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 'wcstok.3p' command

$ man wcstok.3p

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

wcstok ? split a wide-character string into tokens

SYNOPSIS

```
#include <wchar.h>
wchar_t *wcstok(wchar_t *restrict ws1, const wchar_t *restrict ws2,
    wchar_t **restrict ptr);
```

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.

A sequence of calls to wcstok() shall break the wide-character string pointed to by ws1 into a sequence of tokens, each of which shall be de? limited by a wide-character code from the wide-character string pointed to by ws2. The ptr argument points to a caller-provided wchar_t pointer into which the wcstok() function shall store information neces? sary for it to continue scanning the same wide-character string.

The first call in the sequence has ws1 as its first argument, and is followed by calls with a null pointer as their first argument. The separator string pointed to by ws2 may be different from call to call.

The first call in the sequence shall search the wide-character string pointed to by ws1 for the first wide-character code that is not contained in the current separator string pointed to by ws2. If no such wide-character code is found, then there are no tokens in the wide-character string pointed to by ws1 and wcstok() shall return a null pointer. If such a wide-character code is found, it shall be the start of the first token.

The wcstok() function shall then search from there for a wide-character code that is contained in the current separator string. If no such wide-character code is found, the current token extends to the end of the wide-character string pointed to by ws1, and subsequent searches for a token shall return a null pointer. If such a wide-character code is found, it shall be overwritten by a null wide character, which terminates the current token. The wcstok() function shall save a pointer to the following wide-character code, from which the next search for a token shall start.

Each subsequent call, with a null pointer as the value of the first argument, shall start searching from the saved pointer and behave as described above.

The implementation shall behave as if no function calls wcstok().

RETURN VALUE

Upon successful completion, the wcstok() function shall return a pointer to the first wide-character code of a token. Otherwise, if there is no token, wcstok() shall return a null pointer.

ERRORS

No errors are defined.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

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

COPYRIGHT