# Systems Introduction:
# The Basics [*]

Roger Burroughes <roger@inf.ed.ac.uk>
Alison Downie <alisond@inf.ed.ac.uk>

September 2015

## Overview

This document is an introductory guide to the **D**istributed **I**nformatics **C**omputing **E**nvironment (**DICE**). It also aims to give you a short introduction to a few simple UNIX commands and explains how you can get help should you have any problems.

## Jargon

These notes assume that you are familiar with current, common computing terms. The world of Unix has a slightly different 'jargon' to that of the Windows PC or Mac. For example, *folders* are usually referred to as **directories** and *terminal windows* may be referred to as **shell windows** or just **shells**. These alternative terms, and new ones, will be explained when they first occur. If you are unfamiliar with any term that is used or don't understand the explanation given, then please ask.

## Feedback

This document is under continuous revision, and any feedback about errors, omissions, or other content suggestions, would be appreciated. Please send comments to either of the e-mail addresses above.

## So what is DICE?

The term DICE encompasses all those computing services developed, maintained and supported by the School's computing staff.

The term DICE desktop is used to refer to the Linux desktop PCs managed by the School. These can be found in the public spaces and meeting rooms of the Informatics Forum and in the Forrest Hill teaching labs. Many staff and research postgraduates also choose to have a DICE desktop.

You will require a DICE account to login to any DICE desktop or to use any of the DICE services.

---

[*]   Version for 2015/2016

# 1.    Getting Started

## Logging on

Everyone who uses the DICE facilities has a *username* and a *password* that uniquely identifies them to the Informatics system, and allows access to those facilities. You should already have received an e-mail with instructions on how to create a DICE password on-line.  We use the same username (UID) that the University allocates to you, but it is a ***different account**.* Your EASE[1] password (the one issued by University Central Services) will *not* give you access to the Informatics systems and, similarly, your Informatics password will *not* give you access to University Central Services. As a matriculated student, you are bound by the University Computing Regulations. You should read and make sure you understand them - they can be found at:

> http://www.ed.ac.uk/schools-departments/information-services/about/policies-and-regulations/computing-regulations

Network traffic is monitored and any irregularities are investigated and can result in disciplinary action (for example, downloading illegal or otherwise dubious material; accessing music or images via file-sharing protocols in an attempt to avoid copyright restrictions). You should also take note of the guidelines found in our "Best Practice Guide":

> http://computing.help.inf.ed.ac.uk/good-practice

Within the School of Informatics we have a customised version of a standard Linux environment (currently Scientific Linux 7) installed on all machines, plus additional software and utilities. These machines are referred to as **DICE machines**. All DICE machines should have the same login screen. This changes from time to time, but is currently a view of the Informatics Building and cityscape:
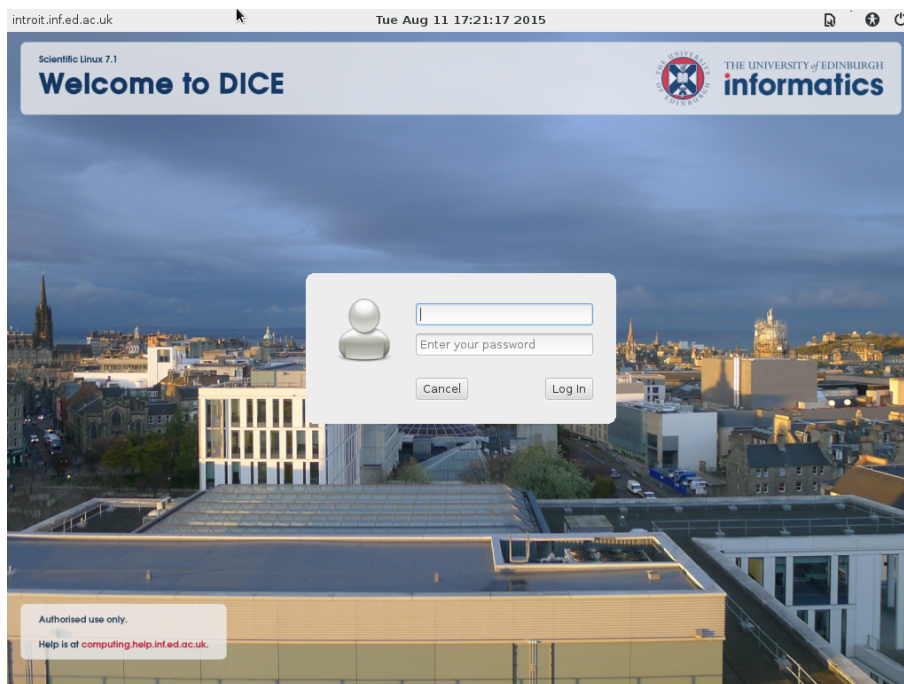


*Figure 1: DICE login screen*

---

1   **E**dinburgh **A**uthentication **SE**rvice

The login screen prompts you for a username and password. Once you have entered your username and password correctly, your session will start.

