

## Linux sort command - Sort lines of text files

[Himanshuz.chd](#) | Dec 7 2012 | [Visits](#)

(156960)

While going through an article on Linux text processing commands, I came across Linux sort command. I found this command interesting enough to read more about it and try out some of its examples. So here in this article, I'll share my understanding on Linux sort command through some examples. Before jumping on to examples, here is an excerpt of information from the [man page](#) of sort command:



### NAME

sort - sort lines of text files

### SYNOPSIS

sort [OPTION]... [FILE]... sort  
[OPTION]... --files0-from=F

### DESCRIPTION

Write sorted concatenation of all FILE(s) to standard output.

So we see that the main purpose of this command is to produce a sorted output.

## Linux sort command examples

### 1. A basic example

The very first input that I tried consisted of some random alphabets. Here is what I tried:

```
$ sort
b
z
a
w
s
```

And here is the output :

```
a
b
s
w
z
```

So we see that the output produced was in sorted form.

### 2. Sort numbers

In the following example, I filled a text file (sort.txt) with some random numbers.

```
$ cat sort.txt
8
2
6
```

```
1
5
3
```

Then I used the sort command with sort.txt as input file to the command.

```
$ sort sort.txt
1
2
3
5
6
8
```

So we see that sorted list of numbers was produced in output.

### 3. Sorting words

In this example, the sort.txt file is filled with some words.

```
$ cat sort.txt
UK
Australia
Newzealand
Brazil
America
```

Now, this file is given as input to the sort command:

```
$ sort sort.txt
America
Australia
Brazil
Newzealand
UK
```

So we see that words were sorted according to dictionary ordering. Even the words beginning with same alphabet were sorted according to succeeding alphabets.

### 4. Use sort to directly write data in sorted manner

This command can be used to write unsorted input data to a file directly in sorted manner. Here is how this can be done :

```
$ sort > sort.txt
9
Hello
4
Why
8
Bye
```

After the above operation, let's check the file contents :

```
$ cat sort.txt
4
8
9
Bye
```

```
Hello  
Why
```

So the output suggests that the input was first sorted and then written to file.

## 5. Write sorted concatenation of all input files to standard output

If more than one file is provided as input, the sort command produces a sorted concatenation on stdout.

Here is an example:

```
$ cat sort1.txt  
7  
4  
9  
1  
  
$ cat sort2.txt  
8  
5  
6  
2
```

Here is the output :

```
$ sort sort1.txt sort2.txt  
1  
2  
4  
5  
6  
7  
8  
9
```

So we see that a sorted concatenation was produced in output.

## 6. Write result of sort in a file

The output of sort command can be written to a file by using -o option.

Here is how it's done :

```
$ sort -o sort.txt  
4  
9  
2  
8  
1
```

Now let's check the file :

```
$ cat sort.txt  
1  
2  
4  
8  
9
```

So we see that the output was actually written to the file whose name was supplied as input to sort through -o option.

## 7. Sort months

There is an interesting option -M through which the month names can be sorted.

Here is an example :

```
$ sort -M > sort.txt
DEC
JAN
FEB
```

Now, let's check the file contents for output :

```
$ cat sort.txt
JAN
FEB
DEC
```

So we see that sort command actually sorted the month names.

## 8. Sort human readable numbers

Another interesting option -h is provided by sort command through which human readable numbers.

Here is an example :

```
$ sort -h > sort.txt
2G
1K
3M
```

Now, let's check the file for output:

```
$ cat sort.txt
1K
3M
2G
```

So we see that the numbers were sorted.

## 9. Produce reverse sorted results

Using -r option provided by sort command, the results can be produced in reverse order.

```
$ sort -h -r > sort.txt
2G
1K
3M
```

Here is the output of file :

```
$ cat sort.txt
2G
3M
1K
```

So we see that this time the sorting results were written in reverse sorted order.

## 10. Compare according to string numerical value

This can be done using -N option.

Here is the input :

```
$ cat > sort.txt  
7 mangoes  
4 oranges  
9 grapes  
1 apple
```

Here is the output :

```
$ sort -n sort.txt  
1 apple  
4 oranges  
7 mangoes  
9 grapes
```