Table of Contents

$ bzip2

Beginner

Compress files using Burrows-Wheeler block sorting

bzip2 compresses files using the Burrows-Wheeler algorithm, typically producing better compression than gzip but running slower. Compressed files get the .bz2 extension.

bzip2 offers significantly better compression ratios than gzip, typically 10-20% smaller files, at the cost of being 2-3x slow...

Options & Flags

`-d`	Decompress
`-k`	Keep original file
`-1 to -9`	Compression level
`-z`	Force compression
`-v`	Verbose
`-c`	Write to stdout

Practical Examples

Example: Compress a file

```
$ bzip2 large_dump.sql
```

Compresses the file, replacing it with large_dump.sql.bz2.

Example: Decompress

```
$ bzip2 -d file.bz2
```

Decompresses back to the original. Or use bunzip2.

Example: Keep original

```
$ bzip2 -k data.csv
```

Compresses while keeping the original file.

Example: Create tar.bz2

```
$ tar cjf archive.tar.bz2 /data/
```

Creates a bzip2-compressed tar archive.

Example: Maximum compression

```
$ bzip2 -9 distribution.tar
```

Uses best compression for the smallest output.

Tips & Best Practices

Note: gzip vs bzip2 vs xz: Speed: gzip > bzip2 > xz. Compression: xz > bzip2 > gzip. For daily backups use gzip. For distribution use xz.

Warning: Slow decompression: bzip2 decompression is also slower than gzip. For files accessed frequently, gzip may be more practical.

Pro Tip: pbzip2 for parallel: Install pbzip2 for multi-threaded bzip2 compression. Dramatically faster on multi-core systems.

$ gunzip

Beginner

Decompress .gz files

gunzip decompresses files compressed with gzip. It is equivalent to gzip -d. gunzip replaces the .gz file with the decompressed original by default.

gunzip can also decompress files created by compress (.Z files) and pack (.z files). It handles multiple files and supports keeping the original co...

Options & Flags

`-k`	Keep compressed file
`-f`	Force decompression (overwrite existing)
`-r`	Recursively decompress
`-l`	List compressed file info
`-v`	Verbose output
`-c`	Write to stdout (keep original)

Practical Examples

Example: Decompress file

```
$ gunzip access.log.gz
```

Decompresses to access.log, removing the .gz file.

Example: Keep original

```
$ gunzip -k backup.sql.gz
```

Decompresses while keeping the .gz file.

Example: Decompress to stdout

```
$ gunzip -c data.csv.gz | head -10
```

Decompresses and shows first 10 lines without creating a file.

Example: Decompress all in directory

```
$ gunzip -r /var/log/old/
```

Decompresses all .gz files recursively.

Example: Force overwrite

```
$ gunzip -f file.gz
```

Decompresses even if the output file already exists.

Tips & Best Practices

Pro Tip: zcat for viewing: Use zcat file.gz to view contents without decompressing. Or zless for paging, zgrep for searching.

Note: gunzip = gzip -d: gunzip is exactly the same as gzip -d. Use whichever you remember.

Warning: Replaces .gz file: gunzip removes the .gz file by default. Use -k to keep it.

$ gzip

Beginner

Compress files using Lempel-Ziv coding

gzip compresses files using the Lempel-Ziv coding algorithm. It is the most common compression tool in Linux, typically producing .gz files. gzip replaces the original file with the compressed version by default.

gzip offers a good balance of compression speed and ratio. It is the standard compr...

Options & Flags

`-d`	Decompress
`-k`	Keep original file
`-r`	Recursively compress files in directory
`-1 to -9`	Compression level (1=fast, 9=best)
`-l`	List compressed file info
`-v`	Verbose - show compression ratio
`-c`	Write to stdout (keep original)

Practical Examples

Example: Compress a file

```
$ gzip access.log
```

Compresses the file, replacing it with access.log.gz.

Example: Decompress

```
$ gzip -d access.log.gz
```

Decompresses back to access.log. Or use gunzip.

Example: Keep original

```
$ gzip -k large_file.txt
```

Compresses while keeping the original file.

Example: Best compression

```
$ gzip -9 database.sql
```

Uses maximum compression. Slower but smallest file.

Example: Check compressed file

```
$ gzip -l backup.sql.gz
compressed  uncompressed  ratio  name\n    12345       98765    87.5%  backup.sql
```

Shows compressed/uncompressed sizes and ratio.

