

NAME

wc — word, line, character, and byte count

SYNOPSIS

wc [**--libxo**] [**-Lclmw**] [*file* . . .]

DESCRIPTION

The **wc** utility displays the number of lines, words, and bytes contained in each input *file*, or standard input (if no file is specified) to the standard output. A line is defined as a string of characters delimited by a ⟨newline⟩ character. Characters beyond the final ⟨newline⟩ character will not be included in the line count.

A word is defined as a string of characters delimited by white space characters. White space characters are the set of characters for which the `iswspace(3)` function returns true. If more than one input file is specified, a line of cumulative counts for all the files is displayed on a separate line after the output for the last file.

The following options are available:

--libxo

Generate output via `libxo(3)` in a selection of different human and machine readable formats. See `xo_parse_args(3)` for details on command line arguments.

-L Write the length of the line containing the most bytes (default) or characters (when **-m** is provided) to standard output. When more than one *file* argument is specified, the longest input line of *all* files is reported as the value of the final “total”.

-c The number of bytes in each input file is written to the standard output. This will cancel out any prior usage of the **-m** option.

-l The number of lines in each input file is written to the standard output.

-m The number of characters in each input file is written to the standard output. If the current locale does not support multibyte characters, this is equivalent to the **-c** option. This will cancel out any prior usage of the **-c** option.

-w The number of words in each input file is written to the standard output.

When an option is specified, **wc** only reports the information requested by that option. The order of output always takes the form of line, word, byte, and file name. The default action is equivalent to specifying the **-c**, **-l** and **-w** options.

If no files are specified, the standard input is used and no file name is displayed. The prompt will accept input until receiving EOF, or [^D] in most environments.

ENVIRONMENT

The `LANG`, `LC_ALL` and `LC_CTYPE` environment variables affect the execution of **wc** as described in `environ(7)`.

EXIT STATUS

The **wc** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

Count the number of characters, words and lines in each of the files `report1` and `report2` as well as the totals for both:

```
wc -mlw report1 report2
```

Find the longest line in a list of files:

```
wc -L file1 file2 file3 | fgrep total
```

COMPATIBILITY

Historically, the **wc** utility was documented to define a word as a “maximal string of characters delimited by <space>, <tab> or <newline> characters”. The implementation, however, did not handle non-printing characters correctly so that “`^D^E`” counted as 6 spaces, while “`foo^D^Ebar`” counted as 8 characters. 4BSD systems after 4.3BSD modified the implementation to be consistent with the documentation. This implementation defines a “word” in terms of the `iswspace(3)` function, as required by IEEE Std 1003.2 (“POSIX.2”).

The **-L** option is a non-standard FreeBSD extension, compatible with the **-L** option of the GNU **wc** utility.

SEE ALSO

`iswspace(3)`, `libxo(3)`, `xo_parse_args(3)`

STANDARDS

The **wc** utility conforms to IEEE Std 1003.1-2001 (“POSIX.1”).

HISTORY

A **wc** command appeared in Version 1 AT&T UNIX.