FIRST STEPS LINUX BEGINNERS SERIES

Filesystem Organise your partitions and folders

How to set your machine up so that you'll always know where to find any piece of information, photo, song or document. Andy Channelle is your guide to control-freakery.

LAST TIME

Buzzword fever took over, as we looked at instant messaging, *Skype*, IRC and *Google Talk*. If you missed the issue, call 0870 837 4773 or +44 1858 438795 for overseas orders.

Everyone loses some data at some point: a picture, document, address or vital soundfile. It's a fact of computer life that most people have come to accept – wrongly. Regular backups, for example, are one way of ensuring that any lost data be retrieved without much fuss. But how can we make sure that we don't lose a file in the first place?

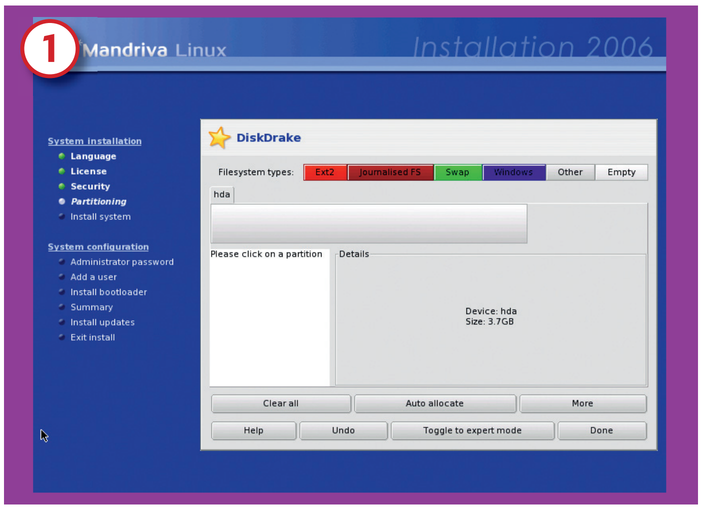
Like many things, it comes down to being organised, which means designing a file structure that takes into account the ways in which we use a computer; and – and this is the tricky bit – making sure that we always drop, rip, save and import files into the correct folders.

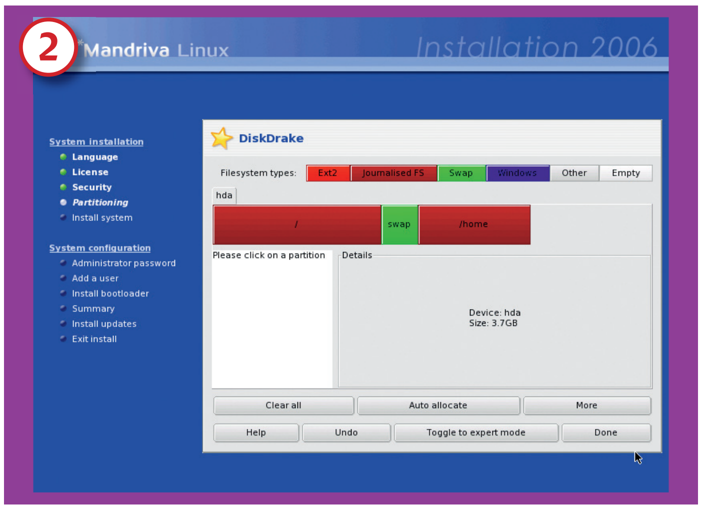
The first thing to consider, and the first part of this tutorial, is how to partition your drive. Dividing your filesystem into two or more partitions is one of the best ways of keeping in control of your data. It's common practice to reformat the entire hard drive when installing a new version of Linux, but this wipes all your personal files as well as configuration settings, and it's then a laborious process to set up your new system to work just as the old one did. However, if you go the trouble of putting your home directory on a separate partition, all this hassle can be avoided. Simply reformat the partition that contains the operating system, and leave all your files and settings safely intact in your home partition.

The (slightly) bad news is that, if you already have a fully configured Linux system with **/home** on the main partition (also known as / or the root partition) then the only really practical way to do this is to back everything up and reinstall Linux with a new set of partitions. This could either be a blessing or a curse, depending how much stuff you've accumulated in your /**home** directory. But either way, don't worry – this is not as scary as it sounds. I'll step you through the process, using Mandriva Linux 2006.