Tips & Best Practices

Warning: Original file is replaced: gzip replaces the original file by default. Use -k to keep it, or gzip -c file > file.gz to write to stdout.

Pro Tip: Use with pipes: gzip works great in pipes: command | gzip > output.gz for on-the-fly compression.

Note: gzip vs xz vs bzip2: gzip is fastest. xz has best compression ratio. bzip2 is in between. For daily use and logs, gzip is recommended.

$ tar

Beginner

Archive files using tape archive format

tar (tape archive) creates and extracts archive files. While tar itself does not compress, it is almost always combined with compression: tar.gz (gzip), tar.bz2 (bzip2), tar.xz (xz). tar is the standard archive format for Linux backup and distribution.

tar preserves file permissions, ownership, ...

Options & Flags

`-c`	Create archive
`-x`	Extract archive
`-z`	Compress/decompress with gzip
`-j`	Compress with bzip2
`-J`	Compress with xz
`-f`	Specify archive filename
`-v`	Verbose - list files processed
`-t`	List archive contents
`-C`	Change to directory before operation
`--exclude`	Exclude files matching pattern

Practical Examples

Example: Create gzipped archive

```
$ tar czf backup.tar.gz /home/user/
```

Creates a compressed archive of the home directory.

Example: Extract archive

```
$ tar xf archive.tar.gz
```

Extracts the archive. Modern tar auto-detects compression.

Example: Extract to specific directory

```
$ tar xf release.tar.gz -C /opt/
```

Extracts the archive into /opt/ directory.

Example: List contents

```
$ tar tf archive.tar.gz
dir/file1.txt\ndir/file2.txt
```

Lists all files in the archive without extracting.

Example: Exclude files

```
$ tar czf deploy.tar.gz --exclude='node_modules' --exclude='.git' --exclude='*.log' ./project/
```

Creates archive excluding unnecessary directories and logs.

Tips & Best Practices

Pro Tip: Remember: c-create, x-extract, t-list: tar czf = create gzip file. tar xf = extract file. tar tf = list (tell) file. f always comes last before filename.

Warning: Always list before extract: Use tar tf archive first to check contents. Some archives extract files without a top-level directory, cluttering your current directory.

Note: Compression comparison: Speed: gzip > bzip2 > xz. Compression ratio: xz > bzip2 > gzip. gzip is the default for most uses. xz for distribution.

$ unzip

Beginner

Extract files from ZIP archives

unzip extracts files from ZIP archives. It supports listing contents, extracting specific files, testing archive integrity, and handling password-protected archives.

unzip is the complement to zip - it extracts what zip creates. It handles ZIP files from any platform (Windows, macOS, Linux) and ...

Options & Flags

`-l`	List archive contents
`-d`	Extract to specific directory
`-o`	Overwrite without prompting
`-n`	Never overwrite existing files
`-t`	Test archive integrity
`-j`	Ignore directory structure (extract flat)
`-P`	Provide password on command line

Practical Examples

Example: Extract archive

```
$ unzip archive.zip
```

Extracts all files from the archive.

Example: Extract to directory

```
$ unzip release.zip -d /opt/app/
```

Extracts to a specific directory.

Example: List contents

```
$ unzip -l archive.zip
  Length  Date    Time   Name\n  12345  01-15-24 14:30  file.txt
```

Lists all files without extracting.

Example: Test integrity

```
$ unzip -t archive.zip
No errors detected in compressed data of archive.zip.
```

Verifies archive is not corrupted.

Example: Extract specific file

```
$ unzip archive.zip path/to/file.txt
```

Extracts only the specified file from the archive.

Tips & Best Practices

Pro Tip: Always list first: Use unzip -l first to check contents. Some archives extract directly into the current directory without a subdirectory.

Note: Wildcard extraction: Extract specific types: unzip archive.zip '*.txt' extracts only .txt files.

Warning: Password on command line: Using -P password is insecure (visible in process list). Let unzip prompt for the password instead.

$ xz

Intermediate

Compress files using LZMA2 algorithm (best compression)

xz compresses files using the LZMA2 algorithm, providing the best compression ratios of the common Linux compression tools. xz-compressed files use the .xz extension.

xz produces significantly smaller files than gzip or bzip2, but at the cost of higher CPU usage and longer compression time. Deco...

Options & Flags

`-d`	Decompress
`-k`	Keep original file
`-0 to -9`	Compression level (0=fast, 9=best)
`-T`	Number of threads
`-v`	Verbose
`-c`	Write to stdout
`-l`	List compressed file info

Practical Examples

Example: Compress a file

```
$ xz large_backup.tar
```

Compresses with best default settings. Creates large_backup.tar.xz.

