Royal Holloway Computer Science Department: Hints on using the Undergraduate computer facilities

Bob Vickers and Jon Butt

November 18, 2014

1 Account passwords

It is a good idea to change your password reasonably often. Just type

```
newpassword
```

at a Unix prompt and follow the instructions. You will be asked to enter your old password once (so as to confirm it is really you and not a passing stranger) and your new password twice. Make sure you don't have the `Caps Lock` key active, and avoid using the numeric keypad (because these keys vary from terminal to terminal).

It is very important to choose a good password. Our computers are all on the Internet, and so are potentially open to attack from anywhere in the world. Attackers use automated brute force techniques to try very large numbers of plausible passwords until they find one that works. If they break into your account then they may exploit operating system bugs and damage the entire system.

Passwords are case-sensitive and must be at least 10 characters long. They must not be all letters or all numbers.

Here are some passwords to AVOID:

- Words from dictionaries, encyclopaedias, lists of names etc (whether English or not).
- Words prefixed or suffixed by 1 or 2 digits.
- Words with letters replaced by similar-looking digits (e.g. letter l replaced by number 1). Hackers know this trick.
- Anything based on your name.
- Backward words.

Some ideas for good passwords:

- Mixed letters, numbers and punctuation.
- Short phrases, with or without spaces, for example `I forgot 2 get up this morning`.

Don't use the examples above (no matter how clever you think they are), nor any other password suggested in a book.

2 Disc storage

The computer systems have a number of storage areas you can use for storing programs and data. Disc storage is a valuable resource; our systems are shared by hundreds of users and it is important you don't waste it.

A word about terminology; some people confuse *disc space* with *memory*. Disc space is a permanent storage area, whereas memory (also known as *RAM*) is very fast transient storage used for short periods by programs you run.

2.1 Storage areas available

2.1.1 Home directory

The most important storage area is your *home directory*. This is your current directory when you login, and belongs to you alone. It is backed up every working day so in the event of a disc failure it can be recovered. Your default allocation is 20 Gbytes.

The file system containing the undergraduates' home directories is called */home/ugrads* and it lives on our file server csfiles. If your username is ermintrude then the following are equivalent ways of referring to it:

```
$HOME
~
~ermintrude
/home/ugrads/ermintrude
/rmt/csfiles/ugrads/ermintrude
```

2.1.2 Work areas

We also have some areas which are NOT backed up. These are places to store big files which you could recreate yourself if necessary.

On Teaching there is */nobackup/ugrads/$USER*, also known as */nobackup/work/$USER*. This lives on a fast local disc, so is ideal for applications which need fast temporary storage.

You also have access to */nobackup/CS/$USER* which is slower but has more space.

2.2 Disc quotas

We run *disc quota* software which limits the amount of disc space and the number of files you can use. Each controlled file system has independent limits for each user. You need to understand the following terms in order to interpret the reports you receive:

- A *block* is a kilobyte, i.e. 1024 bytes.
- A *quota* is a soft limit which you can exceed temporarily.
- A *limit* is a hard limit which you can never exceed
- A *grace period* is the time you are allowed to exceed your quota, typically 7 days.

The command *quota* displays your current usage and quotas. Here is a sample output from a user who urgently needs to delete some files:

```
teaching$ quota
Disk quotas for user bobvstu (uid 10972):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
/dev/mapper/UGRAD-ugradhome
                101114* 100000  120000    none    1387   20000   25000
```

Notes:

1. The asterisk indicates a quota violation.
2. The usage on ugradhome is 101114 kilobytes, which is over-quota. The grace period is *none* which means the user can no longer create or edit files.
3. A quota or limit of zero should be interpreted as infinity.
4. The display shows that the number of files is controlled as well as the space they consume. In this case the user is well under quota.

2.3 What to do if you are using too much disc space

Obviously it is good practice to avoid storing any unneeded files, but there are various techniques which will help you pinpoint the most fruitful areas to look at when you need to delete files in a hurry.

A very useful command is *du* (which stands for Disc Usage). It searches the current directory recursively and displays the blocks consumed by each subdirectory. If you have a complicated directory structure this may be a bit overwhelming so you can type

```
du | sort -n
```

so that the output is sorted.