PART 1 – REINSTALL LINUX TO MAKE A SEPARATE /HOME PARTITION

Warning: this process will DESTROY all the data on your computer. Make a complete backup of your precious documents, photos and anything else before you start the installation. *LXF* can't take responsibility for lost stuff, and we can't retrieve documents from formatted disks. Back up now.



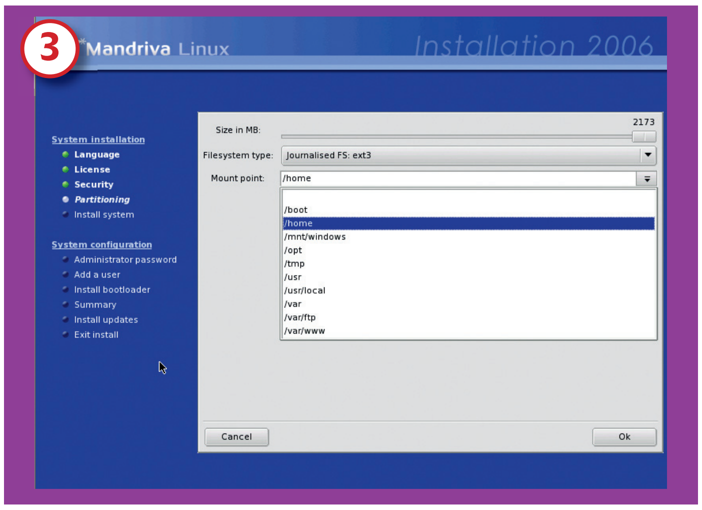


Reboot and find the partition manager

Once you've backed up, reboot with the Linux installation CD, and go through the bootup sequence until you come to the partition manager. This is where you decide if the Linux system will live beside an already installed Windows partition, use the current partition setup or create a new scheme. In this screen shot, Mandriva is addressing a small and unused hard disk. To install on to a completely blank hard drive, select Clear All from the tools along the bottom of the screen. WARNING! This will destroy all your data – so back up first. You can also select each partition in turn and opt to remove it or edit it.

Set up your three partitions

Select the disk to be partitioned and hit Auto Allocate. This will create three partitions by default: / (root), **/home**, and **/swap** as a sort of overflow. Most other distros will create two partitions – / and **swap** – so you will need to add a third using either Ext3 or ReiserFS, accessed as **/home**. You don't have to accept the Auto Allocate partitions just as they are. To adjust them, select one of the planned partitions with the mouse and use the Resize option that appears in the Actions box on the left-hand side of the window. The root partition should be at least 2GB in size, but if you have the space I would recommend 8–10GB. The swap partition needs to be approximately twice the PC's RAM, and the rest can be used for **/home.**



Size in MB: 2173
Filesystem type: Journalised FS: ext3
Mount point: /home

Set the mount point

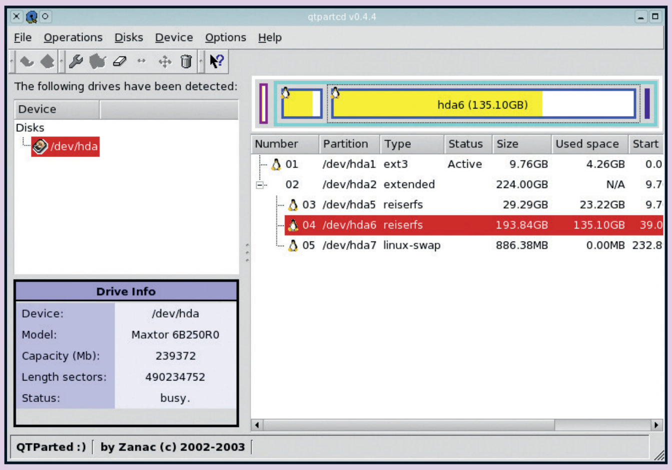
When you create a separate **/home** partition, it must be mounted as such. From the main screen, select the partition to be edited and choose the Mount option from the Actions section. In the resulting dialog box there is a drop-down list containing the most common set of mount points. Select the **/home** option. If you are installing Linux alongside Windows you can use this space to make sure that the Windows partition is automatically mounted on bootup. Select the Windows partition, hit the Mount button and choose **/mnt/windows** from the drop-down list.

