



*Full credit is given to the above companies including the OS that this PDF file was generated!*

## **Red Hat Enterprise Linux Release 9.2 Manual Pages on 'strcmp.3p' command**

**\$ man strcmp.3p**

STRCMP(3P)                    POSIX Programmer's Manual                    STRCMP(3P)

### PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

### NAME

strcmp ? compare two strings

### SYNOPSIS

```
#include <string.h>

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

### DESCRIPTION

The functionality described on this reference page is aligned with the ISO C standard. Any conflict between the requirements described here and the ISO C standard is unintentional. This volume of POSIX.1?2017 defers to the ISO C standard.

The strcmp() function shall compare the string pointed to by s1 to the string pointed to by s2.

The sign of a non-zero return value shall be determined by the sign of the difference between the values of the first pair of bytes (both interpreted as type unsigned char) that differ in the strings being compared.

### RETURN VALUE

Upon completion, `strcmp()` shall return an integer greater than, equal to, or less than 0, if the string pointed to by `s1` is greater than, equal to, or less than the string pointed to by `s2`, respectively.

## ERRORS

No errors are defined.

The following sections are informative.

## EXAMPLES

### Checking a Password Entry

The following example compares the information read from standard input to the value of the name of the user entry. If the `strcmp()` function returns 0 (indicating a match), a further check will be made to see if the user entered the proper old password. The `crypt()` function shall encrypt the old password entered by the user, using the value of the encrypted password in the `passwd` structure as the salt. If this value matches the value of the encrypted `passwd` in the structure, the entered password `oldpasswd` is the correct user's password. Finally, the program encrypts the new password so that it can store the information in the `passwd` structure.

```
#include <string.h>
#include <unistd.h>
#include <stdio.h>
...
int valid_change;
struct passwd *p;
char user[100];
char oldpasswd[100];
char newpasswd[100];
char savepasswd[100];
...
if (strcmp(p->pw_name, user) == 0) {
    if (strcmp(p->pw_passwd, crypt(oldpasswd, p->pw_passwd)) == 0) {
        strcpy(savepasswd, crypt(newpasswd, user));
        p->pw_passwd = savepasswd;
```

```
        valid_change = 1;
    }
    else {
        fprintf(stderr, "Old password is not valid\n");
    }
}
...

```

#### APPLICATION USAGE

None.

#### RATIONALE

None.

#### FUTURE DIRECTIONS

None.

#### SEE ALSO

`strncmp()`

The Base Definitions volume of POSIX.1?2017, <string.h>

#### COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see [https://www.kernel.org/doc/man-pages/reporting\\_bugs.html](https://www.kernel.org/doc/man-pages/reporting_bugs.html) .