

6. To revert the changes back, use the following command:

```
$ patch -p1 version1.txt < version.patch
patching file version1.txt
Reversed (or previously applied) patch detected! Assume -R? [n] y
#Changes are reverted.
```

As shown, patching an already patched file reverts back the changes. To avoid prompting the user with `y/n`, we can use the `-R` option along with the `patch` command.

There's more...

Let's go through additional features available with `diff`.

Generating difference against directories

The `diff` command can also act recursively against directories. It will generate a difference output for all the descendant files in the directories. Use the following command:

```
$ diff -Naur directory1 directory2
```

The interpretation of each of the previous options is as follows:

- ▶ `-N` is for treating absent files as empty
- ▶ `-a` is to consider all files as text files
- ▶ `-u` is to produce unified output
- ▶ `-r` is to recursively traverse through the files in the directories

Using head and tail for printing the last or first 10 lines

When looking into a large file, which consists of thousands of lines, we will not use a command such as `cat` to print the entire file contents. Instead we look for a sample (for example, the first 10 lines of the file or the last 10 lines of the file). We may need to print the first n lines or last n lines and even print all the lines except the last n lines or all lines except first n lines.

Another use case is to print lines from m th to n th lines.

The commands `head` and `tail` can help us do this.

How to do it...

The `head` command always reads the header portion of the input file.

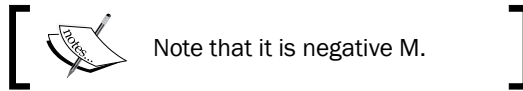
1. Print the first 10 lines as follows:
`$ head file`
2. Read the data from `stdin` as follows:
`$ cat text | head`
3. Specify the number of first lines to be printed as follows:

```
$ head -n 4 file
```

This command prints four lines.

4. Print all lines excluding the last `M` lines as follows:

```
$ head -n -M file
```



For example, to print all the lines except the last five lines, use the following command line:

```
$ seq 11 | head -n -5
```

```
1
2
3
4
5
6
```

The following command will, however, print from 1 to 5:

```
$ seq 100 | head -n 5
```

5. Printing by excluding the last lines is a very important usage of `head`. Now, let us see how to print, last few lines. Print the last 10 lines of a file as follows:

```
$ tail file
```

6. In order to read from `stdin`, you can use the following command line:

```
$ cat text | tail
```