REPARTITION WITHOUT MANDRIVA ON the DISC

SUSE comes with its own partitioning utility, found under System > Partitioner in the *Yast* configuration tool. It's not that easy to use, and only really worth considering for a fresh install.

The most comprehensive partitioning tool (and also the most difficult to use) is *Parted*. This is a command line tool for creating, destroying, resizing, checking and copying partitions. Luckily, there are a few graphical front-ends to this monstrous tool, the most popular of which is called *QtParted*. If you've ever used *Partition Magic* on Windows, *QtParted* will feel familiar. It provides a horizontal box that represents your hard disk, with each partition coloured in to indicate how much data it's holding. You then drag each coloured partition to resize and create new blocks.

Find *Parted*, *QtParted* and *GParted* (for Gnome) on the coverdisc.



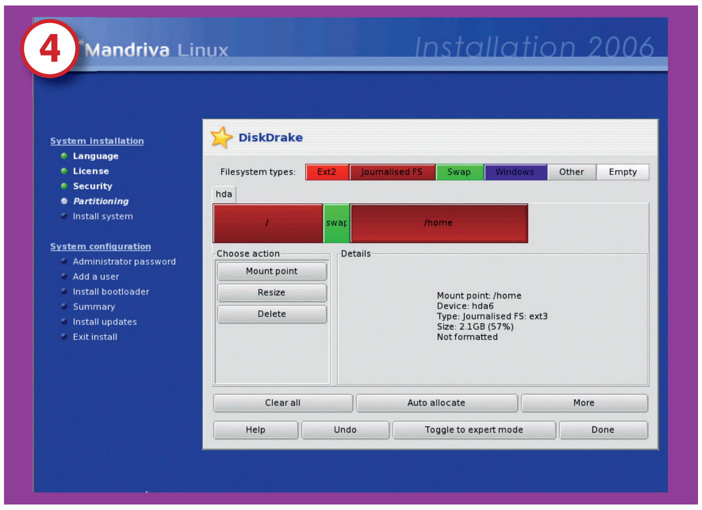
The following drives have been detected:

Number	Partition	Type	Status	Size	Used space	Start
01	/dev/hda1	ext3	Active	9.76GB	4.26GB	0.0
02	/dev/hda2	extended		224.00GB	N/A	9.7
03	/dev/hda5	reiserfs		29.29GB	23.22GB	9.7
04	/dev/hda6	reiserfs		193.84GB	135.10GB	39.0
05	/dev/hda7	linux-swap		886.38MB	0.00MB	232.8

Drive Info	
Device:	/dev/hda
Model:	Maxtor 6B250R0
Capacity (Mb):	239372
Length sectors:	490234752
Status:	busy.

***QtParted* takes its inspiration from *Partition Magic* on Windows.**

»



Mount point: /home
Device: hda6
Type: Journalised FS: ext3
Size: 2.1GB (57%)
Not formatted

Check the setup before confirming

Here is the result. From the automatic setup created in Step 2, the **root** partition has been shrunk slightly, **swap** has been moved along and a new **/home** partition created to house the previously backed-up documents. This is the whole point of the exercise: when you come to reinstall or upgrade the Linux system, you can simply select Use Existing Partitions and make sure **/home** isn't set to format. That way, once the system is reinstalled, the documents, mail and other files will be exactly as you left them.

If you're happy with everything, you've reached the critical step; the point of no return. When you click Done and hit the OK button, the selected hard disk will be completely erased and a new partition table will be written. So this is a good time to make sure that you really have backed all your files up. Once this is done, the installation will continue.

PART 2 – STRUCTURE YOUR DATA

QUICK TIPS

■ If you're using a computer on a network, it's worthwhile making /Shared available to users on your network. If you're still dual booting, it could even be sited on a Windows partition.

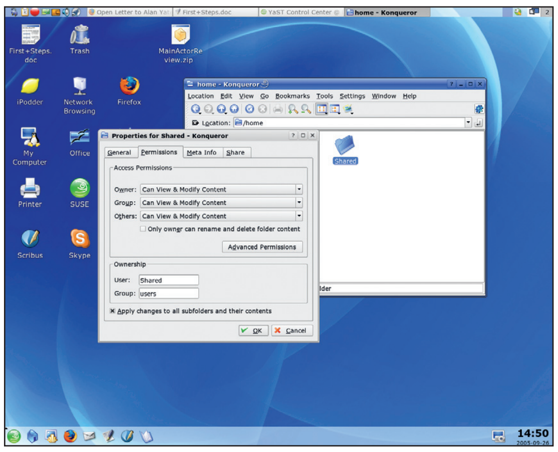
■ To keep items in folders private, it is possible to save documents from many Linux applications, including *OpenOffice.org*, in an encrypted format so that anyone attempting to access the file would need to input the right password before it was opened.

