Useful Unix Topics

Jeremy Sanders

October 2011

1 How to change your password

The command to change your password is `passwd`. Note that you have to log into a Solaris system to chage your password (e.g. `ssh muon3.ast.cam.ac.uk`). Updated passwords are only copied to Linux computers hourly.

The length of the password should be 8 characters. You *should not* choose dictionary words (in any language) or simple combinations of words and numbers. Ideally your password should be a combination of upper and lower case letters, numbers, and symbols, e.g. eFx!2RxJ. Only the first 8 characters are significant. All passwords are easy to remember when you get used to them! A good way to choose are the initial letters of a phrase, randomly capitalised, and with symbols and numbers randomly added.

2 email

You have access to email at the department. Your email address is `username@ast.cam.ac.uk`.

You have two standard options to read your email. The first is a graphical email client called `thunderbird`. The second is the text-based `alpine` or `pine` email client.

Some people may be interested in the webmail interface here `https://www.ast.cam.ac.uk/webmail/`

If you wish to forward your IoA email to another account, you can create a `.forward` file containing an email address. Your mail will be forwarded to this address. Be careful not to create a mail loop!

3 Browsing the web

Although there are several web browsers you can use, I would advise the use of *Firefox*. You can start the latest version with the `firefox &` command.

4 Quotas

You are limited to storing around 2GB of data in your home space (under /home/username). The home space is backed up using ZFS snapshots, allowing you to recover data from over several days past. You can access old snapshots of your home directory under the `.zfs` subdirectory - see the user guide.

ZFS does not use disk quotas. To see how much space you have remaining, use the `df` command, e.g.

```
> df -h /home/username/
```

You are also entitled to additional space (send an email to `helpdesk@ast.cam.ac.uk` to request it) which is ‘mirrored’: a copy of it is saved to another disk every night. That means if the main disk breaks or you accidentally delete it in a 24-hour period you should be able to recover it.

In addition to this you may request space on one or more ‘data disks’ from helpdesk. Your data *will not* be backed up automatically—you must do this yourself.

You need to use the df command if your disk uses ZFS, otherwise use the command `quota -v`. You have a current usage, a quota (where you will receive warnings) and a limit (where you cannot create new files). Going over your quota leads to strange errors which are hard to diagnose, so keep an eye on it.

Note that firefox stores its cache in your homespace. You can try to clear its cache to free up some space. You can set the cache size to a small value to stop it getting too large. A further trick is to move it to a data disk and use a symbolic link to map the data disk directory to where the cache usually resides.

5 Backing-up

If you have data on a data disk *it is vital* you back this up if you care about it. People often learn this fact the hard way. *You may lose several years of research if you do not back up, and you won’t get any allowances for this from the university*. Disks can often break and it is easy to type `rm -r`.

Realise that RAID protected directories don’t protect you from stupid mistakes. You can accidently delete files very easily and you can never get them back.

Alternative ways of backing up data are to copy data to different disks on different computers in other locations. The `rsync` command is a very useful tool to only copy the files that have changed on the source system.

5.1 rsync

You need a command like this to copy files to a different directory:

```
rsync -av /home/foo/indir/ /data/foo/outdir/
```

This will replicate any files in indir to outdir. It will not delete any files in outdir that no longer appear in indir. You need to add the `--delete` option in order to get this.

If you want to copy files to another computer over ssh, you can do this:

```
rsync -av /home/foo/indir/ username@somecomputer.cam.ac.uk:/data/foo/outdir/
```

The username@ part is optional if you have the same username on the destination computer.

5.2 Tape backups

To back up your data find a machine with a tape drive and get hold of a tape of the right size from helpdesk (use `du -k` to see how much space you have used). *Do not try to clean up your directory* *before backing it up: that way you could accidentally delete what you intend to backup.* It's a good idea to ask a fellow experienced student for help the first time you do this.

The tape is accessed using `tar` and the *tape device*, which tells tar what tape drive to use and what level of compression to use on the tape. The command used to backup the current directory is:

```
tar cvf tapedevice ./
```

where tapedevice specifies the device. tapedevice takes the form `/dev/rmt/[0-127] [l,m,h,u,c][b][n]` (no spaces) (see the users' guide for details on what the letters do), e.g. `/dev/rmt/0`. This command will overwrite what is on the tape and rewind the tape afterwards (add n to 0 to stop the rewind). This command backs up the current directory and its contents (and subdirectories inside it, but not symlinks). *Do not use an absolute directory instead of ./ to back up - there is a bug on Sun tar*.

