

Introduction to Linux and the shell

Graham Markall

<http://www.doc.ic.ac.uk/~grm08>

grm08@doc.ic.ac.uk

Imperial College Linux Users' Group

8th November

Starting point, questions, etc.

- Basic assumption: You've just installed Linux
- I'm going to try and move through a lot of things fairly quickly
- In order to introduce as many useful things about the shell as possible
- Please shout out throughout the talk
- Examples usages of any of the shell commands?
- Slides will be available after the talk

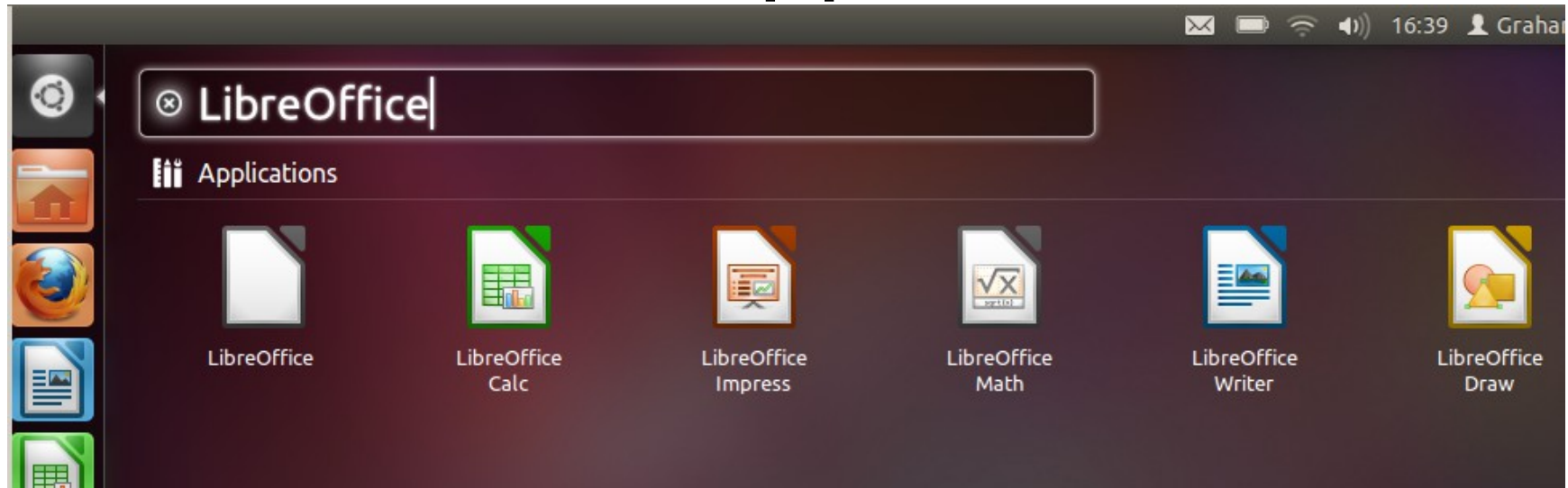
Contents

- Usage - Applications, Windows Compatibility, Connecting to Windows networks
- Introduction to shell scripting: Linux directory structure, Manipulating files, Shell scripts
- Useful shell utilities: zipping and unzipping files, Connecting to remote machines, apt-get, etc.

Part 1

Usage & Applications

Office Applications



- Powerpoint: LibreOffice Impress
- Excel: LibreOffice Calc
- Word: LibreOffice Writer
- Compatibility: Office 97-2003 formats OK, 2007+ (.docx, .pptx) a little dodgy

Office Applications (con.t)