After you've created a separate partition for /home, the first level in the file structure is the users' own directory. A Linux system (like other Unixes) is a multi-user operating system: everyone who uses the computer can have their own account that can be configured right down to the desktop image. Moreover, each user has a folder inside **/home** labelled with their username. You can set up user accounts at install time or post-install, though you will need root privileges to do the latter.

We have three users to add (Rita, Sue and Bob) and each of these will have their own password. I've always found it useful to add an extra account – and I usually do this with a very simple password – called Shared, which I use to store documents or files that every user might need to access. Of course, this isn't the place to store file downloads from your bank, but it makes the perfect repository for digital photos, music and video files, as it saves wasting storage space by replicating files. The /**Shared** folder will be set up in a similar way to the user folders, though with different permissions *(see Read, Write And Execute Simplified box, far right)*.

So we should have four user directories (Rita, Sue, Bob and Shared), which can now be populated with subdirectories for storing various files. We'll start with the **/Shared** folder.

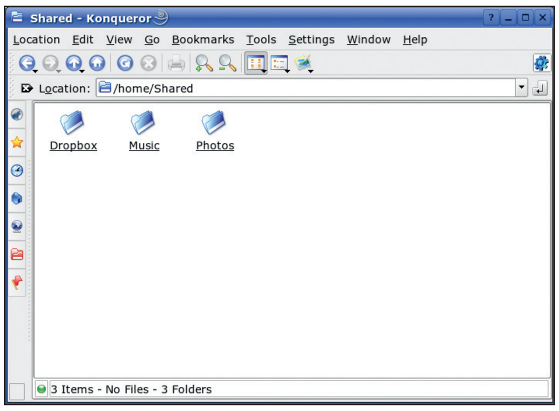
This is going to have three subdirectories: **Music**, **Photos** and **Dropbox**. The first two are obviously going to hold music and photos, but the third is going to be a space where the three



home - Konqueror
Location: /home

Properties for Shared - Konqueror
Access Permissions
Owner: Can View & Modify Content
Group: Can View & Modify Content
Others: Can View & Modify Content
☐ Only owner can rename and delete folder content
Ownership
User: Shared
Group: users
☒ Apply changes to all subfolders and their contents

Setting permissions in KDE is as simple as selecting options from a drop-down list.



Shared - Konqueror
Location: /home/Shared
3 Items - No Files - 3 Folders

The /Shared folder should look something like this.

users can drop files for each other – so if Rita wanted Sue to check over a piece of written work, she could drop it into **/Dropbox** and it would be accessible by all.

To make **/Shared** universally accessible you will have to adjust the permissions for the directory. You may find that **/Shared** is set by default as read-only for other users, so we'll have to change the permissions as the root (or super) user. This is possible by opening either *Konqueror* or *Nautilus* in Superuser mode, which should be under the K menu or Gnome panel in the System > File Manager section. Once the file manager has opened, navigate to **/home**, right-click on the Shared icon and select the Properties option. Under the Permissions tab, set each option so that Owner and Group can view and modify content. Also ensure that the option marked Only Owner Can Rename And Delete Folder Content is not selected, but that Apply Changes To All Subfolders And Their Content is.

Hit the OK button and close down the superuser file manager. You should now be able to open a normal file manager, access the **Shared** directory and add, remove and manage files within it from any account. We are creating three directories. Right-click anywhere in the directory window, select Create New > Folder... and enter **Music** as the name. Do the same for the **Photos** and **Dropbox** folders.

The next stage is to add some symbolic links (often shortened to 'symlinks') to each user's directory so they can access the folders within **/Shared** as though they were part of their **/home** directory. To do this, go into one of the user accounts and open up the file manager. Now right-click anywhere within the main window and select Create New > Link To Location (URL)... to open the symlink dialog. Then use the Browse button (with the small file icon on it) to open a file selector and choose the **/home/Shared/Dropbox** folder. Once this is accepted, right-click on the newly arrived icon and select Properties again, but this time click on the big icon image to open the icon browser where you can select an image appropriate for each user.

