

NAME

strsep - separate strings

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <string.h>
```

*char **

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

DESCRIPTION

The **strsep()** function locates, in the string referenced by **stringp*, the first occurrence of any character in the string *delim* (or the terminating `'\0'` character) and replaces it with a `'\0'`. The location of the next character after the delimiter character (or NULL, if the end of the string was reached) is stored in **stringp*. The original value of **stringp* is returned.

An "empty" field (i.e., a character in the string *delim* occurs as the first character of **stringp*) can be detected by comparing the location referenced by the returned pointer to `'\0'`.

If **stringp* is initially NULL, **strsep()** returns NULL.

EXAMPLES

The following uses **strsep()** to parse a string, and prints each token in separate line:

```
char *token, *string, *tofree;

tofree = string = strdup("abc,def,ghi");
if (string == NULL)
    err(1, "strdup");
while ((token = strsep(&string, ",")) != NULL)
    printf("%s\n", token);

free(tofree);
```

The following uses **strsep()** to parse a string, containing tokens delimited by white space, into an argument vector:

```
char **ap, *argv[10], *inputstring;
```

```
for (ap = argv; (*ap = strsep(&inputstring, "\t")) != NULL;)
    if (**ap != '\0')
        if (++ap >= &argv[10])
            break;
```

SEE ALSO

memchr(3), strchr(3), strcspn(3), strpbrk(3), strrchr(3), strspn(3), strstr(3), strtok(3)

HISTORY

The **strsep()** function is intended as a replacement for the **strtok()** function. While the **strtok()** function should be preferred for portability reasons (it conforms to ISO/IEC 9899:1990 ("ISO C90")) it is unable to handle empty fields, i.e., detect fields delimited by two adjacent delimiter characters, or to be used for more than a single string at a time. The **strsep()** function first appeared in 4.4BSD.