

NAME

`gzexe` – compress executable files in place

SYNOPSIS

gzexe *name* ...

DESCRIPTION

The *gzexe* utility allows you to compress executables in place and have them automatically uncompress and execute when you run them (at a penalty in performance). For example if you execute “*gzexe* /usr/bin/gdb” it will create the following two files:

```
-rwxr-xr-x 1 root root 1026675 Jun  7 13:53 /usr/bin/gdb
```

```
-rwxr-xr-x 1 root root 2304524 May 30 13:02 /usr/bin/gdb~
```

/usr/bin/gdb~ is the original file and /usr/bin/gdb is the self-uncompressing executable file. You can remove /usr/bin/gdb~ once you are sure that /usr/bin/gdb works properly.

This utility is most useful on systems with very small disks.

OPTIONS

-d Decompress the given executables instead of compressing them.

SEE ALSO

gzip(1), *znew*(1), *zmore*(1), *zcmp*(1), *zforce*(1)

CAVEATS

The compressed executable is a shell script. This may create some security holes. In particular, the compressed executable relies on the *PATH* environment variable to find *gzip* and some standard utilities (*basename*, *chmod*, *ln*, *mkdir*, *mktemp*, *rm*, *sleep*, and *tail*).

BUGS

gzexe attempts to retain the original file attributes on the compressed executable, but you may have to fix them manually in some cases, using *chmod* or *chown*.

NAME

gzip, *gunzip*, *zcat* – compress or expand files

SYNOPSIS

gzip [**-acdfhlLnRrtvV19**] [**-S suffix**] [*name* ...]

gunzip [**-acdfhlLnRrtvV**] [**-S suffix**] [*name* ...]

zcat [**-fhLV**] [*name* ...]

DESCRIPTION

Gzip reduces the size of the named files using Lempel-Ziv coding (LZ77). Whenever possible, each file is replaced by one with the extension **.gz**, while keeping the same ownership modes, access and modification times. (The default extension is **-gz** for VMS, **z** for MSDOS, OS/2 FAT, Windows NT FAT and Atari.) If no files are specified, or if a file name is "-", the standard input is compressed to the standard output. *Gzip* will only attempt to compress regular files. In particular, it will ignore symbolic links.

If the compressed file name is too long for its file system, *gzip* truncates it. *Gzip* attempts to truncate only the parts of the file name longer than 3 characters. (A part is delimited by dots.) If the name consists of small parts only, the longest parts are truncated. For example, if file names are limited to 14 characters, *gzip.msdos.exe* is compressed to *gzi.msd.exe.gz*. Names are not truncated on systems which do not have a limit on file name length.

By default, *gzip* keeps the original file name and timestamp in the compressed file. These are used when decompressing the file with the **-N** option. This is useful when the compressed file name was truncated or when the time stamp was not preserved after a file transfer.

Compressed files can be restored to their original form using *gzip -d* or *gunzip* or *zcat*. If the original name saved in the compressed file is not suitable for its file system, a new name is constructed from the original one to make it legal.

gunzip takes a list of files on its command line and replaces each file whose name ends with **.gz**, **-gz**, **.z**, **-z**, **_z** or **.Z** and which begins with the correct magic number with an uncompressed file without the original extension. *gunzip* also recognizes the special extensions **.tgz** and **.taz** as shorthands for **.tar.gz** and **.tar.Z** respectively. When compressing, *gzip* uses the **.tgz** extension if necessary instead of truncating a file with a **.tar** extension.

gunzip can currently decompress files created by *gzip*, *zip*, *compress*, *compress -H* or *pack*. The detection of the input format is automatic. When using the first two formats, *gunzip* checks a 32 bit CRC. For *pack*, *gunzip* checks the uncompressed length. The standard *compress* format was not designed to allow consistency checks. However *gunzip* is sometimes able to detect a bad **.Z** file. If you get an error when uncompressing a **.Z** file, do not assume that the **.Z** file is correct simply because the standard *uncompress* does not complain. This generally means that the standard *uncompress* does not check its input, and happily generates garbage output. The SCO *compress -H* format (lzh compression method) does not include a CRC but also allows some consistency checks.

