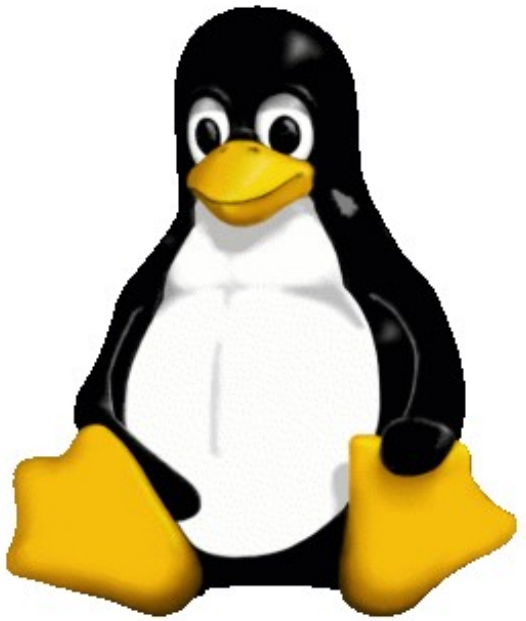
Computing in Fusion

THE UNIVERSITY of York

Dr Ben Dudson

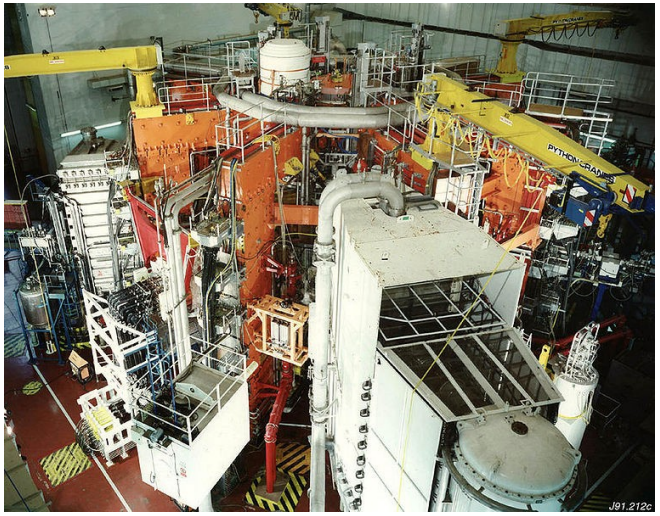


Don't fear the penguin...

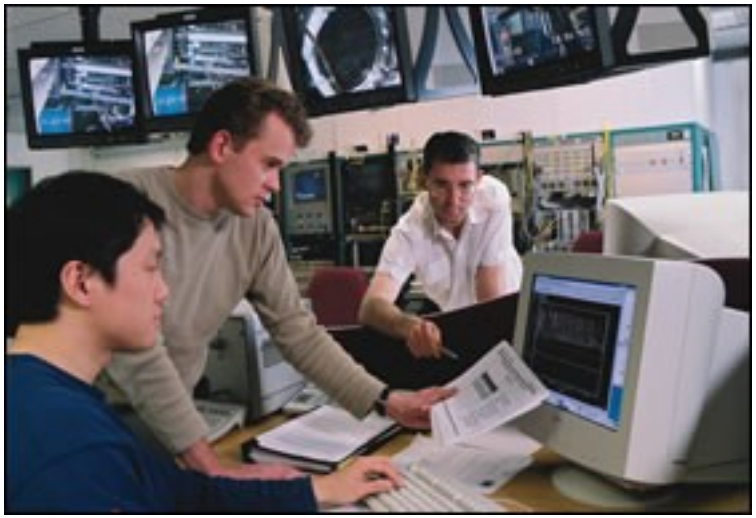
Computing in Fusion

Experiments

- Experiments too large/complex to be done “hands-on”
  - Everything is computer controlled
- Each plasma discharge generates gigabyes of data
  - Usually too much to analyse manually
  - Same analysis needs to be performed hundreds of times
  - Usually automated by writing analysis codes using IDL



Joint European Torus 1991. Photo EFDA/JET

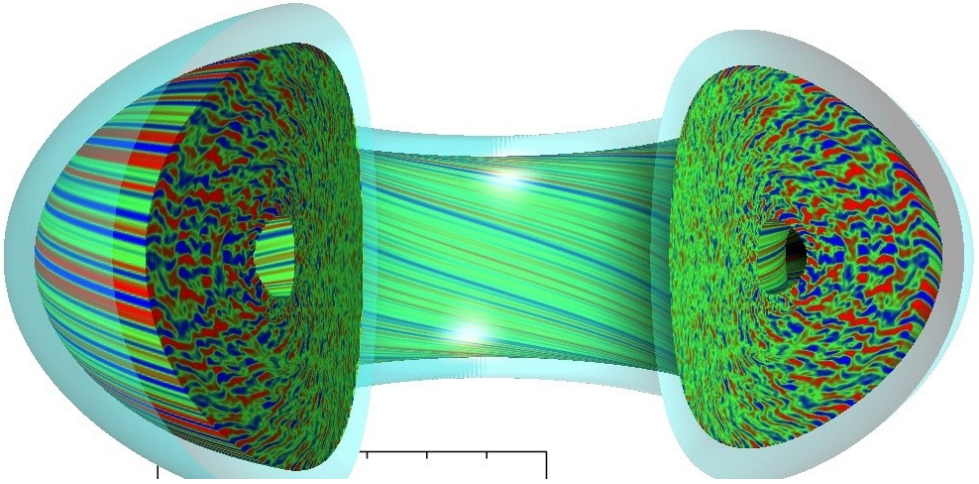


MAST control room 2001/2. Photo UKAEA

Computing in Fusion

Theory

- Some theory can be done analytically
  - Often need to solve resulting equations numerically
  - Applying to experimental situations usually requires some numerical work
- Simulations are now a key part of fusion research
  - Gain understanding of physical processes
  - Predict performance of future devices to aid design



GYRO simulaiton. Image: G.Hammet, General Atomics

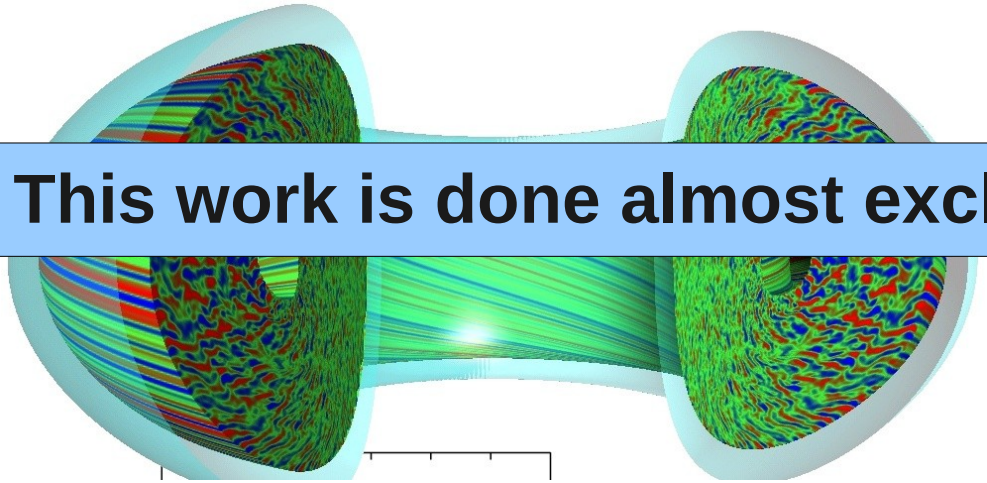
Franklin Cray XT-4. Photo: NERSC

Computing in Fusion

Theory

- Some theory can be done analytically
  - Often need to solve resulting equations numerically
  - Applying to experimental situations usually requires some numerical work
- Simulations are now a key part of fusion research
  - Gain understanding of physical processes
  - Predict performance of future devices to aid design

This work is done almost exclusively on Linux machines.



GYRO simulaiton. Photo: G.Hammet, General Atomics

Franklin Cray XT-4. Photo: NERSC

Computing at York

Courses involving computing:

- Experimental lab. ICF this term, MCF next term
  - Analysis of experimental data using IDL under Linux
- Computational lab. Term 2
  - Programming simulation codes in C or FORTRAN (Linux)
- High Performance Computing. Term 1 (optional)
  - Parallel programming in C or FORTRAN (Linux)
- Projects. Term 3, summer

This is a Fusion course, not a computing course.

