



Linux Ubuntu 26.04 Manual Pages on command 'strverscmp.3'

\$ man strverscmp.3

strverscmp(3) Library Functions Manual strverscmp(3)

NAME

strverscmp - compare two version strings

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

```
#define _GNU_SOURCE      /* See feature_test_macros(7) */
#include <string.h>

int strverscmp(const char *s1, const char *s2);
```

DESCRIPTION

For a dataset like jan1, jan2, ..., jan9, jan10, ... sorting it lexicographically yields jan1, jan10, ..., jan2, ..., jan9. The task of strverscmp() is to compare two strings yielding the former order, while strcmp(3) finds only the lexicographic order. This function does not use the locale category LC_COLLATE, so is meant mostly for situations where the strings are expected to be in ASCII. This is different from the ordering produced by sort(1) -V.

What this function does is the following. If both strings are equal, return 0. Otherwise, find the position between two bytes with the property that before it both strings are equal, while directly after it there is a difference. Find the largest consecutive digit strings containing (or starting at, or ending at) this position. If one or both of these is empty, then return what strcmp(3) would have returned (numerical ordering of byte values). Otherwise, compare both digit strings numerically, where digit strings with one or more leading zeros are interpreted as if they have a decimal point in front (so that in particular digit

RETURN VALUE

ATTRIBUTES

[illegible][illegible][illegible]

STANDARDS

GNU.

EXAMPLES

```
$ ./a.out jan1 jan10;
```

```
jan1 < jan10
```

Program source

Page 2/3

```
res = strverscmp(argv[1], argv[2]);  
printf("%s %s %s\n", argv[1],  
      (res < 0) ? "<" : (res == 0) ? "==" : ">", argv[2]);  
exit(EXIT_SUCCESS);  
}
```

SEE ALSO

rename(1), strcasecmp(3), strcmp(3), strcoll(3)

Linux man-pages 6.17

2026-02-08

strverscmp(3)