

2. Generate bulk files with a different name pattern as follows:

```
for name in {1..100}.txt
do
    touch $name
done
```

In the preceding code `{1..100}` will be expanded as a string "1, 2, 3, 4, 5, 6, 7...100". Instead of `{1..100}.txt`, we can use various shorthand patterns such as `test{1..200}.c`, `test{a..z}.txt`, and so on.

If a file already exists, the `touch` command changes all timestamps associated with the file to the current time. However, if we want to specify that only certain stamps are to be modified, we use the following options:

- ❑ `touch -a` modifies only the access time
- ❑ `touch -m` modifies only the modification time

3. Instead of using the current time for the timestamp, we can specify the time and date with which to stamp the file as follows:

```
$ touch -d "Fri Jun 25 20:50:14 IST 1999" filename
```

The date string that is used with `-d` need not always be in the same format. It will accept any simple date formats. We can omit time from the string and provide handy date formats such as "Jan 20 2010".

## Finding symbolic links and their targets

Symbolic links are very common in Unix-like systems. There are various reasons you want to use them, ranging from convenient access to maintaining different versions of the same library or program. This recipe will discuss the basic techniques of handling symbolic links.

Symbolic links are just pointers to other files, they are similar in function to aliases in Mac OS X or shortcuts in Windows. When symbolic links are removed, they will not cause any harm to the original file.

### How to do it...

The following steps will help you handle symbolic links:

1. We can create a symbolic link as follows:

```
$ ln -s target symbolic_link_name
```

For example:

```
$ ln -l -s /var/www/ ~/web
```

This creates a symbolic link (called **web**) in the current user's home directory, which points to `/var/www/`.

2. To verify that the link was created, run the following command:

```
$ ls -l web
lrwxrwxrwx 1 slynux slynux 8 2010-06-25 21:34 web -> /var/www

web -> /var/www specifies that web points to /var/www.
```

3. In order to print symbolic links in the current directory, use the following command:

```
$ ls -l | grep "^l"
```

4. Use `find` to print all symbolic links from the current directory and subdirectories:

```
$ find . -type l -print
```

5. To read the target path for a given symbolic link, use the `readlink` command:

```
$ readlink web
/var/www
```

### How it works...

When looking for symbolic links in the current directory, `grep` will filter the lines from the `ls -l` output such that it displays only lines starting with `l` using `^`, which is the start marker for the string. This utilizes the fact that for every symbolic link, the permission notation block (`lrwxrwxrwx`) starts with letter `l`.

While using `find`, we use the argument `type` with `l`, which will instruct the command to search only for symbolic link files. The `-print` option is used to print the list of symbolic links to the standard output (`stdout`). The path from which the file search should begin is given as `.` which means it is the current directory.

## Enumerating file type statistics

There are many file types. It will be an interesting exercise to write a script that can enumerate through all the files inside a directory, its descendants, print a report that provides details on types of files (files with different file types), and the count of each file type present. This recipe is an exercise on how to write scripts that can enumerate through many files and collect details.