

- ▶ **Difference:** The difference operation will print the lines that the specified files contain and that are not the same in all of those files
- ▶ **Set difference:** The set difference operation will print the lines in file "A" that do not match those in all of the set of files specified ("B" plus "C" for example)

How to do it...

Note that `comm` takes only sorted files as input. Take a look at the following example:

```
$ cat A.txt
apple
orange
gold
silver
steel
iron
```

```
$ cat B.txt
orange
gold
cookies
carrot
```

```
$ sort A.txt -o A.txt ; sort B.txt -o B.txt
```

1. First, execute `comm` without any options:

```
$ comm A.txt B.txt
apple
      carrot
      cookies
           gold
iron
           orange
silver
steel
```

The first column of the output contains lines that are only in `A.txt`. The second column contains lines that are only in `B.txt`. The third column contains the common lines from `A.txt` and `B.txt`. Each of the columns are delimited using the tab (`\t`) character.

2. In order to print the intersection of two files, we need to remove the first and second columns and print the third column only as follows:

```
$ comm A.txt B.txt -1 -2
gold
orange
```

3. Print lines that are uncommon in two files as follows:

```
$ comm A.txt B.txt -3
apple
      carrot
      cookies
iron
silver
steel
```

In this output, columns have their fields blank for each of the unique lines. Hence, both columns will not have the content on the same line. In order to make it more usable, we need to remove the blank fields and make two columns into a single-column output as follows:

```
apple
carrot
cookies
iron
silver
steel
```

4. In order to produce a unified output, use the following command line:

```
$ comm A.txt B.txt -3 | sed 's/^\t//'
```

```
apple
carrot
cookies
iron
silver
steel
```