# Dual boot Linux–Mac OS X with partition for common home[User]

## Background and Overview

Believe it or not, for some, Apple's interface may be more restrictive than enabling for many applications. I am one of the perverse individuals who, having inherited a top of the range Mac Pro, found the inflexible interface and proprietary straightjacket built into the Mac Operating System (OS) too constraining after prolongued exposure to the Linux environment. I therefore set out on the frustrating and non-trivial, but ultimately mainly successful quest to make my machine capable of dual booting into both. For added convenience I took advantage of the underlying unix structure of both OS to set up a shared partition for the home–User files. Hence when working on files in either OS there are no problems with keeping track of changes.

Documentation on the web is mixed as is the success of the various options advocated. They are machine dependent and depend also on which versions of Mac OS X or Linux are being applied. I shall specify the hardware and OS I have used, but some of the processeses could well be generic. Nevertheless it is likely other setups, may respond unexpectedly. All of my efforts have been assisted by sources on the web, and I shall indicate those, which have been most effective in case you need to refer to the originals to address anything not covered here.

### Outline in brief

In principle it is possible to obtain the desired changes without losing an existing Mac OS and User files. However a first attempt will likely involve much trial and error and be time consuming. In conclusion I recommend that you save any User data from your existing Mac OS to external media such as a hard drive and reinstall it afterwards. This allows you to work much faster and experiment in the full knowledge that everything can be restored. You can also be assured that you will start with a very clean setup and in the long run you will probably save much more time and frustration.

Therefore this outline assumes that both systems will be installed from scratch using the OS installation disks as follows.

- Save your User files to an external hard drive or similar.
- Use Disk Utility to create three partions on the Mac hard drive:
  - HFS+ journaled for Mac OS X
  - HFS+ non-journaled for common user spaces
  - Fat32 for Linux OS (only size matters at this stage)
- Install Mac OS X
- Configure Mac Users to mount in shared partion
- Install refit on Mac
- Use Try Ubuntu to re-partition FAT32 to ext4 and linux-swap
- Install Ubuntu to ext4 partition
- Configure efi on Mac to select default OS
- Configure Linux home to mount on shared partition
- Synchronize user permissions on Linux with Mac uesr permissions
- Restore User files from external media to Mac

**This configuration**

| | | |
|---|---|---|
| Mac Pro 4.1 | Snow Leopard 10.6 | LED Cinema Display |
| Ubuntu 12.04LTS | 64-bit PC (AMD64) alternate install CD | |

Before you proceed ensure you have the following:

- An external drive to store your data back up
- A copy of the *Mac Pro Mac OS X Install DVD*
- Download and copy an ISO image of Ubuntu 12.04LTS *64-bit PC (AMD64) alternate install CD* to CD/DVD [1]
- A spare monitor with DVI connectivity (precautionary)
- A usb connected keyboard and mouse (precautionary)

[1] `http://releases.ubuntu.com/precise/ubuntu-12.04.1-alternate-amd64.iso`
see e.g. `http://hints.macworld.com/article.php?story=20060619181010389` how to burn the ISO
Check the Ubuntu installer is correctly formatted by testing it will boot from the super drive.

# Back up

It is possible to resize the Mac partition using Disk Utility without loss of data and in principle there is no need to start from scratch. My experience was that progress was much slower, because I had to tread so carefully in case I lost any data. In addition trial and error meant that the files left over from aborted attempts left a messy footprint, that I wasted a lot of time trying to remove. So save yourself the grief and start again.

## Time machine

I never fell in love with Mac long enough to have spent any time setting up cookies and personal preferences. If these are important to you, time machine will allow you to save them and restore them after you have completed this installation, along with your files.