Once you have selected a particular directory you can display the biggest files by typing

```
ls -las | sort -n
```

and deleting the unwanted ones with the *rm* command.

Sometimes you will find large files in your *mail* directory, especially if you have been sending or receiving large attachments. Use your favourite mail program to delete large messages; with Pine you can do this very easily by sorting the index of a large folder: select *'$'* to start the sort dialogue then *'z'* to sort by message size. Sorting has no effect on the actual folder, only the current view is sorted.

As well as deleting files you don't need you can also save space by *compressing* files you want to keep...see the man pages for *gzip* and *gunzip*. Postscript files especially benefit from this, and they can still be read afterwards by Ghostview. For example:

```
gzip myfile.ps
gv myfile.ps.gz
gunzip myfile.ps.gz
```

Lastly, if you have saved as much space as you can but still need more then you should e-mail Support and ask for your quota to be raised. The Support team is happy to grant bigger allocations when they are needed for coursework, but we do expect you to do some housekeeping before you ask.

3 Copying your files to and from other computers

The student server *teaching.cs.rhul.ac.uk* supports the SSH and SFTP protocols. As a result, your files can be accessed from anywhere on the Internet via a number of applications, for example:

- From a Unix system such as Linux or MacOS (or from the Cygwin environment on Windows) you can use the command-line programs *scp*, *sftp* and *rsync*. See the man pages for examples.
- On Windows you can install the *WinSCP* program from `http://winscp.net`
- Most modern Linux systems have a graphical file manager with built in support for SFTP protocol. Specify *sftp://teaching.cs.rhul.ac.uk* as the remote URL.
- on a Mac you can install a third-party frontend to sftp, for example *Fugu SSH* from `http://rsug.itd.umich.edu/software/fugu`.

4 Customising your Unix environment

Like other operating systems, Unix gives you great power to customise your environment. For example you can create shortcuts for commonly used operations, or you can change the layout and colours of your windows.

Very often there is more than one way to achieve the same result; if you follow the guidelines here you will find it easier to get help when you need it. If you have experience of other Unix systems you may well find different conventions apply; don't be surprised if you copy configuration files from different machines and find they no longer work.

Many customisations are performed by editing files in your home directory whose names begin with a dot, for example `.profile` and `.bashrc`. When your account is set up some of these are created for you; if you make a mistake when editing them you can retrieve the originals from either `/CS/skel/students` (student accounts) or `/CS/skel/staff` (staff accounts). Note that dot files are *hidden*; they do not appear on `ls` output unless you include the `-a` option.

4.1 Setting environment variables

Many options can be configured by setting *environment variables*. For example, you can tell the system that your favourite editor is `emacs` by giving the environment variable *EDITOR* the value "`emacs`".

The best place to define these variables is the file `.profile` in your home directory. This file is read whenever you login using `bash` (the default shell). A sample is created for you when your account is set up, and it contains hints on how to use it.

If you are not sure what your login shell is, type

```
finger $USER
```

One important variable you may want to change is *PATH*. This variable tells the system which directories to search when you type the name of a program. So, for example, if you type

```
cat
```

the shell executes the command `/bin/cat` because `/bin` is one of the directories in the PATH variable.

The default PATH is set in `/etc/profile`. The sample `.profile` explains how to add new directories.

4.2 Defining bash aliases

The bash shell allows you to save typing by defining aliases for commonly used commands. These aliases should be placed in `.bashrc`; this file is executed by bash whenever it starts an interactive shell.

Some people like to define environment variables in `.bashrc`, but this can cause confusion because `.bashrc` is not executed when programs are started by clicking with the mouse. But one exception to this rule is the prompt `PS1` which is only useful for interactive shells.

4.3 Customising your X session

When you login to the system via the X windows system (e.g. from an X-terminal) then the script `.xsession` controls your session. It defines which programs run at login time and what happens when you logout. When the script terminates then your X session is over.

