

NAME

change – Copy a file replacing all occurrences of one string by another.

SYNOPSIS

`${HOME}/bin/exe/change 1=input 2=output`

DESCRIPTION

Often it is necessary to replace multiple occurrences of one string by another. This is easy to do using sed and other command-line tools when neither string contains characters that are meaningful to the shell. When either string contains characters such as `-$&\#^!;`, complicated escape sequences are often required to keep the string from being misinterpreted or mangled by the shell. This program reads the old and new strings using Fortran I/O that is not visible to the shell, so the old and new strings can simply be typed verbatim.

The program begins by attaching files for the input and output text. Then it prompts for and reads the old string to be replaced and the new string that it is to be replaced by. Next it transcribes lines from the input file into the output file one line at a time. If an input line is blank or does not contain the old string it is copied unchanged (except that any trailing blanks are trimmed off). If an input line contains the old string, the old string is blanked out, the resulting blanks are shifted out, blanks are shifted in to make room for the new string, and the new string is copied into that space; this process is repeated until there are no occurrences of the old string left.

UNITS and FILES

0	prompts and error messages
1	input file
2	output file
5	keyboard responses to prompts

DIAGNOSTICS

If an EOF is sent in response to a prompt, the program stops with return code 1; otherwise it stops with return code 0.

BUGS

As currently compiled the program works only for lines up to 80 characters long.

If changing the first occurrence of old to new results in another occurrence of old, that will also be changed to new on the next pass through the line; this might or might not be the behavior that is desired.

AUTHOR

Michael Kupferschmid

EXAMPLES

```
unix[1] cat fyle1
Here is a file containing -*$&/ some special characters.
unix[2] change
Name of input file: fyle1
Name of output file: out
old: -*$&/
new: /&$*-
unix[3] cat out
Here is a file containing /&$*- some special characters.
unix[4] more fyle2
Here are some ~!@#$$%^&*(){}[]<>? characters to change.
unix[5] change 1=fyle2 2=out
The file "out" is to be written, but it exists; ok? yes
old: !@#$
new: abcd
unix[6] cat out
Here are some ~abcd%&*(){}[]<>? characters to change.
```