## rsync Users

I inherited a machine with my predecessor's data, both work and personal, still installed. Using rsync preserves the permissions and date of the files being transferred. Prepare the external hard drive with a sufficiently large HFSplus unjournalled partition, so it can be read/written by either OS. Open a terminal and from the command line type

```
>sudo rsync -avu /Users/fred/ /Volumes/Untitled/fred/
```

`sudo` invokes super user privileges so hidden files and files for which you do not have user permissions can be copied. `-avu` applies "archive" mode, which ensures that symbolic links, devices, attributes, permissions, ownerships, etc. are preserved; "verbose", printing to screen; and "update" so that only older files with the same name and location are overwritten.

Replace `fred` with the user whose data you wish to retain and `Untitled` with the label of the external hard drive. (See note 4 below)

Altenatively do not invoke `sudo` and only the unhidden files will be transferred. This will be faster, and require much less storage, but will not retain cookies and other User settings. Or instead of copying the whole User data, you could be selective of only important directories.

# Repartition the Mac hard drive

Insert the *Mac Pro Mac OS X Install DVD* and boot from the DVD.
(To do so, restart the machine; when the chimes sound hold down the 'c' until the disk starts to load. Notes 2 and 3 below.)

Ignore the options to *Install Mac OS X* and from the tab menu at the top of the screen select `Utilities->Disk Utility`.

Note, so far no changes have been made to your machine. **The next step is the point of no return.** However proceed in the confidence that if this does not work you can re-install the operating system and restore your files from your back up and all you will have lost is some hours (days) of your life. :-)

## Create three partitions

Select the MAC OS Hard Drive (not the partition) and select the 2nd tab *erase*. This erases everything on the disk. Open the tab current and select 3 partitions. You will have three equal partitions.

### Partition for MAC OS X

Select the first partition. With Snow Leopard 10.6 The full OS X installation used about 13GB. With 2TB of disk space I allowed 64GB for this partition, but 30GB or maybe even less should still allow more than enough to handle the most extreme OS X upgrades.

Label the partition 'MacOSX' (optional) and ensure its type is HFSplus journalled. This is necessary for a stable file system.

### Partition for Linux

The Linux OS is a similar size to Mac OS X. It also requires a swap partition of at least 1GB. With 32GB of RAM it is unlikely I would require swap memory, however I chose to match that and opted to make this partition 96GB in total. A total of 30GB however, should certainly be more than adequate.

Select the next partition. Label the partiton 'linuxOS' (optional) and ensure its type is FAT32. The type is less important than the size as it will be changed later.

### Partition for the shared user files

The last partition will expand to fill the rest of the disk. In my case about 1.8TB. Relabel this 'common' (optional) and ensure its type is HFSplus non-journalled. This is critical, because if it is journalled you will only be able to read it and not write to it from linux.

Note the optional labels are not necessary, but it is useful if they are unambiguously identified. Save the settings and exit `Disk Utility`.

# Install Mac OS X

This should return you to the installation menu. Select *Install Mac OS X* and follow the standard installation. Once completed boot up as normal. At this stage do not add any updates or install the 2nd disk *Mac Pro Applications Install Disc* as these are not yet essential and take over an hour to complete. (The disk appears redundant in retrospect.)

# Edit OS X to mount Users in shared partition

Here I referred to the following:
`http://www.lnx2mac.blogspot.co.uk/2010/09/moving-os-x-users-to-separate-partition.html`

## Log in as root

You need to edit the settings on your User account, so to do so you need to be logged out and log into your machine as root. To do so open a terminal. From finder select `Applications->Utilities->Terminal` and open a terminal window. At the prompt `#` type

```
# sudo passwd root
```

You will be prompted for your user password to enable super user (su) privileges. Then you will be prompted to enter a password for the root user and confirm it again. When you have done this close all applications and log out. At the new login screen select 'Other User', enter username `root` and then for password enter the password you have just set for it. You are now logged in as the root user so are free to edit your User settings.