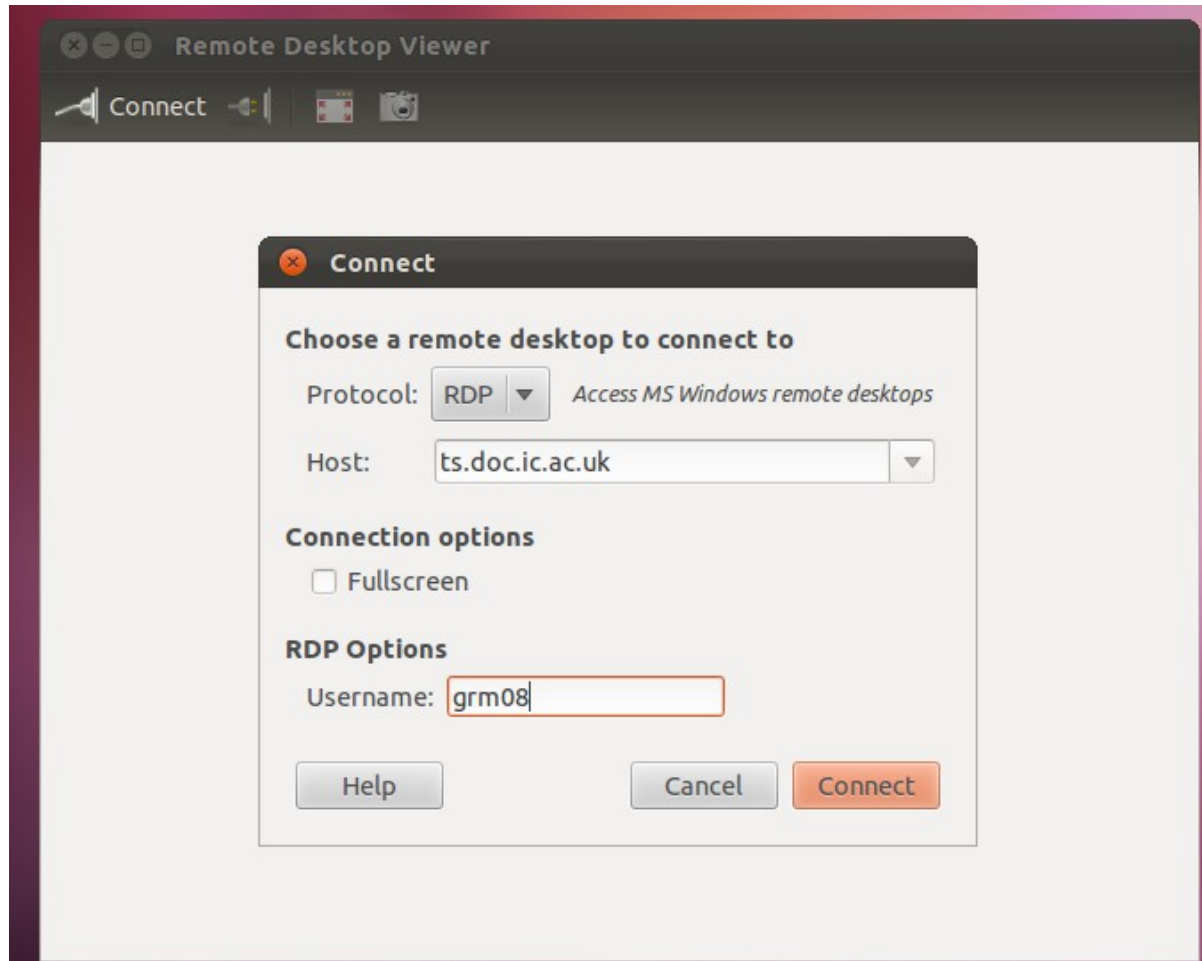
- Access: Openoffice.org base
 - Connects to remote Access databases on Linux
 - Doesn't open .mdb files directly
 - See FAQ at <http://www.openoffice.org/FAQs/ms-access/ms-access.htm>
- Draw: Vector graphics drawing (Coreldraw, or Adobe Illustrator)
- Note: Future versions of Linux distributions might use LibreOffice

Internet Applications

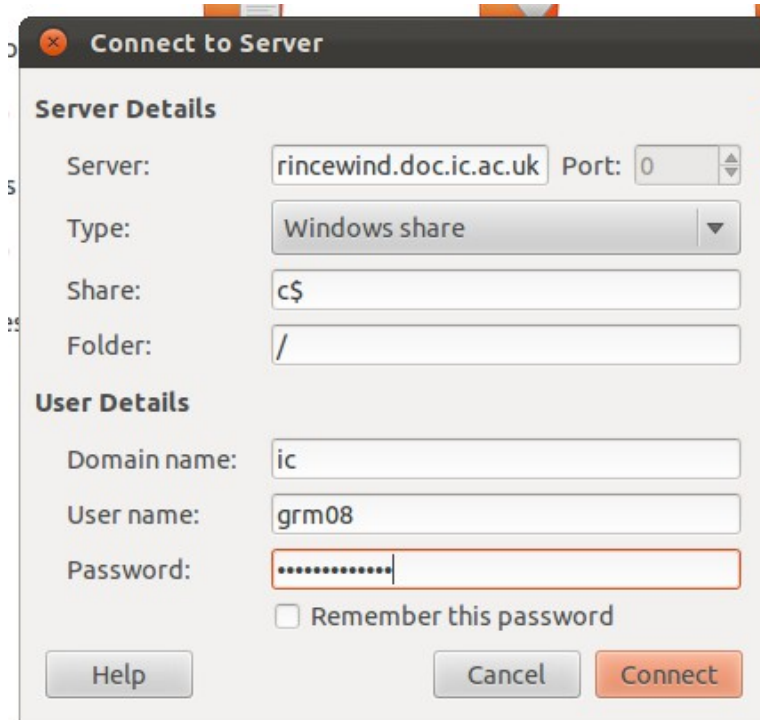


- Empathy – chat, MSN, AIM, etc...
- Thunderbird – email, works with IC email
- Firefox – familiar web browser :-)