The *appearance* of the screen, icons, windows, etc., during a login session (called the **Desktop Environment**) is controlled by a **Desktop Environment Manager** and the one available by default in DICE is called **Gnome**. You can configure the look of your session yourself, and you only need to do this once - the configuration will be stored in your home directory, and will appear on whichever DICE machine you choose to use.

If you are familiar with any other *Window* or *Desktop Managers* that are available on DICE machines, then feel free to use them. You can select alternatives from the drop-down menu in the very top right-hand corner of the login screen - but you *must* have already typed your login name into the central dialogue box, or your selection will not take effect.



*Figure 2: Window Manager selection*

By default, first and second-year students will be allocated a quota of 4 GB **home directory space**, all other students are allocated 10 GB.

Home directory filespace is automatically backed-up overnight, and this backed-up data appears in your home directory as a **read-only** (non-modifiable) directory called "`Yesterday`". As it suggests, this is a copy (mirror) of your home directory from the day before[2], so is very handy should you delete any files by accident. If you lose a file, the required file can be copied from this read-only directory to its original (or any other) location.

NOTE that files in your `Yesterday` directory are *not* included in your quota.

---

2  The actual timing of the backup may vary - if a lot of data is being backed up, it may be more recent than yesterday.

## 2.      Desktop Environment (Gnome)

Once you have successfully logged in using the default Window Manager, you should see an empty desktop screen, with a menu bar at the top containing the ***Activities Overview*** button on the left-hand side, a clock and calendar, and system status icons in the centre, and the system menu on the right-hand side. Clicking on the *Activities* button displays default applications icons on the left-hand side (the "**dash**"), and the screen should then look something like *Figure 3*:
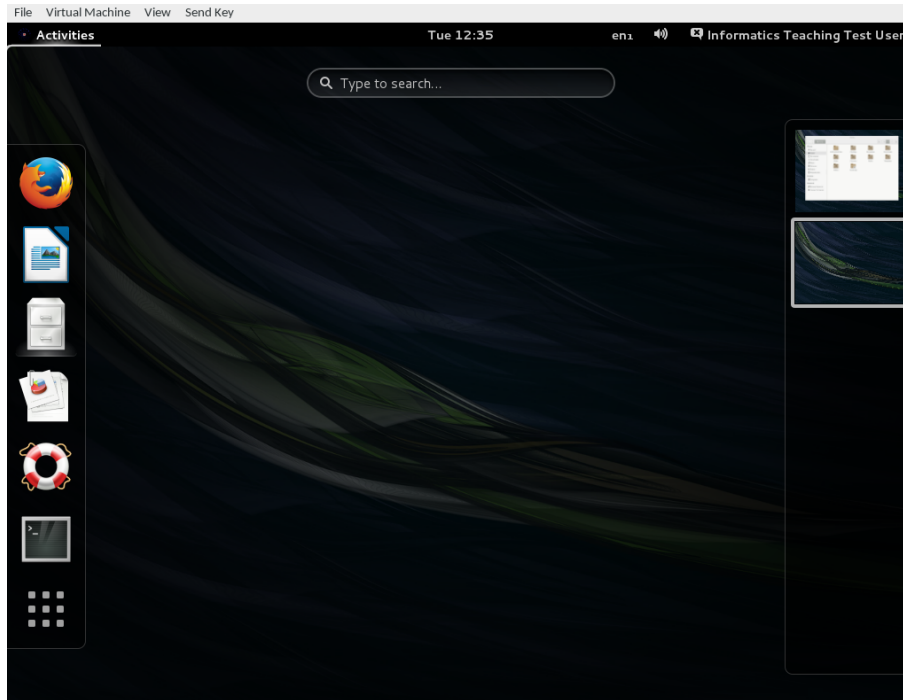


*Figure 3: The default Gnome desktop*

The DICE Gnome environment defaults to "click to type" (you need to move the mouse cursor over a window ***and click*** to make it active)[3].

