www.photocase.com

The Diffutils tool collection

VIVE LA DIFFÉRENCE

The Diffutils tool set helps you compare text files, discover and display the differences between files, and even automatically synchronize files. **BY HEIKE JURZIK**

The Diffutils are a collection of useful tools. Diff compares two text files, outputting the differences in the command line, and storing suggested changes in a patch file, which you can use to automatically modify a file, if needed. If you need to compare three text files, Diff3 is the right tool for you. Read on for more tips and tricks.

A Different View

The Diff program compares two text files (or two files from different directories) line by line and outputs the differences, if any, in the command line. To find out if there are any differences between two files, you can launch the program with the *-q* parameter and the names of the text files:

```
$ diff -q file1 file2
Files file1 and file2 differ
```

If you are interested in the difference, just leave out the option, and Diff will write its findings to the shell (Figure 1). Besides the lines that differ, you will also see a slightly cryptic reference that tells you what you need to change to ensure that the files are absolutely identical.

The suggested changes always follow the same pattern: besides the line numbers or range (line numbers separated by commas), you will see one of three letters, *a*, *c*, or *d*, where *a* stands for "append", *c* for "change", and *d* for "delete". The following line

```
2,3c2,3
```

in Figure 1 means that you will have to modify lines 2 and 3 in the first or second file to synchronize the files. Diff would use an expression such as

```
3a,13,15
```

to tell you that you need to insert lines 13 through 15 from the second file, after line three in the first file. If Diff tells you

```
4-7,d8
```

This means you need to delete lines 4 through 7 in the first file, or insert the lines after line 8 in the second file, if you prefer to modify the second file.

Much Ado about Nothing

If the two targets for comparison contain large numbers of empty lines at different positions, this can considerably bloat the Diff output. However, you can tell Diff to ignore blank lines by setting another parameter, *-B*.

There are also various options for handling blanks. For example, you can tell Diff to ignore tabs (*-E*), single blanks (*-b*), or any blanks (*-w*). If you do not wish to differentiate between upper and lower case, you can use the *-i* option in Diff.

The Right Context

The Diff program has various other command line parameters to help improve

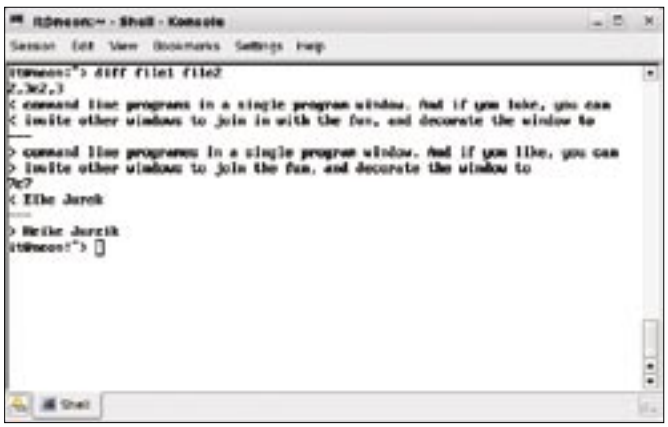

Figure 1: Diff can show you the differences between two text files in the command line.

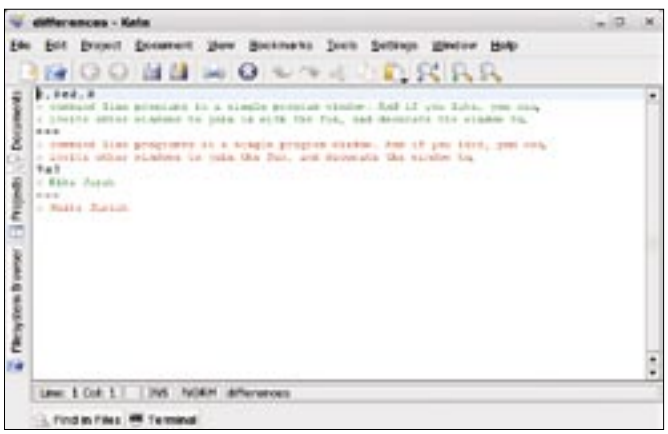

Figure 2: Syntax highlighting makes Diff output much easier to understand.

readability if the output is hard to understand:

```
diff -c file1 file2
```

This adds some formatting, and, by default, three lines of context to the output. If you need more context, just add the line number with the *-C* flag:

```
diff -C 5 file1 file2
```

Diff marks the first file with asterisks, and the second file with dashes. The first things you see are the file names, and the last change date:

```
*** file1      2006-10-04
00:11:09.000000000 +0200
--- file2      2006-10-04
03:05:03.000000000 +0200
```

The remaining output is also organized by file.

Lines with differences are tagged with an exclamation mark, and there are no mark ups for matching lines. New sections are marked with a plus, and missing lines with a minus (Listing 1).

