It is also possible to exclude a list of files provided in a list file with the `-X` flag as follows:

```
$ cat list
filea
fileb

$ tar -cf arch.tar * -X list
```

Now it excludes `filea` and `fileb` from archiving.

Excluding version control directories

We usually use tarballs for distributing source code. In general, most source code is maintained using version control systems such as subversion, Git, mercurial, cvs, and so on. Code directories under version control will contain special directories used to manage versions like `.svn` or `.git`. However, these directories aren't needed by the code itself and so should be eliminated from the tarball of the source code.

In order to exclude version control related files and directories while archiving use the `--exclude-vcs` option along with `tar`. For example:

```
$ tar --exclude-vcs -czvvf source_code.tar.gz eye_of_gnome_svn
```

Printing total bytes

It is sometimes useful if we can print total bytes copied to the archive. To print the total bytes copied after archiving use the --`totals` option as follows:

```
$ tar -cf arc.tar * --exclude "*.txt" --totals
Total bytes written: 20480 (20KiB, 12MiB/s)
```

See also

- *Compressing with gzip*, explains the gzip command

Archiving with cpio

cpio is another archiving format similar to `tar`. It is used to store files and directories in a file with attributes such as permissions, ownership, and so on. But, it is not commonly used as much as `tar`. However, `cpio` is used in RPM package archives (which are used in distros such as Fedora), initramfs files for the Linux kernel which contain the kernel image, and so on. This recipe will give minimal usage examples of `cpio`.

How to do it...

`cpio` takes input filenames through `stdin` and it writes the archive into `stdout`. We have to redirect `stdout` to a file to receive the output `cpio` file as follows:

1. Create test files:

   ```
   $ touch file1 file2 file3
   ```

2. We can archive the test files as follows:

   ```
   $ echo file1 file2 file3 | cpio -ov > archive.cpio
   ```

3. In order to list files in a `cpio` archive use the following command:

   ```
   $ cpio -it < archive.cpio
   ```

4. In order to extract files from the `cpio` archive use:

   ```
   $ cpio -id < archive.cpio
   ```

How it works...

For the archiving command:

- `-o` specifies the output
- `-v` is used for printing a list of files archived

By using `cpio`, we can also archive using files as absolute paths. `/usr/somedir` is an absolute path as it contains the full path starting from root (`/`).

A relative path will not start with `/` but it starts the path from the current directory. For example, `test/file` means that there is a directory `test` and the `file` is inside the `test` directory.

While extracting, `cpio` extracts to the absolute path itself. But in case of `tar` it removes the `/` in the absolute path and converts it as a relative path.

In the command for listing all the files in the given `cpio` archive:

- `-i` is for specifying the input
- `-t` is for listing

While using the command for extraction, `-d` stands for extracting and `cpio` overwrites files without prompting.