Archiving and compressing with zip

ZIP is a popular compression format used on many platforms. It isn't as commonly used as `gzip` or `bzip2` on Linux platforms, but files from the Internet are often saved in this format. In this recipe we will see how to use `zip` to perform compression and extraction.

How to do it...

Let's see how to use various options with `zip`:

1. In order to archive with ZIP, the following syntax is used:

   ```
   $ zip archive_name.zip [SOURCE FILES/DIRS]
   ```

   For example:

   ```
   $ zip file.zip file
   ```

   Here, the `file.zip` file will be produced.

2. Archive directories and files recursively as follows:

   ```
   $ zip -r archive.zip folder1 folder2
   ```

   In this command, `-r` is used for specifying recursive.

3. In order to extract files and folders in a ZIP file, use:

   ```
   $ unzip file.zip
   ```

It will extract the files without removing `filename.zip` (unlike `unlzma` or `gunzip`).

1. In order to update files in the archive with newer files in the filesystem, use the `-u` flag:

   ```
   $ zip file.zip -u newfile
   ```

2. Delete a file from a zipped archive, by using `-d` as follows:

   ```
   $ zip -d arc.zip file.txt
   ```

3. In order to list the files in an archive use:

   ```
   $ unzip -l archive.zip
   ```

How it works...

While being similar to most of the archiving and compression tools we have already discussed, `zip` unlike `lzma`, `gzip`, or `bzip2` won't remove the source file after archiving. Most importantly, while `zip` is similar to `tar`, it performs both archiving and compression while `tar` by itself does not perform compression.

Faster archiving with pbzip2

Most modern computers today are equipped with at least two CPU cores - for the user it means almost the same as two real CPUs doing your work. However, just having a multicore CPU doesn't mean your programs will run faster, it is important that the programs themselves have been designed to run faster on multicore processors.

Most of the compression commands that we saw up to now will use only one CPU and, hence, won't be very fast. `pbzip2` can use multiple cores, hence decreasing overall time taken to compress your files.

Getting ready

`pbzip2` usually doesn't come preinstalled with most distros, you will have to use your package manager to install it.

How to do it...

Let's see how to use `pbzip2` to compress files and extract them:

1. Compress a single file like this:

   ```
   pbzip2 myfile.tar
   ```

   pbzip2 will automatically detect the number of cores on your system and compress myfile.tar, to myfile.tar.bz2

2. To compress and archive multiple files or directories, we use `pbzip2` in combination with tar as follows:

   ```
   tar cf myfile.tar.bz2 --use-compress-prog=pbzip2 dir_to_compress/
   ```

   Or:

   ```
   tar -c directory_to_compress/ | pbzip2 -c > myfile.tar.bz2
   ```

3. Extracting a pbzip2'd file

   If it's a tar.bz2 file, we can perform the decompression and extraction in one step:

   ```
   pbzip2 -dc myfile.tar.bz2 | tar x
   ```

   If the archive is a single file which was pbzip2'd, use this:

   ```
   pbzip2 -d myfile.tar.bz2
   ```