Question and Answer

The first session of 2010 was devoted to a few questions and answers. This paper represents the attempted answers to the questions given in advance of the session, which were:

What is Linux? What are packages?

What is the difference between a file system and a partition? What are the pros and cons of partitions?

What is “Linux”?

by Andy Pepperdine

When you use a general purpose computer, like the one on your desk, to do various tasks, it is running a variety of pieces of software; software written to help you with the jobs to be done.

This software has a certain structure to it, no matter what system you employ. Part of it interfaces directly to the hardware itself; the screen, the keyboard, the mouse, the internal memory (RAM), the files held on a hard disk, the network connection to enable you to use the internet, the sound system, the memory sticks and cameras that you attach to it, etc. None of this could be done without the underlying software that is called the **operating system**, the most important part of which is known as the **kernel**.

Linux is, strictly speaking, a kernel. On its own it is pretty useless because it does not have any knowledge of what you want to do when you type. It merely routes the key presses through to another piece of software called an **application**, but it does keep track of which application is to receive the key strokes.

Applications are what makes the computer useful. These are things like word processors, image manipulators, financial packages, e-mail clients, internet browsers, etc. On any Linux system, there is a fundamental set of applications that enable you to execute commands from a command line process, or terminal emulator. This very basic set of commands comes from the GNU stable, and for this reason, the proper name of these systems should be **GNU/Linux**, and you will often see them referred to by this name. They are fundamental because they provide the set of utilities that all other parts of the applications, like installation, rely on to perform their functions.

Everything else is optional, including the display of your desktop you see on your screen.

Packages

In almost all versions of GNU/Linux, applications come in the form of **packages**. One package per application.

However, a particular application may rely on other packages in order to operate correctly. If you are familiar with Windows, then this is akin to buying an application which comes with DLLs to be installed as well as the program you actually want to use. And if you are experienced, you may well remember the issues that come from “DLL hell” when two different pieces of software want different versions of the DLL.

In GNU/Linux, the packages all reside on a server for you to access to obtain what you want. When you install a package, it knows what else it needs to have available for it work.

Distributions, or "Distros"

When you get a GNU/Linux system, say Ubuntu, you are not only receiving a set of packages when you install the whole system, but you are also pointed at a server containing all the packages that were not on the CD, but can be obtained by downloading and installing. The complete set of packages, both on the CD and the server, have been put together by a distributor, and us usually called a **distro**. In the case of Ubuntu, this is organised by Canonical Ltd. Other distros are built by other companies (e.g. Red Hat Inc.) or individuals (e.g. PCLinuxOS). The distributor has done all the necessary work to ensure that compatible versions of all the packages are present. The packages will work because the other packages they depend on will be of the right version, and this has been checked as far as possible by the creator of the distribution.

Synaptic, or Package Manager

Each package contains a list of what it needs to work. The major distributions, like Ubuntu, contain an application, called a **package manager**, that looks at the requirements of the package you are asking for and selects all other packages that are needed. Some may already be installed, others it will add to the total list of what it has to download and install. This is why when you install one application, it may bring in automatically several more. The package manager will keep a record of what is installed and ensure that the right sets of packages are installed for each application.

On Ubuntu, the package manager is called **Synaptic**.

Updates

When the developers of a package fix a bug, or make a new version with few changes, they will issue a new updated package and it will be placed onto the distributor's server. Ubuntu checks regularly against the server to see whether there have been any updates, and will fetch and install them when you request it to do so.

But when a complete revision of the whole system and all the application packages happens, then it is called an upgrade to the next version. There is in this case a danger that the major changes cause an incompatibility and will require new versions of the applications that depend on them. The distributor will then issue either a new complete CD for re-installation, or will use a special process to upgrade all the necessary applications. This of course must be done very carefully, and it is not clear sometimes in what order it can be done without causing errors. For this reason, an upgrade will almost always require access to the servers as there is insufficient room on a CD for all the packages that might be involved. At best, an upgrade without a connection to the internet may need some applications to be upgraded separately afterwards.