Connecting to a Windows machine

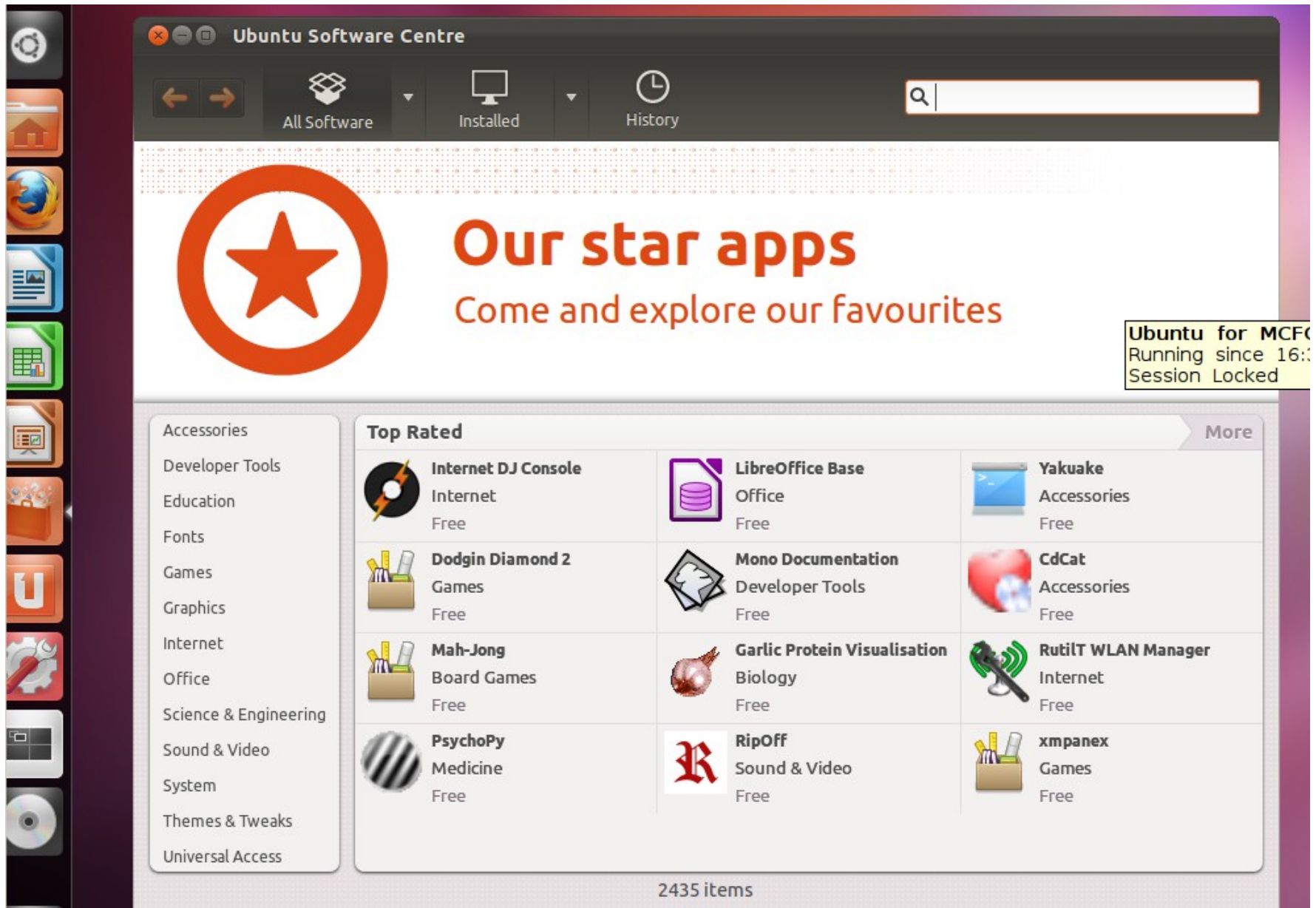


Connecting to a Windows network

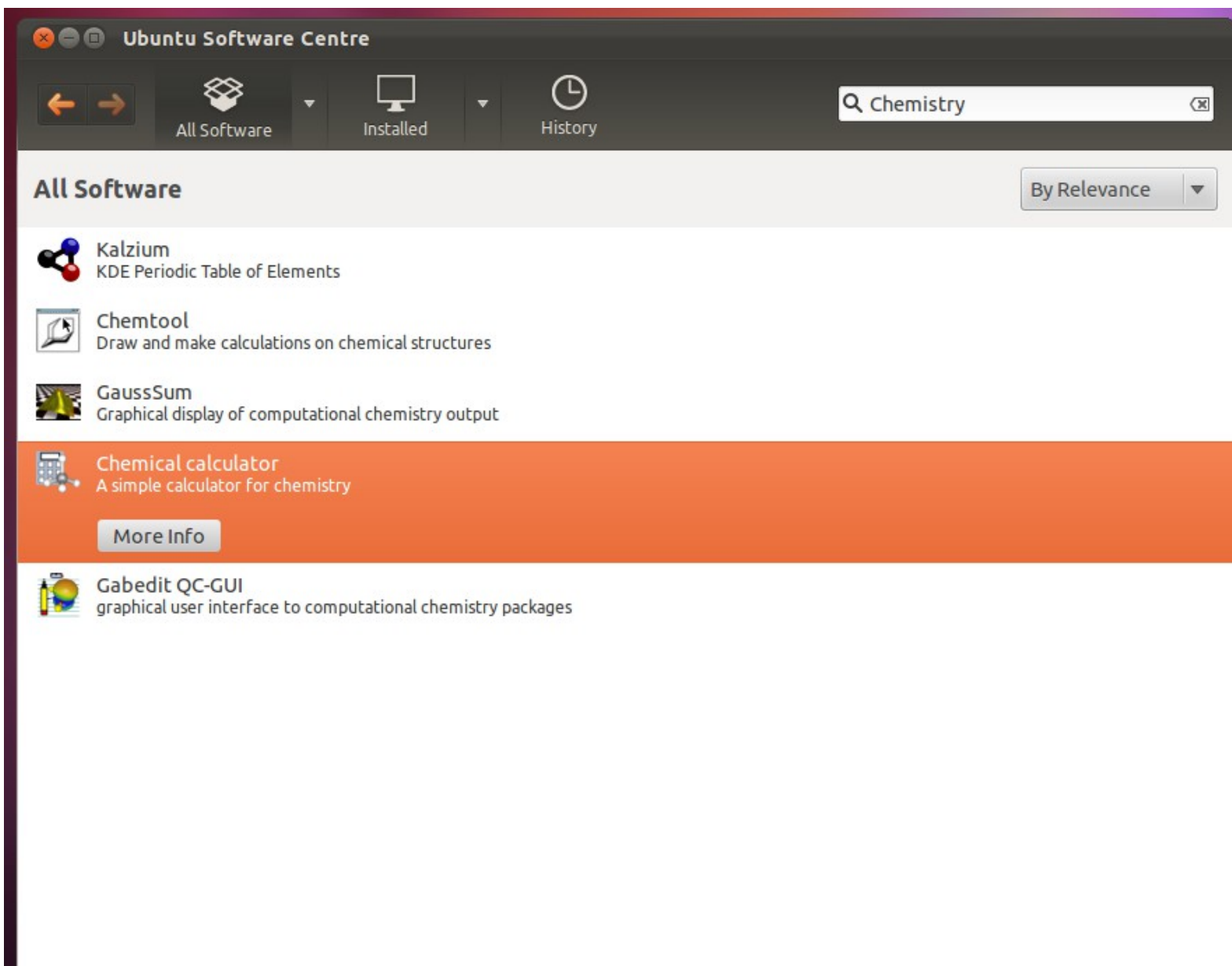


- Open up home folder 
- File menu → Connect to Server

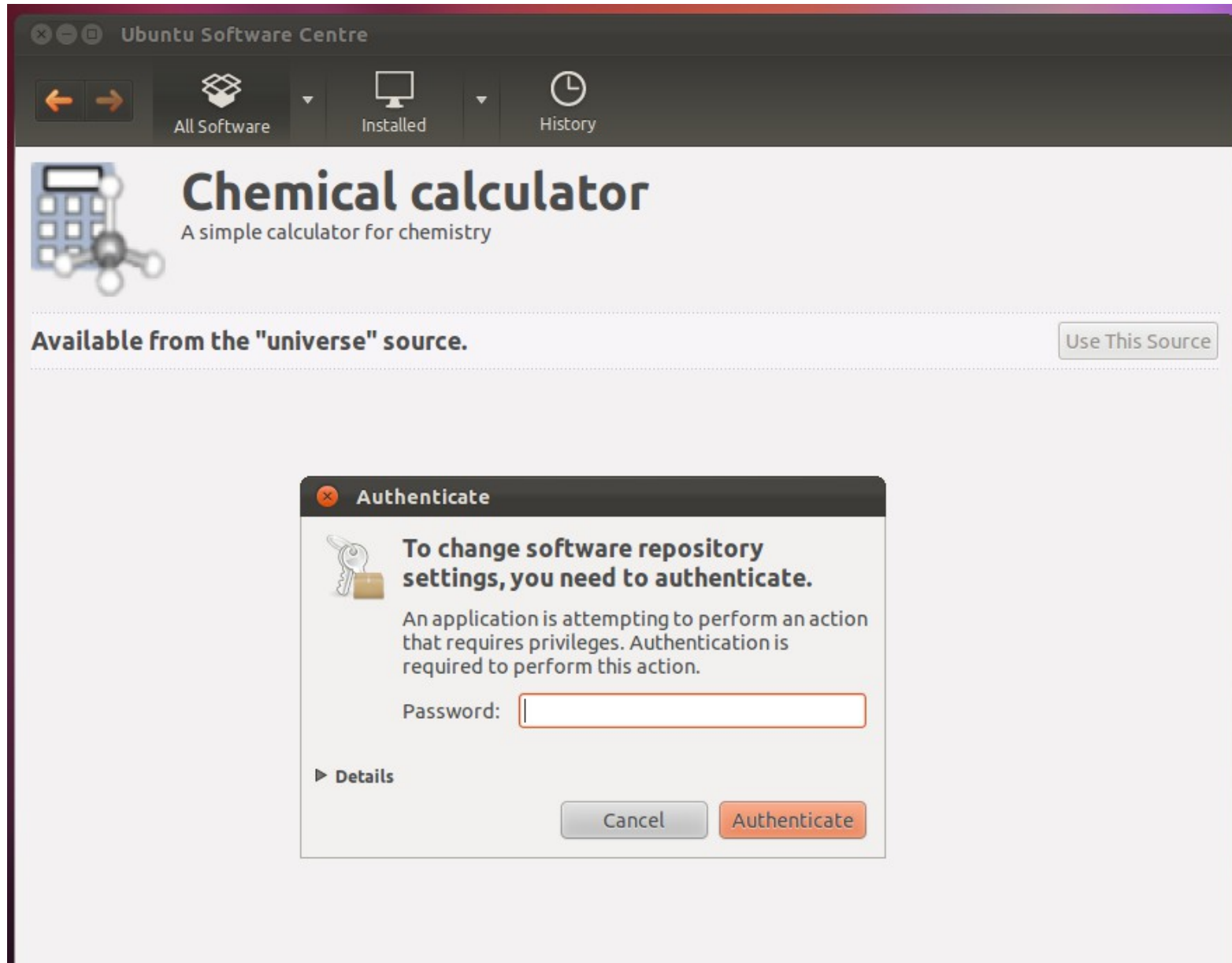
Installing - Ubuntu Software Centre



Installing Applications




Installing Applications




Ubuntu Software Centre

All Software Installed History

 **Chemical calculator**
A simple calculator for chemistry

Available from the "universe" source. [Use This Source](#)

Authenticate

 **To change software repository settings, you need to authenticate.**

An application is attempting to perform an action that requires privileges. Authentication is required to perform this action.

Password:

