How to Find Things on Linux
by Andy Pepperdine

This month we will look at various applications and methods we can use to discover things about the system that you might want to know some time. I will be referring in these notes only to systems that use APT to manage the Linux packages. Ubuntu, LinuxMint, Peppermint and many others use this method. If you have some other variant of Linux, then most of these ideas will still transfer directly. If not there will be some similar method that you can find by searching with Google.

There are an infinite number of questions that you might want an answer to, and the following are a fairly arbitrary selection, in no particular order.

Because most of the methods described here start from a command line, I'll just say a couple of quick words first about commands in general.

How do I write terminal command?

Commands often have a lot of different options to control what they do and how they do it. You may wonder how to find out what options and parameters a command has.

The traditional answer to this is yet another command:

```
    man [command]
```

where the [] indicate to replace the word “command” with the name of the command you want to know about. But it is not exactly the most readable manner of investigating a topic. There is a website [4] that contains all the commands, although it is not easy to do searches unless you know what you are doing. However, a Google search will usually take you directly to the right page on that site.

If you are going to do this sort of thing, it is worthwhile becoming familiar with the style of writing, and the way a command is described. Almost always, a command is written as a one-word *command* name, followed by one of more *options*, followed in turn by one or more *parameters*.

With very few exceptions, options are indicated by their first character being a minus sign and modify the type or method of operation.

Parameters are the names of things, like the names of files, which constitute the data being operated on by the command.

Is that installed?

If you see in some forum or webpage that a particular application might be what you are looking for, then the first question is whether it is already installed.

Applications are shipped in *packages*, and these are managed automatically by a Package Manager. You can normally find such an entry in the menu of applications. For our purposes, the package manager is also known as *synaptic*.

When synaptic starts up it will present you with a list of all packages, possibly modified by the default selection shown in the left hand pane. If you select Status and Installed on the left side, the list of packages on the right will all be preceded by a green square, indicating that the named package is installed. Now by putting into the Quick filter box the name of the package you are looking for, it will restrict the names to display to those containing that string of characters. You will then see whether or not you have the package installed.

What package am I looking for?

Very often the name of the application is the same as the package. Occasionally it is not. In that case you will have to identify the package to search for. But before you do that, you can check whether the application is present by another route. It will mean using the command line.

In LinuxMint, open up a terminal window by Menu → Terminal. [For other systems, look for something akin to the word “terminal” under Accessories, System Tools, etc.]

Then, for example to see whether you the gimp application, issue the command:

```
which gimp
```

This will either produce a line showing where to find the program, or will produce no output if there is no program by that name that you can execute.

If this fails, and you do not have the program already installed, then it is time to use more serious tools. One of the most useful packages in cases like this is *apt-file*, which will have to be installed first, and you will also have to download a large file of data it will use in the search. (See reference [1]).

An alternative is to use a website that can do the search for you, as in [2]. Although that is Ubuntu, the differences between that and the other systems I've mentioned are minimal for this purpose.

In fact, the same process can search for the packages containing any file, not just applications.

Where did that space go?

You might be running out of space somewhere, or think the backup is taking an inordinate amount of time. So how can you how find where all the files are, and how big they are? The answer in this case is the command *du* [3] which has a number of useful options. For instance, suppose you have a directory *Pictures*, and you want to find which part of that structure is the largest. Perhaps you want to move part of it out to some other device. So do the following

```
du -cs Pictures/*
```

and you will see a list of directories, each preceded by the size in numbers of “blocks”. If you want it in human readable form, try:

```
du -csh Pictures/*
```

Or maybe you want to see a sorted list:

```
du -csh Pictures/* | sort -n
```

For those who prefer a diagrammatic representation of file usage, then Ubuntu-based systems come with a Disk Usage analyser. For Linux Mint, it is found under System Tools in the Menu and is known as Baobab. When you have started it up, the Scan Home button will scan the whole of the home directory, and could take some time depending on the speed of your disk drive. You can also get it to scan any individual folder and its subfolders.

For those who understand Unix file systems, I noticed that it treats encrypted directories as both the encrypted and unencrypted forms, so that they are counted twice.