Now when Rita double-clicks on this icon it will open up the **Dropbox** folder within **/Shared**. She can open and save things normally in this folder and can also simply drop files in there so they are accessible to Sue and Bob.

Create links to **/Shared/Music** and **/Shared/Photos** to ensure Rita can also access the other content too.

I want some privacy

By default, Rita's **/home** directory will be accessible to the other users on the system, though only she can change or delete the content of any of her folders. Of course, she can save documents into **/Shared** if she wants to let Sue and Bob read and amend them – but what if she wants to keep some files to herself?

Once again, permissions can be used to restrict Sue and Bob from browsing into sensitive areas. In the **Rita** account, create a new directory and call it Private (or even PRIVATE) and give it a nice icon. Right-click on the resulting icon and select **Properties**. You can set the Permissions tab so that only the owner can read and write to the directory; other users are forbidden.

So now we have a directory structure which links to a shared folder and a private folder to keep all our secret stuff. All that's left is to set up sensible folders for what you might call 'local' content – word-processing files, pictures, downloads and so on. The best way to make sure the system doesn't become cluttered is to go for sensible divisions in the structure of your home directory (and to always make sure you use the schema you've set up). At the very least, you will need folders for Documents, Downloads, Apps, Archive and Junk.

■ **Documents** For work in progress and documents you know you'll need in the near future.

■ **Downloads** This will probably end up full of .tar.gz and RPM files, sounds, desktop wallpaper and other assorted stuff. Take care to clean it out occasionally.

■ **Apps** Some applications are distributed as static binaries. If you're the only user who needs access to them, you can store them in here. If you wanted to follow traditional Unix naming you could call this one **bin**.

■ **Archive** When a job or piece of work is finished and discarded, shove it in here. At the end of the month, turn the whole thing into a zip file and burn it to a CD – no more backup hassles.

■ **Junk** This is a paranoid step, but **Junk** is the space I use before Trash – which is a prelude to unrecoverable loss. At any given time, the **/Junk** folder will contain three months of zips from **/Archive** and other assorted bits of nonsense. When it hits 4GB I burn the directory to a DVD and move the contents to **Trash**.

And that's it! An organised way of working awaits... LXF

NEXT MONTH
We'll look at ways of accessing your files over a local network (from any machine) and over the internet.

READ, WRITE AND EXECUTE SIMPLIFIED

Access state	File	Folder
Read	The user can open and view the contents of the file. No changes can be made. Many system files you encounter will be configured to allow users read-only access.	A listing of the folder contents can be displayed either through a file manager or CLI command such as *ls*.
Write	The user can read and edit the contents of a file. The owner should have these privileges by default for most documents in your home directory.	Folder contents can be displayed, moved and deleted. In a multi-user system it is sensible to deny other users write access to your home directory.
Execute	This permission is used for applications or executable binaries.	A folder can be have execute permissions if it needs to be accessed by an application or script.

When talking about Linux file permissions, each file has three possible states: read, write and execute. These states are user-specific, which means that the permissions may change depending upon who is attempting to access a file or folder. In our Rita, Sue, Bob and Shared setup, for example, when Sue is trying to access **/Sue** she will have read and write access, but if Bob attempted to access it he would only have the option to view the content.

Permissions can also be expressed as r (read), w (write), x (execute) and - (deny) and these will be displayed as a nine character text string in some file managers. It's easy to break this down into three sets of three to cover **owner**, **group** and **others** as we did above. In the scheme **rwxrw-r--**, the owner can read, write and execute; members of the group can read and write; and everyone else can only read.

Number	Read	Write	Execute
0			
1			✓
2		✓	
3		✓	✓
4	✓		
5	✓		✓
6	✓	✓	
7	✓	✓	✓

Some hardened hackers might try to baffle you with numbers, but number permissions are really quite easy – especially if you have a handy table stuck to the wall above your computer. Each set of permissions is given a numerical value.

Our **/Shared** directory above would have the permissions **777**, because it allows full, unadulterated access to the owner, the group – that is, all the other registered users – and everyone else. Rita's
/PRIVATE directory (when viewed by Rita herself) would be set to **700** or, in terms of a text string, **rwx------**.