Files created by *zip* can be uncompressed by *gzip* only if they have a single member compressed with the 'deflation' method. This feature is only intended to help conversion of *tar.zip* files to the *tar.gz* format. To extract a *zip* file with a single member, use a command like *gunzip <foo.zip* or *gunzip -S .zip foo.zip*. To extract *zip* files with several members, use *unzip* instead of *gunzip*.

zcat is identical to *gunzip -c*. (On some systems, *zcat* may be installed as *gzcat* to preserve the original link to *compress*.) *zcat* uncompresses either a list of files on the command line or its standard input and writes the uncompressed data on standard output. *zcat* will uncompress files that have the correct magic number whether they have a **.gz** suffix or not.

Gzip uses the Lempel-Ziv algorithm used in *zip* and PKZIP. The amount of compression obtained depends on the size of the input and the distribution of common substrings. Typically, text such as source code or English is reduced by 60–70%. Compression is generally much better than that achieved by LZW (as used in *compress*), Huffman coding (as used in *pack*), or adaptive Huffman coding (*compact*).

Compression is always performed, even if the compressed file is slightly larger than the original. The worst case expansion is a few bytes for the *gzip* file header, plus 5 bytes every 32K block, or an expansion ratio of 0.015% for large files. Note that the actual number of used disk blocks almost never

increases. *gzip* preserves the mode, ownership and timestamps of files when compressing or decompressing.

The *gzip* file format is specified in P. Deutsch, GZIP file format specification version 4.3, <ftp://ftp.isi.edu/in-notes/rfc1952.txt>, Internet RFC 1952 (May 1996). The *zip* deflation format is specified in P. Deutsch, DEFLATE Compressed Data Format Specification version 1.3, <ftp://ftp.isi.edu/in-notes/rfc1951.txt>, Internet RFC 1951 (May 1996).

OPTIONS

-a --ascii

Ascii text mode: convert end-of-lines using local conventions. This option is supported only on some non-Unix systems. For MSDOS, CR LF is converted to LF when compressing, and LF is converted to CR LF when decompressing.

-c --stdout --to-stdout

Write output on standard output; keep original files unchanged. If there are several input files, the output consists of a sequence of independently compressed members. To obtain better compression, concatenate all input files before compressing them.

-d --decompress --uncompress

Decompress.

-f --force

Force compression or decompression even if the file has multiple links or the corresponding file already exists, or if the compressed data is read from or written to a terminal. If the input data is not in a format recognized by *gzip*, and if the option *--stdout* is also given, copy the input data without change to the standard output: let *zcat* behave as *cat*. If *-f* is not given, and when not running in the background, *gzip* prompts to verify whether an existing file should be overwritten.

-h --help

Display a help screen and quit.

-l --list For each compressed file, list the following fields:

compressed size: size of the compressed file
 uncompressed size: size of the uncompressed file
 ratio: compression ratio (0.0% if unknown)
 uncompressed_name: name of the uncompressed file

The uncompressed size is given as -1 for files not in *gzip* format, such as compressed .Z files. To get the uncompressed size for such a file, you can use:

```
zcat file.Z | wc -c
```

In combination with the *--verbose* option, the following fields are also displayed:

method: compression method
 crc: the 32-bit CRC of the uncompressed data
 date & time: time stamp for the uncompressed file

The compression methods currently supported are deflate, compress, lzh (SCO compress -H) and pack. The crc is given as ffffffff for a file not in *gzip* format.

With *--name*, the uncompressed name, date and time are those stored within the compress file if present.

With *--verbose*, the size totals and compression ratio for all files is also displayed, unless some sizes are unknown. With *--quiet*, the title and totals lines are not displayed.

-L --license

Display the *gzip* license and quit.

-n --no-name

When compressing, do not save the original file name and time stamp by default. (The original name is always saved if the name had to be truncated.) When decompressing, do not restore the original file name if present (remove only the *gzip* suffix from the compressed file name) and do not restore the original time stamp if present (copy it from the compressed file). This option is the default when decompressing.

-N --name

When compressing, always save the original file name and time stamp; this is the default. When decompressing, restore the original file name and time stamp if present. This option is useful on systems which have a limit on file name length or when the time stamp has been lost after a file transfer.

-q --quiet

Suppress all warnings.

-r --recursive

Travel the directory structure recursively. If any of the file names specified on the command line are directories, *gzip* will descend into the directory and compress all the files it finds there (or decompress them in the case of *gunzip*).

--rsyncable

While compressing, synchronize the output occasionally based on the input. This increases size by less than 1 percent most cases, but means that the **rsync**(1) program can much more efficiently synchronize files compressed with this flag. *gunzip* cannot tell the difference between a compressed file created with this option, and one created without it.

