diff & diff3

Different Tasks

The diff and diff3 programs from the diffutils package help you compare text files, displaying any variants in the command line and even automatically merging the files, if needed. BY HEIKE JURZIK

Do you make lots of changes to configuration files and other ASCII texts, but prefer to keep the original versions? If so, you will end up with an enormous collection of very similar files sooner or later. In this case, you may appreciate utilities that quickly find text variants, such as the tools in the *diffutils* package:

- *diff* compares two text files,
- *diff3* displays the difference between three files,
- *sdiff* merges two files interactively,
- and *cmp* compares binary files.

In this article we will be looking at using the *diff* and *diff3* tools, or your favorite editor for that matter, to spot the differences.

Different

The *diff* program compares two text files and displays the results in a terminal window. If you are simply interested in discovering whether two files are different, just type *diff -q file1.txt file2.txt*. The program will then either report that *Files file1.txt and file2.txt differ* or it will not say anything. If you leave out the option, the differences are displayed on screen:

```
hj@asteroid:~$ diff file1.txt
 file2.txt
1c1
< Hen
---
> Hens
```

1c1 indicates that line 1 in these files is different. The *c* in this output stands for change. In other words, you need to change line 1 to make the files match. This is followed by the content of line one from the first file, a separating line —, and the same line from the second file.

diff also shows new sections:

```
5a6,10
>
> the diffutils include the
> programs diff (for comparing
> text files), diff3 (outputs
> the differences between three
> files), sdiff
...
```

5a6,10 means that the first file would need to insert lines 6 through ten from the second after line 5 (*a* stands for "append") if the files are to match.

Of course, the output quickly scrolls off screen, especially if you are comparing two longer files. In this case you can use a pager, such as *less* or *more*, to display the output page by page (*diff file1.txt file2.txt | less*), or redirect the output to a file (*diff file1.txt file2.txt > differences.diff*). The vim editor has a practical syntax-highlighting feature for *diff* files, using colors to show the differences (see Figure 1). The vim package also includes a vimdiff tool (see Box 1), which can compare and manipulate up to four different files at the same time. If you prefer (X)emacs, check out Box 2 for a few ediff tips.

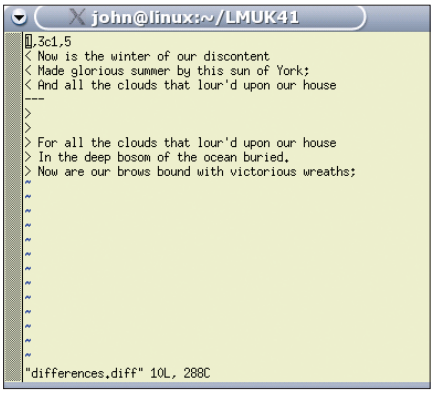

Figure 1: Displaying the "differences.diff" file in the vim editor.

More Context

The output from *diff* is easier to read if you specify the *-c* flag. The first line in

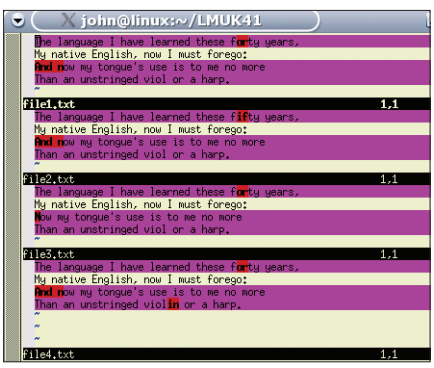

Figure 2: Vimdiff displays up to four files.

Command Line

Although GUIs such as KDE or GNOME are useful for various tasks, if you intend to get the most out of your Linux machine, you will need to revert to the good old command line from time to time. Apart from that, you will probably be confronted with various scenarios where some working knowledge will be extremely useful in finding your way through the command line jungle.

Box 1: Vimdiff

Vimdiff allows you to edit up to four files at the same time, showing text variants. This mode is launched by entering *vim -d file1.txt file2.txt* or *vimdiff file1.txt file2.txt*. Vim displays each file in a separate window, using vertical tiling by default. The *-o* parameter allows you to specify horizontal tiling. The program has an excellent help feature, which is launched by typing *:help diff* in vim.

the output is the last change date for both files. The first file (indicated by an asterisk) appears first. Lines where *diff* has found discrepancies start with an exclamation mark. Lines without any differences are not tagged. After the output for the first text file, *diff* draws a dotted line and then outputs the second. This kind of comparison can be more complex depending on the length of the file.

