NAME

wordexp - perform shell-style word expansions

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <wordexp.h>

int
wordexp(*const char * restrict words*, *wordexp_t * restrict we*, *int flags*);

void
wordfree(*wordexp_t *we*);

DESCRIPTION

The **wordexp**() function performs shell-style word expansion on *words* and places the list of words into the *we_wordv* member of *we*, and the number of words into *we_wordc*.

The *flags* argument is the bitwise inclusive OR of any of the following constants:

WRDE_APPEND	Append the words to those generated by a previous call to **wordexp**().
WRDE_DOOFFS	As many NULL pointers as are specified by the *we_offs* member of *we* are added to the front of *we_wordv*.
WRDE_NOCMD	Disallow command substitution in *words*. See the note in *BUGS* before using this.
WRDE_REUSE	The *we* argument was passed to a previous successful call to **wordexp**() but has not been passed to **wordfree**(). The implementation may reuse the space allocated to it.
WRDE_SHOWERR	Do not redirect shell error messages to */dev/null*.
WRDE_UNDEF	Report error on an attempt to expand an undefined shell variable.

The *wordexp_t* structure is defined in <*wordexp.h*> as:

```
typedef struct {
        size_t    we_wordc;         /* count of words matched */
```

```
                char      **we_wordv;       /* pointer to list of words */
                size_t    we_offs;  /* slots to reserve in we_wordv */
        } wordexp_t;
```

The **wordfree**() function frees the memory allocated by **wordexp**().

IMPLEMENTATION NOTES

The **wordexp**() function is implemented using the undocumented **freebsd_wordexp** shell built-in command.

RETURN VALUES

The **wordexp**() function returns zero if successful, otherwise it returns one of the following error codes:

WRDE_BADCHAR	The *words* argument contains one of the following unquoted characters: <newline>, ‘|’, ‘&’, ‘;’, ‘<’, ‘>’, ‘(’, ‘)’, ‘{’, ‘}’.
WRDE_BADVAL	An error after successful parsing, such as an attempt to expand an undefined shell variable with WRDE_UNDEF set in *flags*.
WRDE_CMDSUB	An attempt was made to use command substitution and WRDE_NOCMD is set in *flags*.
WRDE_NOSPACE	Not enough memory to store the result or an error during fork(2).
WRDE_SYNTAX	Shell syntax error in *words*.

The **wordfree**() function returns no value.

ENVIRONMENT

IFS Field separator.

EXAMPLES

Invoke the editor on all *.c* files in the current directory and */etc/motd* (error checking omitted):

```
wordexp_t we;

wordexp("${EDITOR:-vi} *.c /etc/motd", &we, 0);
execvp(we.we_wordv[0], we.we_wordv);
```

DIAGNOSTICS

Diagnostic messages from the shell are written to the standard error output if WRDE_SHOWERR is set in *flags*.

SEE ALSO

sh(1), fnmatch(3), glob(3), popen(3), system(3)

STANDARDS

The **wordexp**() and **wordfree**() functions conform to IEEE Std 1003.1-2001 ("POSIX.1").

BUGS

The current **wordexp**() implementation does not recognize multibyte characters other than UTF-8, since the shell (which it invokes to perform expansions) does not.

SECURITY CONSIDERATIONS

Pathname generation may create output that is exponentially larger than the input size.

Although this implementation detects command substitution reliably for WRDE_NOCMD, the attack surface remains fairly large. Also, some other implementations (such as older versions of this one) may execute command substitutions even if WRDE_NOCMD is set.