Is there an equivalent on Windows?

No. On Windows, you will purchase each application separately from a different vendor and it is up to the provider of the application to tell you what else you need. Microsoft attempts to keep things clean by providing installation tools, but in practice it is nothing like as clean as the GNU/Linux way of doing things. To be fair, all the software in a distro is free and the builder of the distro can ensure that all the bits fit together, if necessary by re-building the applications from scratch; something that Microsoft cannot do because it does not have access to the individual applications.

Why are there so many distros?

Since anyone can begin to work on generating a distro, it is impossible to say how many there are at any given time. I've seen claims that there are 600 (www.consortiuminfo.org), over 500 (lwn.net/Distributions), 212 (www.linux.org), or numbers in between.

At distrowatch.com you can find a fairly comprehensive list of distros available at any one time, together with the number of downloads from some of the servers. The last time I checked, it was keeping statistics on 306 free operating systems, which includes some non-Linux based systems, so the number of GNU/Linux systems is a bit less than that; and not all of the GNU/Linux systems are packaged up into distros – there are some for the real techies who want to build their own from source code, like Slackware, but even they have done some work to co-ordinate the versions of applications.

By selecting different options when creating the kernel and applications, the resulting system can be tailored for different environments. Even for general desk use, some people want to do only e-mail, browsing and letter writing; whereas others want a system that can play music and organise photographs. These applications put different strains on the total system in terms of disk access and cpu usage required. The builder of a distro will have some idea of what they think their typical user wants, and will create the system appropriately. They will select their favourite desktop display and build the applications around it.

Some take a strict view of licensing and will only use packages that can be confirmed to be free everywhere in the world. But others will incorporate things which are less secure from a legal point of view in order to give a good impression when playing music and video files.

Also, Linux is the most flexible kernel ever written. It can be adapted for small mobile phones, or the largest supercomputers. It can be built so it will run on many different chips, from different manufacturers. Intel does not have a monopoly of computer chips; ARM chips are used extensively in phones, and other chips can be found in various parts of cars, broadband routers, televisions, and many other devices. GNU/Linux can be built for each of them, and is increasingly being used in them.

What are Partitions and File Systems?

by Andy Pepperdine.

File systems

On Linux, all the data you use on a computer resides in **files**. Files are collected into **directories**. Directories can also contain other directories to create a tree structure which is known as a **file system**. On Windows, documents are files, and folders are directories. A file system is represented by a drive letter (e.g. C:).

So far as the user is concerned, on Linux, *every* file is in a single tree structure and can be found by a path starting from the root of the structure designated by a slash. So the path name `/home/andy/list.odt` is a file `list.odt` in the directory `andy`, which in turn is in the directory `home`. The initial `/` indicates that the path can be traced from the root. This structure is known as the **user file system** because it is the view that the user sees of where all the files are.

It does not indicate where they in fact reside so far as hardware is concerned. A file might be on a hard disk, a USB memory stick, an external hard drive, a CD or DVD, a camera, etc. To the user it is somewhere in this tree, and one of the functions of the operating system is to keep this illusion and make it all look like part of the same whole.

On each physical device, though, the files are arranged for the convenience of the hardware, not for the convenience of the user. In practice, there is a piece of software called a **driver** which understands how the hardware is laid out and can interpret the data on it so it can be presented as though it is a tree structure. When a device is **mounted**, the driver is located and invoked to create this illusion, and the tree is slotted into the user file system at a suitable point, which varies according to the type of device, and the use to which the contents are to be put. Each device will contain a self-contained tree of directories and files, and so the whole will appear in the user file system under a suitable directory, like `/media/USB2`, for example, for a USB memory stick. On most modern Linux systems, plugging in a device will automatically mount it in a suitable place.

