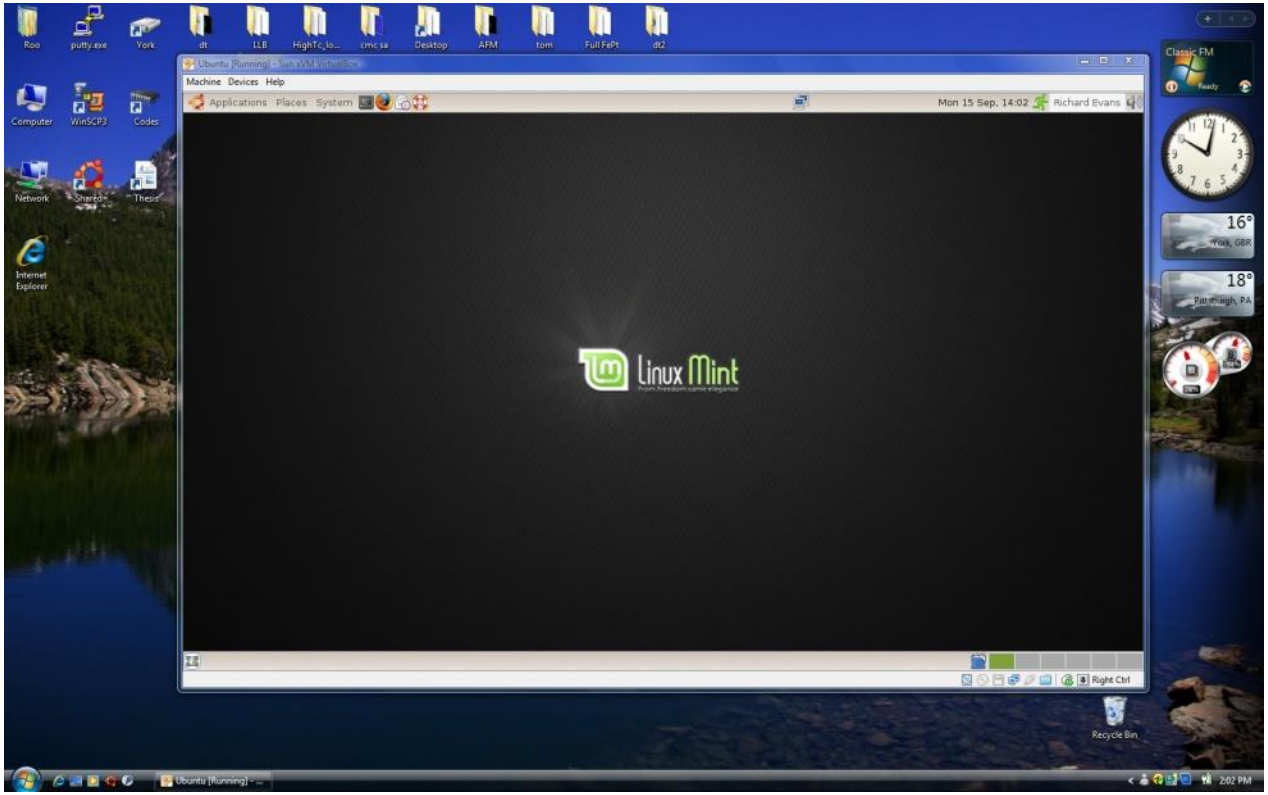
Starting Ubuntu

- This will start a *virtual* Linux machine in windows, much like any other program



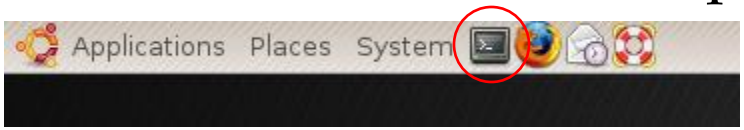
Starting Ubuntu

- Click on the VirtualBox window and press right Ctrl+F
- This will put Ubuntu into full screen mode
- To get back to Windows, press right Ctrl+F again