But

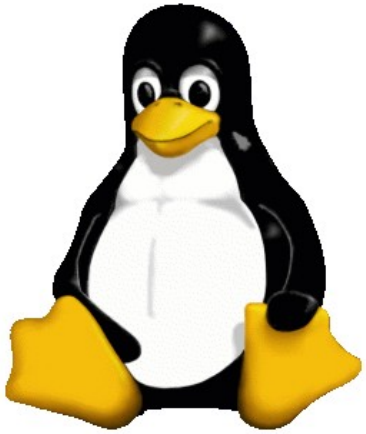
Programming is one of the vital tools you need to study Fusion physics

You need to be comfortable with using Linux

What is Linux?

- Started as a hobby by Linus Torvalds (1991)
- Combined with tools from the GNU project (started 1983)
- One of a family of UNIX-like operating systems
  - Others include *BSD, Solaris and Mac OS X




What is Linux?

- Started as a hobby by Linus Torvalds (1991)
- Combined with tools from the GNU project (started 1983)
- One of a family of UNIX-like operating systems
  - Others include *BSD, Solaris and Mac OS X
- Freely available (including source code) under GPL
- Easily customisable, so comes in many flavours
- Developed by companies and thousands of volunteers
- Installed on everything from mobile phones to supercomputers
- Supported by companies such as IBM, Compaq, Oracle, Sun, Novell, HP, ...

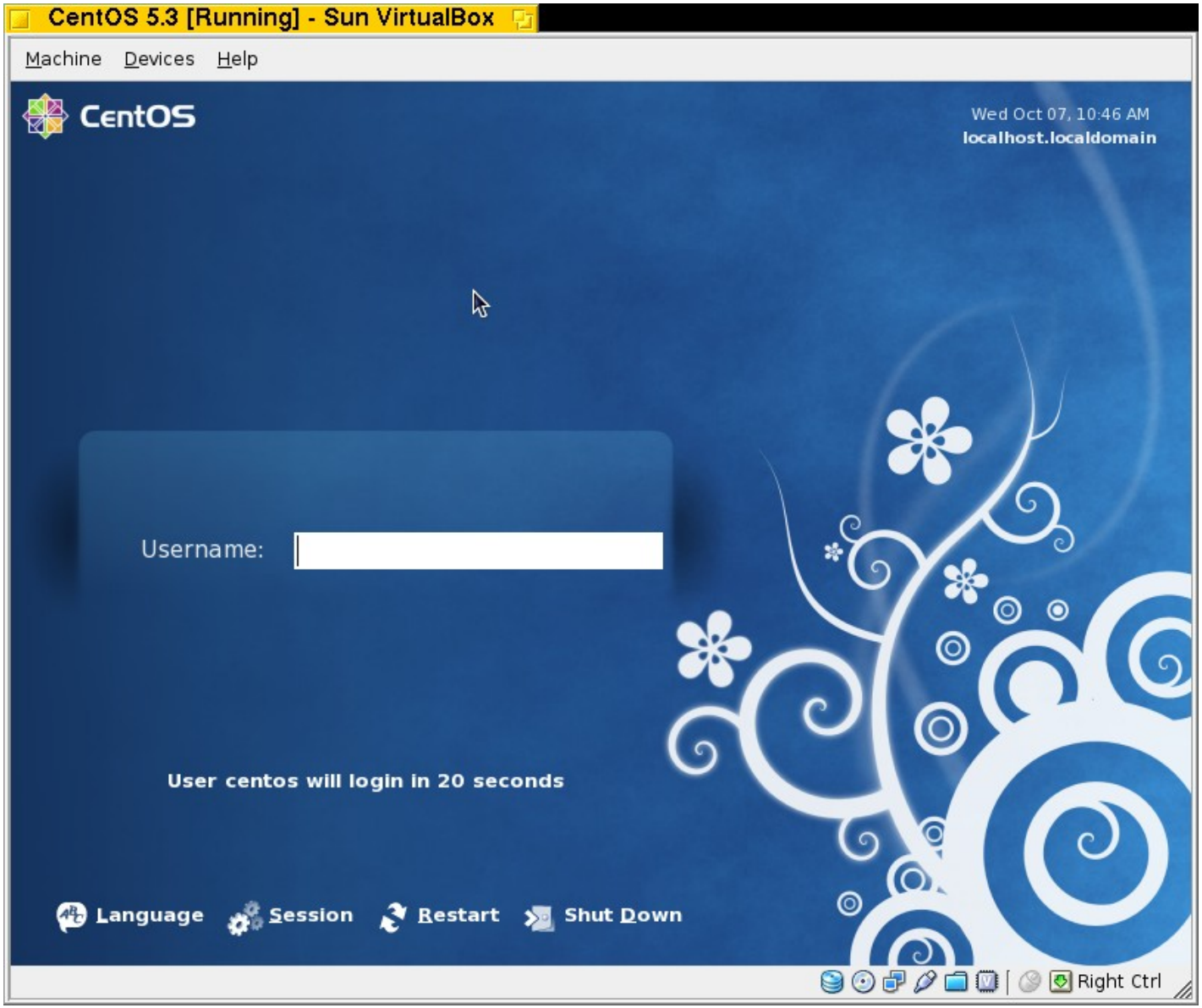
Linux distributions

- Linux can be easily customised for different purposes
- Many different “distributions” of linux
- Some popular ones:
  - Ubuntu (Home, ease of use)
  - Red hat, CentOS (Enterprise)
  - Debian
  - OpenSUSE

- If you can use one, you'll have no problem using another
- Bigger distinction is between desktop managers.
  - Determine appearance, layout of desktop
  - Common ones are Gnome, KDE and XFCE
  - Many other options to suit all tastes

