

**NAME**

**compress**, **uncompress** - compress and expand data

**SYNOPSIS**

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

**compress -c** [-b *bits*] [*file*]

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

**uncompress -c** [*file*]

**DESCRIPTION**

The **compress** utility reduces the size of files using adaptive Lempel-Ziv coding. Each *file* is renamed to the same name plus the extension *.Z*. A *file* argument with a *.Z* extension will be ignored except it will cause an error exit after other arguments are processed. If compression would not reduce the size of a *file*, the file is ignored.

The **uncompress** utility restores compressed files to their original form, renaming the files by deleting the *.Z* extensions. A file specification need not include the file's *.Z* extension. If a file's name in its file system does not have a *.Z* extension, it will not be uncompressed and it will cause an error exit after other arguments are processed.

If renaming the files would cause files to be overwritten and the standard input device is a terminal, the user is prompted (on the standard error output) for confirmation. If prompting is not possible or confirmation is not received, the files are not overwritten.

As many of the modification time, access time, file flags, file mode, user ID, and group ID as allowed by permissions are retained in the new file.

If no files are specified or a *file* argument is a single dash ('-'), the standard input is compressed or uncompressed to the standard output. If either the input and output files are not regular files, the checks for reduction in size and file overwriting are not performed, the input file is not removed, and the attributes of the input file are not retained in the output file.

The options are as follows:

- b** *bits*    The code size (see below) is limited to *bits*, which must be in the range 9..16. The default is 16.
- c**        Compressed or uncompressed output is written to the standard output. No files are modified. The **-v** option is ignored. Compression is attempted even if the results will be larger than the original.

- f** Files are overwritten without prompting for confirmation. Also, for **compress**, files are compressed even if they are not actually reduced in size.
- v** Print the percentage reduction of each file. Ignored by **uncompress** or if the **-c** option is also used.

The **compress** utility uses a modified Lempel-Ziv algorithm. 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** option or its default is reached.

After the limit is reached, **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.

The **-b** option is unavailable 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 the historical command pack), or adaptive Huffman coding (as used in the historical command compact), and takes less time to compute.

If *file* is a soft or hard link **compress** will replace it with a compressed copy of the file pointed to by the link. The link's target file is left uncompressed.

## EXIT STATUS

The **compress** and **uncompress** utilities exit 0 on success, and >0 if an error occurs.

The **compress** utility exits 2 if attempting to compress a file would not reduce its size and the **-f** option was not specified and if no other error occurs.

## EXAMPLES

Create a file *test\_file* with a single line of text:

```
echo "This is a test" > test_file
```

Try to reduce the size of the file using a 10-bit code and show the exit status:

```
$ compress -b 10 test_file
$ echo $?
2
```

Try to compress the file and show compression percentage:

```
$ compress -v test_file
test_file: file would grow; left unmodified
```

Same as above but forcing compression:

```
$ compress -f -v test_file
test_file.Z: 79% expansion
```

Compress and uncompress the string 'hello' on the fly:

```
$ echo "hello" | compress | uncompress
hello
```

## SEE ALSO

gunzip(1), gzexe(1), gzip(1), zcat(1), zmore(1), znew(1)

Welch, Terry A., "A Technique for High Performance Data Compression", *IEEE Computer*, 17:6, pp. 8-19, June, 1984.

## STANDARDS

The **compress** and **uncompress** utilities conform to IEEE Std 1003.1-2001 ("POSIX.1").

## HISTORY

The **compress** command appeared in 4.3BSD.

## BUGS

The program does not handle links well and has no link-handling options.

Some of these might be considered otherwise-undocumented features.

**compress**: If the utility does not compress a file because doing so would not reduce its size, and a file of the same name except with an *.Z* extension exists, the named file is not really ignored as stated above; it causes a prompt to confirm the overwriting of the file with the extension. If the operation is confirmed, that file is deleted.

**uncompress**: If an empty file is compressed (using **-f**), the resulting .Z file is also empty. That seems right, but if **uncompress** is then used on that file, an error will occur.

Both utilities: If a '-' argument is used and the utility prompts the user, the standard input is taken as the user's reply to the prompt.

Both utilities: If the specified file does not exist, but a similarly-named one with (for **compress**) or without (for **uncompress**) a .Z extension does exist, the utility will waste the user's time by not immediately emitting an error message about the missing file and continuing. Instead, it first asks for confirmation to overwrite the existing file and then does not overwrite it.