On the right-hand side of the screen is a **workspace list** (it expands when you move the mouse over it) which keeps track of all the *workspaces* (*virtual screens*) that you have open. Click on any window icon in that list to make that window current.

The *Activities* button acts as a toggle for displaying applications. Click once, and you see the applications dash on the left-hand side, workspace list on the right-hand side. Click again, and only the blank desktop is displayed. Note the menu icon at the bottom of the applications list - click on this, and you get more applications - either ***Frequent*** or ***All*** applications (depending on which option you select at the bottom of the screen). Click on any application to run it.

When in *Activities Overview* mode, application windows can be dragged to a specific workspace in the workspace  list on the right. There is always an empty workspace at the bottom of the list - drag an application here, and a new, empty workspace will appear below it.

---

3    This can, if desired, be changed from the "Windows Focus Mode" option of the "Windows" section of the "Activities: Applications (All): Utilities: Tweak Tool" utility, where you can select "Mouse" or "Click".

## Doing Something

Although a windowing environment is easy to use, it does not always lend itself to making use of the full flexibility of Unix. To do this, instructions (*commands*) must be entered via the **command line** interface (CLI). This can be done using a **terminal window** (shell window). Click on *Activities*, and then on the terminal application icon in the dash on the left-hand side.

A terminal – or **shell** – window is where you type commands, and is similar to the Windows/DOS "Command Prompt" window. There should be a **prompt** at the left-hand edge of the shell window (which will appear when input is possible *in that window*), and anything that you type in that window will appear next to the prompt. By default, the prompt on a DICE machine is of the form:

```
[hostname]username:
```

(where *hostname* is the name of the machine, and *username* is the account name of the person logged in). This, like most things in Unix (Linux), is completely configurable.

## Doing Nothing

The default behaviour is for a DICE machine to sleep when it is not actively in use. This is a big energy saver. The machine will wake up automatically only when it needs to adjust its configuration, or with manual intervention. If you spot a lab machine with a flashing amber light, this usually means it is asleep, and can be woken up either by pressing the 'On' button once or by pressing any key on the keyboard.

If you do not type anything or move the mouse for several minutes, then your screen will go blank  and your session will lock to prevent unauthorised access.  If your screen is locked, hit a key or move the mouse and you will be prompted to enter your password to unlock the screen and continue your session.

**Please do NOT lock your screen if using the computer labs** - they can get very busy, so *do not* leave a machine unattended hoping that you'll be able to come back to it later (you may well have been logged off!).

The system allows you to kill a session by means of *Ctrl + Alt + Backspace* (destroying all existing user processes and re-displaying the login screen). Please *only* use this facility when there are no other lab machines available and someone has left themselves logged in to a machine with the screen locked for an unreasonable period of time.

## Exiting Gnome (logging off)

Click on your name (system menu, top right) and you should see the drop-down menu containing a "Log Out" option. Select this, and you'll be prompted to "Log Out" or "Cancel" - if you click on "Log Out", your session will end and the system will display the login screen.

Please do *not* choose the "Power Off" option from the system menu - all machines are configured to go to sleep after a period of idleness, and will wake up if the flashing amber light at the front of the machine is pressed.  DICE machines *receive updates overnight*, so it is important that they remain switched on - otherwise there may be some delay when the machine is restarted as it searches for, and receives, any necessary updates. We appreciate that there are limited free power sockets for laptops in some of the labs but please *do not* unplug any of the DICE machines.

### 3.    Changing your password

Your password is the "key" to your DICE account and your files. It should not be made known to anyone else, and should not be guessable (either by being something obvious about you, or by a brute-force dictionary search).

You initially set your password using the password portal. Should you feel the need to change your password at a later date, you can do this using the `passwd` command:

```
% passwd
Changing password for user infteach:
Current password:
New password:
Retype new password:
passwd: All authentication tokens updated successfully
```

The `passwd` command is very strict about types of passwords and will give you an error message if it doesn't think your new password is appropriate e.g. "BAD PASSWORD: it is WAY too short".

If you forget your password, then you can contact *Informatics Computing Support* (see Section 5) and they will generate a new one for you.

**IMPORTANT**: Your password to access University central resources is *completely different* from your Informatics account password and the systems *do not* interact at all.


### 4.    Web Browsing

There are many different web browsers, but we recommend that you use Firefox - this can be accessed via the "Firefox Web Browser" icon in the *Activities* dash, or by typing "`firefox`" at the command prompt in a terminal window.