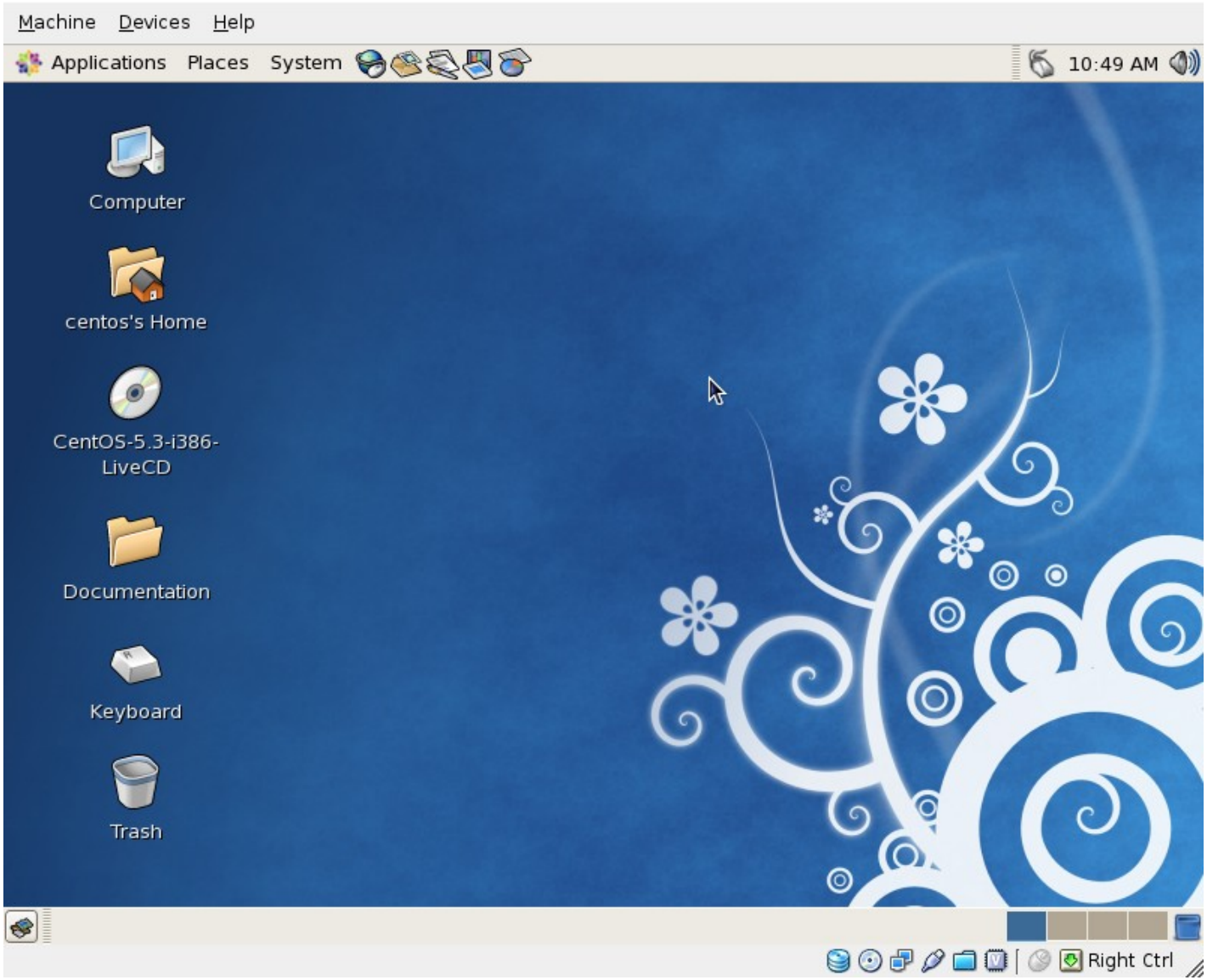
Linux on your laptops

- Your laptops can run Linux in a Virtual Machine



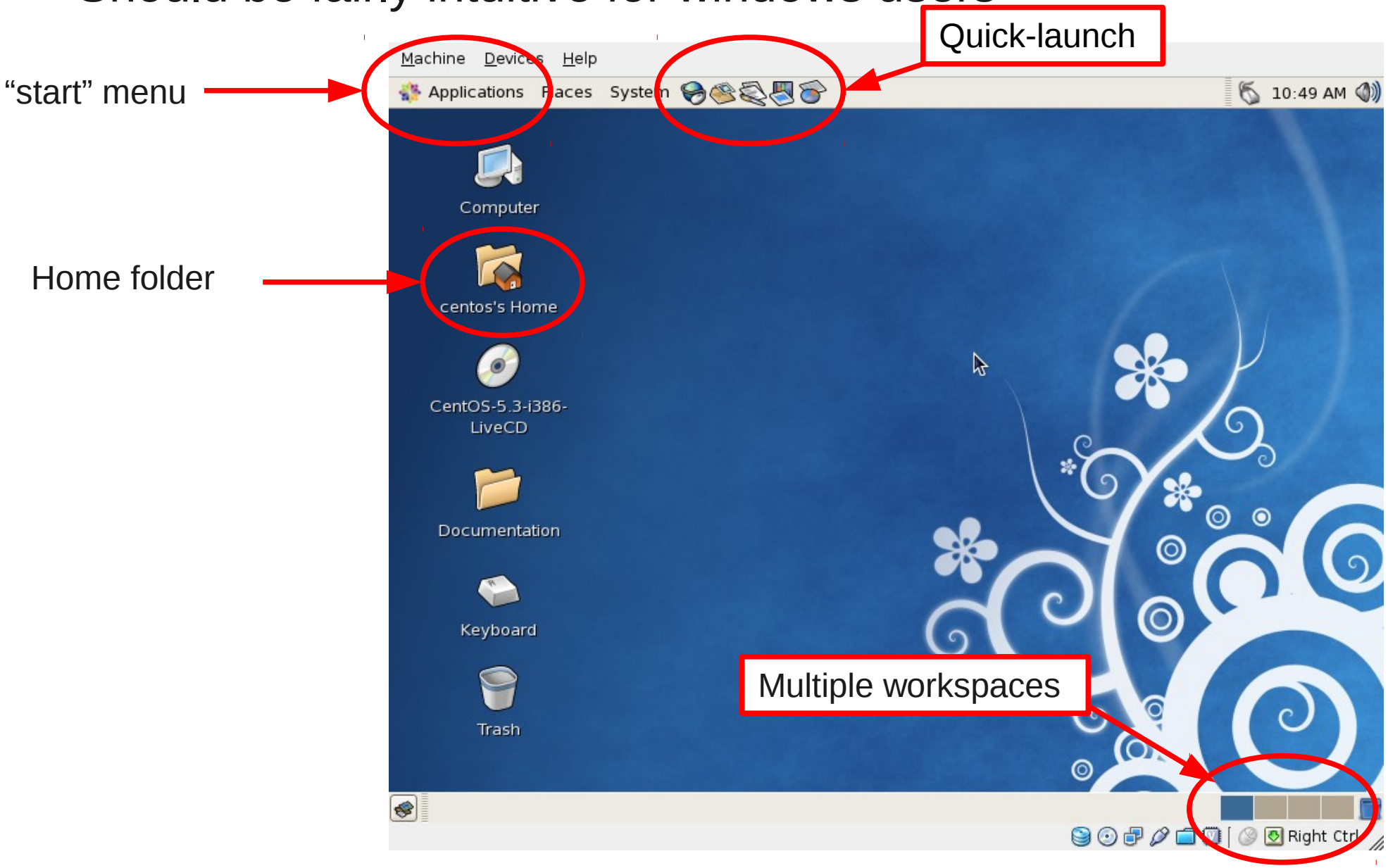
Gnome desktop

- Default is Gnome desktop, but KDE is also installed



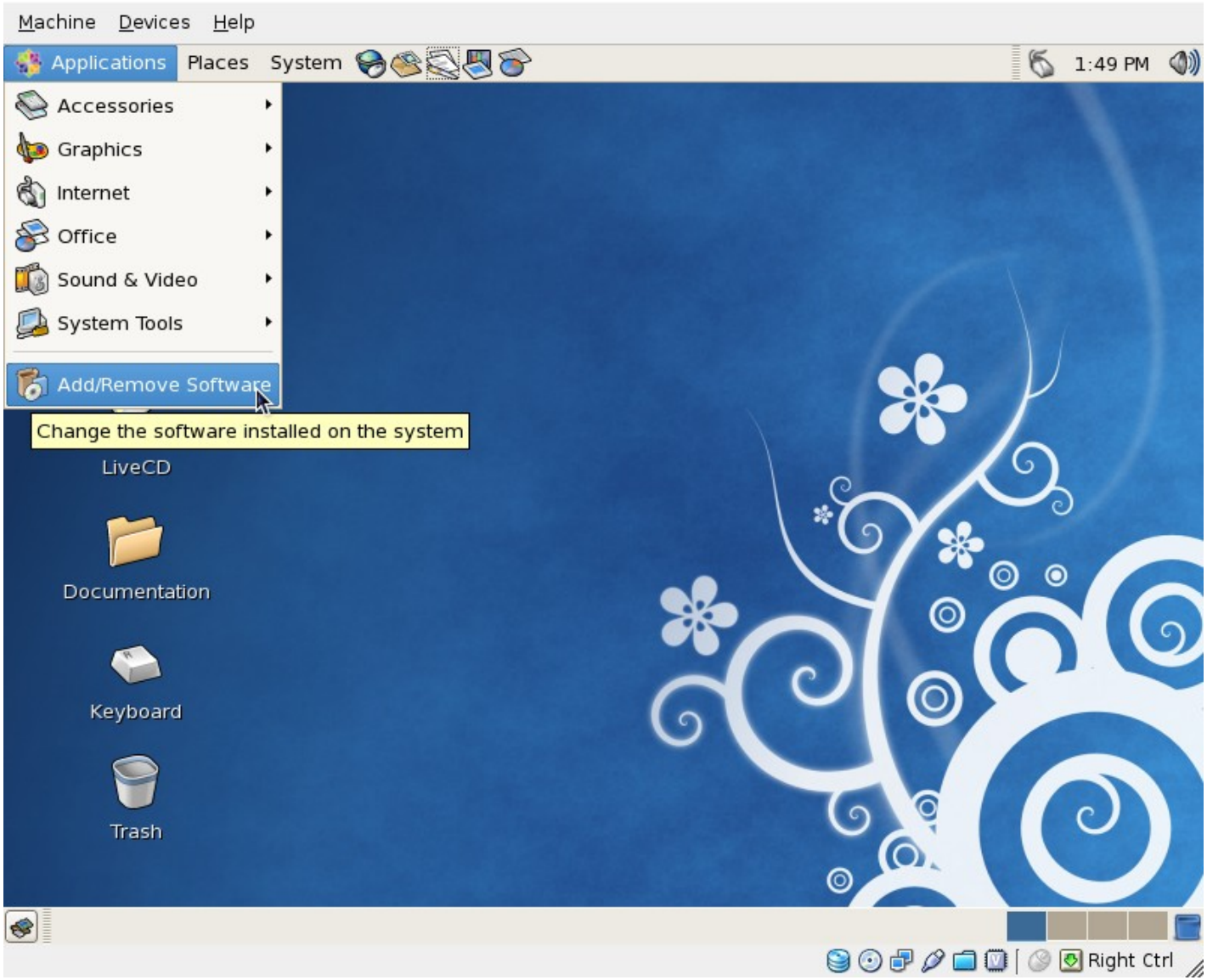
Gnome desktop

- Should be fairly intuitive for windows users



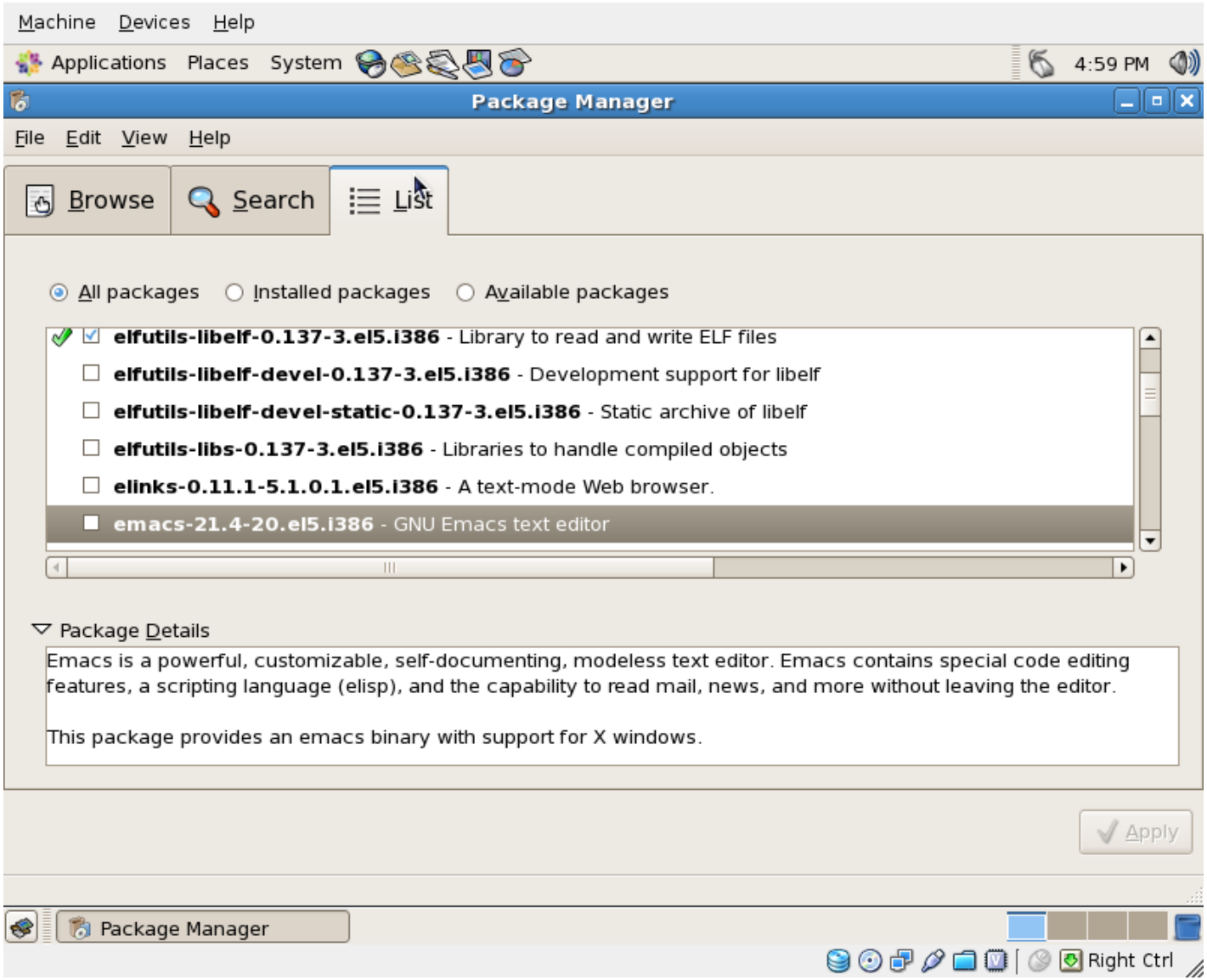
Installing software

- Package managers contain thousands of free programs



Installing software

- Package managers contain thousands of free programs