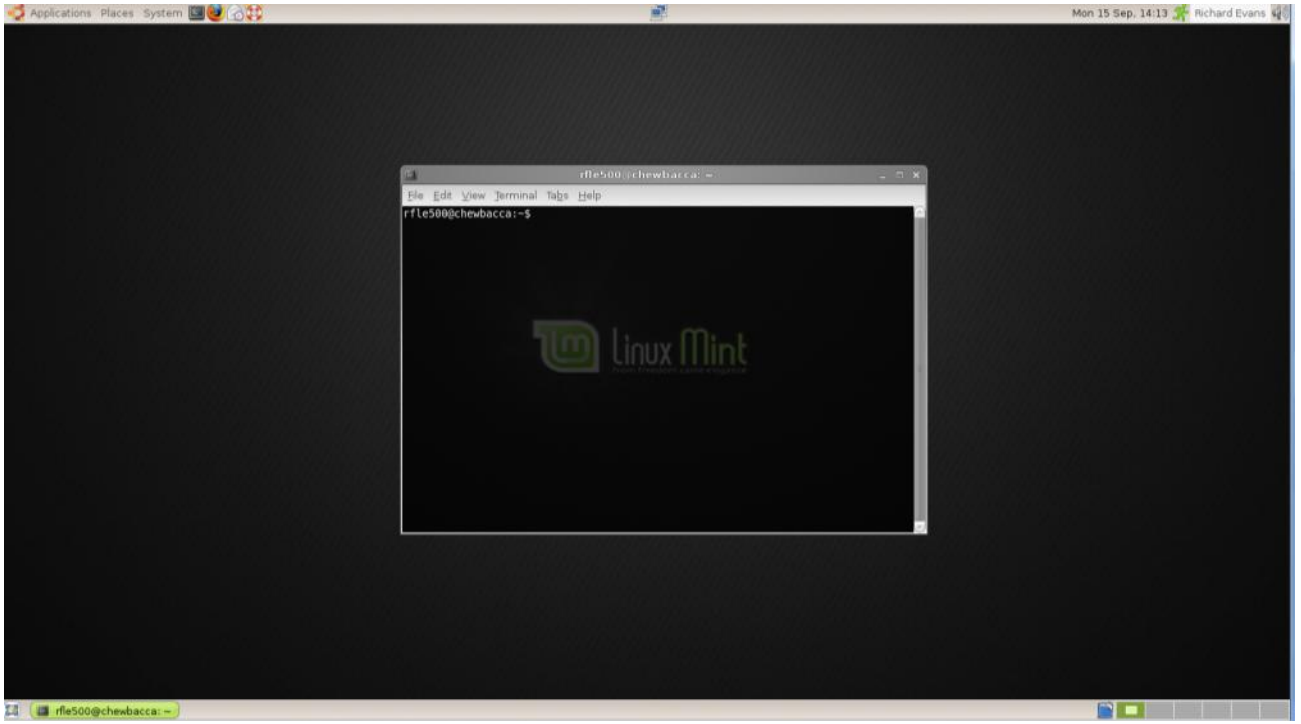


Starting the Command Line Terminal

- Click on the Terminal button at the top of the desktop

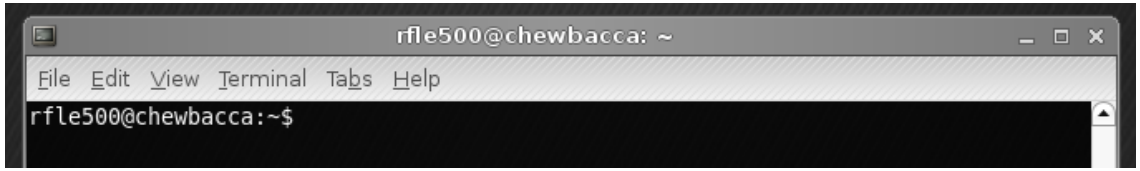


- This will open up a command line terminal



The basics of the Command Line Terminal

- The Terminal gives a blinking cursor waiting for a command:



- All commands have the following form:

```
command [options] [files]
```

This is the command you wish the computer to execute

These are the (optional) options you want to run the command with

This is the file(s) that the command requires in order to run

File Management Commands

- There are hundreds of command line programs available, but some are particularly useful for managing files
- "*pwd*" lists in which directory you are on the machine
  - All user files are stored in **`/home/user`** by default
- "*mkdir*" creates a directory in the directory given by *pwd*
  - **`mkdir new`**
- "*cd*" changes into the directory you specify
  - **`cd new`**
- "*ls*" lists the files in your current directory

File Management Commands

- There are some special files which help you change directory:
  - "." – this is a shortcut for the current directory
  - ".." – this is a shortcut for the directory above the current one
  - "~" – this is a shortcut for your home directory
  - "*" – this is a shortcut for all files in the current directory
- These shortcuts can be used with ls, pwd, and cd, eg:
  - `pwd .` print the current working directory
  - `cd ..` change directory to the above
  - `cd ../../` change directory to two above
  - `ls ~` list all files in my home directory

File Management Commands

- There are also commands which manage files directly:
  - **`cp file1 file2`** copy *file1* onto *file2* (overwrites *file2*)
  - **`mv file1 file2`** move *file1* to *file2* (deletes *file1*, overwrites *file2*)
  - **`rm file1`** remove *file1* (permanently!)