Use `tar tvf tapedevice` to list the contents of a tape. Use `tar xvf tapedevice` to extract its contents into the current directory. Other tape commands are (see `man mt`):

```
mt -f /dev/rmt/0 offline  <-- rewind and eject tape
mt -f /dev/rmt/0 rewind   <-- rewind tape
mt -f /dev/rmt/0 status   <-- what is the drive doing?
mt -f /dev/rmt/0 eom      <-- go to end of tape contents
```

6 Logging into other machines

All the computers (machines) in the Department are given a *hostname*, such as `cass00.ast.cam.ac.uk`. You can always access your directories on other computers due to the Network Filing System (NFS). The machines (such as the one on your desk) are labelled with a sticker with its name. Some of the machines in the department are *private* as they were paid by private funds, some are *semi-private* (private money but run by IoA), and most are *public*. If the machine you have isn't good enough to run the program you want to, then you can log into a specific machine to run it (providing the machine is public or you have permission to use a private/semi-private machine).

If your work is data intensive you should run your programs on the machine hosting the data if possible, otherwise it will be slow as the data will have to be transferred over the network.

The command to log into another machine is `ssh` (secure-shell). It is even possible to run graphical programs over ssh from a remote computer (thanks to the wonders of the X Windows System). See the Users' Guide and `man ssh` for much more information. Note that the computers you can log into remotely are restricted.

```
ssh cass80  [or ssh cass80.ast.cam.ac.uk]
[type your commands here in this new shell]
[type exit to return to your computer]
```

If you are planning to run a long job (longer than a few minutes), you should *nice* your job. Niced jobs run at a lower priority level than interactive jobs (such as the session of the person sitting in front of the computer). Nice levels run between -19 (high priority) and 19 (low priority). Standard programs run at level 0. Normally you would you a command like

```
nice +19 myjob
```

to run the job at level 19.

The easiest way to see how loaded a computer (how many things are running on it) is to use the `top` command which interactively shows the ‘topmost’, or most CPU intensive, jobs. Press `q` to exit it (press `?` to get other possible commands).

It is hard to find a computer free on which to run a job (or several), but there is another, easier, way to do this called the *Sun Grid Engine*. It allows you to send a job, setting the requirements for the computer you need, adding it to a queue to be run. Jobs are run fairly (nobody can hog the queue) from the queue among the free computers. The best documentation is in the Users’ Guide.

You can make terminal sessions that you can disconnect from and reconnect to later using the GNU `screen` command, which also allows multiple terminals in the same connection. Type `screen` to start, and run your programs. If you want to disconnect, type `ctrl+a` followed by `d` (most screen commands use `ctrl+a` followed by something - try using ctrl+a then ?). You can then reconnect later by logging into the computer and typing `screen -r`.

7 Printing

You can print Postscript and text files. You have to print other files using an application which can print them. *Do not try sending dvi and gif etc files to the printer!*. The Unix command to print files is `lp` (or alternatively the `lpr` command), short for line-print (don’t ask why). If you print using a Unix application, you will normally get the option of editing the lp command so you can change which printer or options to use. Do try to view a PostScript file with `gv` before printing it to make sure it looks okay. I suggest printing text files with `a2ps` rather than sending them directly to `lp` or `lpr`.

The `lp`-style printing commands are rather random. I apologise for them (even though they are not my fault). The `lpr`-style commands are more coherent, but don’t allow you to set printer options.

To determine which printer you want to print on, ask someone which is nearest, or note down the name which is stuck on each one. The command to view what is on the *queue* - list of jobs to be printed - is:

```
lpstat -o printer_name
```

where printer_name is the name you should remember. To print a file on the printer:

```
lp -d printer_name filename
```

(I told you they were random!). If you realise you didn’t want to print that file, find the number of the job you printed (see the output of the `lp` command or repeat the `lpstat` command), and do

```
cancel job_name
```

(who knows why they called it cancel and not lpcancel!!).