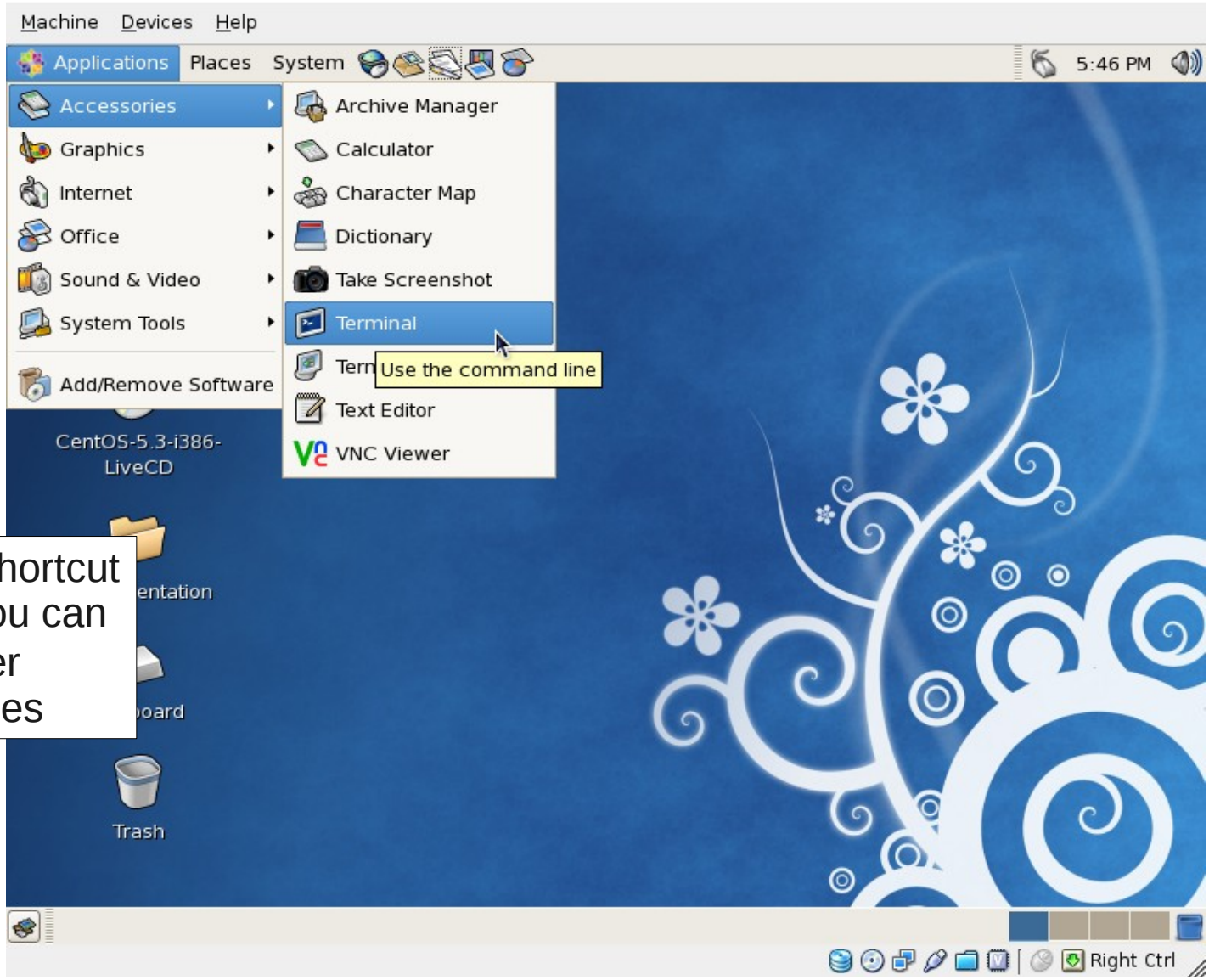
Some useful programs

- Coming from another operating system (Windows), you will find many programs you're used to aren't available
- Most of the time there will be other programs available with similar (or better!) functionality

- Text editors: kate, gedit, emacs, vim, nano
- Image/figure editors: GIMP, IPE, inkscape, xfig
- Office suite: OpenOffice, koffice, abiword
- Desktop publishing: scribus
- Music/video: amarok, totem, rhythmbox, vlc, ...

Terminal window

- Most useful program on your laptop (more later...)