Example: Parallel compression

```
$ xz -T$(nproc) -9 huge_file.tar
```

Uses all CPU cores for faster compression.

Example: Decompress

```
$ xz -d archive.tar.xz
```

Decompresses back to archive.tar. Or use unxz.

Example: Create tar.xz

```
$ tar cJf release.tar.xz source/
```

Creates an xz-compressed tar archive.

Example: Keep original

```
$ xz -k database.sql
```

Compresses while keeping the original.

Tips & Best Practices

Pro Tip: Use -T for multi-threading: xz -T0 uses all available cores. This dramatically speeds up compression: xz -T0 -9 large_file.

Note: Best compression ratio: xz typically compresses 20-30% better than gzip and 10% better than bzip2, at the cost of speed.

Warning: High memory usage: xz -9 uses significant memory (~674MB). Lower levels use less. Ensure sufficient RAM for the chosen level.

$ zcat

Beginner

View contents of compressed files without decompressing

zcat displays the contents of gzip-compressed files without decompressing them to disk. It is equivalent to gzip -dc (decompress to stdout). zcat is the compressed equivalent of cat.

zcat is essential for viewing, piping, and processing compressed files without creating temporary decompressed co...

Options & Flags

`file.gz`	Display decompressed contents
`multiple`	Concatenate multiple compressed files
`with pipe`	Pipe to other commands
`-f`	Force (handle non-gz files too)

Practical Examples

Example: View compressed file

```
$ zcat /var/log/syslog.1.gz
```

Displays the contents of a compressed log file.

Example: Search in compressed file

```
$ zcat access.log.gz | grep "404"
```

Searches for 404 errors in a compressed log without decompressing.

Example: Count lines

```
$ zcat data.csv.gz | wc -l
500000
```

Counts lines in a compressed file without decompressing to disk.

Example: Head of compressed file

```
$ zcat large_file.gz | head -20
```

Shows just the first 20 lines of a compressed file.

Example: Process compressed CSV

```
$ zcat data.csv.gz | awk -F',' '{print $1, $3}'
```

Extracts specific columns from a compressed CSV file.

Tips & Best Practices

Pro Tip: Use zgrep, zless for common tasks: zgrep searches directly in .gz files. zless pages through them. Both are more convenient than zcat | grep or zcat | less.

Note: zcat vs bzcat vs xzcat: zcat for .gz files, bzcat for .bz2 files, xzcat for .xz files. Each handles its respective compression format.

Warning: Large output: zcat outputs the entire decompressed file to stdout. For large files, always pipe to head, less, or grep.

$ zip

Beginner

Package and compress files into ZIP archives

zip creates compressed archive files in the ZIP format, widely compatible across Windows, macOS, and Linux. Unlike tar+gzip, zip combines archiving and compression in one step.

zip is the best choice when creating archives that need to be opened on Windows or macOS, as ZIP is universally support...

Options & Flags

`-r`	Recursively include directories
`-e`	Encrypt with password
`-9`	Maximum compression
`-x`	Exclude files matching pattern
`-u`	Update existing archive
`-d`	Delete files from archive
`-j`	Junk (strip) directory paths
`-s`	Split into parts of specified size

Practical Examples

Example: Create zip archive

```
$ zip -r project.zip project/
```

Creates a zip file from a directory recursively.

Example: Zip with exclusions

```
$ zip -r deploy.zip app/ -x '*.git*' 'node_modules/*' '*.log'
```

Creates archive excluding git files, node_modules, and logs.

Example: Password-protected zip

```
$ zip -e sensitive.zip report.pdf data.xlsx
```

Creates an encrypted archive that requires a password to open.

Example: Maximum compression

```
$ zip -9r archive.zip documents/
```

Uses maximum compression for smallest file size.

Example: Update archive

```
$ zip -u archive.zip updated_file.txt
```

Adds or replaces a file in an existing archive.

Tips & Best Practices

Pro Tip: Use for cross-platform sharing: zip is the best format when sharing with Windows/macOS users. Use tar.gz for Linux-only archives.

Warning: zip encryption is weak: Standard zip encryption (ZipCrypto) is weak. For secure encryption, use 7z or GPG instead.

Note: -x exclude syntax: Exclude patterns need quoting: zip -r arch.zip dir/ -x '*.log' '*.tmp' 'dir/cache/*'