

```
1: #!/usr/local/bin/perl -w
2: # $Id: edfile.perl,v 1.1 2013-04-11 18:53:50-07 - - $
3: # See the listedit utility man page for specifications.
4:
5: use strict;
6: use warnings;
7: use Getopt::Std;
8: use IO::Handle;
9:
10: $0 =~ s|^(\.*/)?([^\/]+)/*$|$2|;
11: my $exit_status = 0;
12: END{ exit $exit_status; }
13: sub note(@){ print STDERR "$0: @_"; };
14: $SIG{'__WARN__'} = sub{ note @_; $exit_status = 1; };
15: $SIG{'__DIE__'} = sub{ warn @_; exit; };
16:
17: autoflush STDOUT 1;
18: autoflush STDERR 1;
19: my ($dotline, @buffer);
20:
21: sub readfile($) {
22:     my ($filename) = @_;
23:     warn "$filename: $!\n" and return unless open my $file, "<$filename";
24:     my @input = <$file>;
25:     close $file;
26:     printf "%d lines read from $filename\n", scalar @input;
27:     splice @buffer, $dotline + 1, 0, @input;
28:     $dotline += @input;
29: }
30:
31: sub writefile($) {
32:     my ($filename) = @_;
33:     warn "$filename: $!\n" and return unless open my $file, ">$filename";
34:     print $file @buffer;
35:     close $file;
36:     $dotline = $#buffer;
37:     printf "%d lines written to $filename\n", scalar @buffer;
38: }
39:
40: sub printcurr() {
41:     print "no lines in file\n" and return unless $dotline >= 0;
42:     printf "%6d %s", $dotline, $buffer[$dotline]
43: }
44:
45: my %switch = (
46:     #ch => [hasopnd, printcurr, sub]
47:     '#' => [1, 0, sub {}],
48:     '$' => [0, 1, sub {$dotline = $#buffer}],
49:     '*' => [0, 0, sub {$dotline = $_, printcurr for 0..$#buffer}],
50:     '.' => [0, 1, sub {}],
51:     '0' => [0, 1, sub {$dotline = 0}],
52:     '<' => [0, 1, sub {--$dotline}],
53:     '>' => [0, 1, sub {++$dotline}],
54:     'a' => [1, 1, sub {splice @buffer, ++$dotline, 0, "$_[0]\n"}],
55:     'd' => [0, 1, sub {splice @buffer, $dotline, 1 if @buffer}],
56:     'i' => [1, 1, sub {if ($dotline < 0) {@buffer = "$_[0]\n"}
57:                     else {splice @buffer, $dotline, 0, "$_[0]\n"}}],
58:     'r' => [1, 0, sub {readfile $_[0]}],
```

```
59:     'w' => [1, 0, sub {writefile $_[0]}],
60:
61: );
62:
63: my $wantecho = ! (-t STDIN && -t STDOUT);
64: for my $filename (@ARGV) {
65:     open FILE, "<$filename" or warn "$filename: $!\n" and next;
66:     push @buffer, <FILE>;
67:     close FILE;
68: }
69: $dotline = $#buffer;
70: printf "%d lines read\n", scalar @buffer if @buffer;
71:
72: for(;;) {
73:     print "$0: ";
74:     last unless defined (my $line = <STDIN>);
75:     print $line if $wantecho;
76:     chomp $line;
77:     next if $line =~ m/^\s*$/;
78:     my ($key, $operand) = $line =~ m/(.)(.*)/;
79:     my ($hasopnd, $printcurr, $sub) = @{$switch{$key} || []};
80:     warn "invalid command: $line\n" and next
81:         unless length $key and defined $sub
82:             and ($hasopnd or (length $operand) == 0);
83:     $sub->($operand);
84:     $dotline = 0 if $dotline < 0;
85:     $dotline = $#buffer if $dotline > $#buffer;
86:     printcurr if $printcurr;
87: }
88:
89: print "^D" if $wantecho;
90: print "\n";
91:
```

```
1: # $Id: edscript,v 1.1 2013-04-11 18:53:54-07 - - $
2: rtest.in
3: *
4: aanother line
5: ianother line
6: *
7: <
8: <
9: d
10: <
11: d
12: *
13: wscript.out
```

```
1: #!/bin/sh -x
2: # $Id: mk,v 1.2 2014-04-15 19:25:40-07 - - $
3: FILES="edfile.perl edsript mk test.in"
4: cid + $FILES
5: edfile.perl <edsript 2>&1 | tee edsript.log
6: mkpspdf Listing.test.ps $FILES *.out *.log
```

```
1: # $Id: test.in,v 1.1 2013-04-11 18:53:50-07 - - $  
2: This is an input test file.  
3: It contains a few lines of data.  
4: It will be written out again.  
5: Look at the typescript logs for an example.
```

```
1: # $Id: test.in,v 1.1 2013-04-11 18:53:50-07 - - $  
2: This is an input test file.  
3: It contains a few lines of data.  
4: another line  
5: another line
```

```
1: edfile.perl: # $Id: edscript,v 1.1 2013-04-11 18:53:54-07 - - $
2: edfile.perl: rtest.in
3: 5 lines read from test.in
4: edfile.perl: *
5:     0 # $Id: test.in,v 1.1 2013-04-11 18:53:50-07 - - $
6:     1 This is an input test file.
7:     2 It contains a few lines of data.
8:     3 It will be written out again.
9:     4 Look at the typescript logs for an example.
10: edfile.perl: aanother line
11:     5 another line
12: edfile.perl: ianother line
13:     5 another line
14: edfile.perl: *
15:     0 # $Id: test.in,v 1.1 2013-04-11 18:53:50-07 - - $
16:     1 This is an input test file.
17:     2 It contains a few lines of data.
18:     3 It will be written out again.
19:     4 Look at the typescript logs for an example.
20:     5 another line
21:     6 another line
22: edfile.perl: <
23:     5 another line
24: edfile.perl: <
25:     4 Look at the typescript logs for an example.
26: edfile.perl: d
27:     4 another line
28: edfile.perl: <
29:     3 It will be written out again.
30: edfile.perl: d
31:     3 another line
32: edfile.perl: *
33:     0 # $Id: test.in,v 1.1 2013-04-11 18:53:50-07 - - $
34:     1 This is an input test file.
35:     2 It contains a few lines of data.
36:     3 another line
37:     4 another line
38: edfile.perl: wscript.out
39: 5 lines written to script.out
40: edfile.perl: ^D
```