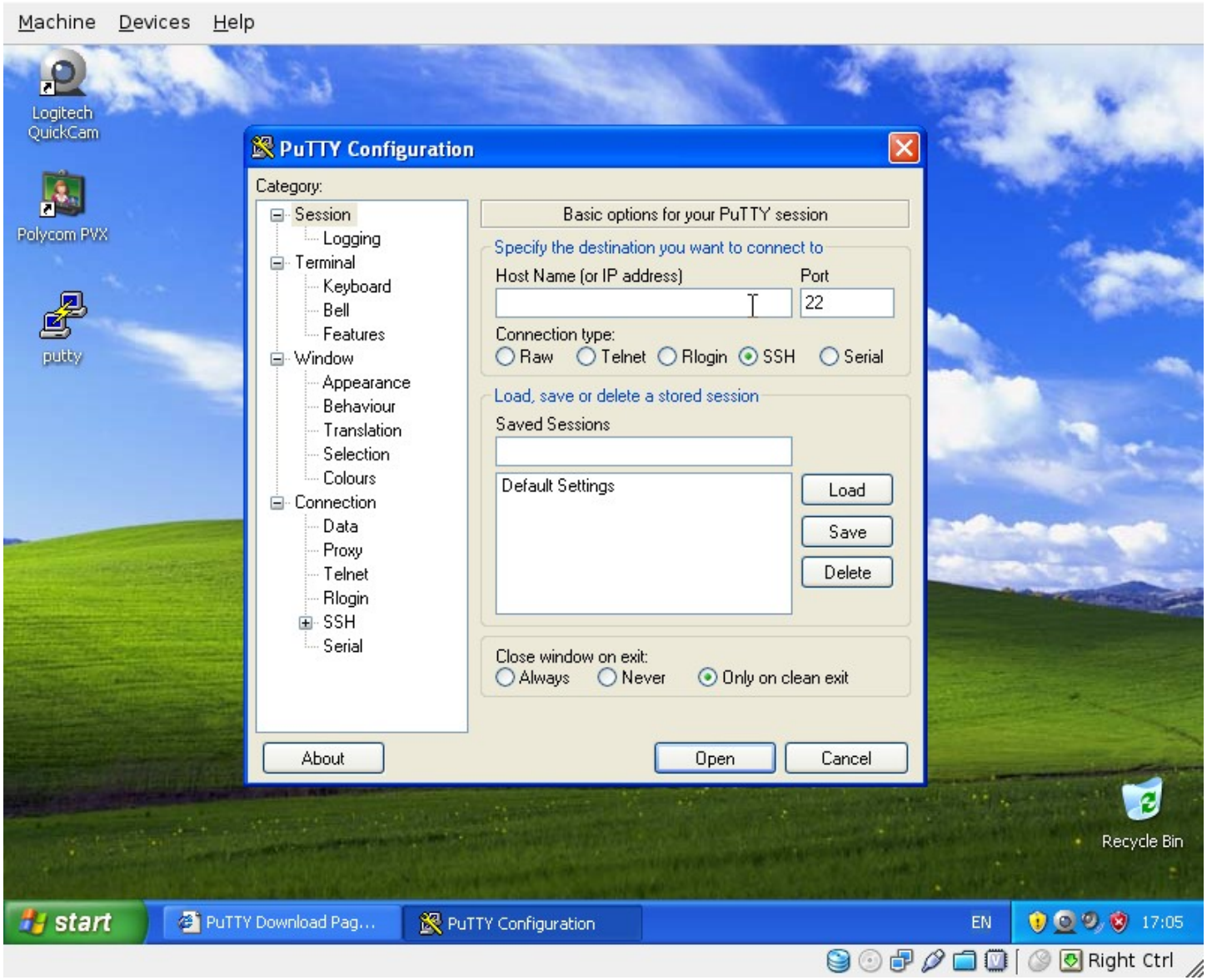
Often has a keyboard shortcut (e.g. `Ctrl-Alt-t`). You can set shortcuts under System->Preferences

Connecting from Windows: Exceed and Putty

- To display graphics, UNIX systems use a protocol called “X-windows”, “X”, or “X11” (since it's the 11th version of the protocol)
- Works over the network, so allows results from servers and supercomputers to be displayed on workstations or laptops.
- For this to work on Microsoft Windows, you need to run a separate program: an X server.
- The one installed on your laptops is Hummingbird **exceed**
- Free alternatives are Xming and Cygwin/X

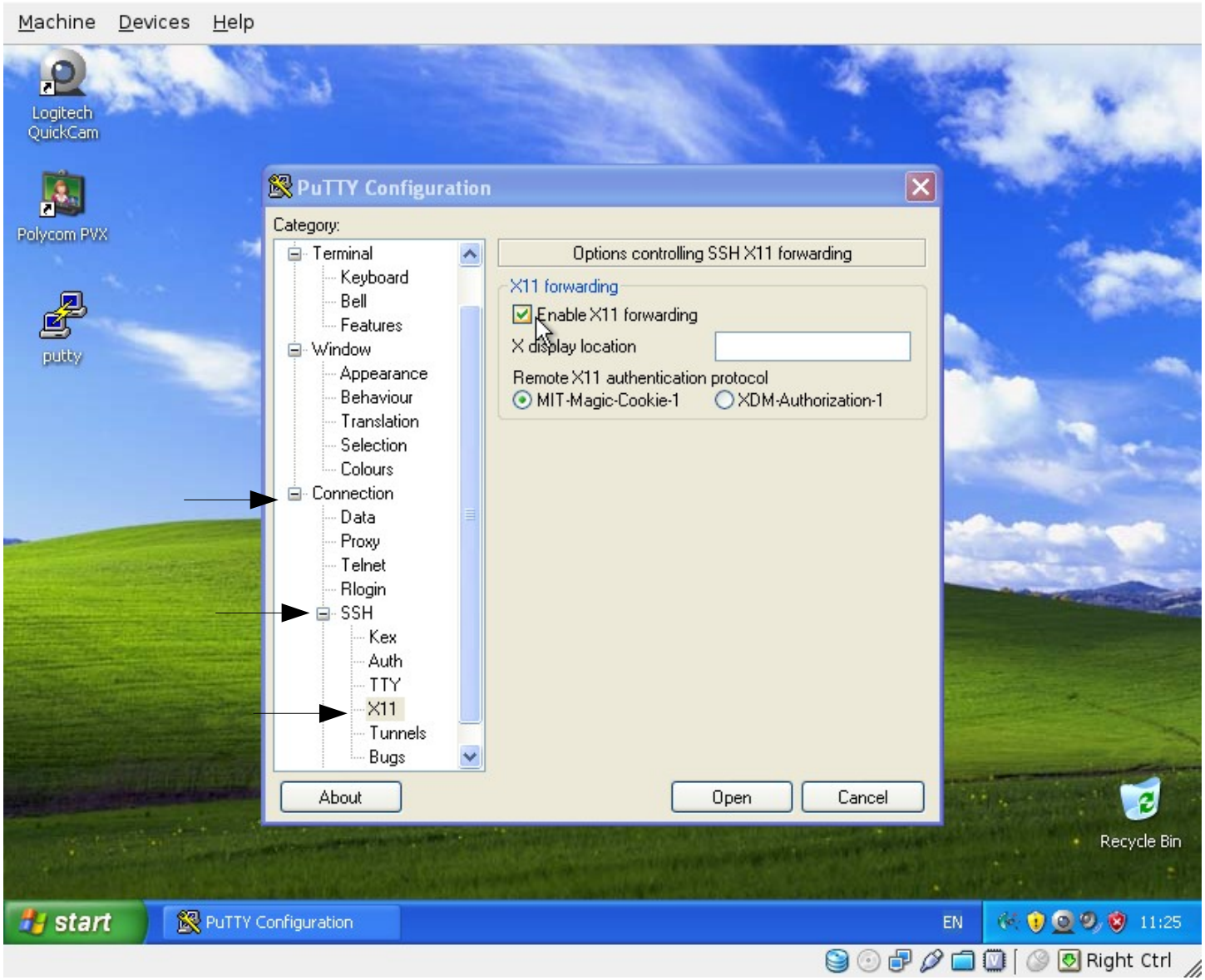
Connecting from Windows: Exceed and Putty