A browser address (or URL) is just a pointer to a document which can either be local (within Informatics or the University) or anywhere in the world. It has the format:

```
protocol://site.name/path/to/document
```

The first portion specifies the protocol used for data transmission when talking to the machine specified in the second part.  The most common are "`http`" and "`https`" (the 's' signifying a more secure transmission). You may also come across "`ftp`", used for remote file transfers, and "`file`" used to access files on the local machine.

When a browser first starts up, it normally displays a *homepage*. There is no local default for this, but with `firefox`  you can set your homepage from the "General" section of Firefox's "Edit => Preferences" menu.

Be aware that *not all Informatics web pages are available from all locations* – most teaching pages are only available from machines on the University networks.

# 5.     Getting Help

 You will hopefully find answers to most of your questions on the support help pages at:

http://computing.help.inf.ed.ac.uk

which also include a handy FAQ:

http://computing.help.inf.ed.ac.uk/FAQ

You can also request help via the *Informatics Computing Support Form* (the preferred option), or by visiting the *Support Desk* in room 3.D09 in Forrest Hill (FH) for undergraduate & MSc support, and room IF-2.07 (Forum) for research postgraduate support. Opening times are advertised on the computing help website, but the support office is generally manned all day (Monday – Friday) during the first couple of weeks at the start of the first semester and thereafter between 2 and 4pm (Monday – Friday).  Outwith semester dates, the support desk in FH is not manned (the support desk in the Forum is manned all year round). You can find the *Support Form* at:

https://www.inf.ed.ac.uk/systems/support/form/

***Please give as much information as you can*** - this will help *Computing Support* in dealing with your query and prevent any unnecessary delays.  Depending on the nature of the problem, the information you provide should include:

*   the name of the machine that you were using
*   any error messages
*   the time that you had the problem (this helps us to look through logs if necessary)

Please report any problems that you find with the computing lab machines via the support form. Although we do monitor lab machines, not all problems will necessarily be noticed, and so it is helpful to have problems reported to us.

## Software

Software packages are usually added and updated automatically on all managed DICE machines. If a piece of software is not available on DICE, but you think it would be useful to have it generally available, you may put in a request for this additional software to be installed DICE-wide. For information on how to make such a request, see:

https://computing.help.inf.ed.ac.uk/requested-software

# 6.     Printing

Sometimes it's helpful to have a paper copy of a document - but please be considerate and **use the printers responsibly**. Taught students are given a ***printing quota*** at the start of each semester and will be **charged** for any *additional* printing. Further details on charges and how taught students can top up their credit can be found at:

http://computing.help.inf.ed.ac.uk/printing

## Initial Setup

There are printers on the first and third floors of Forrest Hill (but not on the ground floor), and there is a label on each printer with its name. Printers in Forrest Hill (FH) all start with '**fh**', followed by an abbreviation for the location, and then optionally a '**c**' to denote a colour printer. Similarly, "**if**" for Informatics Forum printers.

*Undergraduates* and *taught MSc students* cannot use the printers in the Forum, and *research students* cannot use the printers in Forrest Hill, even though the printers may appear on the drop-down list of available printers in most common applications.

Setting up a default printer is done by typing (at the command prompt):

```
echo "export PRINTER=printername" >> ~/.benv
```

(where *printername* is set appropriately). This saves the variable PRINTER in your `.benv` shell configuration file, but it will *not* take effect until you log on again or start a new shell. For it to take immediate effect, you must type (at the command prompt):

```
export PRINTER=printername
```

Most printing tends to be done from within an application, but can also be done from the command line using the **lpr** command:

```
lpr –P<printer> <filename>
```

or, if you have set a default printer:          `lpr <filename>`

...to check a print queue:          `lpq –P<printer>`

...to remove a job from a queue:          `lprm <job-id>`

(The *job-id is* found from the output from `lpq`. You only have permission to remove your own jobs, however!)

At peak times (near submission dates), computing lab printers can get *very* busy. *Informatics Computing Support* will try to ensure that the printers have sufficient toner but, if you notice that a printer (anywhere, not just in the labs) has run out of toner, don't assume that someone else will report it or fix it! Instead, you can report the problem using the web *Support Form*. Paper is the responsibility of admin staff, not  *Computing Support*.

# 7.     Network Access

As stated above, it is possible to view web pages from a variety of locations, *but not necessarily from everywhere*. If you have a computer at home with network access, you may be able to access only some of the University services.

**ssh**

One remote access method, available on Linux, MacOS and Windows platforms, is called **S**ecure **SH**ell client (`ssh`) and can be used to access external-facing servers (i.e. servers visible outside Informatics). Once connected to one of these servers (also known as **gateways**), other internal machines and services can be accessed.