You can alternatively print text with the following commands (which are more coherent), but they are not standard in Solaris, so don’t blame me if they break! Subtly the `lpr` command is different to `lp` in that it is safe to delete the file it is printing before it prints.

To view a printer queue:

```
lpq -Pprinter_name
```

To send a job:

```
lpr -Pprinter_name filename
```

To cancel a print job:

```
lprm -Pprinter_name job_id
```

7.1 Default printer

If you are nearly always printing to the same printer you should set up a default printer. The easiest way to do this is to add a line like the following to your .mytchrc file:

```
setenv PRINTER printer_name
```

to make `printer_name` your default printer. You won't need to use `-d` on the `lp` command for that printer after logging in again. Note that any programs started from firefox will not see this change unless you log out and log in again.

7.2 Printer options

The printers take their individual and bewildering array of strange options (unless you use the X-ray printing system). These might change from time-to-time, so consult the Users' Guide. The options apparently don't work for `lpr`.

```
# print one side of paper only
# (normally prints on both for black and white)
 lp -d printer_name -o simplex filename

# print on both sides of paper
# (normally single for colour printers)
 lp -d printer_name -o duplex filename

# print on transparancies for colour4 and colour5
 lp -d printer_name -o trans filename

# print on transparancies for colour3
 lp -d colour3 -o mtypeTransparency filename

# use manual feed tray (stick your own paper in)
 lp -d printer_name -o mpt filename
```

7.3 Colour printing

Colour printing is relatively *expensive*, so don't use it for frivolous things (e.g. train times web pages) or someone might come round and slap you on the wrists. Therefore preview colour documents on the screen before printing, and use black and white printers for drafts if possible (not checking colour balance obviously!). Unfortunately it is hard to get exactly the colours you want on the printer from the screen!

Also if you use StarOffice/OpenOffice for presentations, don't use a dark background as that will waste lots of the same colour for each page. Leave backgrounds blank if possible.

8 Choosing window manager

I recommend that you use gnome-desktop as your window manager and desktop environment. To do this you need a `.xsession` file containing something similar to this:

```
#!/bin/tcsh
xrdb -load $HOME/.Xdefaults
xmotd &
gnome-session
```

The `xmotd` line is optional and shows the message of the day.

Alternatively you can use the faster fvwm window manager. You need to have the following line in ∼/`.dtprofile`:

```
SESSIONTYPE=xdm
```

You will need to put your X startup script in ∼`/.xsession` rather than ∼`/.xinitrc`

9 Logging in remotely

9.1 NX

NX is the recommended way to log into the systems remotely. It is fast over even modest internet connections and secure. You log in with an NX client. Clients are available for Linux, Windows and Mac OS X. Only certain computers can be logged into with NX - see this guide: `http://www.ast.cam.ac.uk/nx`

See below for details on logging in using other methods (ssh and X over ssh).

9.2 Logging in from Microsoft Windows

To log in with a text based window, you can use the Windows program putty, available from `http://www.chiark.greenend.org.uk/˜sgtatham/putty/`

You can only log in directly to the muon1.ast.cam.ac.uk, muon2 and muon3 gateway machines under the current security arrangements.

If you want to run graphical programs remotely to a Windows computer, you either have to use commercial software (which costs money), or use a free program like Xming. A commercial possibility is WinaXe at `http://xwinpro.com/xwinpro.html` which is available in a 30 minute demo mode.

Xming is free software and is available from `http://www.straightrunning.com/XmingNotes/` Here are some instructions to get going. It's not too bad when installed.

1. Download the Putty Windows installer from `http://www.chiark.greenend.org.uk/˜sgtatham/putty/` and install the programs.

2. Download the Xming installer and font package from `http://www.straightrunning.com/XmingNotes/` and install them.

3. **Important!** You will need to connect to the computer you are intending to run your programs on once with Putty before Mingw will work! Putty needs the ssh keys in order to connect graphically.