- Putty provides a terminal window to another machine



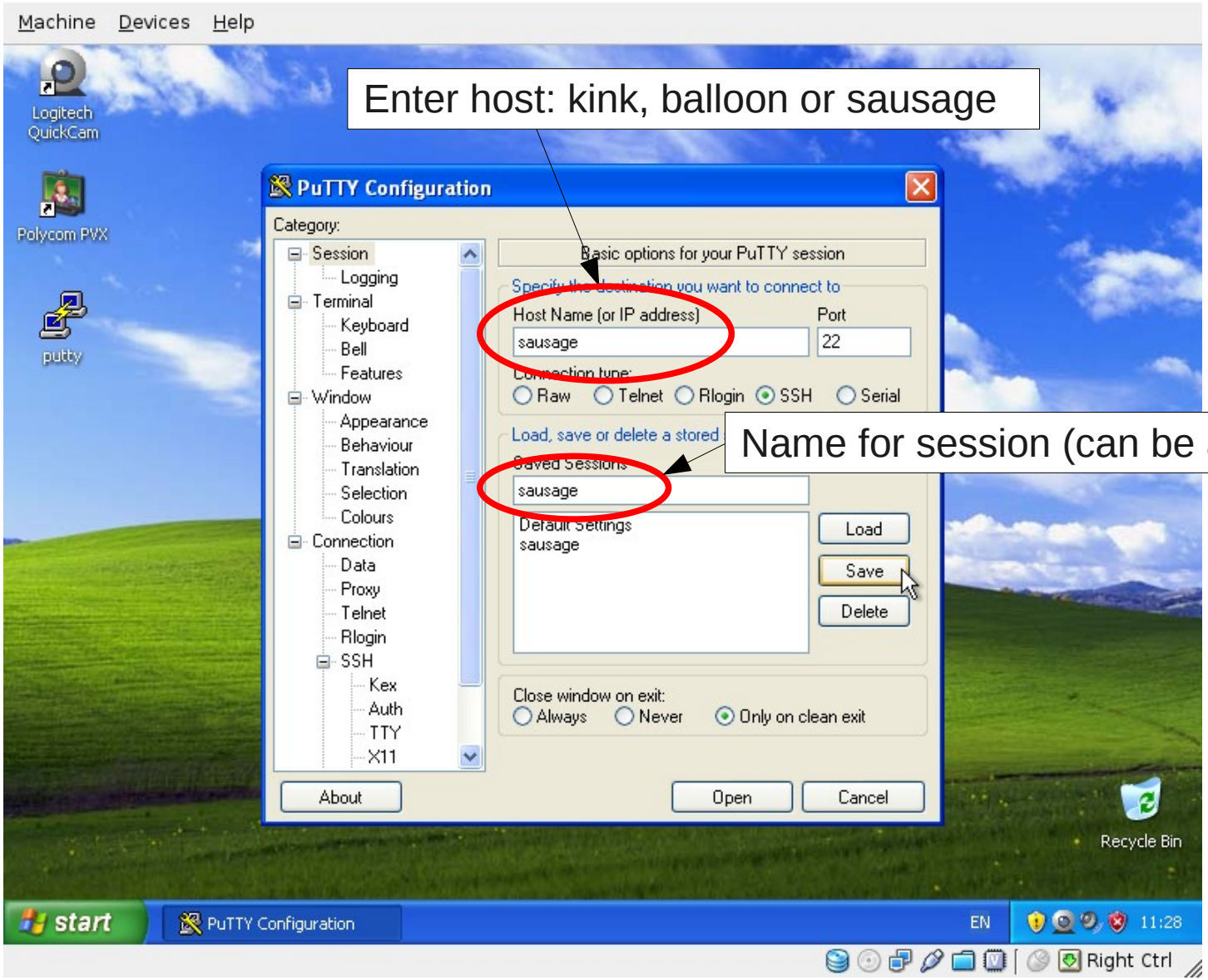
Connecting from Windows: Exceed and Putty

- Make sure you enable X11 forwarding in the settings



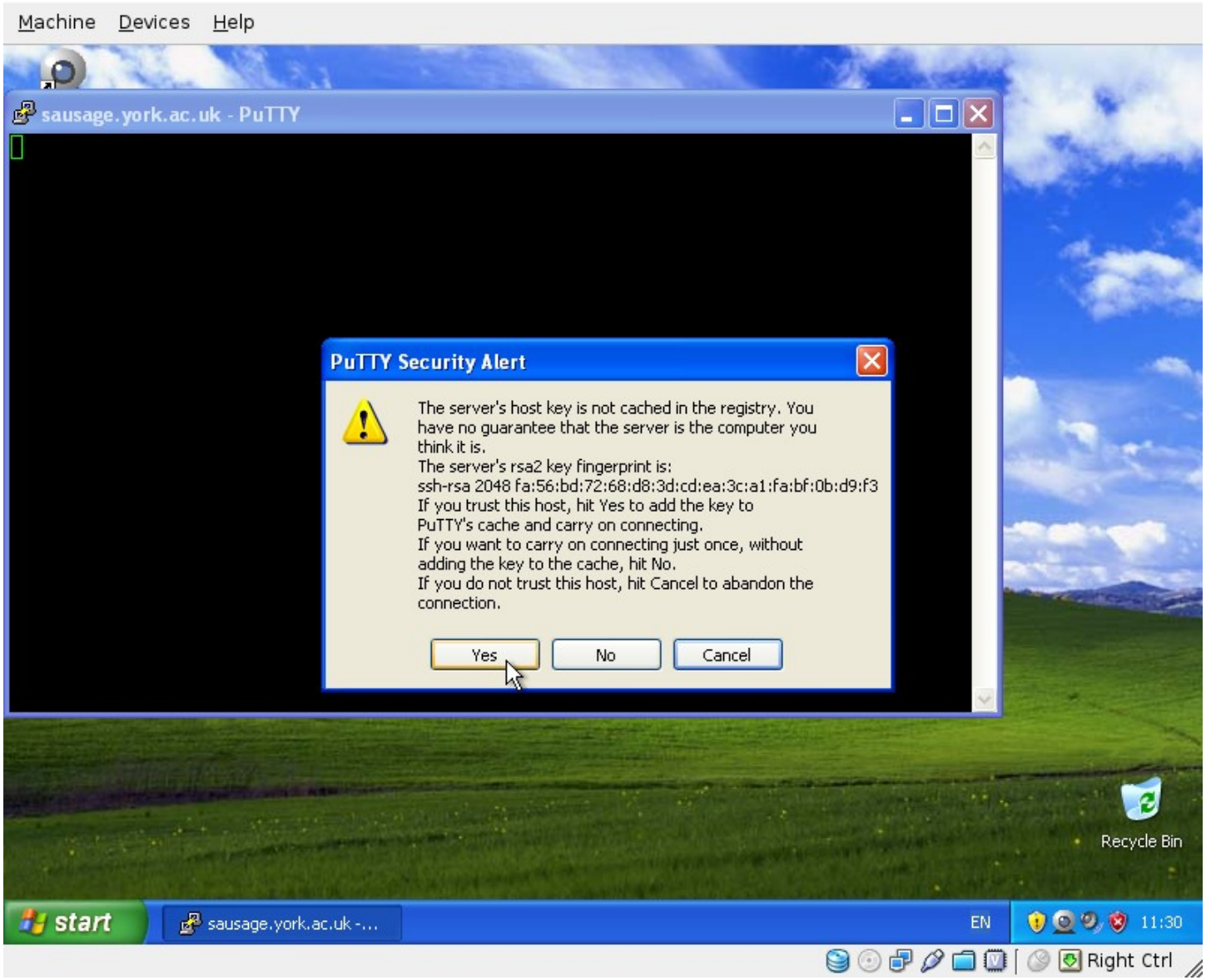
Connecting from Windows: Exceed and Putty

- Save settings to use next time



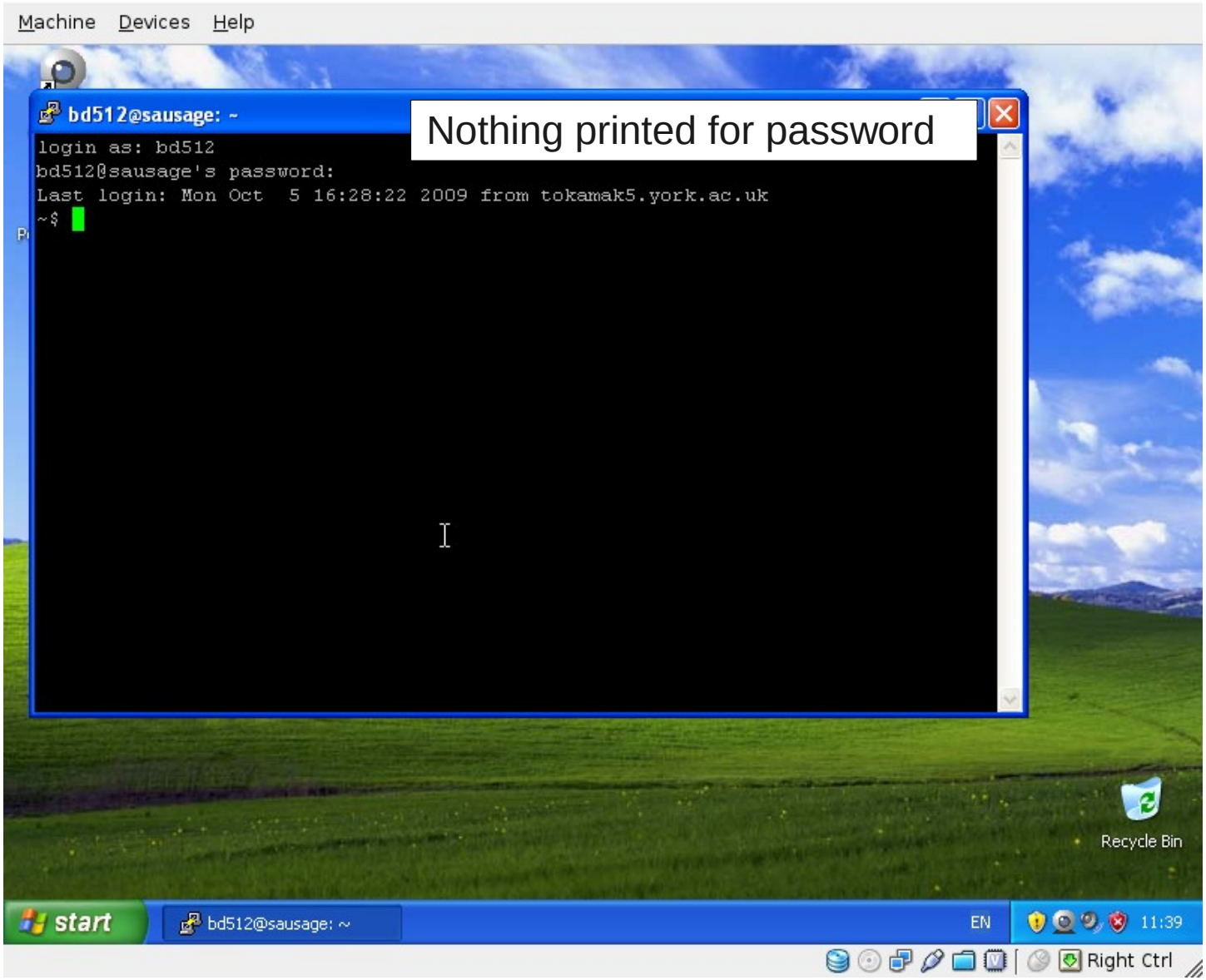
Connecting from Windows: Exceed and Putty

