

Comments on Use of Disk Space

by Andy Pepperdine

Introduction

Linux is a type of Unix. Central to the Unix design is the idea that “everything is a file”. What that means is that all significant data areas associated with the operation of the system can be found somewhere in the “Unix file system”, by which I mean that everything can be found from one central place, the so-called *root* directory /.

But it is also important to be clear that a *file* is just a, possibly virtual, area containing the data. To access that data, you have to follow a *path*. So the text file on a Linux Mint system found at the end of the path: /etc/issue.net is a file containing the name of the Mint release.

In practice, the terms path and file are often used interchangeably, but the concepts are clearly different.

In particular, note that in Linux, it is possible for a given file to be reached via more than one path. This can be arranged, for example, by setting up symbolic or hard links, or by mounting virtual file systems.

Disk Space

The purpose of this note is to describe how you can determine how much space is being used, and where it is. In addition, some hints will be given as to how to find what may be excessive use of space, and places to keep an eye on, should you think that you are running short of space.

Since all files are found via some path, which typically is a list of directories and subdirectories, then what is required is to know how much space is used by a particular directory, and all its subdirectories. But in practice it is often not quite that simple.

In particular, due to the way that data is distributed, you have to consider that data is also held in real disk partitions, or areas of fixed size and usually determined when a system is set up in the first place. Partitions could also be added as external USB devices. If you run out of space, then it means that some partition has reached its limit.

All the known partitions are linked in some way to the root directory through the subdirectory structures of the Unix file system. Almost all of these will be inserted into the structure when the system is booted.

Pictorial tools

The principal tool to use if you want a GUI is the Disk Usage Analyser (found under System Tools). When it starts, it merely shows the summary of the total file system, which generally is not much use.

The dropdown at the top right can change the view it presents between a circular chart and a rectangular tree.

Scan Home

Clicking on this icon will scan your home directory and report on the space used by each subdirectory in your home directory. In this way you can see whether your cache or thumbnail directories are getting too large.

Scan File System

This will analyse the whole file system, and can take a long time to complete, depending on the number of files you have and where they are located.

Scan a Folder

This icon is perhaps the most useful. It will present a browser selection window and you can select the directory you want to analyse in more detail.

Command Line Tools

In certain circumstances, if you really have run out space, then it may not be possible to run a GUI tool. In this case, you will have to resort to the command line and a terminal.

Disk space usage (*df*)

The starting point will be the command *df*. This will immediately show whether any of the separate partitions you have are straining for space. Use the line:

```
df -h
```

to show you numbers in an easy to see form. In particular, one column is the percentage of space used. A large number there is where you can start your investigation.

Disk Usage (*du*)

This command can give more detailed use of particular directories, and is often used in the form

```
du -sh /home/you
```

which will summarise the space used under that directory. Then you can burrow down further when you know where the trouble is.

The form

```
du -mhc /home/you
```

gives much more detail, and could obscure where the space actually is.

Where to look

Data is stored in partitions that have a fixed upper limit for their size. When you delete a file, in most cases, it will be moved over to an area known as *Trash*, or some similar term, like *Wastebasket*. Each partition will have its own trash area, if it supports it. So when you see a full partition, the first stop should be to check whether the trash area is large, and if so delete it, hoping that your backups are good enough to recover any files you recently deleted in error.

It is certainly good practice to often check your trash areas to make sure they are not getting uncomfortably large.

There are some other directories that are always worth keeping an eye on if you think you are running short of space. These are:

```
/home/.cache    [especially thumbnails]
/home/.local
/tmp
/var
```

How you handle these will depend on what is actually there.

Note that in `/var` several directories and files are not accessible by normal users, so you may have to investigate using the “*sudo*” command.

Cleaning up

Generally speaking, data in the cache and thumbnail directories can be removed. If it is needed, it will be created anew on demand.

Data in `/tmp` should be cleaned out every time the system is rebooted, and will contain files that are useful to programs currently running. You should see that they all have today’s date on them, if you booted up today.

Data in `/var/tmp` are temporary files that some tools use to keep information from one boot to the next. If you have closed down ALL your applications, then it ought to be possible to delete all files in this directory that you own. Do not delete anything owned by *root*.

Printing systems in Linux seem not to clean up the spool files very well, at least not on my system, nor on one of the members of the group. Deleting the old spool files is not a trivial matter, since they are not owned by a normal user, but instead by *root*, and accessible by the pseudo-user *lp*. In a terminal, first check what files are in the directory `/var/spool/cups/`, like this:

```
sudo ls -l /var/spool/cups/
```

Now check that you can find all the files that you wish to delete:

```
sudo find /var -path 'cups/c0*'
```

If that shows the right list of files, then do this:

```
sudo find /var -path 'cups/c0*' | xargs sudo rm
```

References

You may get more ideas from: <https://www.makeuseof.com/tag/how-to-analyze-your-disk-usage-pattern-in-linux>

More information on the command line: <https://www.tecmint.com/check-linux-disk-usage-of-files-and-directories/>