NAME

strtok, **strtok_r** - string tokens

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <string.h>

*char **
strtok(*char *str, const char *sep*);

*char **
strtok_r(*char *str, const char *sep, char **last*);

DESCRIPTION

This interface is obsoleted by strsep(3).

The **strtok**() function is used to isolate sequential tokens in a null-terminated string, *str*. These tokens are separated in the string by at least one of the characters in *sep*. The first time that **strtok**() is called, *str* should be specified; subsequent calls, wishing to obtain further tokens from the same string, should pass a null pointer instead. The separator string, *sep*, must be supplied each time, and may change between calls.

The implementation will behave as if no library function calls **strtok**().

The **strtok_r**() function is a reentrant version of **strtok**(). The context pointer *last* must be provided on each call. The **strtok_r**() function may also be used to nest two parsing loops within one another, as long as separate context pointers are used.

RETURN VALUES

The **strtok**() and **strtok_r**() functions return a pointer to the beginning of each subsequent token in the string, after replacing the token itself with a NUL character. When no more tokens remain, a null pointer is returned.

EXAMPLES

The following uses **strtok_r**() to parse two strings using separate contexts:

```
char test[80], blah[80];
char *sep = "\\/:;=-";
```

```
char *word, *phrase, *brkt, *brkb;

strcpy(test, "This;is.a:test:of=the/string\\tokenizer-function.");

for (word = strtok_r(test, sep, &brkt);
     word;
     word = strtok_r(NULL, sep, &brkt))
{
   strcpy(blah, "blah:blat:blab:blag");

   for (phrase = strtok_r(blah, sep, &brkb);
        phrase;
        phrase = strtok_r(NULL, sep, &brkb))
   {
      printf("So far we're at %s:%s\n", word, phrase);
   }
}
```

SEE ALSO

memchr(3), strchr(3), strcspn(3), strpbrk(3), strrchr(3), strsep(3), strspn(3), strstr(3), wcstok(3)

STANDARDS

The **strtok**() function conforms to ISO/IEC 9899:1990 ("ISO C90"). The **strtok_r**() function conforms to IEEE Std 1003.1-2001 ("POSIX.1").

AUTHORS

Wes Peters <*wes@softweyr.com*>, Softweyr LLC

Based on the FreeBSD 3.0 implementation.

BUGS

The System V **strtok**(), if handed a string containing only delimiter characters, will not alter the next starting point, so that a call to **strtok**() with a different (or empty) delimiter string may return a non-NULL value. Since this implementation always alters the next starting point, such a sequence of calls would always return NULL.