- SSH checks the identity of the host, so asks first time



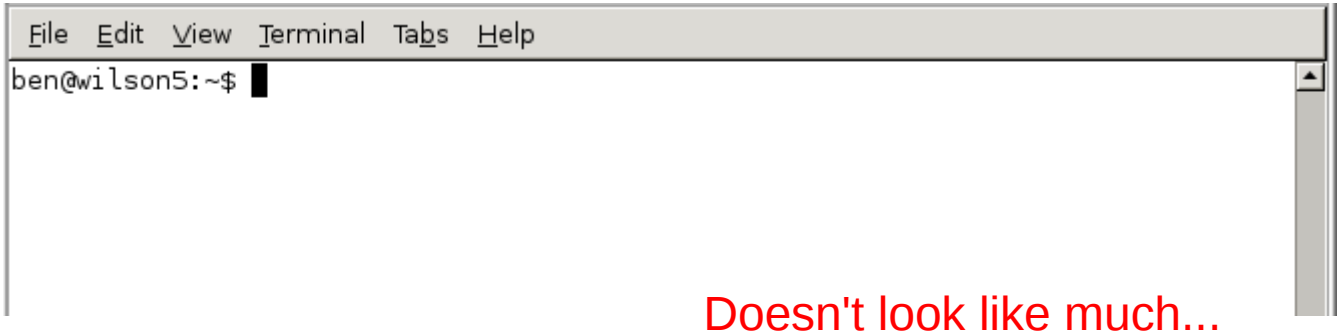
Connecting from Windows: Exceed and Putty

- Now have a Linux shell (terminal) on sausage



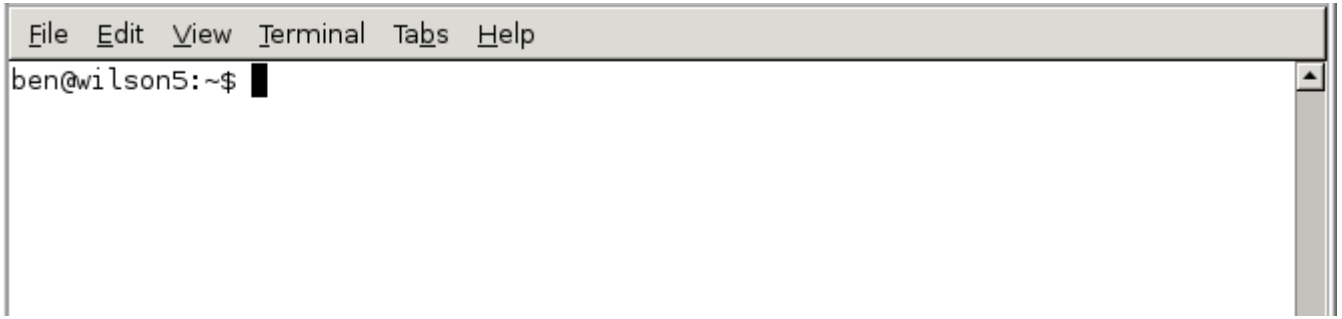
Command line

- Graphical tools are easy to use, but the real power of UNIX is the command line



Command line

- Graphical tools are easy to use, but the real power of UNIX is the command line



- Run a program by typing its name followed by options
- Using remote servers / supercomputers becomes no different to local machine
- Output of one program can be piped into another
- You can write scripts to automate many things
- Steep learning curve initially
- After a while you'll find it much quicker for many tasks

UNIX commands

UNIX commands have the form:

```
$ command  -options  target
```

- List directory: **`ls`**
- Print current (working) directory: **`pwd`**
- Make directory: **`mkdir name`**
- Change directory: **`cd dirname`**
- Copy files: **`cp from to`**
- Remove files: **`rm name`**
- Remove directories and files: **`rm -r name`**

Means recursive
Works for **`rm`** and **`cp`**

Directories in UNIX

Move up one level

```
$ cd ..
```

Directories are separated by forward slashes:

```
$ cd ../usr/bin/
```

To go back to your previous location:

```
$ cd -
```

Your home directory will probably be

```
$ cd /home/<your username>/
```

Which can be abbreviated to

```
$ cd ~
```

Or just

```
$ cd
```

Wildcards

When specifying files, wildcards can be used, so

```
$ ls *.txt
```

Lists all files ending in ".txt"

```
$ rm */*.txt
```

Deletes all files ending in .txt which are in a subdirectory

Other wildcards:

- **?** matches a single character
- **[ ]** matches one of the characters in brackets

so

```
file?.[oc]
```

Matches "file1.o, filea.c" but not "file01.o" or "file1.f"

Searching

- Search the database for a file name: **`locate string`**

- Find files by name: **`find . -name “*string*”`**

Start from the current directory

Search by file name. Many other options

- Find text inside files:

`grep -i “text” file1 file2 file3 ...`

Specifies case insensitive

Redirection and Pipes

- Send output to a file
  `$ date > file`
- Append to a file
  `$ cal >> file`
- Print contents of file
  `$ cat file`
- Pipes: Send output of one program into input of another
  $ **`cat file | wc`**
  $ **`cat file | tr 'a-z' 'A-Z'`**
- Can chain an arbitrary number of programs
- Some useful programs: **`sort`** (sort alphabetically or numerically), **`sed`** (search and replace), **`awk`** (text extraction), **`tee`** (write to file and output), ...

Other things...

Some other things to look up and try (just for a giggle)

- **BASH scripting**. When you enter commands at the terminal, they are interpreted by a “shell” before being passed to the operating system. The shell you're using is called the Bourne Again Shell (improved Bourne Shell)
- **Perl**, the “swiss army chainsaw”. Powerful text processing and scripting language
- **Gnuplot**. A plotting program which can be scripted (so you can automate generating plots)
- Network-related programs: **`curl`** (fetch webpages), **`nc`** (netcat, very flexible), **`mail`** (scriptable mail sending)

Getting help

Man pages

Man command is standard on UNIX-like systems.
See for example:

```
$ man man
```

Course material and links

http://www-users.york.ac.uk/~bd512/

Plasma forum

http://plasmaforum.york.ac.uk

Please post questions to the forum, rather than emailing me directly, so that others can use the replies.

Ubuntu online forums

http://ubuntuforums.org/

Look around... Google Is Your Friend

Programming tools

- IDL `$ idl`
  - Will be used for experimental labs
  - Installed on your laptops under linux
  - Lectures and problem classes on Thursday and Friday
- C
```
$ gcc myfile.c
$ ./a.out
```
  - One option for the computational lab
  - GNU Compiler Collection
- FORTRAN
```
$ gfortran myfile.f
$ ./a.out
```
  - Alternative to C for computational lab
- Python, Octave
  - Widely used programming tools (not part of this course)

Connecting to other machines: SSH

```
$ ssh -X username@host
```

- Secure connection to another machine (host)
- -X (or -Y) options allow you to run graphical programs on the remote machine, but display on your desktop

- Has many other options (see man page)
- A particularly useful feature is port forwarding (-L option)

Copying files around: SCP

- Copy files and directories between machines

```
$ scp [-r] <source> <destination>

$ scp somefile   username@host:~/directory/
$ scp username@host:~/directory/somefile .

$ scp -r somedirectory  username@host:~/
$ scp -r username@host:~/somedirectory .
```

Summary

- Whether you're doing experiments or theory, it's very likely you'll need to do some programming
- At York, as at all Fusion labs, this is done on Linux systems
- In the first term (especially this week), aim to become familiar with using Linux.
- Strengths of UNIX systems are the programming environment, and flexible ways to connect to other machines.

