

du – show disk usage stats

du command is one of the most popular [Unix commands](#). It is used to quickly estimate and report disk usage by a certain directory tree, all figures are reported in n blocks of data consumed by each object. While most commonly used block size is 1024b these days, you can easily override this if you have to. It's flexible enough to report sizes of only specified directories, full directory tree or even size of each file. There is no faster or easier way to find out what's consuming most space on a certain filesystem – you usually run du for a high-level overview, and, once the largest directories are identified, use [find command](#) to highlight the largest files within them.

Basic du command usage

The simplest form of using **du** is when you run it with no additional parameters. In this mode, du will scan your current directory and all the subdirectories of it to calculate usage stats. By default, you get the usage reported by all the directories found under the specified (or default) location.

Important: be careful with default **du** behavior, because depending on your Unix OS, it may use different block size for reporting the size. Older systems would use 512-byte blocks, while most of the recent releases use 1024-byte ones. To avoid any confusion, get used to specifying the desired block size with **-k** parameter (see below).

Default **du** behaviour. I'm doing [pwd](#) before it just so that you know which directory it's going to scan:

```
root@ubuntu# pwd
/root
root@ubuntu# du
4      ./aptitude
```

```
16      ./mc
8       ./ssh
8       ./update-manager
4       ./Mail
92      .
```

Specifying block size for du reports

Like it was mentioned before, **du** calculates all the sizes in blocks of data. All the reports you see are not bytes or kilobytes, but the block sizes – so it's good to always remember this. In most cases these days, the default block size is 1024 bytes, so you are effectively looking at all the sizes reported in kilobytes, but to make sure it's always the case I advise you to use **-k** parameter which overrides you Unix flavour default block size and sets it to 1024 bytes:

```
root@ubuntu# du -k
4       ./aptitude
16      ./mc
8       ./ssh
8       ./update-manager
4       ./Mail
92      .
```

If you wanted to have all your sizes in bytes, then there's a **-b** parameter just for you:

```
root@ubuntu# du -b
4096    ./aptitude
7222    ./mc
5790    ./ssh
6293    ./update-manager
4096    ./Mail
58142   .
```

Finally, most common releases of Linux have **-h** option for you, which reports all the figures in an even easier form, specifying **K** for kilobytes, **M** for megabytes and so on:

```
root@ubuntu# du -h
```

```
4.0K   ./aptitude
16K    ./mc
8.0K   ./ssh
8.0K   ./update-manager
4.0K   ./Mail
92K    .
```

Report sizes of both files and directories

Using `-a` parameter, it is possible to have `du` report sizes of not only the directories, but all the files as well.

Compare this (directory sizes):

```
root@ubuntu# du -h
4.0K   ./aptitude
16K    ./mc
8.0K   ./ssh
8.0K   ./update-manager
4.0K   ./Mail
92K    .
```

against this (files and directories):

```
root@ubuntu# du -ah
0      ./aptitude/config
4.0K   ./aptitude
4.0K   ./mc/history
4.0K   ./mc/Tree
4.0K   ./mc/ini
16K    ./mc
4.0K   ./ssh/known_hosts
8.0K   ./ssh
4.0K   ./update-manager/meta-release
8.0K   ./update-manager
4.0K   ./Mail
16K    ./viminfo
4.0K   ./rnd
12K    ./bash_history
4.0K   ./bashrc
4.0K   ./lessht
```

```
4.0K    ./mysql_history
4.0K    ./profile
92K     .
```

Summarizing du reports

The higher the level of disk usage you're after, the larger lists will be produced by `du`. To cope with this, `du` has a `-s` option, which summarizes all the reports down to only the level of directories you specify. To be exact, it only reports size summaries of the directories you specify in command line. If you use this option but specify no directory, it will produce only one line of output, summarizing the size of the current directory:

```
root@ubuntu# du -sk
92     .
```

And here's how one would specify a few directories to summarize upon:

```
root@ubuntu# du -sk .ssh .mc
8      .ssh
16     .mc
```

But like I said, the real power of this option can only be appreciated when dealing with many more directories. Take `/usr` for example, and compare...

... basic `du` approach, which shows size of each directory and subdirectory under `/usr` (I have stripped off the output):

```
root@ubuntu# du -h /usr/*
4.0K    /usr/X11R6
96M     /usr/bin
24K     /usr/games
4.0K    /usr/include/X11
64K     /usr/include/arpa
52K     /usr/include/asm-generic
180K    /usr/include/asm-i386
232K    /usr/include/asm-x86_64
```

```
192K    /usr/include/asm
636K    /usr/include/bits
8.0K    /usr/include/et
20K     /usr/include/gnu
48K     /usr/include/gssapi
116K    /usr/include/gssrpc
...
```

against the summary **du -sh** output, which matches **/usr/*** mask and essentially shows the summary of only the first level of **/usr** subdirectories:

```
root@ubuntu# du -sh /usr/*
4.0K    /usr/X11R6
96M     /usr/bin
24K     /usr/games
12M     /usr/include
164M    /usr/lib
0       /usr/lib64
96K     /usr/local
25M     /usr/sbin
197M    /usr/share
4.0K    /usr/src
```

Do you see the difference? Summary like this allows me to instantly estimate which of the **/usr** subdirectories take up most space.

See Also

- [Basic Unix commands](#)
- [Unix Commands](#)