-S .suf --suffix .suf

Use suffix .suf instead of .gz. Any suffix can be given, but suffixes other than .z and .gz should be avoided to avoid confusion when files are transferred to other systems. A null suffix forces *gunzip* to try decompression on all given files regardless of suffix, as in:

```
gunzip -S "" *      (*. * for MSDOS)
```

Previous versions of *gzip* used the .z suffix. This was changed to avoid a conflict with *pack*(1).

-t --test

Test. Check the compressed file integrity.

-v --verbose

Verbose. Display the name and percentage reduction for each file compressed or decompressed.

-V --version

Version. Display the version number and compilation options then quit.

-# --fast --best

Regulate the speed of compression using the specified digit #, where **-1** or **--fast** indicates the fastest compression method (less compression) and **-9** or **--best** indicates the slowest compression method (best compression). The default compression level is **-6** (that is, biased towards high compression at expense of speed).

ADVANCED USAGE

Multiple compressed files can be concatenated. In this case, *gunzip* will extract all members at once. For example:

```
gzip -c file1 > foo.gz
gzip -c file2 >> foo.gz
```

Then

```
gunzip -c foo
```

is equivalent to

```
cat file1 file2
```

In case of damage to one member of a .gz file, other members can still be recovered (if the damaged member is removed). However, you can get better compression by compressing all members at once:

```
cat file1 file2 | gzip > foo.gz
```

compresses better than

```
gzip -c file1 file2 > foo.gz
```

If you want to recompress concatenated files to get better compression, do:

```
gzip -cd old.gz | gzip > new.gz
```

If a compressed file consists of several members, the uncompressed size and CRC reported by the --list option applies to the last member only. If you need the uncompressed size for all members, you can use:

```
gzip -cd file.gz | wc -c
```

If you wish to create a single archive file with multiple members so that members can later be extracted independently, use an archiver such as tar or zip. GNU tar supports the -z option to invoke gzip transparently. gzip is designed as a complement to tar, not as a replacement.

ENVIRONMENT

The environment variable **GZIP** can hold a set of default options for *gzip*. These options are interpreted first and can be overwritten by explicit command line parameters. For example:

```
for sh:  GZIP="-8v --name"; export GZIP
for csh: setenv GZIP "-8v --name"
for MSDOS: set GZIP=-8v --name
```

On Vax/VMS, the name of the environment variable is GZIP_OPT, to avoid a conflict with the symbol set for invocation of the program.

SEE ALSO

znew(1), zcmp(1), zmore(1), zforce(1), gzexe(1), zip(1), unzip(1), compress(1), pack(1), compact(1)

The *gzip* file format is specified in P. Deutsch, GZIP file format specification version 4.3, <<ftp://ftp.isi.edu/in-notes/rfc1952.txt>>, Internet RFC 1952 (May 1996). The *zip* deflation format is specified in P. Deutsch, DEFLATE Compressed Data Format Specification version 1.3, <<ftp://ftp.isi.edu/in-notes/rfc1951.txt>>, Internet RFC 1951 (May 1996).

DIAGNOSTICS

Exit status is normally 0; if an error occurs, exit status is 1. If a warning occurs, exit status is 2.

Usage: *gzip* [-cdfhlLnNrtvV19] [-S suffix] [file ...]

Invalid options were specified on the command line.

file: not in gzip format

The file specified to *gunzip* has not been compressed.

file: Corrupt input. Use zcat to recover some data.

The compressed file has been damaged. The data up to the point of failure can be recovered using

```
zcat file > recover
```

file: compressed with *xx* bits, can only handle *yy* bits

File was compressed (using LZW) by a program that could deal with more *bits* than the decompress code on this machine. Recompress the file with *gzip*, which compresses better

and uses less memory.

file: already has .gz suffix -- no change

The file is assumed to be already compressed. Rename the file and try again.

file already exists; do you wish to overwrite (y or n)?

Respond "y" if you want the output file to be replaced; "n" if not.

gunzip: corrupt input

A SIGSEGV violation was detected which usually means that the input file has been corrupted.

xx.x% Percentage of the input saved by compression.

(Relevant only for **-v** and **-l**.)

-- not a regular file or directory: ignored

When the input file is not a regular file or directory, (e.g. a symbolic link, socket, FIFO, device file), it is left unaltered.

-- has *xx* other links: unchanged

The input file has links; it is left unchanged. See *ln(1)* for more information. Use the **-f** flag to force compression of multiply-linked files.

