5. By removing the unnecessary columns, we can produce the set difference for A.txt and B.txt as follows:

    - Set difference for A.txt:

      ```
      $ comm A.txt B.txt -2 -3
      ```

      -2 -3 removes the second and third columns.

    - Set difference for B.txt:

      ```
      $ comm A.txt B.txt -1 -3
      ```

      -2 -3 removes the second and third columns.

How it works...

The command-line options for comm format the output as per our requirement. These are:

- -1 – removes the first column from the output
- -2 – removes the second column
- -3 – removes the third column

While creating a unified output, the sed command is piped to the comm output. The sed removes the \t character at the beginning of the lines. s in the sed script stands for substitute. /^\t/ matches the \t character at the beginning of the lines (^ is the start of the line marker). // (no character) is the replacement string for every \t character at the beginning of the line. Hence, every \t at the start of the line gets removed.

The set difference operation enables you to compare two files and print all the lines that are in the file A.txt or B.txt excluding the common lines in A.txt and B.txt. When A.txt and B.txt are given as arguments to the comm command, the output will contain column-1 with the set difference for A.txt with regard to B.txt and column-2 will contain the set difference for B.txt with regard to A.txt.

Finding and deleting duplicate files

Duplicate files are copies of the same files. In some circumstances, we may need to remove duplicate files and keep a single copy of them. Identification of duplicate files by looking at the file content is an interesting task. It can be done using a combination of shell utilities. This recipe deals with finding duplicate files and performing operations based on the result.

Getting ready

We can identify the duplicate files by comparing file content. Checksums are ideal for this task, since files with exactly the same content will produce the same checksum values. We can use this fact to remove duplicate files.

How to do it...

1. Generate some test files as follows:

   ```
   $ echo "hello" > test ; cp test test_copy1 ; cp test test_copy2;
   $ echo "next" > other;
   # test_copy1 and test_copy2 are copy of test
   ```

2. The code for the script to remove the duplicate files is as follows:

   ```
   #!/bin/bash
   #Filename: remove_duplicates.sh
   #Description:  Find and remove duplicate files and keep one sample
   of each file.

   ls -lS --time-style=long-iso | awk 'BEGIN {
     getline; getline;
     name1=$8; size=$5
   }
   {
     name2=$8;
     if (size==$5)
     {
       "md5sum "name1 | getline; csum1=$1;
       "md5sum "name2 | getline; csum2=$1;
       if ( csum1==csum2 )
       {
         print name1; print name2
       }
     };

     size=$5; name1=name2;
   }' | sort -u > duplicate_files


   cat duplicate_files | xargs -I {} md5sum {} | sort | uniq -w 32 |
   awk '{ print "^"$2"$" }' | sort -u >  duplicate_sample

   echo Removing..
   ```