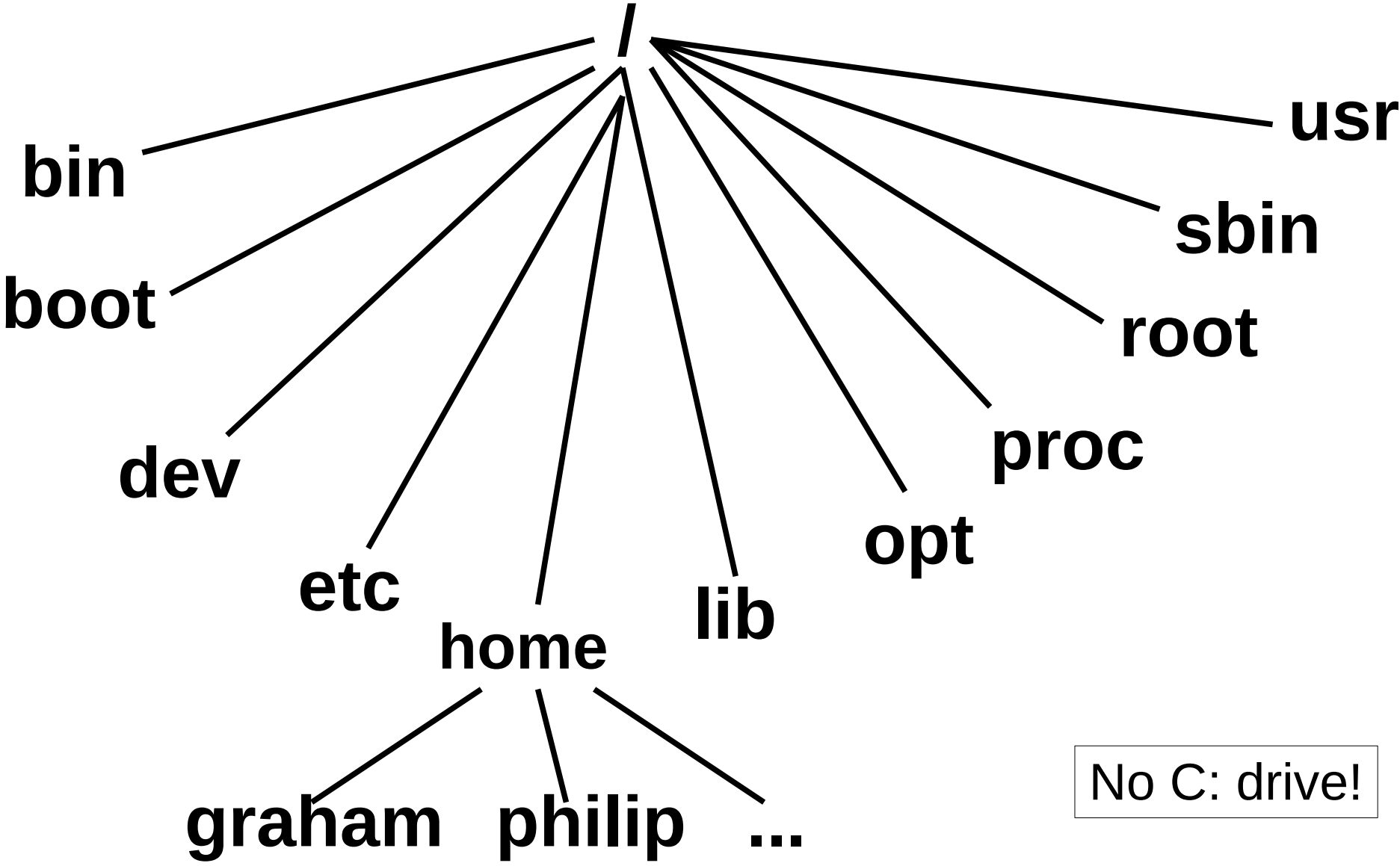
► Details

Cancel Authenticate

Part 2

Shell scripting

Directory structure



No C: drive!