The external-facing servers (gateways) for Informatics are currently:

- `staff.ssh.inf.ed.ac.uk`
- `student.ssh.inf.ed.ac.uk`
- `ssh.inf.ed.ac.uk`

These gateways must *not* be used for any heavy computational work - they are merely gateways to allow you to access other Informatics resources. For more information, see:

http://computing.help.inf.ed.ac.uk/remote-working

**Wireless**

The University offers a wireless network service (called ***eduroam***), and wireless access points are available throughout Forrest Hill and the Informatics Forum. Registration is required - for more information, see:

http://www.ed.ac.uk/information-services/computing/desktop-personal/wireless-networking

Note that ***the wireless network is NOT a 'secure' network,*** and as such may not provide access to all DICE services. You cannot, for example, print to any of our printers while connected to the Central wireless service.

Note also that support for this service is provided by the *Central IS Support Team,* and that *Informatics Computing Support* does ***not*** provide support for this service.

**Virtual DICE**

Whilst not, strictly speaking, "network access", using *Virtual DICE* is another way of working at home. A disk image is available that can be used by virtualisation software[4] to run a copy of the DICE environment on your own machine. For more details, see:

http://computing.help.inf.ed.ac.uk/vdice-intro

---

4   Such as VirtualBox.

**NX and OpenVPN**

**NX** is a remote graphical login facility which makes it possible to use a standard DICE desktop environment from any location and on any device which can run the NX client application. This enables you to connect to an existing session. You'll need an NX client such as *OpenNX* or *NoMachine* to be able to connect (which you'll have to install yourself if you don't already have it). For more information, see:

http://computing.help.inf.ed.ac.uk/nx

**OpenVPN** allows access from a remote machine to Informatics servers, such that the remote machine appears to be on the Informatics network. For more information, see:

http://computing.help.inf.ed.ac.uk/openvpn

## Access from Student Accommodation

*Accommodation Services* provides internet and telephone facilities to the vast majority of students staying in University residential accommodation. More details can be found at:

http://www.accom.ed.ac.uk/services/it-facilities

Note that support for this service is provided by the *Central IS Support Team*, and that *Informatics Computing Support* does **not** provide support for this service.

# 8.      Unix File System Navigation and Simple Unix Commands

## Files and Directories

Information is stored in files, and Linux doesn't care what's in a file - it doesn't need suffixes (or file extensions) like .doc, .exe, .txt. You can use them to help you recognise what's in a file, but it's not required.

Files can be grouped into ***directories*** (*folders* in the non-Unix world), a special type of file which contains other files and may also contain other directories. In this way, they create a tree-like structure – see *Figure 4*.
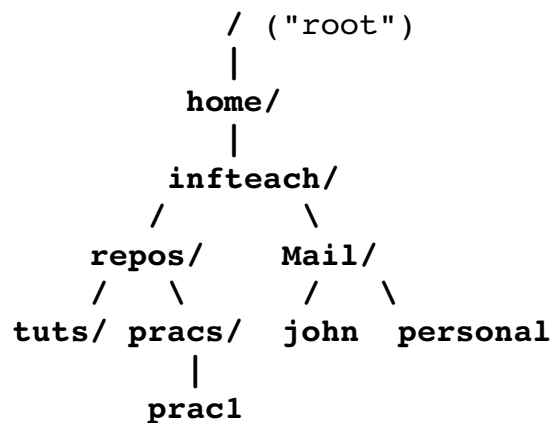
```
                / ("root")
                |
              home/
                |
             infteach/
          /           \
        repos/      Mail/
        /   \        /   \
    tuts/ pracs/  john  personal
                |
              prac1
```

*Figure 4: Linux Tree Structure*

Here, 'prac1' is a file which is contained in the directory 'pracs'. The directory 'pracs' is a subdirectory of the directory 'repos' which in turn is a subdirectory of the directory 'infteach'. When you first log in, you are taken to what is known as your ***home directory***. Its pathname is /home/*<username>* which can be shortened to ~*<username>*. If you have navigated to a different directory and want to get back quickly to your home directory, you use the command "**cd**" without any arguments. The "cd" command (short for 'change directory') also accepts an ***argument,*** which is either the full pathname or relative pathname of the directory that you want to move to.

