

How to compare two version numbers in a shell script

Author : Dan Nanni

Categories : [Shell](#), [Utilities](#)

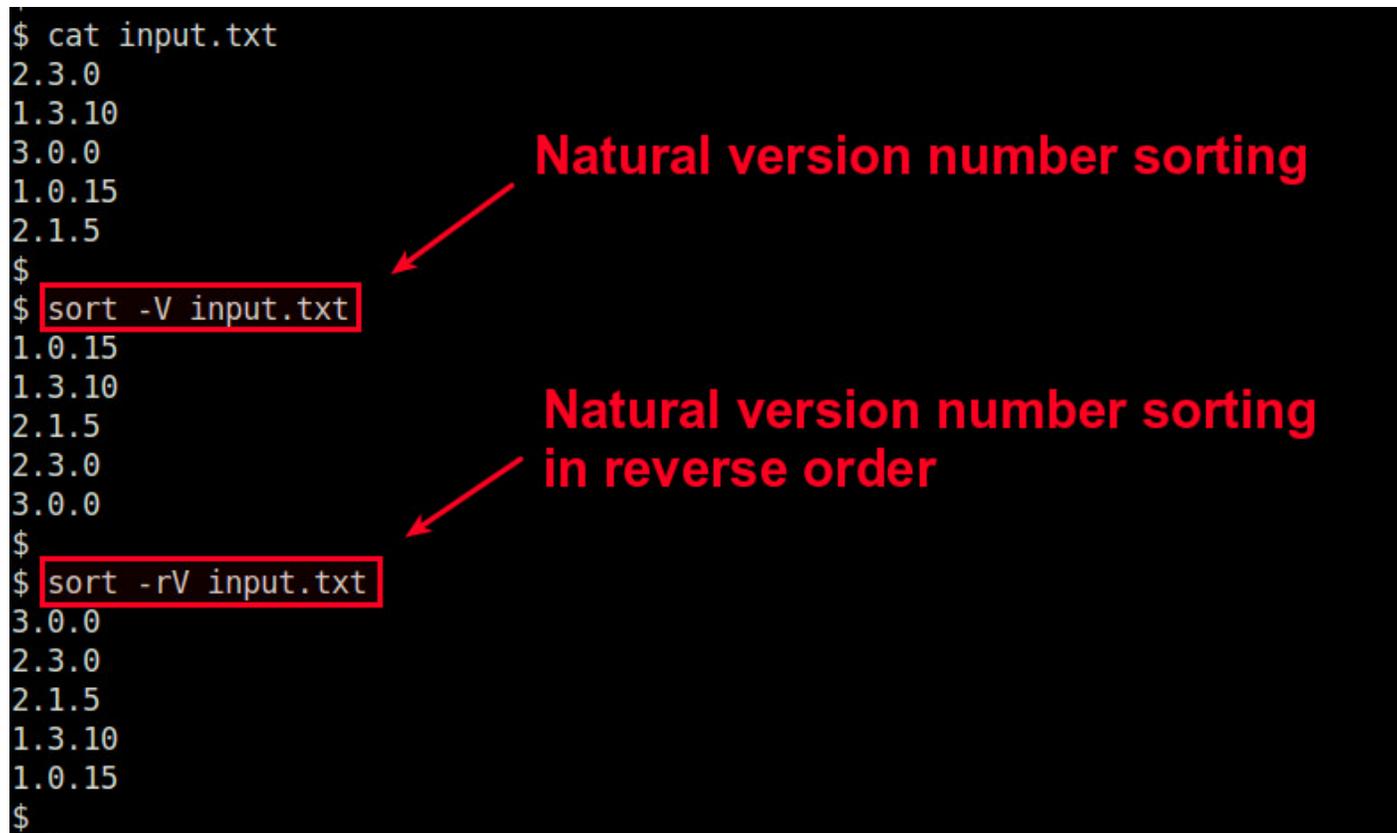
Tagged as : [cliversion](#)

Question: I am writing a shell script in which I need to compare two version number strings (e.g., "1.2.30" and "1.3.0") to determine which version is higher or lower than the other. Is there a way to compare two version number strings in a shell script?

When you are writing a shell script, there are cases where you need to compare two version numbers, and proceed differently depending on whether one version number is higher/lower than the other. For example, you want to check for the minimum version requirement (i.e., `$version ? 1.3.0`). Or you want to write a conditional statement where the condition is defined by a specific range of version numbers (e.g., `1.0.0 ? $version ? 2.3.1`).

If you want to compare two strings in version format (i.e., "X.Y.Z") in a shell script, one easy way is to use `sort` command. With "-V" option, the `sort` command can sort version numbers within text (in an increasing order by default). With "-rV" option, it can sort version numbers in a decreasing order.

```
$ cat input.txt
2.3.0
1.3.10
3.0.0
1.0.15
2.1.5
$
$ sort -V input.txt
1.0.15
1.3.10
2.1.5
2.3.0
3.0.0
$
$ sort -rV input.txt
3.0.0
2.3.0
2.1.5
1.3.10
1.0.15
$
```

The image shows a terminal window with a black background and white text. It displays the output of two shell commands. The first command is `cat input.txt`, which outputs a list of version numbers: 2.3.0, 1.3.10, 3.0.0, 1.0.15, and 2.1.5. The second command is `sort -V input.txt`, which outputs the same version numbers sorted in ascending order: 1.0.15, 1.3.10, 2.1.5, 2.3.0, and 3.0.0. The third command is `sort -rV input.txt`, which outputs the version numbers sorted in descending order: 3.0.0, 2.3.0, 2.1.5, 1.3.10, and 1.0.15. Two red arrows point from red text annotations to the respective `sort` command lines. The first annotation, "Natural version number sorting", points to the `sort -V` command. The second annotation, "Natural version number sorting in reverse order", points to the `sort -rV` command.

Ask Xmodulo

Find answers to commonly asked Linux questions

<http://ask.xmodulo.com>

Now let's see how we can use the `sort` command to compare version numbers in a shell script.

For version number string comparison, the following function definitions come in handy. Note that these functions use the `sort` command.

These functions perform, respectively, "greater-than", "less than or equal to", "less than", and "greater than or equal to" operations against two specified version numbers. You will need to use `bash` shell due to function definitions.

Below is an example bash script that compares two version numbers.