See also

- The *Column-wise cutting of the file with cut* recipe in this chapter explains how to extract data from text files

Printing the nth word or column in a file or line

We may have a file having a number of columns, and only a few will actually be useful. For example, in a list of students in an order of their scores, we want to get, for instance, the fourth highest scorer. In this recipe, we will see how to do this.

How to do it...

The most widely-used method is to use `awk` for doing this task. It can be also done using `cut`.

1. To print the fifth column, use the following command:

```
$ awk '{ print $5 }' filename
```

2. We can also print multiple columns and insert our custom string in between columns.

   For example, to print the permission and filename of each file in the current directory, use the following set of commands:

```
$ ls -l | awk '{ print $1 " :  " $8 }'
-rw-r--r-- :  delimited_data.txt
-rw-r--r-- :  obfuscated.txt
-rw-r--r-- :  paste1.txt
-rw-r--r-- :  paste2.txt
```

See also

- The *Using awk for advanced text processing* recipe in this chapter explains the `awk` command
- The *Column-wise cutting of the file with cut* recipe in this chapter explains how to extract data from text files

Printing text between line numbers or patterns

We may require to print a certain section of text lines, based on conditions such as a range of line numbers, and a range matched by a start and end pattern. Let's see how to do it.

Getting ready

We can use utilities such as `awk`, `grep`, and `sed` to perform the printing of a section based on conditions. Still, I found `awk` to be the simplest one to understand. Let's do it using `awk`.

How to do it...

1. To print the lines of a text in a range of line numbers, `M` to `N`, use the following syntax:

   ```
   $ awk 'NR==M, NR==N' filename
   ```

   Or, it can take the `stdin` input as follows:

   ```
   $ cat filename | awk 'NR==M, NR==N'
   ```

2. Replace `M` and `N` with numbers as follows:

   ```
   $ seq 100 | awk 'NR==4,NR==6'
   4
   5
   6
   ```

3. To print the lines of a text in a section with `start_pattern` and `end_pattern`, use the following syntax:

   ```
   $ awk '/start_pattern/, /end _pattern/' filename
   ```

   For example:

   ```
   $ cat section.txt
   line with pattern1
   line with pattern2
   line with pattern3
   line end with pattern4
   line with pattern5

   $ awk '/pa.*3/, /end/' section.txt
   line with pattern3
   line end with pattern4
   ```

   The patterns used in `awk` are regular expressions.