- Other commands
  - **`man [command]`** prints the manual for the command to screen
  - **`top`** program to monitor current cpu usage
  - **`less [file]`** simple text viewer
  - **`more [file]`** less simple text file viewer

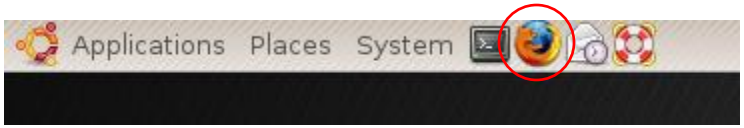
Connecting to The Outside World

- ssh – secure shell command
  - **`ssh myusername@remotemachine.york.ac.uk`**
  - Opens a secure "shell" to a remote linux machine, allowing you to type and execute commands remotely
  - eg: **`ssh abc500@ludwig.york.ac.uk`**

- sftp – secure file transfer protocol
  - **`sftp myusername@remotemachine.york.ac.uk`**
  - When logged in we can use "*put*" and "*get*" to transfer files
  - **`put file1`** put *file1* from local machine to remote machine
  - **`get file1`** get *file1* from remote machine to local machine

Getting Started

- You need to register on the unix service so we can get access to our files in linux
- Start Firefox by clicking on the icon



- Go to http://www.york.ac.uk/services/cserv/myitaccount/
- Click on “Account facilities” and enter your username and password
- Register for the “Unix Service”

- http://www.york.ac.uk/proxy.config

SETTING UP YOUR FILES

- This next step will link your university files to your Ubuntu file system
  - At the command line type:
  - **`sshfs yourusername@unix0:w2k ~/files`**
  - Enter your university password
  - Your university files are now linked to the directory ~/files/ on the local machine
- cd into files and make a new directory named complab by typing:
  - **`mkdir complab`**
- cd into complab so that all the files you create from now on are stored where you can find them
- At the end of the session close all open files and type
  - `fusermount -u /home/yourusername/files`

  to unlink your files

Editing a Source File

- First we are going to get a source file
- Use sftp to connect to unix0

```
sftp yourusername@unix0
cd /usr/transfer
get hello.f90
```

- To view and edit the source file we are going to use a text editor called emacs
  - Emacs features syntax highlighting and a Fortran menu button
  - Open emacs by typing: **`emacs hello.f90 &`**
- The ampersand (&) places the application in the background so that you can still type commands into the command line

Compiling a Program

- Now we are going to compile the source file with a compiler which converts the code to an executable which can be understood by the computer
- The following command compiles the source file:

```
gfortran hello.f90 -O0 -o hello.exe
```

This is the compiler command which calls the intel fortran compiler

This is the source file you are compiling

This is an option for the compiler, telling it how much optimisation to perform on your code

This is the executable output file

Running a Program

- Now that the program is compiled we can run it by typing the following:

```
./hello.exe
```

This . is the shortcut for the current directory

This is the executable file we want to the computer to run

- This program outputs "hello world" to the command line, and writes a data file "function.txt" to the local folder

Plotting Data with GnuPlot

- Now that we have some data we can plot that with gnuplot

```
plot [-100:100][-100:100] "function.txt" u 1:2 w lp
```

This is the plot command, use splot for 3D

These are the x and y ranges respectively (leave blank for auto)

This is the file we want to plot

These are the columns we want to use in the file

This is how we want the data displayed (l=lines, p=points)

Plotting Data with GnuPlot

- Gnuplot can also plot functions

```
plot [-100:100][-100:100] x**2 w l
```

- Labels can be given to axes

```
set xlabel "x label" ; set ylabel "y label"
```

- Also title for the graph

```
set title "title"
```

Plotting Data with GnuPlot

- You can plot multiple graphs and assign each one a label for the legen

```
plot [-100:100][-100:100] x**2 title "data label" w
    l,"function.txt" u 1:2 ti "data label 2" w p
```

- You can also post/pre multiply columns or multiply columns together

```
plot [-100:100][-100:100] "yourdata.dat" u 1:($2*0.1)
                          w lp
        plot "yourdata.dat" u 1:($2*$3) w l
```

Plotting Data with GnuPlot

- Printing from gnuplot: to print you must change the terminal type, by default it is xterm or X11. Some good ones are png and postscript

```
set term postscript eps enhanced color solid
set output "filename.eps"
replot
```

- You can put all of these commands in a text file to save it
- You can then use the load command to replot your data
- More information can be found at:

http://t16web.lanl.gov/Kawano/gnuplot/index-e.html