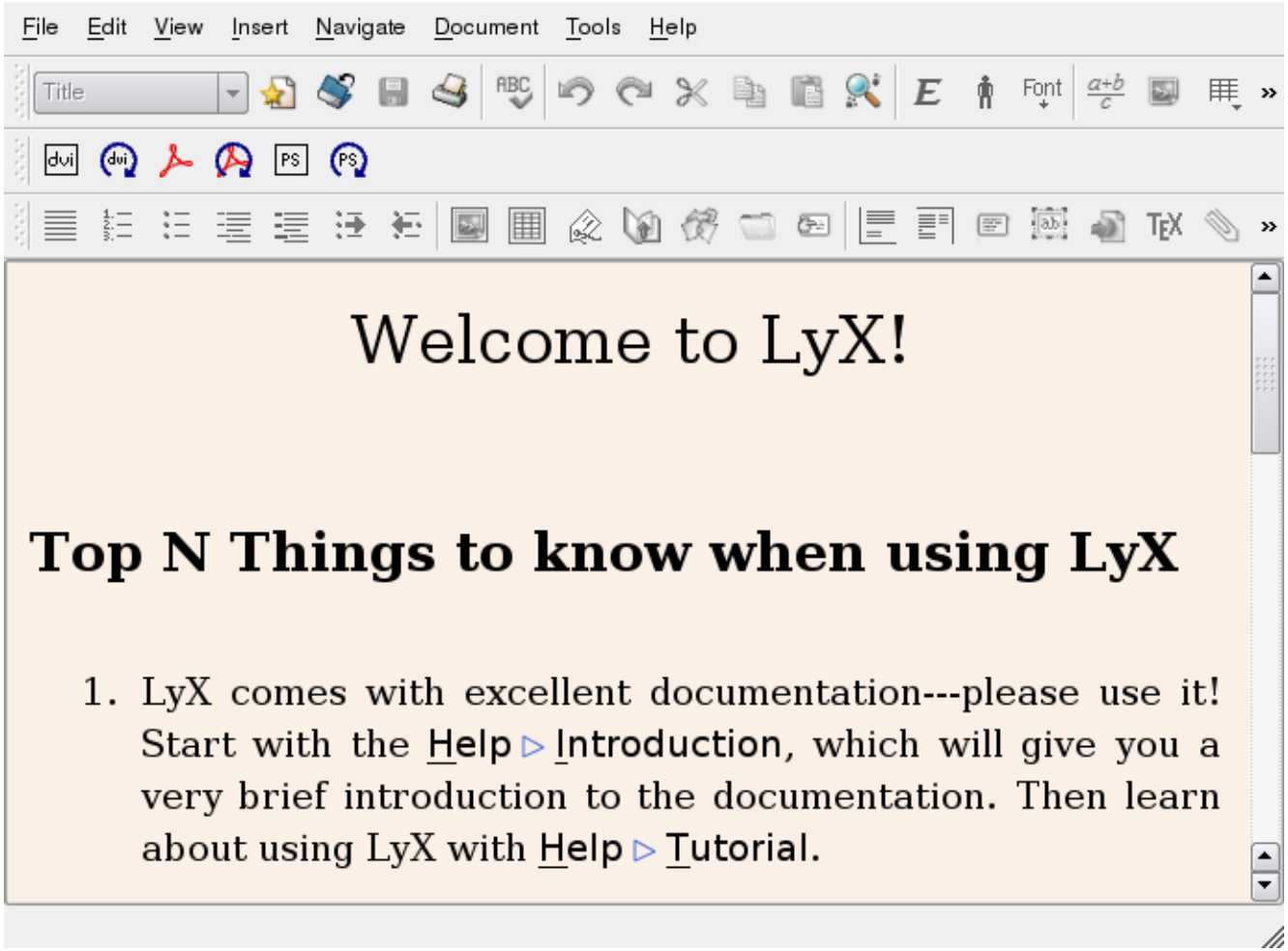
- You can connect to York Linux servers either through the virtual machine on your laptop, or using Exceed & Putty.

Side note: LaTeX

- Widely used system for typesetting scientific documents
- Not required for any part of the course
  - You're free to use any system you want: MS Office, OpenOffice, etc.
- Highly recommended, particularly for preparing large documents (e.g. Theses) with lots of references and equations.
- Makes high-quality typesetting much easier
- As with Linux, programming etc. a bit of a learning curve

LaTeX tools

- LyX – an document preparation system using LaTeX



Provides an easy-to-use interface if you don't know LaTeX

Can show you the source (View -> Source), so useful for learning LaTeX

LaTeX tools

- LyX – an document preparation system using LaTeX
- TeXnicCenter (http://www.texniccenter.org/)

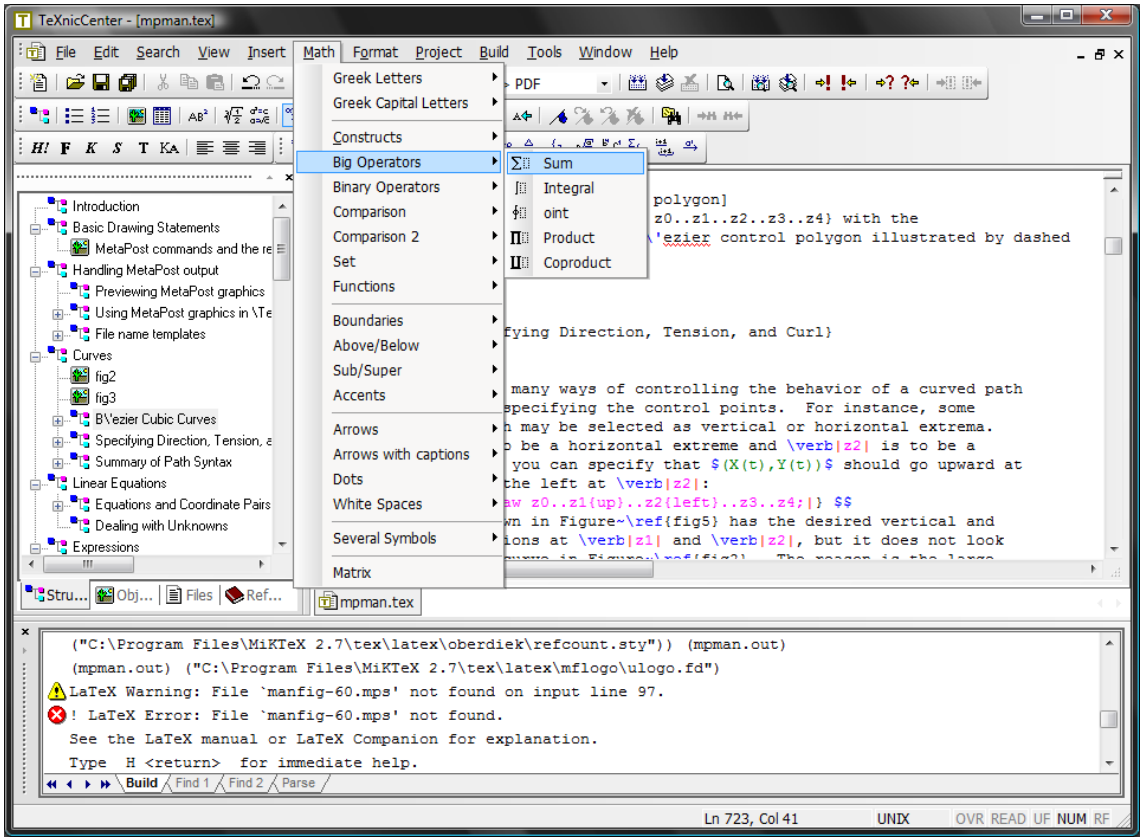


Image:
www.texniccenter.org

An integrated environment for LaTeX editing.
Requires some knowledge of LaTeX, but includes lots of help

Resources

See my teaching and links pages:

http://www-users.york.ac.uk/~bd512/

York plasma wiki page (internal only):

http://plasmawiki.york.ac.uk

Introduction to UNIX

http://www.doc.ic.ac.uk/~wjk/UnixIntro/

Have a look at lectures 1-6

→ Try some of the exercise sheets