In general, a device might be able to support several different ways of laying out the data. These different ways are also often confusingly known as file systems, but really they are **file system types**. Linux can handle many such types, and the commonest have names ext2, ext3 and ext4. The type vfat is one Microsoft has invented and is the one usually found on external devices (including cameras and hard drives) when purchased.

Each of the file system types has its own characteristics and is more suited to particular situations. Some, for instance, keep a record of activity on the device which can be used to correct its state should it suddenly be turned off. These are often known as journaling systems, named after the journal they keep. The vfat type is useful if the device is to be read by many different systems, and so is favoured by small memory devices and appliances, like cameras. The vfat type is suitable for this purpose because it does not record the identity of the user who has permission to read and write files on it.

Partitions

The total available space on many devices can be divided into separate regions, which can then be treated as though they were separate devices in themselves. These regions are known as **partitions**.

If you want an analogy, imagine that you have a note book full of blank pages in which you will write a number of short stories. These will be edited over time, and so you will not know how long they are to be when you start a story. What you can do then is construct a contents list at the front containing details of the stories and where they are. All the pages are numbered to help you find them. If you lengthen a story, then you will have to put some sort of link from the end of it to where the extensions starts, as there might be another story occupying the next page after the existing text of a story. The notebook is like a partition. How the contents page and index, etc. are laid out are akin to the details of the type of file system that it contains.

Splitting a partition into two others is like dividing your book into two parts. But then you would have to re-arrange all the text pages, and make sure that the contents, indexes and cross-references are all updated properly. This can be difficult. If you want to split a partition that already contains data, then you **must** take a backup first in case the shuffling of the data is not done correctly, or the power fails while it is being done.

When to use different partitions

Some parts of the user file system might be better placed on their own file system types in order to make use of the characteristics of that type. Sensitive data should be given into the care of a journaling file system because it is safer to do so in the event of a power failure, for example.

Other files might need faster access, but the data rarely changes and can be re-constructed easily, and then a simpler type of file system would be more appropriate.

In practice, for the normal user, the more important features would be the ease of taking a backup of the data, and whether it would need to be saved and restored if the system is upgraded to the next version.

Swap area

A special sort of disk area is used for the **swap area**. This space is not organised in a file system as it contains no files. Instead it contains space for the operating system to pretend to be an extension of its internal memory. In this way, if all the programs running need more space than is in physical memory, then it can use this disk area to “swap” pieces of internal memory onto disk and so allow the physical memory to be used only when the relevant part of memory is required.

The swap area contains no data that persists between starts of the system, and can usefully be kept in its own partition. Some distros require it be on its own.

Users’ data areas

Users will normally be interested in keeping their own data cleanly separated from the system in order that the system can be upgraded easily without impacting the data. One way to do this is to put the users' data in its own partition, providing clear access to the system partitions for re-installing the system.

System area

If the operating system, and all its associated data, is placed in its own partition, then the system can be completely re-installed without affecting any of the data that belongs to the users. In this way, if an upgrade fails, you can instead install afresh.

Alternatively, if you are looking to switch distros and try another, then it can be installed into the system partition over the old one without affecting the users' data.

Encryption

Partitions can be encrypted as a whole, so that any data written to them can be hidden. If you are concerned to keep your data safe from prying eyes or theft (e.g. on an easily portable device), then separating the data into stuff to hide and stuff that is public may be what you want. You might, for instance, wish to keep your backups encrypted if you think they might be compromised.

There are descriptions of how to encrypt a partition at
http://www.linuxconfig.org/Partition_Encryption

and at http://gentoo-blog.de/ubuntu/encrypted-home-and-swap-partition-on-ubuntu-9-10-karmic/

[If any one does this, it might be worth a paper to keep for others to look at.]

Other considerations

If speed is your thing, then partitions do not gain much, if anything. A hard disk is slowest when the reading heads have to travel long distances. To reduce this it is best to use more than one head for each part of the file system, and put the swap, system and user data on separate devices. Do not share a device in this case.

There is little to be gained from any other splitting of the user file system for most users of personal computers.