Each file can be uniquely identified either by a full path name (which always starts with a "**/**" to signify that it starts at the top of the tree) or relative to any other position in the tree.  For example, in *Figure 4*, the file prac1 has a full pathname of /home/infteach/repos/pracs/prac1. Had you been *in* the 'repos' directory already, the file could be identified as pracs/prac1 (note there is no "/" in front of repos).  Had you been *in* the 'Mail' directory, then the file's relative pathname would be "**../**repos/pracs/prac1". The "**..**" signifies to go up a level in the tree before descending again.

## Listing files

To see what files you have in a directory, you use the command "**ls**", which will list both files and directories. The command has several useful flags (options), which you should experiment with. The majority of Linux commands have an associated manual (man) page. So, if you try "man ls", you should see a list of the flags that you can use.

## Manipulating files

You can copy files from anywhere to anywhere else (provided that you have permissions to do so) with the "**cp**" command. It has the form "cp *<source> <destination>*". So, to make a copy of the file prac1 (calling it prac1.bak) in the same directory, you would type:

**cp prac1 prac1.bak**

If you wanted to copy it to a different directory, then you would type something like:

**cp prac1 ../tuts/prac1.bak**

A word of warning - by default, Linux will normally carry out an instruction *without* checking with you first so, if the file that you want to write to already exists, it will overwrite it (if it doesn't exist, it creates it).


## Viewing a file

The way that you view a file depends on the format of that file. For plain text files, the command **less** can be used. The less command displays the contents of a file one screenful at a time, and you can move on to the next screenful by pressing the space bar (you can press "q" at any time to return to the command prompt).

Some documents files are held in low-level page description languages such as *PostScript*, (which often has the suffix .ps) or *Portable Document Format* (with the suffix .pdf) and such files require a special viewer. For example, PDF documents can be read using the command "evince *<filename>*" (or by opening the file from within the "Utilities: Document Viewer" application).


## Creating Directories Of Your Own

You can create directories of your own so that you can group similar files together. For example, you may want to put your programs and practicals in separate directories. You use the **mkdir** command to do this, for example using:

**mkdir programs**

This will create the directory "programs" in your current working directory - which is wherever Linux thinks you are in the filesystem hierarchy. To check where you are in the hierarchy, you can type the command **pwd** (print working directory).

Assuming you have now made a directory called programs, typing:

**cp example1 programs**

will make a copy of the file example1 in the directory programs. The copy command (cp) will only overwrite a file if it is not a directory. (If the directory programs did not exist, then the above command would merely have made a copy of the file example1 and called it programs in the same directory as example1.)

## Moving and Removing Files

If you don't want to copy a file but merely move it to another location (or, in fact, just change its name), you use the **mv** command. In the example above, had we substituted cp with mv:

```
mv example1 programs
```

the result would have been that we would not have left a copy of example1 in the current directory but merely moved it to the programs directory. Had the directory programs not existed, the file example1 would have been *renamed* "programs".

If you want to remove a file, use the **rm** command. To remove a directory, use **rmdir**. However, you can't remove a directory using the rmdir command *unless it is empty*. You can do it by using the "**rm −r**" command (the "−r" means "do a recursive remove"), but *use this with caution*!

## Command Line Shortcuts

When typing names of files and programs, it is possible to avoid typing the whole name. Using the *<tab>* key asks the shell to complete the filename for you - if it can find only one possible solution, then it completes it for you. Otherwise, pressing *<tab>* again will show all possibilities.

The **history** command lists your most recent commands. You can then use the up and down arrow keys to select the command that you want and then just hit *<Return>* to execute it.

If you want to cancel a line that you are typing, or are in the middle of executing a command and want to cancel it, you can do this by typing <ctrl><c> (sometimes written ^C).

## Quotas

First and second-year undergraduates have a quota of 4Gb, all other students have a quota of 10Gb. If you go over this quota, you won't be able to log on (the logging-on process needs to be able to write to your home directory) and *Computing Support* will only increase your quota under *exceptional circumstances*. It is therefore important that you keep an eye on this to avoid any unnecessary interruptions.

To check your quota, from the command line, run the command:

```
fs lq
```

which should show something like:

```
Volume Name              Quota      Used %Used   Partition
user.infteach          1000000     19455   2%         61%
```

If you are close to - or have exceeded - your quota, the most likely place to look for unwanted files is in a directory called .Trash in your home directory.

Note that, as already stated, the contents of ~/Yesterday are *not* included in your quota allowance (and therefore attempting to delete anything in that location will have no effect on filespace usage).