Typically the script does 4 main things:

1. Set up the environment variables your applications expect. This is best done by calling `.profile` .
2. Run the xrdb program to customise your X resources. See below.
3. Start xterm and other windows you like to have on your desktop.
4. Start the window manager

This is easier than it sounds. The important thing to remember is that all applications should be started in the background, otherwise the script will get stuck halfway before the window manager is started.

4.4 Setting X resources

When you have become experienced at running X clients you may want to customise their look and feel. The man pages for X clients tell you how to set *X resources*; the place to set them is in the file `.Xresources` (by changing .xsession you can use a different name, but you may confuse yourself if you do).

4.5 Configuring the window manager

The default window manager is *Xfce*. If you click the Xfce icon (a blue X with a rat on top) there is a Settings option.

4.6 Configuring the emacs editor

Your emacs configuration file is called `.emacs`. See the emacs online reference manual (type `C-H i` within emacs) for hints on what to put there.

4.7 Testing customisation changes

Sometimes an attempted customisation fails to achieve the desired effect, or if you are really unlucky it may prevent you from logging on. So you should always try and test your changes before logging off.

The `.profile` and `.bashrc` files can be tested by typing

```
xterm -ls &
```

to start a new login shell window.

The `.xsession` script is harder to test. The safest way is to run two sessions simultaneously so that you can undo your change if the second session fails to start.

If you do get in a situation where you can't login try using an ssh client (e.g. Putty) from a different computer. This will give you a single command window.

5 Terminal Room Rules and Etiquette

These rules are based on requests from students.

5.1 Taking breaks

If you want to take a short break without relinquishing your terminal then take the following steps:

1. Type `xlock` on the keyboard. The screen will go blank.
2. Take your break.
3. When you come back, hit any key to revive the terminal, then type your normal Unix password.

If you stay away for more than 10 minutes then a public logout button will appear. If no other terminals are available then waiting students are free to log you out.

The xlock program has been configured for optimum use in the lab environment, so please do not use other terminal locking programs. Use of screen saver options can generate significant network traffic and CPU load.

5.2 Consideration for others

1. Avoid unnecessary noise. If someone asks you to keep quiet please respect their wishes.
2. Don't play games during busy periods. Others may have work to do and will resent not being able to get at a terminal.

6 e-mail

Every student at the college is given an e-mail address by the Computer Centre; this address will be of the form

```
I.Surname.yearnumber@rhul.ac.uk
```

and allows you to read your mail from any lab run by the Computer Centre.

7 Printing

The department has a number of printers but not all are public. If you print to the wrong one you may find yourself presented with a hefty bill.

7.1 Configuring your default printer

Programs which print (such as the `lpr` command) use the environment variable `$PRINTER` to define your default printer. You can change this by editing the file `.profile` in your home directory.

To find out the current value type

```
echo $PRINTER
```

7.2 Printing files from computer labs

Printing is a **chargeable** service managed by the IT department, and use of a printer will draw funds from your College printing account. Problems should normally be reported directly to the Computer Centre.

An exception is the laser printer in McCrea 103. Charging is managed by IT, but problems with the printer itself should be reported to CIM Support.

The two print queues you will normally use are called *_rhul_bw* and *_rhul_colour*.

7.3 Printing Problems and Paper

The McCrea laser printer is managed by CIM. If you have a problem with it or it runs out of paper please contact a member of the support team.

Please do **not** attempt to fix printer problems yourself! You could make the problem worse or even injure yourself in the process. Always report problems to the appropriate staff.

7.4 Printing to the Computer Centre

All the self-service and operator-service print queues are available from Teaching; most accept either plain text or postscript. It is your responsibility to ensure you pick an appropriate queue or you may find yourself paying to print on an expensive colour printer in a locked room!

Printer names (which are all lower-case) can be obtained from the Computer Centre or from the Computer Science website.

8 Logging in from outside the college

All our servers are on the internet, and can be connected to from anywhere in the world. Provided your computer has an *ssh* client you can connect to one of our Unix servers and type Unix commands in a shell window. Undergraduates should connect to `teaching.cs.rhul.ac.uk`.

If you need to run graphical programs such as *firefox* we recommend running an NX client: see below.

