

NAME

diffmark – mark changes between versions of a file

SYNOPSIS

diffmark [**code**"string"] ... [*name*]

DESCRIPTION

Diffmark is a filter used to modify the editor command version of *diff(I)* output so that it can be used to mark the changes between successive versions of a file. Its most common use is to automatically insert change mark commands into a file of text for *nroff(I)* or *troff(I)*. The following is a typical command sequence:

```
diff -e oldfile newfile | diffmark markedfile | ed - oldfile
```

The generated file *markedfile* is the same as *newfile*, except that it has the needed change mark requests inserted. The user would normally print *markedfile*, and later remove it and *oldfile* when no longer needed.

Diffmark adds extra lines to the output of *diff*. It inserts one line at both the beginning and end of each sequence of appended or changed lines, and appends two lines following each deletion. The default values of these lines are chosen to make use of the “margin character” request of the formatters. The user may override any such value by supplying an option string, which is concatenated with a newline to make up the line. Any null option string causes its corresponding line to be omitted completely. The option codes and their defaults are as follows:

-ab".mc |" – “append” beginning – insert at beginning of an addition

-ae".mc" – “append” end – insert at end of an addition

-cb".mc |" – “change” beginning – insert at beginning of a change

-ce".mc" – “change” end – insert at end of a change

-db".mc *" – “delete” 1st line – insert as first line of deletion

-de".mc" – “delete” 2nd line – insert as 2nd line of deletion

Although not a necessity, the following option is convenient:

name causes *diffmark* to append "w *name*" to the end of its output. For safety’s sake, this name should *not* be the same as that of the file being edited.

Here is an example. Suppose you run the following sequence of commands:

```
diff -e oldfile newfile >diff1
diffmark diff3 -cb".mc +" <diff1 >diff2
ed - oldfile <diff2
nroff diff3 >diff4
```

Of course, the only reason for doing this rather than using pipelines is to see what all the files look like:

oldfile:

```
.nf
ccc
eee
```

ggg
hhh
zzz

newfile:

.nf
aaa
eee
fff
ggg
zzz

diff1 (output from diff):

5d
3a
fff
.
2c
aaa
.

diff2 (output from diffmark):

5c
.mc *
.mc
.
3a
.mc |
fff
.mc
.
2c
.mc +
aaa
.mc
.
w diff3

diff3 (edited version of oldfile):

.nf
.mc +
aaa
.mc
eee
.mc |
fff
.mc
ggg
.mc *
.mc
zzz

diff4 (formatted output, with line length set to 10):

```
aaa      +
eee
fff      |
ggg
zzz      *
```

If you are so inclined, you can use *diffmark* to produce listings of C (or other) programs with changes marked. A typical shell procedure is:

```
:      cdiffmk: shell proc to show C program differences
:      called: cdiffmk old new
diff -e $1 $2 | (diffmark;echo `1,$p`) | ed - $1 | nroff macs - | pr -h $2
```

The file macs looks like this:

```
.pl1
.ll 77
.nf
.eo
.nc `
```

The **ll** request might specify a different line length, depending on the nature of the program being printed. The **eo** and **nc** requests are probably needed only for C programs.

DIAGNOSTICS

"input not from diff"

"line too long" (>512 characters)

Up to 72 characters of the offending line are printed immediately following the diagnostic.

EXIT CODES

0 – normal completion

1 – input did not appear to be from *diff*, or other error

SEE ALSO

diff(I), nroff(I), troff(I)

BUGS

Esthetic considerations may dictate manual adjustment of some output. File differences involving only formatting requests may produce undesirable output. I.e., replacing ".sp" by ".sp2" will produce a change mark on the preceding or following line of output.

For those who use *diffmark* to produce UNIX Manual pages, extra handling may be needed to get vertical bars to appear. This results from the choice of the bar as the character translated to a nonadjustable blank for use with tabs. When you use *diffmark*, override the default choice of "|" by "!" instead, causing the latter to appear in your final output. If you prefer the vertical bar, you can get it on the final output by adding the following to the beginning of your file:

```
.if n .tr !|
.if n .ds v !
```

which may be mysterious, but works.