## Backups

Each night a copy of your home directory is created ('mirrored'), and this backed-up data appears in your home directory as a **read-only** (non-modifiable) directory called "`Yesterday`" - this is updated every night, hence the name. This location is handy if you accidentally delete a file by mistake and notice the error quickly.

Home directories are also backed up to tape, so if you remove an important file and don't notice your mistake for a few days, you can contact *Informatics Computing Support* using the web support form and ask them to restore it for you. ***Please note that this may take some time***.
For more detail, see:

http://computing.help.inf.ed.ac.uk/backups-and-mirrors

## Other Things To Consider

The following is only a *very* brief overview of Linux. The topics are only covered at a very basic level, and you should make yourself more familiar with them.

### Linux filename names and types

*Filenames* can consist of all alphanumeric characters plus "`-`" and "`_`". The following characters have special meanings, and should be avoided (to ensure that unexpected things don't happen):

```
& ( ) < > $ ' " ` ; / | \ * ! ?
```

Files can be broadly categorised into two types – those viewable as plain text, and those that can't. Although Linux makes no distinction between files, there are certain naming conventions that are widely used, and *some* applications do make assumptions about files based on their suffix (extension). The following list gives a few examples:

| File suffix | File Type | Viewer |
|---|---|---|
| `.ps` | PostScript | evince |
| `.pdf` | Portable Document Format | evince |
| `.jpg, .png, .gif` | Image formats | gimp, Image Viewer |
| `.c` | C source code | pager or text editor |
| `.html` | Web (HTML) source | pager or text editor |
| `.tex` | Latex | pager or text editor |
| `.gz` | compressed (gzip) files | not appropriate |
| `.jav, .java` | Java source | pager or text editor |

There is a useful file-identification utility called **file**. Running this command with the name of the file will make a guess at the file type:

```
% file README
README:                 English text
% file planning_quiz.ps
planning_quiz.ps:       PostScript document
%
```

**File permissions/access**

Your files are stored using a **filesystem** called **AFS** (Andrew File System), which controls access to files with *Access Control Lists.* These work on a per-directory basis (permissions apply to all files within a directory). More information can be found at:

http://computing.help.inf.ed.ac.uk/afs-acls

Your files initially will have secure permissions by default, but at some later point you may wish to change these.


**Linux user configuration files**

Most applications use hidden files to configure themselves to a specific user's requirements. These files tend to start with a dot (for example, `.emacs`), and can also be referred to as "dot files". These are normally hidden when using `ls`, and you can only see them if you use "`ls -a`" (see the individual application documentation for usage).


**Other Linux commands**

There are too many to cover in this short introduction. Remember the `man` command. Have a look at "`man bash`" which describes the bash shell in a little more detail. Also look at the `ps` and `grep` commands.


**Unix Job Control**

When you type a command, the usual behaviour is for that command to be executed and then, once completed, you are returned to the command prompt. This is fine for commands that run for just a short time, but can be problematic for commands that run for longer periods (such as browsers and editors).

In general, each command can be referred to as a "**job**" – it starts a process, performs some action, and then dies when finished. While it is running, it takes control of the terminal and is said to be running "**in the foreground**".  This is the default behaviour, but if you want a job to run "**in the background**" (so that it returns control of the terminal before the job completes), you append an ampersand ("**&**") to the command.  To see what jobs are active, type `jobs` at the command line. To bring a job to the foreground, type `fg` at the command line (see the JOB CONTROL section of the "`bash`" man page).

> **Please make sure that you are familiar with the job control documentation (man page) before experimenting – otherwise you may end up with numerous processes (jobs) running on your machine that you are unaware of!**

Should you need to run jobs for more than 18 hours, you should first of all read the instructions about setting up a long-running job. These jobs should always be run in the background and using the `nohup` command so that you can log out to allow someone else to use the machine. Do not put up signs on lab machines saying "Experiment Running" - you will be logged off!

## 9.      Emacs

To create a text file or edit the contents of an existing one, we use an **editor**. There are several available for Linux, but we recommend that you use **Emacs** (if you have no other preference).