CAVEATS

When writing compressed data to a tape, it is generally necessary to pad the output with zeroes up to a block boundary. When the data is read and the whole block is passed to *gunzip* for decompression, *gunzip* detects that there is extra trailing garbage after the compressed data and emits a warning by default. You have to use the **--quiet** option to suppress the warning. This option can be set in the **GZIP** environment variable as in:

```
for sh: GZIP="-q" tar -xfz --block-compress /dev/rst0
```

```
for csh: (setenv GZIP -q; tar -xfz --block-compr /dev/rst0
```

In the above example, *gzip* is invoked implicitly by the **-z** option of GNU *tar*. Make sure that the same block size (**-b** option of *tar*) is used for reading and writing compressed data on tapes. (This example assumes you are using the GNU version of *tar*.)

BUGS

The *gzip* format represents the input size modulo 2^{32} , so the **--list** option reports incorrect uncompressed sizes and compression ratios for uncompressed files 4 GB and larger. To work around this problem, you can use the following command to discover a large uncompressed file's true size:

```
zcat file.gz | wc -c
```

The **--list** option reports sizes as **-1** and **crc** as **fffffff** if the compressed file is on a non seekable media.

In some rare cases, the **--best** option gives worse compression than the default compression level (**-6**). On some highly redundant files, *compress* compresses better than *gzip*.

COPYRIGHT NOTICE

Copyright © 1998, 1999, 2001, 2002 Free Software Foundation, Inc.

Copyright © 1992, 1993 Jean-loup Gailly

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Foundation.

NAME

zcmp, zdiff – compare compressed files

SYNOPSIS

zcmp [cmp_options] file1 [file2]

zdiff [diff_options] file1 [file2]

DESCRIPTION

Zcmp and *zdiff* are used to invoke the *cmp* or the *diff* program on files compressed via *gzip*. All options specified are passed directly to *cmp* or *diff*. If only 1 file is specified, then the files compared are *file1* and an uncompressed *file1.gz*. If two files are specified, then they are uncompressed if necessary and fed to *cmp* or *diff*. The exit status from *cmp* or *diff* is preserved.

SEE ALSO

cmp(1), diff(1), zmore(1), zgrep(1), znew(1), zforce(1), gzip(1), gzexe(1)

BUGS

Messages from the *cmp* or *diff* programs refer to temporary filenames instead of those specified.

NAME

`zforce` – force a '.gz' extension on all `gzip` files

SYNOPSIS

zforce [name ...]

DESCRIPTION

zforce forces a .gz extension on all *gzip* files so that *gzip* will not compress them twice. This can be useful for files with names truncated after a file transfer. On systems with a 14 char limitation on file names, the original name is truncated to make room for the .gz suffix. For example, 12345678901234 is renamed to 12345678901.gz. A file name such as foo.tgz is left intact.

SEE ALSO

`gzip(1)`, `znew(1)`, `zmore(1)`, `zgrep(1)`, `zdiff(1)`, `gzexe(1)`

NAME

zgrep – search possibly compressed files for a regular expression

SYNOPSIS

zgrep [grep_options] [**-e**] *pattern filename...*

DESCRIPTION

Zgrep invokes *grep* on compressed or gzipped files. All options specified are passed directly to *grep*. If no file is specified, then the standard input is decompressed if necessary and fed to *grep*. Otherwise the given files are uncompressed if necessary and fed to *grep*.

If the GREP environment variable is set, *zgrep* uses it as the *grep* program to be invoked.

AUTHOR

Charles Levert (charles@comm.polymtl.ca)

SEE ALSO

grep(1), gzexe(1), gzip(1), zdiff(1), zforce(1), zmore(1), znew(1)

NAME

zless – file perusal filter for crt viewing of compressed text

SYNOPSIS

zless [name ...]

DESCRIPTION

Zless is a filter which allows examination of compressed or plain text files one screenful at a time on a soft-copy terminal. It is the equivalent of setting the environment variable LESSOPEN to "|gzip -cdfq -- %s", and then running *less*. However, enough people seem to think that having the command *zless* available is important to be worth providing it.

Note that *zless* does not work with files piped to it on stdin, it only works with files specified as arguments. If you really want to use *less* in a pipe, then do the uncompress explicitly and pipe to *less* instead of *zless*.

SEE ALSO

zmore(1), *less*(1)

BUGS

Zless does not work with compressed data that is piped to it via standard input; it requires that input files be specified as arguments.

COPYRIGHT NOTICE

Copyright © 2006 Free Software Foundation, Inc.

