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 'tail.1p' command

$ man tail.1p

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

tail ? copy the last part of a file

SYNOPSIS

tail [-f] [-c number|-n number] [file]

DESCRIPTION

The tail utility shall copy its input file to the standard output be? ginning at a designated place.

Copying shall begin at the point in the file indicated by the -c number or -n number options. The option-argument number shall be counted in units of lines or bytes, according to the options -n and -c. Both line and byte counts start from 1.

Tails relative to the end of the file may be saved in an internal buf? fer, and thus may be limited in length. Such a buffer, if any, shall be no smaller than {LINE_MAX}*10 bytes.

OPTIONS

The tail utility shall conform to the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines, except that '+'

may be recognized as an option delimiter as well as '-'.

The following options shall be supported:

-c number The application shall ensure that the number option-argument is a decimal integer, optionally including a sign. The sign shall affect the location in the file, measured in bytes, to begin the copying:

Sign	Copying Starts
+	Relative to the beginning of the file.
-	Relative to the end of the file.
none	Relative to the end of the file.

The application shall ensure that if the sign of the number option-argument is '+', the number option-argument is a non-zero decimal integer.

The origin for counting shall be 1; that is, -c +1 represents the first byte of the file, -c -1 the last.

-f If the input file is a regular file or if the file operand specifies a FIFO, do not terminate after the last line of the input file has been copied, but read and copy further bytes from the input file when they become available. If no file operand is specified and standard input is a pipe or FIFO, the -f option shall be ignored. If the input file is not a FIFO, pipe, or regular file, it is unspecified whether or not the -f option shall be ignored.

-n number This option shall be equivalent to -c number, except the starting location in the file shall be measured in lines in? stead of bytes. The origin for counting shall be 1; that is, -n +1 represents the first line of the file, -n -1 the last.

If neither -c nor -n is specified, -n 10 shall be assumed.

OPERANDS

The following operand shall be supported:

file A pathname of an input file. If no file operand is specified, the standard input shall be used.

STDIN

The standard input shall be used if no file operand is specified, and shall be used if the 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

If the -c option is specified, the input file can contain arbitrary data; otherwise, the input file shall be a text file.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of tail:

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).

LC_MESSAGES

Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard er? ror.

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

ASYNCHRONOUS EVENTS

Default.

STDOUT

The designated portion of the input file shall be written to standard

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 -c option should be used with caution when the input is a text file containing multi-byte characters; it may produce output that does not start on a character boundary.

Although the input file to tail can be any type, the results might not be what would be expected on some character special device files or on file types not described by the System Interfaces volume of POSIX.1?2017. Since this volume of POSIX.1?2017 does not specify the block size used when doing input, tail need not read all of the data from devices that only perform block transfers.

When using tail to process pathnames, and the -c option is not speci? fied, it is recommended that LC_ALL, or at least LC_CTYPE and LC_COL? LATE, are set to POSIX or C in the environment, since pathnames can contain byte sequences that do not form valid characters in some lo? cales, in which case the utility's behavior would be undefined. In the POSIX locale each byte is a valid single-byte character, and therefore this problem is avoided.

EXAMPLES

The -f option can be used to monitor the growth of a file that is being

written by some other process. For example, the command:

tail -f fred

prints the last ten lines of the file fred, followed by any lines that are appended to fred between the time tail is initiated and killed. As another example, the command:

tail -f -c 15 fred

prints the last 15 bytes of the file fred, followed by any bytes that are appended to fred between the time tail is initiated and killed.

RATIONALE

This version of tail was created to allow conformance to the Utility Syntax Guidelines. The historical -b option was omitted because of the general non-portability of block-sized units of text. The -c option historically meant ``characters'', but this volume of POSIX.1?2017 in? dicates that it means ``bytes''. This was selected to allow reasonable implementations when multi-byte characters are possible; it was not named -b to avoid confusion with the historical -b.

The origin of counting both lines and bytes is 1, matching all wide? spread historical implementations. Hence tail -n +0 is not conforming usage because it attempts to output line zero; but note that tail -n 0 does conform, and outputs nothing.

Earlier versions of this standard allowed the following forms in the SYNOPSIS:

tail -[number][b|c|l][f] [file]

tail +[number][b|c|l][f] [file]

These forms are no longer specified by POSIX.1?2008, but may be present in some implementations.

The restriction on the internal buffer is a compromise between the his? torical System V implementation of 4096 bytes and the BSD 32768 bytes.

The -f option has been implemented as a loop that sleeps for 1 second and copies any bytes that are available. This is sufficient, but if more efficient methods of determining when new data are available are developed, implementations are encouraged to use them.

Historical documentation indicates that tail ignores the -f option if

the input file is a pipe (pipe and FIFO on systems that support FIFOs).

On BSD-based systems, this has been true; on System V-based systems, this was true when input was taken from standard input, but it did not ignore the -f flag if a FIFO was named as the file operand. Since the -f option is not useful on pipes and all historical implementations ig? nore -f if no file operand is specified and standard input is a pipe, this volume of POSIX.1?2017 requires this behavior. However, since the -f option is useful on a FIFO, this volume of POSIX.1?2017 also re? quires that if a FIFO is named, the -f option shall not be ignored.

Earlier versions of this standard did not state any requirement for the case where no file operand is specified and standard input is a FIFO. The standard has been updated to reflect current practice which is to treat this case the same as a pipe on standard input. Although histor? ical behavior does not ignore the -f option for other file types, this is unspecified so that implementations are allowed to ignore the -f op? tion if it is known that the file cannot be extended.

FUTURE DIRECTIONS

None.

SEE ALSO

head

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

COPYRIGHT

most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.ker?nel.org/doc/man-pages/reporting_bugs.html .