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 'compress.1p' command

$ man compress.1p

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

compress ? compress data

SYNOPSIS

compress [-fv] [-b bits] [file...]

compress [-cfv] [-b bits] [file]

DESCRIPTION

The compress utility shall attempt to reduce the size of the named files by using adaptive Lempel-Ziv coding algorithm.

Note: Lempel-Ziv is US Patent 4464650, issued to William Eastman, Abraham Lempel, Jacob Ziv, Martin Cohn on August 7th, 1984, and assigned to Sperry Corporation.

Lempel-Ziv-Welch compression is covered by US Patent 4558302, issued to Terry A. Welch on December 10th, 1985, and assigned to Sperry Corporation.

On systems not supporting adaptive Lempel-Ziv coding algorithm, the in? put files shall not be changed and an error value greater than two shall be returned. Except when the output is to the standard output,

each file shall be replaced by one with the extension .Z. If the in?
voking process has appropriate privileges, the ownership, modes, access
time, and modification time of the original file are preserved. If ap?
pending the .Z to the filename would make the name exceed {NAME_MAX}
bytes, the command shall fail. If no files are specified, the standard
input shall be compressed to the standard output.

OPTIONS

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

The following options shall be supported:

-b bits Specify the maximum number of bits to use in a code. For a
conforming application, the bits argument shall be:

9 <= bits <= 14

The implementation may allow bits values of greater than 14.
The default is 14, 15, or 16.

-c Cause compress to write to the standard output; the input
file is not changed, and no .Z files are created.

-f Force compression of file, even if it does not actually re?
duce the size of the file, or if the corresponding file.Z
file already exists. If the -f option is not given, and the
process is not running in the background, the user is
prompted as to whether an existing file.Z file should be
overwritten. If the response is affirmative, the existing
file will be overwritten.

-v Write the percentage reduction of each file to standard er?
ror.

OPERANDS

The following operand shall be supported:

file A pathname of a file to be compressed.

STDIN

The standard input shall be used only if no file operands are speci?
fied, or if a file operand is '-'.

INPUT FILES

If file operands are specified, the input files contain the data to be compressed.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of com? press:

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_COLLATE

Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements used in the extended regular expression defined for the yesexpr locale keyword in the LC_MESSAGES category.

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), the behavior of character classes used in the extended regular expression defined for the yesexpr locale keyword in the LC_MESSAGES category.

LC_MESSAGES

Determine the locale used to process affirmative responses, and the locale used to affect the format and contents of di? agnostic messages, prompts, and the output from the -v option written to standard error.

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

ASYNCHRONOUS EVENTS

Default.

STDOUT

If no file operands are specified, or if a file operand is '-', or if the -c option is specified, the standard output contains the compressed output.

STDERR

The standard error shall be used only for diagnostic and prompt mes? sages and the output from -v.

OUTPUT FILES

The output files shall contain the compressed output. The format of compressed files is unspecified and interchange of such files between implementations (including access via unspecified file sharing mecha? nisms) is not required by POSIX.1?2008.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

0 Successful completion.

1 An error occurred.

2 One or more files were not compressed because they would have in? creased in size (and the -f option was not specified).

>2 An error occurred.

CONSEQUENCES OF ERRORS

The input file shall remain unmodified.

The following sections are informative.

APPLICATION USAGE

The amount of compression obtained depends on the size of the input, the number of bits per code, and the distribution of common substrings. Typically, text such as source code or English is reduced by 50?60%. Compression is generally much better than that achieved by Huffman cod? ing or adaptive Huffman coding (compact), and takes less time to com? pute.

Although compress strictly follows the default actions upon receipt of a signal or when an error occurs, some unexpected results may occur. In some implementations it is likely that a partially compressed file is

left in place, alongside its uncompressed input file. Since the general operation of compress is to delete the uncompressed file only after the .Z file has been successfully filled, an application should always carefully check the exit status of compress before arbitrarily deleting files that have like-named neighbors with .Z suffixes.

The limit of 14 on the bits option-argument is to achieve portability to all systems (within the restrictions imposed by the lack of an ex? plicit published file format). Some implementations based on 16-bit ar? chitectures cannot support 15 or 16-bit uncompression.

EXAMPLES

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

uncompress, zcat

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