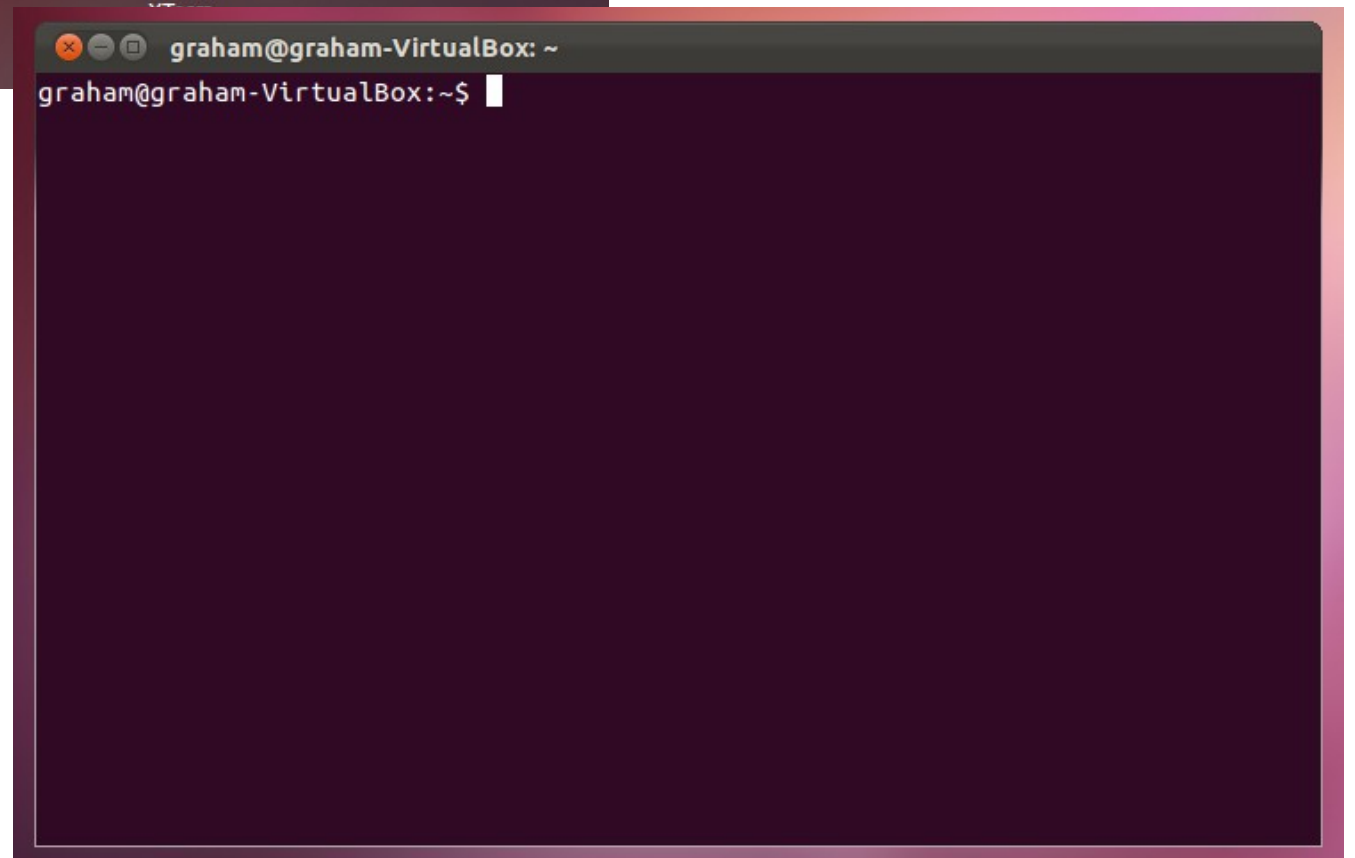
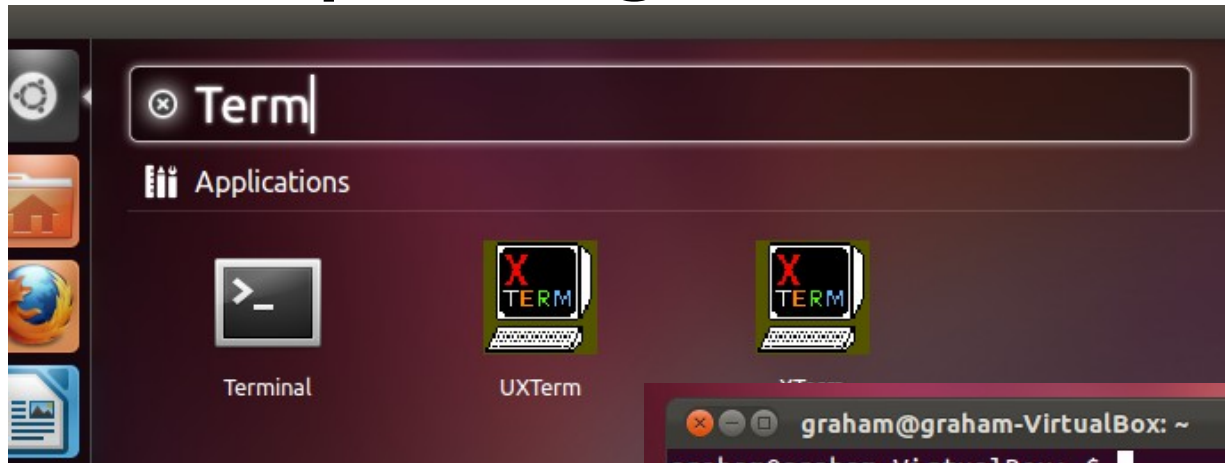
Directories

- Bin: essential system programs
- Boot: files needed for booting including the kernel
- Dev: “virtual” files that correspond to the devices in the computer
- Etc: configuration files
- Home: Users' home directories
- Lib: essential libraries and kernel modules
- Opt: third-party software sometimes installed here
- Proc: “virtual” files that contain the state of the system
- Root: Home directory for the superuser
- Sbin: essential system binaries, usually for the use of the superuser
- Usr: A similar structure to /, without /homes, /proc, /dev, /boot, etc...
Most of the system programs and libraries are installed under here.