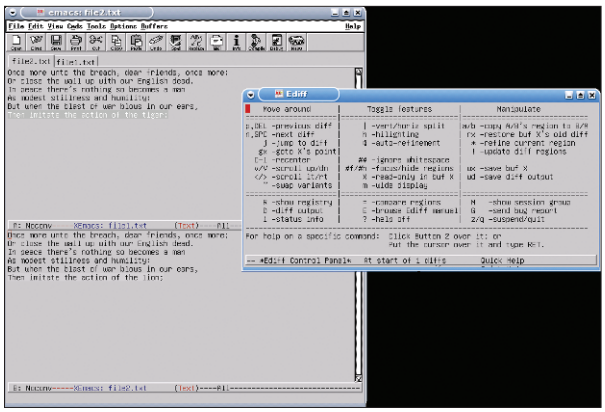

Figure 3: Ediff finding file differences.

The context option tells *diff* to tag new sections with a plus character. A minus character indicates sections that have been dropped from the second file.

Three of a Kind

diff3 compares three files. The output is not what you are used to from *diff*. The easiest way to explain this is to look at an example. Listing 1 shows three short text files, followed by the output from the *diff3* command.

diff3 indicates differences by outputting three = signs, followed by 1, 2 or 3, depending on the file with the text variant. Three equals signs without a number indicate that all three files are different.

In Listing 1 you can see that the second file is different (= = = = *2*). The first line (*1c*) in both the first (*1:*) and the third file (*3:*) is different. This output is followed by a quote from the first line in file 2.

The third file = = = = *3* has an additional line in comparison to the first two. To make the files match, you would have to add the extra line from file three (*3:4c Frog*) to both the first and the second file (*1:3a, 2:3a*).

Patchwork

Of course, you can view the variants found by *diff* in a pager or editor, and use them to perform manual changes. This is extremely tedious with longer files. If you want to tell another user about the changes, assuming that this user has the original file, you can simply store the differences. The *patch* command can then be used to apply the changes to the original file, thus updating it.

Patches contain the changes created by *diff* in "unified diff format" when you set the *-u* flag. The changes are displayed in sequence, and tagged with plus and minus signs:

```
--- file1.txt  Sun Jan 25 ➥
16:12:59 2004
+++ file2.txt  Sun Jan 25 ➥
16:13:33 2004
@@ -1,5 +1,11 @@
-You can perform many
-tasks using KDE or GNOME
+Though you can perform many
+tasks using GNOME or KDE
...
```

If you redirect the output to a file, called *patch* for example, you can then use this file to apply the changes in the second file to the first file. The following section shows you how to do this:

```
hj@asteroid:~$ diff -u ➥
file1.txt file2.txt > patch
hj@asteroid:~$ patch -b -p0 ➥
< patch
```

These options tell *patch* to create a backup of the original file (*-b*) and perform the patch operation in the current directory (*p0*). *patch* has a very detailed manpage, which is definitely worth your while reading if you need more information on the command.

Listing 1: Comparing files

file1.txt:

```
Hen
Cockerel
Chicken
```

file2.txt:

```
Hens
Cockerel
Chicken
```

file3.txt:

```
Hen
Cockerel
Chicken
Frog
```

diff3 output:

```
01 hj@asteroid:~$ diff3 file1.txt
    file2.txt file3.txt
02 ====2
03 1:1c
04 3:1c
05    Hen
06 2:1c
07    Hens
08 ====3
09 1:3a
10 2:3a
11 3:4c
12    Frog
```

Box 2: Ediff in (X)Emacs

The ediff tool is integrated in the (X)emacs editor. Ediff can compare two or three text files. The editor displays these files in separate windows (tiled vertically or horizontally) or in separate frames. Depending on your (X)emacs version, you can either launch ediff via the (*Tools / Compare / ...*) menu, or use the keyboard shortcut [Alt-x], *ediff* ("M-x ediff" in emacs notation).

Then type *?* in the separate ediff window to display an overview of the available commands. You might also like to look at the info pages (*info ediff*) for a detailed guide.

Ediff does not use color highlighting by default, using various grayscales to indicate variants instead. If you prefer a more colorful approach, as provided by vim, press [Alt-x] and type *customize ediff-highlighting*. The *Even Diff Face A*, *Even Diff Face B*, and *Even Diff Face C* entries are responsible for highlighting among other things. Click on the arrow to the right to modify the color and style preferences. Don't forget to click on *Save* to save your changes when you are finished. Then select *Done* to quit configuration mode. The new color settings are stored in the configuration files for the editor below your own home directory, this is *~/.xemacs/custom.el* for Xemacs.