## Remount /Users in shared partition

Move the current `/Users` to a safe temporary place. Open a terminal and at the prompt type

```
# mv /Users /Usersold
```

then prepare the new /Users mount point with appropriate ownership and permissions

```
# mkdir /Users
# chown root:admin /Users
# chmod 755 /Users
```

We require the UUID (Universal Unique Identifier) for the shared partition. Open `Disk Utility` in `Applications->Utilities->Disk Utility` and right click on the partition you shall use for `/Users`, which I labelled as 'common'. Select 'Information' (or key i). Select the value of 'Universal Unique Identifier' and copy it ( c). Close the information window and if the partition is mounted, right-click on it, and 'Unmount' it. In the terminal

```
# open -a TextEdit /etc/fstab
```

which will open an empty text file. Make sure it is in plain text mode (from the `Format` menu select 'Make Plain Text'). Append the following line to the file

```
UUID=TheValueCopiedAbove /Users hfs auto
```

so that after editing you should have something similar to

```
UUID=84BA91DE-C37F-F13D-B5C9-FECA5184DEB7 /Users hfs auto
```

Save the file and quit TextEdit.

## Mounting the new /Users for the first time

At this stage, our `/etc/fstab` file is ready, and, on the next boot, Mac OS X should be able to automatically mount our `/Users` partition.

Verify that it can be mounted, as expected, on `/Users`. In `DiskUtility`, find the partition you designated for `/Users`, right-click it and select 'mount'. The new `/UsersVolume` should appear in `Finder` (and in the Desktop if you selected that in your Finder's Preferences).

As we want to keep the same level of permissions as we had before the move, we'll make sure file ownership is preserved. In `Finder`, locate the 'Users' Volume, right-click on it and select 'Get Info'. Verify that the 'Ignore ownership on this volume' is *NOT* checked.

### Moving the home directories contents to the new /Users partition

With our partition mounted, and permissions properly set, it's time to move all users' home directories to the new partition. In the terminal

```
# mv /Usersold/* /Usersold/.[^.]* /Users/
# rmdir /Usersold
```

### Reboot and check /Users

We can now reboot and verify the `/Users` is correctly mounted. At this time, all our users home directories are in the new partition, and as we instructed OS X to mount it automatically (via the `/etc/fstab` file), it should be accessible just as before. If you open the Volume in `Finder` containing the shared partition, `Users` will be contained within it and also accessible directly from `Finder`. Mac is now ready, but defer updates or other changes until everything else is working to save time in case you need to repeat anything.

# Install rEFIt and select default OS

In Mac OS X download the .dmg installer from `http://refit.sourceforge.net/` Unpack and install the package. Then reboot and log in again to Mac OS X. At this stage rEFIt will not have changed the boot menu. If you want your default OS to be Mac OS X do nothing. Configuring Linux, will require rebooting a few times so for now you probably want Linux as default. If you want your dafault OS to be Linux open a terminal and at the prompt type

```
# sudo vi /etc/efi/refit/refit.conf
```

replacing `vi` with your preferred editor, find the line

```
#default_selection L
```

and remove the comment `#` at the start of the line, then save and close. If there is no such line just add it at the end without the `#` in front of it. Note the default can be returned to Mac by replacing `L` with `M`. Next time you reboot the rEFIt menu will appear.

# Try Ubuntu and prepare the partitions

At this stage it is worth connecting a usb keyboard and mouse. I never found a problem with the wireless keyboard and linux, but my mouse had already been *borrowed* before I inherited the machine so on that I cannot comment.

I encountered problems with the monitor connected via the miniport to the graphics card, which could not get past the grub2 menu. I think this procedure should now work without the second monitor connected to a dvi socket, but I recommend that as a precaution you have one ready. If you find a blank screen after booting Linux just plug in the spare and try again. The spare should be redundant at the end of the installation.

Note Ubuntu has the advantage that it supports working with GPT, it has gptsync and it provides a means to install GRUB to the root partition, so it simplifies the process of synchronizing the Linux installation with the Mac hardware and OS X.