Copyright © 1992, 1993 Jean-loup Gailly

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Foundation.

NAME

zmore – file perusal filter for crt viewing of compressed text

SYNOPSIS

zmore [name ...]

DESCRIPTION

Zmore is a filter which allows examination of compressed or plain text files one screenful at a time on a soft-copy terminal. *zmore* works on files compressed with *compress*, *pack* or *gzip*, and also on uncompressed files. If a file does not exist, *zmore* looks for a file of the same name with the addition of a .gz, .z or .Z suffix.

Zmore normally pauses after each screenful, printing --More-- at the bottom of the screen. If the user then types a carriage return, one more line is displayed. If the user hits a space, another screenful is displayed. Other possibilities are enumerated later.

Zmore looks in the file */etc/termcap* to determine terminal characteristics, and to determine the default window size. On a terminal capable of displaying 24 lines, the default window size is 22 lines. To use a pager other than the default *more*, set environment variable *PAGER* to the name of the desired program, such as *less*.

Other sequences which may be typed when *zmore* pauses, and their effects, are as follows (*i* is an optional integer argument, defaulting to 1) :

- i*<space> display *i* more lines, (or another screenful if no argument is given)
- ^D display 11 more lines (a “scroll”). If *i* is given, then the scroll size is set to *i*.
- d same as ^D (control-D)
- iz* same as typing a space except that *i*, if present, becomes the new window size. Note that the window size reverts back to the default at the end of the current file.
- is* skip *i* lines and print a screenful of lines
- if* skip *i* screenfuls and print a screenful of lines
- q or Q quit reading the current file; go on to the next (if any)
- e or q When the prompt --More--(Next file: *file*) is printed, this command causes *zmore* to exit.
- s When the prompt --More--(Next file: *file*) is printed, this command causes *zmore* to skip the next file and continue.
- = Display the current line number.
- i*/*expr* search for the *i*-th occurrence of the regular expression *expr*. If the pattern is not found, *zmore* goes on to the next file (if any). Otherwise, a screenful is displayed, starting two lines before the place where the expression was found. The user's erase and kill characters may be used to edit the regular expression. Erasing back past the first column cancels the search command.
- in* search for the *i*-th occurrence of the last regular expression entered.
- !command invoke a shell with *command*. The character ‘!’ in “command” is replaced with the previous shell command. The sequence “\!” is replaced by “!”.
- :q or :Q quit reading the current file; go on to the next (if any) (same as q or Q).
- . (dot) repeat the previous command.

The commands take effect immediately, i.e., it is not necessary to type a carriage return. Up to the time

when the command character itself is given, the user may hit the line kill character to cancel the numerical argument being formed. In addition, the user may hit the erase character to redisplay the --More-- message.

At any time when output is being sent to the terminal, the user can hit the quit key (normally control-\\). *Zmore* will stop sending output, and will display the usual --More-- prompt. The user may then enter one of the above commands in the normal manner. Unfortunately, some output is lost when this is done, due to the fact that any characters waiting in the terminal's output queue are flushed when the quit signal occurs.

The terminal is set to *noecho* mode by this program so that the output can be continuous. What you type will thus not show on your terminal, except for the / and ! commands.

If the standard output is not a teletype, then *zmore* acts just like *zcat*, except that a header is printed before each file.

FILES

/etc/termcap

Terminal data base

SEE ALSO

more(1), gzip(1), zdiff(1), zgrep(1), znew(1), zforce(1), gzexe(1)

NAME

`znew` – recompress .Z files to .gz files

SYNOPSIS

znew [-ftv9PK] [name.Z ...]

DESCRIPTION

Znew recompresses files from .Z (compress) format to .gz (gzip) format. If you want to recompress a file already in gzip format, rename the file to force a .Z extension then apply *znew*.

OPTIONS

- f** Force recompression from .Z to .gz format even if a .gz file already exists.
- t** Tests the new files before deleting originals.
- v** Verbose. Display the name and percentage reduction for each file compressed.
- 9** Use the slowest compression method (optimal compression).
- P** Use pipes for the conversion to reduce disk space usage.
- K** Keep a .Z file when it is smaller than the .gz file

SEE ALSO

gzip(1), *zmore(1)*, *zdiff(1)*, *zgrep(1)*, *zforce(1)*, *gzexe(1)*, *compress(1)*

BUGS

Znew does not maintain the time stamp with the -P option if *cpmod(1)* is not available and *touch(1)* does not support the -r option.