4. Start the Xlauncher wizard (make sure X isn't already running - see note below - there should be no X in the Windows status bar)

5. Make sure ‘Multiple Windows’ is selected and click Next
6. Choose ‘Start a program’ and click Next
7. Choose the program to start (usually xterm or gnome-terminal)
8. Choose ‘Using Putty’
9. Enter computer to connect to (muon1 or muon2 or muon3.ast.cam.ac.uk)
10. Click ‘compression’ to get a faster connection over a slow link
11. Enter your username
12. Enter password
13. Click Next. You will now have to locate the Putty ‘plink’ program, which should be in Program Files, PuTTY. Select it and press Open.
14. Click Next, then optionally save the entered settings.
15. Click Finish to start the terminal.

If something goes wrong, close the X server by right clicking on the X in the bottom right if it is there, then choose exit. Start again by running Xlauncher.

You can start a complete desktop environment by choosing ‘Single window’ in the first dialog box, and ‘gnome-session’ as the program to start running.

9.3 Logging in from Linux

It is pretty trivial to log in and run programs remotely with linux, although this method requires a fast network for graphical programs (NX would make this faster but are not supported. VNC is supported — see the users’ guide — but you need to tunnel over SSH for safety).

You can simply type

```
ssh -Y -C username@muonX.ast.cam.ac.uk
```

where `username` is your username and `muonX` is the gateway system to log into (muon1 or muon2). `-Y` means forward the X connection (so you can start graphical applications), and `-C` means compress the data sent over the network (which speads things up over a slow network). You can start further applications (including gnome-terminal) from this logged-in window.

If you want to run an entire desktop from the remote system, there is a slightly more complex way of doing this. Firstly start a second X session by typing in a terminal window (make sure you have the spaces correct):

```
xinit -- :1
```

When the second session has started, you should get a single terminal window on a grey background. Move the mouse over that terminal, and run the ssh command above. After logging in type `gnome-session`. This should start up the remote session.

`ctrl+alt+backspace` should kill off this second session should you need to.

9.4 Logging in from Mac OS X

Logging in from Mac OS X is pretty similar to Linux. Firstly, you need to make sure that X11 is installed on your Mac. I believe from Mac OS X 10.5 (Leopard) it is installed by default. For 10.4 you can install it from the Mac OS install disc (under Optional Installs). You may have to download it online for previous Mac versions.

To run it, go to ‘Applications’, then ‘Utilities’ and start ‘X11’. A terminal window should then pop up. From that terminal window you can type the ssh command to log in remotely and forward connections:

```
ssh -Y -C username@muon1.ast.cam.ac.uk
```

muon1 could be muon2 here.

9.5 Reading mail remotely

If you are at a remote site, there is a webmail interface available at `https://www.ast.cam.ac.uk/webmail/`

10 Office packages

To do word processing, make presentations or use spreadsheets, there are two very similar offices packages available, OpenOffice and StarOffice. Linux systems use only openoffice, so I would recommend you stick to this (start with the `openoffice.org` command).

StarOffice includes some elements not present in OpenOffice, but OpenOffice is available to download for free on your home system. You can open Microsoft Office documents with either. OpenOffice is very handy for making presentations (like Powerpoint). See the Users’ Guide for help in setting up both.

11 Module environment

The system has a variety of optional components which can be loaded as modules (e.g. compilers or different programming language versions). In order to use these modules you need to set up the module environment first.

If you are using the tcsh shell (the default), you need to run the command

```
source /opt/ioa/Modules/default/init/tcsh
```

See the users’ manual for other shells. This command needs to be run in each shell, so add it to your `.mytcshrc` file to get it run by default.

After you have done this, you can use the module commmand to see what is available (at the time of writing):

```
> module avail
...
--------------- /opt/ioa/Modules/modulefiles ------------------------
fftw/2.1.5_double  gsl/1.14              openmpi/gcc/4.5.0/1.4.2
fftw/2.1.5_float   hdf5/1.6.10_parallel  openmpi/intel/11.1/1.4.2
```

```
fftw/3.2.2_double  hdf5/1.6.10_serial     openmpi/sunstudio/ss12u1/1.4.2
fftw/3.2.2_float   intel/11.0             python/2.7
gcc/4.5.0          intel/11.1             pythonmod/pyfits_2.3.1
gsl/1.13           intel-mkl/10.2.4       sunstudio/ss12u1
```

If you want to know what one of these module files does, use the `module help` command. To load a module you can use `module load`, e.g.

```
> module load gcc          # load latest gcc module
```

You can add a version if you want a specfic version of the software.