## Boot from the Ubuntu install disk

Insert the Ubuntu disk in the superdrive and retart the machine. when the chimes start hold down the 'c' key until the drive starts loading (notes 2 and 3 below). When prompted select your keyboard language and then choose 'Try Ubuntu'.

Open a terminal by holding simultaneously the keys `ctrl-alt-t` (cmd, option, t on the Mac keyboard). At the command prompt `>` type:

```
> sudo gparted
```

which will launch a partition console.

## Linux and swap partition

Select the FAT32 partition and delete it. Select the free space and create an `ext4` partition the size you require for your Linux OS, leaving at least 1GB unused. In my case as mentioned earlier I used 64GB with 32GB left over. Select the remaining free space and create a linux-swap partition. You should now have an EFI partition, the Mac OS X partition in HFSplus journalled, the shared partition in HFSplus non-journalled, an ext4 partion and a linux-swap partition. The EFI partition is created during the installation of the Mac OS X.

Note the order of the partitions is not entirely arbitrary. GPT can ordinarily only handle 4 positions. Although there are work arounds, which are unavoidable for a triple boot arrangement including a Windows OS, here we can keep it simple. EFI and Mac OS X take up positions `sda1` and `sda2` respectively. Ensure you allocate the `ext4` partition to either `sda3` or `sda4` and leave `sda5` to either the swap or HFSplus partition. Apply the changes then exit gparted.

## Converting Ubuntu into EFI mode

I found this step essential to solve the issues with booting into Linux using the miniport connection on the monitor (thus making the spare monitor redundant).

Install and run `Boot-Repair` (see `https://help.ubuntu.com/community/UEFI`) as follows:

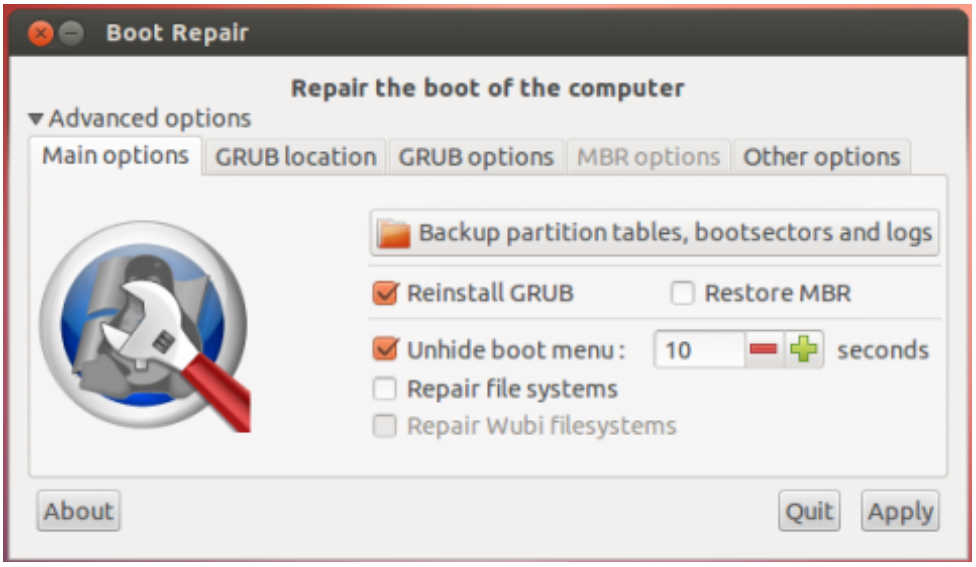
Open a terminal (`ctrl-alt-t`) and enter the command

```
> sudo add-apt-repository ppa:yannubuntu/boot-repair && sudo apt-get update
```

