Compressing data with gzip

gzip is a commonly used compression format in the GNU/Linux platform. It is one of the utilities (such as `gzip`, `gunzip`, and `zcat`) that handle gzip compression. However, `gzip` can be applied only on a single file or data stream. This means that it cannot archive directories and multiple files. Hence, we must first create a `tar` archive and compress it with `gzip`. Let's see how to operate with `gzip`.

How to do it...

`gzip` can be used both to compress files and decompress them back to the original:

1. In order to compress a file with `gzip` use the following command:

   ```
   $ gzip filename
   $ ls
   filename.gz
   ```

2. Extract a `gzip` compressed file as follows:

   ```
   $ gunzip filename.gz
   $ ls
   file
   ```

3. In order to list out the properties of a compressed file use:

   ```
   $ gzip -l test.txt.gz
   compressed        uncompressed  ratio uncompressed_name
     35                     6 -33.3% test.txt
   ```

4. The `gzip` command can read a file from `stdin` and also write a compressed file into `stdout`.

   Read data from `stdin` and output the compressed data to `stdout` as follows:

   ```
   $ cat file | gzip -c > file.gz
   ```

   The `-c` option is used to specify output to `stdout`.

5. We can specify the compression level for `gzip` using `--fast` or the`--best` option to provide low and high compression ratios, respectively.

There's more...

The gzip command is often used with other commands and also has advanced options to specify the compression ratio. Let's see how to work with these features.

Gzip with tarball

A gzipped tarball is basically a tar archive compressed using gzip. We can use two methods to create such tarballs:

- The first method is as follows:

  ```
  $ tar -czvvf archive.tar.gz [FILES]
  ```

  or

  ```
  $ tar -cavvf archive.tar.gz [FILES]
  ```

  The -a option specifies that the compression format should automatically be detected from the extension.

- Alternatively, here's the second method:

  First, create a tarball:

  ```
  $ tar -cvvf archive.tar [FILES]
  ```

  Compress the tarball as follows:

  ```
  $ gzip archive.tar
  ```

If many files (a few hundreds) are to be archived in a tarball and need to be compressed, we use the second method with few changes. The issue with giving many files as command arguments to tar is that it can accept only a limited number of files from the command line. In order to solve this issue, we can create a tar file by adding files one by one using a loop with an append option (-r) as follows:

```
FILE_LIST="file1  file2  file3  file4  file5"

for f in $FILE_LIST;
do
tar -rvf archive.tar $f
done

gzip archive.tar
```