How much space is left?

A quick way of determining how full partitions are is to use the simple command *df*. This will produce some output that will show all used partitions and other parts of the filesystem, and the amount of data they occupy and what is free to use. However, it will only show what space is being used in the partitions that are currently accessible. They may be other space on a disk that is not in use, but which has not been made accessible.

What parts are accessible?

To see which partitions are in use currently, the *mount* command will show you which parts of all the devices are being made available. Use it without any options to get the list of everything.

```
mount
```

In the output the lines starting with */dev* represent the individual devices and partitions that contain your data. This will help to interpret the lines in the output from the *mount* command.

Also in the output are lines for other parts of the file system, such as remote directories, encrypted directories and special system directories. If there is interest, these can be examined in more detail.

What hardware do I have?

This can be thought of in two separate parts. First, the processor that does all the work of executing the applications. Most of the details about that can be found in a file in the */proc* directory. If you navigate to */proc/cpuinfo* and open it in a text editor, you can see information about each cpu core, it's current execution speed and other data. If you see any notice about the file being changed on disk, ignore it.

Information about the amount of RAM you have, and what it is used for is found in the file */proc/meminfo* so you can see whether there is a need to expand it or not.

External devices

The second, and most extensive, part is the other physical devices attached to the machine, such as wireless connections, disks, etc. A start to identify things is the command *lshw*. You may need to install the package by that name to get it. This must be run under *sudo* and will provide an initial list of what is available, but if you want to know details connected to what driver you need for it, then only the manufacturer's specification will do, and can be got from their websites. You will need the exact model number and type, often provided definitively on a label on the device.

The *lshw* command produces voluminous output, and is best saved and scanned with a text editor, for example by:

```
sudo lshw > file.txt
```

which will put the results in the file name *file.txt*.

However, in some cases (depending on how you have set up your system), it can be difficult to save the output from a *sudo* command. For such a case, here is one way to do so:

```
sudo lshw | cat - > file.txt
```

What is the machine doing?

If you want to see what processes are using time on the cpu at any time, a useful command is *top*. This will display some information about the use of the cpu and memory. Then there is a list of running processes, each with some data on how much space and cputime it is using.

Important: the display typically will update frequently. To stop the program, hit the 'q' key.

There are many command line options available, and the display can be tailored to what you want to watch, all of which is given in the man pages.

Sometimes, you want to know all of the running processes, not just the busy ones. The way to do that is with the ps command, like this:

```
ps auwx
```

This can be useful if some program is looping, or stuck in some way and you want to get rid of just that one. It will also show you just how many processes are started up to run the system. Most of these are waiting for something to happen, which may or may not occur – like plugging in a memory stick.

Why is it slow to start up?

There is a way to see what is going on during the boot process, if that matters to you. The relevant package is called *bootchart*. Install that, then when you next boot, it will generate data and a chart for you to examine at your leisure. You will find the chart in */var/log/bootchart*.

Since this will happen on every boot, you may want to disable it after you've done your analysis. The reference on how is at [5].

The best way to speed things up is to disable any processes you do not need. However, it is not at all obvious what is needed and what is not, and it is NOT a good idea to remove something and try it as you may get a system that will no longer boot at all. If you do not know what something does, try Google to see if it is relevant, and make sure you understand it. Only then should you try to suppress the process from starting.

Some of the services are listed by the *Services* menu item in Ubuntu based systems, but not all of them. Whether they show up is defined in the directory */etc/xdg/autostart*. There you can see a file for each service. Some of them will contain the line

```
NoDisplay=true
```

and those will *not* be shown in the Services list. Change *true* to *false* and they should appear.

References

[1] Finding which package to install: http://www.debianhelp.co.uk/findfile.htm

[2] Remote site to search packages: http://packages.ubuntu.com/

[3] The *du* command is described here: http://linux.die.net/man/1/du

[4] Command descriptions: http://linux.die.net/man/

[5] Disabling *bootchart*: http://askubuntu.com/questions/264935/how-do-i-disable-bootchart