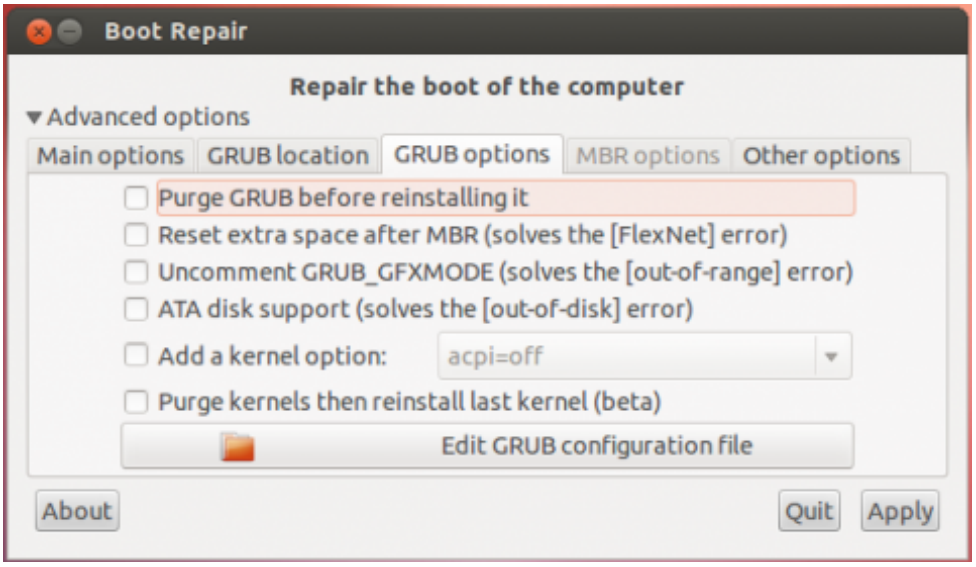
press enter to continue then enter

```
> sudo apt-get install -y boot-repair && boot-repair
```

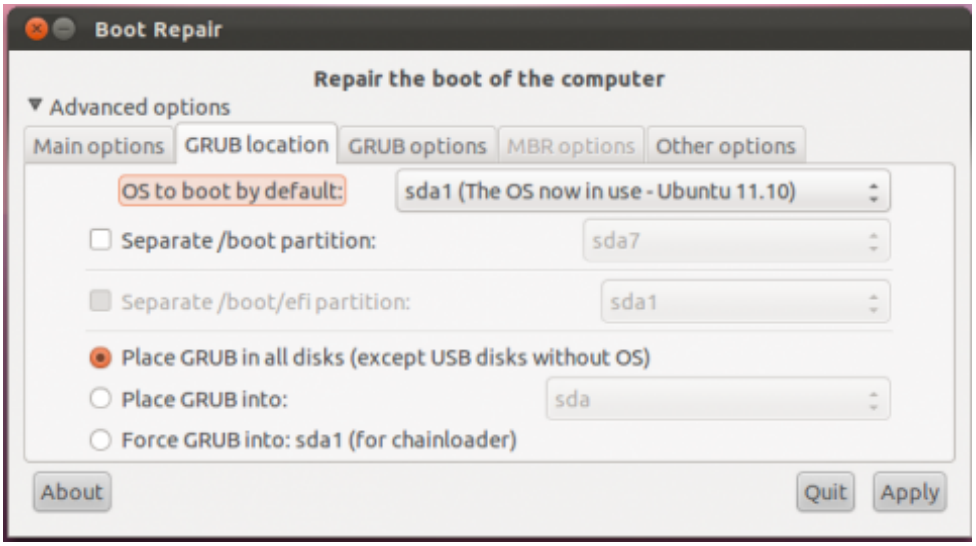
When the dialogue box appears select advanced and the Boot Repair will appear something like this:



From the GRUB options tab tick Uncomment GRUB_GFXMODE(solves the [out-of-range] error), tick Add a kernel option: and then select from the menu nomodeset radeon mode=0.