Listing 1: Diff Output

```
01 *** file1      2006-10-04 ➥
   00:11:09.000000000 +0200 ➥
02 --- file2      2006-10-04
   03:05:03.000000000 +0200
03 ***************
04 *** 3,17 ****
05 [...]
06 !    Heike Jurzik
07 [...]
08 --- 3,9 ----
09 [...]
10 !    Elke Jurek
11 [...]
12 ***************
13 *** 25,31 ****
14 [...]
15 -   Setting up the ladder
16 [...]
17 --- 252,256 ----
18 [...]
19 +
20 + Here's some more text!
```

Side by Side

Another interesting Diff option is *-y*, which tells the program to display the two files side by side. You need a large terminal window for this output format, and even if you have one, the lines may still be curtailed. The whole thing becomes more readable if you add the *--suppress-common-lines* parameter to remove matching lines from the output. In this scenario, try the *-W* option followed by numeric value for the number of columns you wish to view:

```
diff -y --suppress-common-lines
-W 70 file1 file2
```

Patching

The real power of Diff becomes apparent if you use the tool for fully automated file synchronizing – this avoids the need for manual patching. Diff has a number of options that handle the following:

```
diff -u file1 file2 > patching
```

This creates output in unified format, and the resulting file is referred to as a patch file. Both files are listed below each other in the output, along with the last change dates and three lines of context to improve readability. You can specify a different number of lines. The following command tells Diff to add ten unchanged lines to the output:

```
diff -U 10 file1 file2 > ➥
patching
```

Time to get out your needle and thread, and apply the patch. The tool for doing this at the command line is aptly named Patch. For safety reasons, specify the *-b* option to tell patch to keep a backup of the original file. The < option points to the patch file:

```
$ patch -b < patching
patching file file1
```

Additionally, a backup of the original *file1* is stored with a suffix of *.orig*.

Listing 2: The Diff3 Tool

```
01 $ diff3 file1 file2 file3
02 ====2
03 1:6c
04 3:6c
05 _____________________________
   _____________________________
   _______
06 2:6c
07      Elke Jurek
08 [...]
09 ====
10 1:14,16c
11      Heike Jurzik
12 2:10,14c
13 The screen program gives➥ you
   the ability to run➥
   multiple virtual consoles
14 [...]
15 3:14,16c
16 Heike Jurzig
17 [...]
18 ====3
19 1:257a
20 2:255a
21 3:255,256c
```

Allrounder

If you run Diff with two directory names, the program will compare every file that occurs in both directories. As an alternative, you can pass the required file and directory names to Diff. The tool will then search the directories for the file you specified, and compare the two files if the search was successful.

By default, Diff will not create any output if two files are identical, but if you need a complete report, just set the *-s* flag:

```
$ diff -s dir1 dir2
Files dir1/file1 and dir2/file1 ➥
are identical
diff -s dir1/file2 dir2/file2
6c6
<    Elke Jurek
---
>    Elke Jurzig
Only patch in dir1:
```

If you would like to include subdirectories, set the recursion option *-r* to tell Diff to search recursively throughout the underlying tree. You can also exclude files from the comparison. To do so, set the *-x* flag, and specify the exclusion candidates, using quotes when you do so.

To exclude all files that end in *.orig*, do the following:

```
diff -s -x "*.orig" dir1 dir2
```

Good Things Come in Threes

Diffutils include a program that can compare three files. The appropriately named Diff3 tool is run against three text files (Listing 2).

The output from Diff3 is different from that of the normal Diff tool. The Diff3 tool uses different sections, separated by equals signs, to show you where the files differ from each other. Equals signs without a number indicate that all three files are different; the numbers 1, 2, or 3 show which of the files is different from the others.

Just like Diff, Diff3 gives you line numbers and ranges, and tells you whether you need to add (*a*) or change (*c*) something.

Common Denominator

Diff3 has more tricks up its sleeve. Let's assume that two users have modified the same text file; in this case, you can use Diff3 to merge the files. To do so, specify the original file as the second argument in the Diff3 command line, and the two modified files as the first and third arguments:

```
diff3 -m file1 original file2 > ➥
output
```

The *-m* parameter stands for "merge", and the output here is redirected via the > - operator to a file named *output*. If the action causes irresolvable conflicts, the output file will point to the problematic areas:

```
<<<<<<< file1
||||||| original
```

In this case, just fire up your favorite editor and manually modify the files. ■

TIP

If the Diff output scrolls off the screen, you can pipe it to the Less or More pagers, as in *diff file1 file2 | less*. Alternatively, you might prefer to redirect the output to a file, as in *diff file1 file2 > differences*. scrolllf you then open the file in a text editor that supports syntax highlighting, the differences will be highlighted in different colors for easier viewing (Figure 2).