NAME

compress, uncompress, zcat − compress and expand data

SYNOPSIS

compress [**−f**] [**−k**] [**−v**] [**−c**] [**−V**] [**−r**] [**−b** *bits*] [**−−**] [*name ...*]
uncompress [**−f**] [**−k**] [**−v**] [**−c**] [**−V**] [**−−**] [*name ...*]
zcat [**−V**] [**−−**] [*name ...*]

DESCRIPTION

Compress reduces the size of the named files using adaptive Lempel−Ziv coding. Whenever possible, each file is replaced by one with the extension **.Z,** while keeping the same ownership modes, access and modification times. If no files are specified, the standard input is compressed to the standard output. *Compr ess* will only attempt to compress regular files. In particular, it will ignore symbolic links. If a file has multiple hard links, *compress* will refuse to compress it unless the **−f** flag is given.

If **−f** is not given and *compress* is run in the foreground, the user is prompted as to whether an existing file should be overwritten.

Compressed files can be restored to their original form using *uncompress* or *zcat.*

uncompress takes a list of files on its command line and replaces each file whose name ends with **.Z** and which begins with the correct magic number with an uncompressed file without the **.Z.** The uncompressed file will have the mode, ownership and timestamps of the compressed file.

The **−k** option makes *compress/uncompress* keep the input files instead of automatically removing them.

The **−c** option makes *compress/uncompress* write to the standard output; no files are changed.

zcat is identical to *uncompress* **−c.** *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 ha ve the correct magic number whether they have a **.Z** suffix or not.

If the **−r** flag is specified, *compress* will operate recursively. If any of the file names specified on the command line are directories, *compress* will descend into the directory and compress all the files it finds there. When compressing, any files already compressed will be ignored, and when decompressing, any files already decompressed will be ignored.

The **−V** flag tells each of these programs to print its version and patchlevel, along with any preprocessor flags specified during compilation, on stderr before doing any compression or uncompression.

Compress uses the modified Lempel−Ziv algorithm popularized in "A Technique for High Performance Data Compression", Terry A. Welch, *IEEE Computer,* vol. 17, no. 6 (June 1984), pp. 8−19. Common substrings in the file are first replaced by 9−bit codes 257 and up. When code 512 is reached, the algorithm switches to 10−bit codes and continues to use more bits until the limit specified by the **−b** flag is reached (default 16). *Bits* must be between 9 and 16. The default can be changed in the source to allow *compress* to be run on a smaller machine.

After the *bits* limit is attained, *compress* periodically checks the compression ratio. If it is increasing, *compress* continues to use the existing code dictionary. However, if the compression ratio decreases, *compress* discards the table of substrings and rebuilds it from scratch. This allows the algorithm to adapt to the next "block" of the file.

Note that the **−b** flag is omitted for *uncompress,* since the *bits* parameter specified during compression is encoded within the output, along with a magic number to ensure that neither decompression of random data nor recompression of compressed data is attempted.

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 (as used in *pack*), or adaptive Huffman coding (*compact*), and takes less time to compute.

Under the **−v** option, a message is printed yielding the percentage of reduction for each file compressed.

−− may be used to halt option parsing and force all remaining arguments to be treated as paths.

DIAGNOSTICS

Exit status is normally 0; if the last file is larger after (attempted) compression, the status is 2; if an error occurs, exit status is 1.

Usage: compress [−dfvcVr] [−b maxbits] [file ...]
: Invalid options were specified on the command line.

Missing maxbits
: Maxbits must follow **−b**.

file: not in compressed format
: The file specified to *uncompress* has not been compressed.

file: compressed with *xx* bits, can only handle *yy* bits
: *File* was compressed by a program that could deal with more *bits* than the compress code on this machine. Recompress the file with smaller *bits*.

file: already has .Z suffix −− no change
: The file is assumed to be already compressed. Rename the file and try again.

file: filename too long to tack on .Z
: The file cannot be compressed because its name is longer than 12 characters. Rename and try again. This message does not occur on BSD systems.

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.

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

Compression: *xx.xx%*
: Percentage of the input saved by compression. (Relevant only for **−v**.)

−− 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.

−− file unchanged
: No savings is achieved by compression. The input remains virgin.

BUGS

Although compressed files are compatible between machines with large memory, **−b**12 should be used for file transfer to architectures with a small process data space (64KB or less, as exhibited by the DEC PDP series, the Intel 80286, etc.)

SEE ALSO

pack(1), **compact**(1)