Home directories

- Each user has their own, in /home
- e.g. /home/graham, /home/philip
- All user files and settings generally kept in here.
- For security reasons, most normal users can't write to places outside their home folder

Opening the terminal (“bash”)



Beginning bash

- Programs executed in the current directory
 - Your home (abbreviated as ~) by default
- General format of commands:
 - `<command> -<options> <file>`
- Why is the last argument usually a file?
 - “Everything is a file”

Beginning bash

- What's in the current folder?
 - `ls`
- Changing to a new folder
 - `cd <folder>`
- Make a new folder
 - `mkdir <folder>`
- Delete a file:
 - `rm <file>`
- Delete a folder and all contents:
 - `rm -rf <folder>`
- Move a file:
 - `mv <src> <dest>`
- Copy a file:
 - `cp <src> <dest>`

Folder=directory,
interchangeably

Permissions

- Permission format:

- rwx rwx rwx

Type User Group Others

```
graph TD; A["- rwx rwx rwx"] --- B["Type"]; A --- C["User"]; A --- D["Group"]; A --- E["Others"];
```

-:file
d: directory
s: Setuid
l: Link

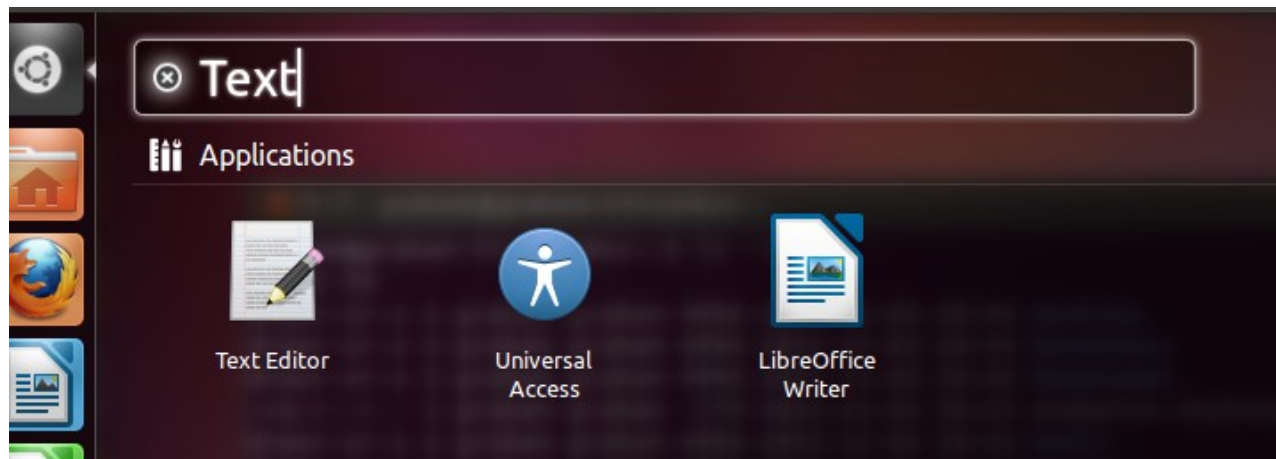
r: Read
w: Write
x: eXecute

Every file has a set of permissions, and an owner user and an owning group

Viewing/changing permissions

- View permissions:
 - `ls -l`
- Change permissions with `chmod`:
 - `chmod {ugo}{+-}{rwx} file` e.g. `chmod o+rx myfile`
- Change owner with `chown`:
 - `chown <user>:<group> <file>`

Text editor



A simple bash script

```
#!/bin/bash
```

```
echo 'I am a bash script. Hello!'
```

Running the bash script

```
$ ./script1.sh
```

```
bash: ./script1.sh: Permission denied
```

```
$ ls -l
```

```
-rw-r--r-- 1 graham graham 47 Nov  8 15:53 script1.sh
```

```
$ chmod u+x script1.sh
```

```
$ ls -l
```

```
-rwxr--r-- 1 graham graham 47 Nov  8 15:53 script1.sh
```

```
$ ./script1.sh
```

```
I am a bash script. Hello!
```

A more complicated bash script

```
#!/bin/bash
```

```
for i in `seq 1 10`; do  
    echo Count: ${i}  
done
```

Back ticks (`)
tell bash to execute the
command in the ticks

seq 1 10 gives a list from
1 to 10

```
for i in 1 2 3 4 5 6 7 8 9 10;
```

`${ }` references the
current value of the variable

Script output

```
$ ./script2.sh
```

```
Count: 1
```

```
Count: 2
```

```
Count: 3
```

```
Count: 4
```

```
Count: 5
```

```
Count: 6
```

```
Count: 7
```

```
Count: 8
```

```
Count: 9
```

```
Count: 10
```

To be continued...

- So far we've covered:
 - Applications that have the functionality of their Windows counterparts
 - Directory structures and permissions, basic shell usage
 - Some simple shell scripts
- Next time we'll look at:
 - Some more advanced shell scripting
 - Useful shell commands