

```
1: #!/usr/local/bin/perl -w
2: my $RCSID = '$Id: edfile.perl,v 1.1 2008-02-15 13:06:04-08 - - $';
3: use strict;
4: use warnings;
5: #
6: # See the listedit utility man page for specifications.
7: #
8:
9: $0 =~ s|^(\./)?(?:[^\./]+)/*$|$2|;
10: my $exit_status = 0;
11: END { exit $exit_status; }
12: sub note(@) { print STDERR "@_"; };
13: $SIG{'__WARN__'} = sub { note @_; $exit_status = 1; };
14: $SIG{'__DIE__'} = sub { warn @_; exit; };
15:
16: select STDOUT; $| = 1;
17: select STDERR; $| = 1;
18:
19: my %Options;
20: my @Linelist = ();
21: my $Dotline = $#Linelist;
22:
23: sub readfile ($) {
24:     my ($filename) = @_;
25:     if (open my $file, "<$filename") {
26:         my @readlines = <$file>;
27:         close $file;
28:         splice @Linelist, $Dotline + 1, 0, @readlines;
29:         $Dotline += @readlines;
30:         note sprintf "$0: %d lines read from %s\n",
31:             scalar @readlines, $filename;
32:     } else {
33:         warn "$0: $filename: $!";
34:     }
35: }
36:
37: sub writefile ($) {
38:     my ($filename) = @_;
39:     $filename = $ARGV[0] unless $filename =~ m/.*/;
40:     if (open my $file, ">$filename") {
41:         print $file @Linelist;
42:         close $file;
43:         $Dotline = $#Linelist;
44:         note sprintf "$0: %d lines written to %s\n",
45:             scalar @Linelist, $filename;
46:     } else {
47:         warn "$0: $filename: $!";
48:     }
49: }
50:
51: sub printcurr () {
52:     $Dotline = 0 if $Dotline < 0;
53:     $Dotline = $#Linelist if $Dotline > $#Linelist;
54:     if ($Dotline >= 0) {
55:         printf "%6d: %s", $Dotline, $Linelist[$Dotline];
56:     } else {
57:         warn "$0: no lines in file";
58:     }
59: }
60:
```

```
61:
62: my %Commandswitch = (
63: #   key => [hasoperand, printcurr, sub].
64:   '#' => [1, 0, sub {}],
65:   '$' => [0, 1, sub {$Dotline = $#Linelist}],
66:   '*' => [0, 0, sub {$Dotline = $_, printcurr for 0..$#Linelist}],
67:   '.' => [0, 1, sub {}],
68:   '0' => [0, 1, sub {$Dotline = 0}],
69:   '<' => [0, 1, sub {--$Dotline}],
70:   '>' => [0, 1, sub {++$Dotline}],
71:   'a' => [1, 1, sub {splice @Linelist, ++$Dotline, 0, "$_[0]\n"}],
72:   'd' => [0, 1, sub {splice @Linelist, $Dotline, 1 if @Linelist}],
73:   'i' => [1, 1, sub {$Dotline = 0 if $Dotline < 0;
74:                     splice @Linelist, $Dotline, 0, "$_[0]\n"}],
75:   'r' => [1, 0, sub {readfile $_[0]}],
76:   'w' => [1, 0, sub {writefile $_[0]}],
77: );
78:
79: unless (@ARGV == 1) {
80:   print STDERR "Usage: $0 filename\n";
81:   $exit_status = 1;
82:   exit;
83: }
84: my $bothttys = -t STDIN and -t STDOUT;
85: readfile $ARGV[0];
86: for (;;) {
87:   print "$0: ";
88:   last unless defined (my $command = <STDIN>);
89:   print $command unless $bothttys;
90:   chomp $command;
91:   next if $command =~ m/^\s*$/;
92:   my ($key, $operand) = $command =~ m/(.)(.*)/;
93:   my $sub = $Commandswitch{$key};
94:   if (defined ($sub) and ($sub->[0] or length ($operand) == 0)) {
95:     $sub->[2] ($operand);
96:     printcurr if $sub->[1];
97:   } else {
98:     warn "$0: $command: invalid command\n";
99:   }
100: }
101: print "^D\n";
102:
```