*Emacs* provides much in the way of functionality, but you can get by with very few commands in the beginning. You start Emacs by typing `emacs` at the command prompt. You can also give it a filename:

```
emacs <filename>
```

and if `<filename>` exists, it will be opened and you will see it in a new window. If it doesn't exist, you'll see a window similar to the one below:
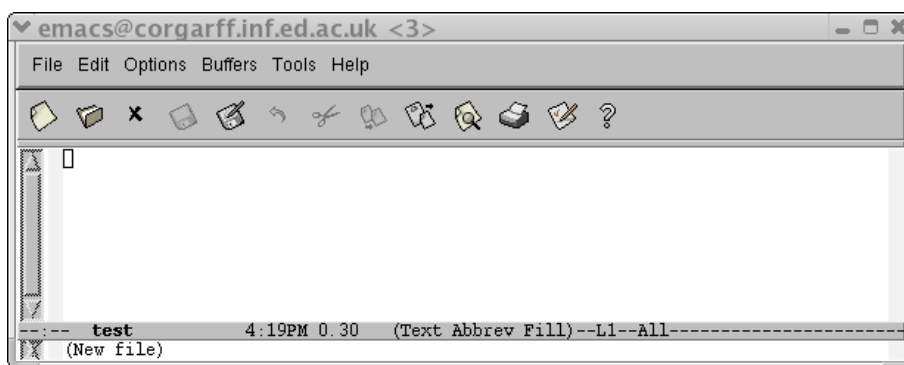


There are pull-down menus and shortcut buttons at the top. One helpful one is the **undo button** – sixth from the left (the backwards-pointing arrow).

If you open an existing file, you can see information about the status of the edit in the status (mode) line at the bottom of the Emacs window:
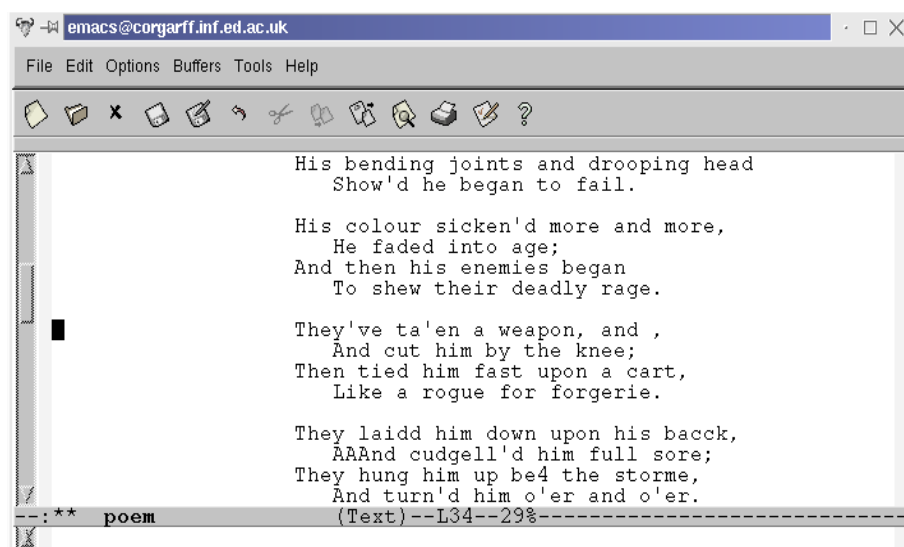


You will note that the status line has two asterisks at the left-hand side – this means that the **buffer** has been edited (i.e. the *memory copy* has been changed) but these changes have *not yet been saved to disk*.

When you start Emacs, a copy of the file is created (in memory) and it is this copy (the buffer) that you edit. The actual *file* doesn't get changed until you explicitly save the buffer (write changes to the file).

On the status line are the name of the file, the type of the file that Emacs thinks it is editing, the line number at which the cursor is currently sitting and the amount of the file above the top of the window. To insert text into the file, all you need to do is position the cursor and type it in. To delete characters, you can use the Delete or Backspace key.

You may notice that when you continue typing at the end of a line, a small down & left arrow will appear, with a complimentary down & right arrow at the start of the next – and your text will continue on the next line. Once you've typed a paragraph (or at any point when text starts to "wrap") typing ESC-q (i.e. press the "ESCape" key and then type "q") will format it: in other words, it will insert line breaks in the text at appropriate points making it easier to read and print.

When you've created or modified a file and want to save it you can:

- Ctrl-x Ctrl-s:        Saves your file without quitting Emacs.
- Ctrl-x Ctrl-c:        Exits Emacs, but prompts you to save file ("y") or not ("n")
- Ctrl-x Ctrl-w:        Allows you to write (save) your file under a different name.

To find out more about Emacs, try the following:

**web**:        extensive information can be found at:

https://www.gnu.org/software/emacs/emacs.html

**info:**        type "info emacs" at the prompt ("q" to quit).

**Help**:        various sources of information can be found under the pull-down menu entitled 'Help' within Emacs. The Emacs tutorial can also be found there and is well worth working through.