8.1 Graphical logins using NX software

You need to install an NX client on your computer, plus a configuration file which includes a special key. There are versions available for Linux, Windows and Mac OS X to download from our FTP server: ftp://ftp.cs.rhul.ac.uk/pub/NX[1].

The configuration file you need in order to connect to *Teaching* can be found at www.cs.rhul.ac.uk/Internal/Computers/teachin
Install it in the NX configuration directory on your home computer. On Unix systems this directory is .nx/config in your home directory, whereas on a Windows system it is .nx\config in your user profile area. This will be something like C:\Users*username* but may vary by Windows version. If you are unsure you can type the following at a command prompt to open it:

```
explorer %USERPROFILE%
```

You can either create the directories by hand, or run the NX client program once to get it created automatically.

Members of staff and research postgraduates should download the StaffLinux version[3] instead.

When you run the client you can further optimise the configuration according to your screen resolution and the speed of your internet connection.

8.2 Connecting from a Unix system

If you only need a line-mode login you can use the ssh program, which is almost universally available on Unix systems. If your computer does not already have it then visit the OpenSSH web site[4] and download the software (which includes the file copying programs *scp* and *sftp* as well).

If you run ssh with the -X parameter you can run graphical programs, but they may run very slowly across the Internet. A better solution is to use NX (see above).

8.3 Windows systems

Some services are only available to clients which are either on campus or else connected using the college's VPN service. VPN is run by the Computer Centre and is described on their website[5]

If you have installed the Cygwin Posix environment then *ssh* and *scp* will be included. Other useful programs you can install include *WinSCP* and *Putty*.

9 Bundling and unbundling Unix directories

It is often useful to bundle up a group of files into a single file, either for backup purposes or for sending to someone else. On Unix this is usually done with the `tar` command, although `zip` is also available.

Software is often distributed as compressed tar files, so you need to be able to unbundle them if you ever pick up software from the network.

[1] ftp://ftp.cs.rhul.ac.uk/pub/NX
[2] http://www.cs.rhul.ac.uk/Internal/Computers/teaching.nxs
[3] http://www.cs.rhul.ac.uk/Internal/Computers/stafflinux.nxs
[4] http://www.openssh.org
[5] http://www.rhul.ac.uk/information-services/computer-centre/faqs/results-keyword.asp?keyword=VPN&Submit02=Go

9.1 Creating a compressed tar file

This section gives instructions for submitting a project to your supervisor, as an illustration of `tar`. Alternatively see the man page for the locall-written archivedir command.

You should put all your project files in a subdirectory, which may of course contain further subdirectories. In the examples I assume the subdirectory is called `rocketry`.

1. Create the directory and move into it:

```
mkdir rocketry
cd rocketry
```

2. When you have finished the project move into the directory *above* it, and create a compressed tar file:

```
cd ..
tar -cvf - rocketry | gzip -c >rocketry.tar.gz
```

3. Your compressed tar file `rocketry.tar.gz` can now be mailed to your supervisor as an *attachment* to an e-mail message. This is easily done when composing a message in `pine` by moving to the `Attchmnt:` line and pressing *cntrl-J*.

9.2 Unbundling a compressed tar file

When you receive a tar file it is wise to check what is on it before unbundling it. To avoid confusing the extracted files with your own ones you will want to put the files in a subdirectory; if the tar file was created as above then this will happen automatically. But occasionally the files will be created in the current directory, in which case you should create a directory and move to it before extracting the files. If in doubt then play safe: make sure you are in an empty directory before unbundling the files.

1. If you receive the file as a mail attachment then save it to a file. Users of mailers which don't support MIME (e.g. `xmh`) should save the entire message to a file, then use `uudeview` to detach the attachment.

2. Check what is in the tar file:

```
gunzip -c rocketry.tar.gz | tar -tvf -
```

3. If you are happy with the directory structure then unbundle the tar file:

```
gunzip -c rocketry.tar.gz | tar -xf -
```

Note that by uncompressing and untarring in a single step you save disc space. If you are close to your quota you can save further space by keeping rocketry.tar.gz in your work area `/nobackup/work/$USER`.