NAME

strsep – extract token from string

SYNOPSIS

#include <string.h>

char *strsep(char ****stringp*, const char ****delim***);**

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

strsep(): Since glibc 2.19: _DEFAULT_SOURCE Glibc 2.19 and earlier: _BSD_SOURCE

DESCRIPTION

If **stringp* is NULL, the **strsep**() function returns NULL and does nothing else. Otherwise, this function finds the first token in the string **stringp*, that is delimited by one of the bytes in the string *delim*. This token is terminated by overwriting the delimiter with a null byte ('\0'), and **stringp* is updated to point past the token. In case no delimiter was found, the token is taken to be the entire string **stringp*, and **stringp* is made NULL.

RETURN VALUE

The **strsep**() function returns a pointer to the token, that is, it returns the original value of **stringp*.

ATTRIBUTES

For an explanation of the terms used in this section, see attributes(7).

Interface	**Attribute**	**Value**
strsep()	Thread safety	MT-Safe

CONFORMING TO

4.4BSD.

NOTES

The **strsep**() function was introduced as a replacement for strtok(3), since the latter cannot handle empty fields. However, strtok(3) conforms to C89/C99 and hence is more portable.

BUGS

Be cautious when using this function. If you do use it, note that:

* This function modifies its first argument.

* This function cannot be used on constant strings.

* The identity of the delimiting character is lost.

SEE ALSO

index(3), memchr(3), rindex(3), strchr(3), string(3), strpbrk(3), strspn(3), strstr(3), strtok(3)

COLOPHON

This page is part of release 4.16 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man–pages/.