
*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 'wc.1p' command*

***$ man wc.1p***

WC(1P) POSIX Programmer's Manual WC(1P)

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

wc ? word, line, and byte or character count

SYNOPSIS

wc [-c|-m] [-lw] [file...]

DESCRIPTION

The wc utility shall read one or more input files and, by default, write the number of <newline> characters, words, and bytes contained in each input file to the standard output.

The utility also shall write a total count for all named files, if more than one input file is specified.

The wc utility shall consider a word to be a non-zero-length string of characters delimited by white space.

OPTIONS

The wc utility shall conform to the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines.

The following options shall be supported:

-c Write to the standard output the number of bytes in each in?

put file.

-l Write to the standard output the number of <newline> charac? ters in each input file.

-m Write to the standard output the number of characters in each input file.

-w Write to the standard output the number of words in each in? put file.

When any option is specified, wc shall report only the information re? quested by the specified options.

OPERANDS

The following operand shall be supported:

file A pathname of an input file. If no file operands are speci? fied, the standard input shall be used.

STDIN

The standard input shall be used if no file operands are specified, and shall be used if a file operand is '-' and the implementation treats the '-' as meaning standard input. Otherwise, the standard input shall not be used. See the INPUT FILES section.

INPUT FILES

The input files may be of any type.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of wc:

LANG Provide a default value for the internationalization vari? ables that are unset or null. (See the Base Definitions vol? ume of POSIX.1?2017, Section 8.2, Internationalization Vari? ables for the precedence of internationalization variables used to determine the values of locale categories.)

LC_ALL If set to a non-empty string value, override the values of all the other internationalization variables.

LC_CTYPE Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files) and which characters are defined as white-space char?

acters.

LC_MESSAGES

Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

NLSPATH Determine the location of message catalogs for the processing of LC_MESSAGES.

ASYNCHRONOUS EVENTS

Default.

STDOUT

By default, the standard output shall contain an entry for each input file of the form:

"%d %d %d %s\n", <newlines>, <words>, <bytes>, <file>

If the -m option is specified, the number of characters shall replace the <bytes> field in this format.

If any options are specified and the -l option is not specified, the number of <newline> characters shall not be written.

If any options are specified and the -w option is not specified, the number of words shall not be written.

If any options are specified and neither -c nor -m is specified, the number of bytes or characters shall not be written.

If no input file operands are specified, no name shall be written and no <blank> characters preceding the pathname shall be written.

If more than one input file operand is specified, an additional line shall be written, of the same format as the other lines, except that the word total (in the POSIX locale) shall be written instead of a pathname and the total of each column shall be written as appropriate.

Such an additional line, if any, is written at the end of the output.

STDERR

The standard error shall be used only for diagnostic messages.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

0 Successful completion.

>0 An error occurred.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

The -m option is not a switch, but an option at the same level as -c. Thus, to produce the full default output with character counts instead of bytes, the command required is:

```
wc -mlw
```

EXAMPLES

None.

RATIONALE

The output file format pseudo-printf() string differs from the System V version of wc:

```
"%7d%7d%7d %s\n"
```

which produces possibly ambiguous and unparsable results for very large files, as it assumes no number shall exceed six digits.

Some historical implementations use only <space>, <tab>, and <newline> as word separators. The equivalent of the ISO C standard isspace() function is more appropriate.

The -c option stands for ``character'' count, even though it counts bytes. This stems from the sometimes erroneous historical view that bytes and characters are the same size. Due to international require? ments, the -m option (reminiscent of ``multi-byte'') was added to ob? tain actual character counts.

Early proposals only specified the results when input files were text files. The current specification more closely matches historical prac? tice. (Bytes, words, and <newline> characters are counted separately and the results are written when an end-of-file is detected.)

Historical implementations of the wc utility only accepted one argument to specify the options -c, -l, and -w. Some of them also had multiple occurrences of an option cause the corresponding count to be written multiple times and had the order of specification of the options affect the order of the fields on output, but did not document either of these. Because common usage either specifies no options or only one op? tion, and because none of this was documented, the changes required by this volume of POSIX.1?2017 should not break many historical applica? tions (and do not break any historical conforming applications).

FUTURE DIRECTIONS

None.

SEE ALSO

cksum

The Base Definitions volume of POSIX.1?2017, Chapter 8, Environment Variables, Section 12.2, Utility Syntax Guidelines