Select the GRUB location tab then tick Place GRUB into: sda, tick the box Separate /boot/efi partition: sda1 and Apply.




The system is now ready to proceed to Linux installation.

# Install Ubuntu

- Select the icon to launch the installer.
- Select the third option to manually partition 'Something Else'.
- Select the ext4 partition and click 'change'.
- Select
  - 'use this space'
  - 'EXT4'
  - mount point root '/' from the drop down menu.
  - the box to format the partition.
- The default location to install the boot loader (grub) is '/dev/sda'
- Complete the installation of Ubuntu. If you have an ethernet internet connection select 'Download updates while installing' and 'Install this third-party software'. Wireless connection may not be supported at this stage.

Reboot and the rEFIt menu will appear. If more than two boot options (Mac and Linux) are presented select the option labelled EFI\ubuntu\grubx64.efi in EFI. From the grub menu select the linux default. At this stage it is worth applying the software updates and add hardware. NVIDIA version experimental-310 worked for me.

The following set of commands in the terminal may help with graphics problems.

```
> sudo cp /etc/X11/xorg.conf /etc/X11/xorg.conf.old
> sudo apt-get update
> sudo apt-get install aptitude
> sudo aptitude reinstall lightdm
> sudo nvidia-xconfig
```

check that the version of xorg.conf matches the driver you have installed. You may also want to check

```
> sudo vi /etc/default/grub
```

and add the uncommented lines below

```
GRUB_GFXPAYLOAD_LINUX=text
    ...
#GRUB_GFXMODE=640x480
GRUB_GFXMODE=1920x1200
```

where 1920x1200 is the default resolution of your monitor and the commented out 640x480 is the grub default resolution. Then apply

```
> sudo update-grub
```

A reboot will be required. Once you have booted successfully with graphics and updates installed proceed to mount the home area on the shared partition.

# Configure Linux home to mount on shared partition

We now need to mount `/home` on the HFSplus non-journalled partition. In my installation the LED Cinema screen switches off when the grub menu initialises and to boot up the 2nd monitor must be connected, otherwise the system stalls. Once logged in it is worth checking for hardware updates and installing the latest proprietary drivers. This might solve the grub problem and you can dispense with the 2nd monitor. Unfortunately I have not so far. Reboot again if necessary.

## Remount /home in shared partition

Once you have finished updates you need to edit the `/etc/fstab` file, which controls what partitions are mounted at boot. Here is an example of its contents following the preceding installation:

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    nodev,noexec,nosuid 0       0
# / was on /dev/sda4 during installation
UUID=66dec0db-5a6b-45b6-ba49-807cfa320504 /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=79323f76-75d0-4dac-af59-6dd8056b12ff none            swap    sw              0       0
```

To find the UUID for the shared partion in a terminal type

```
> sudo blkid
```

Copy the shared partition UUID number from the list. If you labelled it earlier it will be easy to identify, or it will be the `TYPE="hfsplus"` with the later sda number. From the terminal type

```
> sudo vi /etc/fstab
```

replacing `vi` with your preferred editor. Then append to the end of `/etc/fstab` the lines of the form

```
  # /home was on /dev/sda3 following remount
UUID=3035ceba-312e-32ee-bc41-d8c163a4491a /media/home hfsplus nodev,nosuid    0       2
```

Save and close the file then type

```
> sudo mkdir /media/home
```

to create a new directory later used for temporarily mounting the shared partition. The `/etc/fstab` file will be edited later.

Restart the machine into ubuntu and to ensure the partition is mounted correctly type

```
> sudo mount -a
```

followed by

```
> sudo rsync -aXS --exclude='/*/.gvfs' /home/. /media/home/.
```

in order to copy all the files from the current home to the shared home, without overwriting any data already there from Mac OS X. There are likely to be some warning messages, but these do not prevent the transfer.