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 input 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 invoking process has appropriate privileges, the ownership, modes, access time, and modification time of the original file are preserved. If appending 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 IEEE Std 1003.1-2001, 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 reduce 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.

-v Write the percentage reduction of each file to standard error.

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 specified, 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 *compress*:

LANG Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables 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).

LC_MESSAGES

Determine the locale that should be used to affect the format and contents of diagnostic messages 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 messages 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 mechanisms) is not required by IEEE Std 1003.1-2001.

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 increased 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 coding or adaptive Huffman coding (*compact*), and takes less time to compute.

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 explicit published file format). Some implementations based on 16-bit architectures cannot support 15 or 16-bit uncompression.

EXAMPLES

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

uncompress , *zcat*

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1, 2003 Edition, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 6, Copyright (C) 2001-2003 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.open-group.org/unix/online.html> .

NAME

`uncompress` – expand compressed data

SYNOPSIS

uncompress [-c**f**v][*file*...]

DESCRIPTION

The *uncompress* utility shall restore files to their original state after they have been compressed using the *compress* utility. If no files are specified, the standard input shall be uncompressed to the standard output. If the invoking process has appropriate privileges, the ownership, modes, access time, and modification time of the original file shall be preserved.

This utility shall support the uncompressing of any files produced by the *compress* utility on the same implementation. For files produced by *compress* on other systems, *uncompress* supports 9 to 14-bit compression (see *compress*, -b); it is implementation-defined whether values of -b greater than 14 are supported.

OPTIONS

The *uncompress* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

The following options shall be supported:

- c Write to standard output; no files are changed.
- f Do not prompt for overwriting files. Except when run in the background, if -f is not given the user shall be prompted as to whether an existing file should be overwritten. If the standard input is not a terminal and -f is not given, *uncompress* shall write a diagnostic message to standard error and exit with a status greater than zero.
- v Write messages to standard error concerning the expansion of each file.

OPERANDS

The following operand shall be supported:

- file* A pathname of a file. If *file* already has the .Z suffix specified, it shall be used as the input file and the output file shall be named **file** with the .Z suffix removed. Otherwise, *file* shall be used as the name of the output file and **file** with the .Z suffix appended shall be used as the input file.

STDIN

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

INPUT FILES

Input files shall be in the format produced by the *compress* utility.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of *uncompress*:

LANG Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables 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).

LC_MESSAGES

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

NLSPATH

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

ASYNCHRONOUS EVENTS

Default.

STDOUT

When there are no *file* operands or the **-c** option is specified, the uncompressed output is written to standard output.

STDERR

Prompts shall be written to the standard error output under the conditions specified in the **DESCRIPTION** and **OPTIONS** sections. The prompts shall contain the *file* pathname, but their format is otherwise unspecified. Otherwise, the standard error output shall be used only for diagnostic messages.

OUTPUT FILES

Output files are the same as the respective input files to *compress*.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

- | | |
|----|------------------------|
| 0 | Successful completion. |
| >0 | An error occurred. |

CONSEQUENCES OF ERRORS

The input file remains unmodified.

The following sections are informative.

APPLICATION USAGE

The limit of 14 on the *compress* **-b** *bits* argument is to achieve portability to all systems (within the restrictions imposed by the lack of an explicit published file format). Some implementations based on 16-bit architectures cannot support 15 or 16-bit uncompression.

EXAMPLES

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

compress , *zcat*

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1, 2003 Edition, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 6, Copyright (C) 2001-2003 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.open-group.org/unix/online.html> .

NAME

zcat – expand and concatenate data

SYNOPSIS

zcat [*file*...]

DESCRIPTION

The *zcat* utility shall write to standard output the uncompressed form of files that have been compressed using the *compress* utility. It is the equivalent of *uncompress -c*. Input files are not affected.

OPTIONS

None.

OPERANDS

The following operand shall be supported:

file The pathname of a file previously processed by the *compress* utility. If *file* already has the **.Z** suffix specified, it is used as submitted. Otherwise, the **.Z** suffix is appended to the filename prior to processing.

STDIN

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

INPUT FILES

Input files shall be compressed files that are in the format produced by the *compress* utility.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of *zcat*:

LANG Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables 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).

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

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

ASYNCHRONOUS EVENTS

Default.

STDOUT

The compressed files given as input shall be written on standard output in their uncompressed form.

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

None.

EXAMPLES

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

compress , *uncompress*

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1, 2003 Edition, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 6, Copyright (C) 2001-2003 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.open-group.org/unix/online.html> .