The [du](http://linux.about.com/library/cmd/blcmdl1_du.htm) command shows the disk space used by the files and directories in a directory. The -h option makes the output easier to read, and the -s option summarizes the result. For example,

du -h -s

will report the combined disk space used by all subdirectories of the current directory. In order to see the disk space usage of all subdirectories as well you would use:

du -h

Du stands for Disk Usage.

Linux du command is used for summarizing the disk usage in terms of file size. It can be used with folders to get the total disk usage. This article provides some examples on how to use du command effectively.  
  
All the du examples shown here are executed on a directory containing the following contents:

$ ls

linuxKernel redhat testfile.txt ubuntu

**1. A basic example**

$ du -a

0 ./redhat/rh7

4 ./redhat

4 ./testfile.txt

0 ./linuxKernel

0 ./ubuntu/ub10

4 ./ubuntu

16 .

I have used the -a flag in the example above to show the disk usage of all the files and directories. Its because if -a is not used then only directories that are occupying some disk are listed. For example :

$ du

4 ./redhat

4 ./ubuntu

16 .

So, now we get a basic idea about how to use du command but as with me, anyone would find it hard to understand what those numbers in the output mean??  
Lets move on to next examples and the clouds will clear off.

**2. Display output in human readable form using -h**

$ du -ah

0 ./redhat/rh7

4.0K ./redhat

4.0K ./testfile.txt

0 ./linuxKernel

0 ./ubuntu/ub10

4.0K ./ubuntu

16K .

So we see that in the above example, I used the -h flag along with the -a flag. The -h flag is used to get the output in the human readable format. As you can see that above output is more easy to understand as disk usage is listed in terms of ‘K’.

**3. Display grand total in the output using -c**

The example that I am using has a small directory structure. One could easily calculate the total disk usage of the directory by calculating manually. But, in real time scenario manual calculation is not practical. So, there exists a flag through which one can get the total usage in the output.

$ du -ahc

0 ./redhat/rh7

4.0K ./redhat

4.0K ./testfile.txt

0 ./linuxKernel

0 ./ubuntu/ub10

4.0K ./ubuntu

16K .

16K total

So we see that through the -c flag, one can get the total usage in the output.

**4. Display only the total count using -s**

If you wish to fetch only the total size in form of a summary, use the -s flag.

$ du -sh

7.3G .

So we see that only the total size was displayed in the output above.

**5. Feed input files from stdin using ‘–files0-from=-’**

If there arises a situation wherein you want to provide ‘du’ input on the go, then this is possible through the ‘–files0-from’ flag.

$ du -ah --files0-from=-

copy12K copy

file12K file

helloWorld12K helloWorld

^C

$

Note that we passed value as ‘-’ to the flag ‘–files0-from’ in order to indicate ‘du’ to start accepting the file names from stdin. After entering each file name press ctrl+D twice in order to produce the output for that file.

**6. End the output with null byte using -0**

If you wish to produce each output ending without a newline but with a null byte instead then use the -0 flag.

$ du -am -0

0 ./redhat/rh71 ./redhat1 ./testfile.txt0 ./linuxKernel0 ./ubuntu/ub101 ./ubuntu1

So we see that the output above was produced in a single line as each output was terminated with a 0 byte.

**7. Customize the block size in output through –block-size**

Talking of block size, some times it is required to have output in a different way. For example :

$ du -ac

0 ./redhat/rh7

4 ./redhat

4 ./testfile.txt

0 ./linuxKernel

0 ./ubuntu/ub10

4 ./ubuntu

16 .

16 total

The above output is represented in terms of number of 1024 bytes blocks. Now suppose if we require the output to be in number of 2048 bytes block, then in this case the flag ‘–block-size’ can be used.

$ du -ahc --block-size=2048

0 ./redhat/rh7

2 ./redhat

2 ./testfile.txt

0 ./linuxKernel

0 ./ubuntu/ub10

2 ./ubuntu

8 .

8 total

So we see that the above output is in the form of number of 2048 bytes blocks.

**8. Display output in bytes using -b**

To get the output in terms of bytes, the -b flag can be used.

$ du -achb

0 ./redhat/rh7

4096 ./redhat

3 ./testfile.txt

0 ./linuxKernel

0 ./ubuntu/ub10

4096 ./ubuntu

12291 .

12291 total

So we see that the above output is in terms of bytes.

**9. Exclude particular types of file(s) using –exclude**

Suppose we do not want to have the statistics of .txt files. So for these types of requirements there is a ‘–exclude’ flag.

$ du -cbha --exclude="\*.txt"

0 ./redhat/rh7

4.0K ./redhat

0 ./linuxKernel

0 ./ubuntu/ub10

4.0K ./ubuntu

12K .

12K total

So we see that in the output above, the text file entry was not there.

**10. Display the modification time and customize the display style**

This can be achieved through –time and –time-style flags.

$ du -cbha --time

0 2012-05-22 21:52 ./redhat/rh7

4.0K 2012-05-22 21:52 ./redhat

3 2012-06-18 19:23 ./testfile.txt

0 2012-05-22 21:52 ./linuxKernel

0 2012-05-22 21:52 ./ubuntu/ub10

4.0K 2012-05-22 21:52 ./ubuntu

13K 2012-06-18 19:23 .

13K 2012-06-18 19:23 total

$ du -cbha --time --time-style=iso

0 2012-05-22 ./redhat/rh7

4.0K 2012-05-22 ./redhat

3 2012-06-18 ./testfile.txt

0 2012-05-22 ./linuxKernel

0 2012-05-22 ./ubuntu/ub10

4.0K 2012-05-22 ./ubuntu

13K 2012-06-18 .

13K 2012-06-18 total

For –time-style, you can also use full-iso, long-iso, iso.