```
> sudo diff -r /home /media/home
```

should confirm that the only differences are /.gvfs files and some Mac files. Again

```
> sudo vi /etc/fstab
```

and replace the last line in the file with

```
UUID=3035ceba-312e-32ee-bc41-d8c163a4491a /home     hfsplus nodev,nosuid    0       2
```

in which `/media/home` is now replaced by `/home`. Save and close, but do not reboot just yet. At this stage there are two copies of `/home`, the new one mounted as `/media/home` and the old one mounted only as `/home`. The contents of the old one need to be moved out of the way to create an empty "placeholder" directory to act as a mount point for the new partition. In the terminal type the following string of commands to do this all at once:

```
> cd / && sudo mv /home /old_home && sudo mkdir /home
```

Reboot again into linux and the shared partition will now be mounted as home.

## Synchronization of files between OS

There remains a potential conflict over ownership and permssions due to a difference in the default UID (user id's) allocated by the two operating systems. OS X uses 501 by default and Linux uses 1000. To synchronize these one of them needs to be modified and Linux is easier.

### Log in as root

To change the user permissions it is necessary to be logged out as the user and to login as `root` to change the mount point for the `/home`. In Ubuntu `root` is disabled by default, so manual login and root need to be enabled. Open a terminal and type

```
> sudo passwd root
```

Enter your password if requested and create the root password as before on the Mac OS X. Then type

```
> sudo sh -c 'echo "greeter-show-manual-login=true" >> /etc/lightdm/lightdm.conf'
```

To enable manual login. Restart the computer and load Linux. When prompted enter the username 'root' and for the password enter the one you have just created. You are now logged in as `root` and can edit your user home.

### Modify user UIDs

Open a terminal and type at the prompt:

```
> usermod --uid 501 fred
> chown -R 501:fred /home/fred
```

The former sets the UID to 501 for user `fred` (replace with your username). The latter changes the ownership of all files in /home/fred from 1000 to 501.

To ensure in later sessions linux does not reset the UID to 1000 it also advisable to edit the file

```
> vi /etc/login.defs
```

Find the line where `UID_MIN` is set as 1000 and change the value to 501. Save and close the file and reboot into linux logging in as yourself.

### Remove root and manual login

Open a terminal and edit the file replacing `vi` with your preferred editor

```
> sudo vi /etc/lightdm/lightdm.conf
```

and edit the line `greeter-show-manual-login=true` to `=false` to remove manual login then save and close the file. Then type

```
> sudo passwd -l root
```

to disable the root password. You will still be able to invoke sudo when required, but will not be offered to login directly as root.

## Restore files and complete updates

The OS installations are now complete. Next time you load Mac OS X you can restore your files.

If you used `rsync` as above you can restore your files by typing

```
#sudo rsync -avu /Volumes/Untitled/fred/ /Users/fred/
```

providing you kept the same username as you had originally. (If you changed it see note 4 below)

Continue to update and add new software packages to either OS according to your preferences. Bon voyage.

## Supplementary notes

1. Without converting Ubuntu into EFI mode the LED monitor hangs when the Grub menu is invoked and it is impossible to boot into Linux. If you still encounter problems check the settings in `/etc/default/grub` to see if you can fix this.

2. To open the superdrive if the operating systems become inoperable and you need to boot from the installation disk. Turn of the machine and restart it. When the chimes sound hold down the keyboard F12 until the drive opens.

3. To boot from an installation disk in the superdrive Restart the machine with the disk in the supredrive. When the chimes begin hold down the keyboard 'c' until the installation disk starts loading.

4. Linux file names and directories are case sensitive and use of lower case is more convenient. Mac file and directory labels often include spaces, which will not be interpreted as part of the name by Linux, so spaces should be avoided. If you have spaces in your username, then e.g. 'Fred A' would need to be typed as '`Fred\ A`' as spaces otherwise terminate the string. Alternatively choose 'Fred_A', 'FredA' or 'Fred-A' to avoid spaces.