NAME

fv – binary file viewer/editor

SYNOPSIS

fv [**–hrsvw**] [**-k** *STRING*] *filename ...*

DESCRIPTION

fv is a "friendly" editor for binary files. Its function is similar to that of vi for text files, and has a relatively familiar feel for those who are comfortable with vi.

A simple online help window is displayed by pressing ^H [control-H] while running fv. Scrolling up and down in the window is very similar to motion in *more* or *vi*.

OPTIONS

–h Display command parameter syntax.

–k Specify the initial keystrokes to perform

–r Specify this file as read only (do not modify).

–s Perform editing with no screen display

–v Print the current fv version number and exit.

–w Try to force write mode even if the file is read-only.

Main Editor State:

0-9	Enter number of times to perform command
a	Address display on/off
A	Append to file from cursor until mark
b	Back one full display page
^B	Back one full display page
c	Character display on/off
C	Copy to file from cursor position to mark
^C	Exit (do not ask to save)
d	Down one half display page
^D	Down one half display page
f	Forward one full display page
F	Select new file from list on command line
^F	Forward one full display page
g	Go to specific byte position (hex) in file
G	Go to bottom of file
^G	Show current filename and cursor position
h	Cursor left
H	Buffer window left
^H	Help screen
^I	{tab} Change word search mode (string/hex)
j	Cursor down
J	Buffer window down
^J	[Return] Start editing either String or Hex
k	Cursor up
K	Buffer window up
l	Cursor right
L	Buffer window right
^L	Redraw screen
m	Set mark (type mark character)
M	Move cursor to middle of screen
^M	[Return] Start editing either String or Hex

n	Search again in current direction
N	Search again, but in opposite direction
o	Change display char for non-ASCII and zero
O	Change display char for non-ASCII and zero
q	Exit (ask to save if modified)
Q	Exit (ask to save if modified)
r	Reread page
R	Change Read-only mode
s	Change byte swapping mode (off, word, long)
S	Toggle byte swapping relative to chars/ words
t	Toggle word display type (hex/ dec)
T	Cycle word display type (hex/ dec/ float/ dbl)
u	Up one half display page
^U	Up one half display page
w	Toggle Word display on/ off
W	Write current buffer contents to file
^W	Write current buffer contents to file
x	Exit (ask to save if modified)
^X	Exit (ask to save if modified)
z	Set current line at top "z<CR>", middle "z.", bottom "z-", or force bottom to top "z+"
0	Beginning of file/ top of screen
/	Search forward for String/ Hex
?	Search backward for String/ Hex
^	Go to beginning of current display row
$	Go to end of current display row
+	Add more display columns (up to screen size)
=	Add more display columns (up to screen size)
-	Subtract display columns (down to one column)
_	Subtract display columns (down to one column)
’	Go to mark (mark character or ’ for last)
‘	Go to mark (mark character or ’ for last)
(	Go to top/ left of screen
)	Go to bottom/ right of screen
<	Increase cursor alignment (1, 2, 4, 8 bytes)
>	Decrease cursor alignment (1, 2, 4, 8 bytes)
[	Double lower status address display div/ mod
]	Halve lower status address display div/ mod
{	Double left address address display div/ mod
}	Halve left address address display div/ mod
"	Toggle status range address between div/ mod
~	Toggle left address address between div/ mod
.	Go to string editor and repeat last session
:	Enter colon mode command

Editor State, Hex Edit Mode

0-9	Change the current hex digit
a-f	Change the current hex digit
^H	Cursor left
h	Cursor left
j	Cursor down
k	Cursor up
l	Cursor right

q	Quit editing the string, do not store
^M	Finished editing (same as return)
q	Quit editing the string, do not store
^X	Quit editing the string, do not store
?	Display online help information

Editor State, String Edit Mode

ESC Mode:

a		Add after cursor, enter insert mode
A		Add at end of string, enter insert mode
b		Go to previous word
c	f	Delete including character, insert mode
c	t	Delete to character, put in insert mode
c	w	Delete word, put in insert mode
c	^	Delete to start of string, insert mode
c	0	Delete to start of string, insert mode
c	$	Delete to end of string, insert mode
C		Change to end of string
d	f	Delete including character
d	t	Delete to character
d	w	Delete word
c	^	Delete to start of string
c	0	Delete to start of string
c	$	Delete to end of string
D		Delete to end of string
e		Go to end of this word or next
f		Go to next character typed
G		Go to end of string
h		Cursor left
^H		Cursor left
i		Insert at cursor position, enter insert mode
I		Insert at start of string, enter insert mode
j		Cursor down
k		Cursor up
l		Cursor right
^M		Finished editing (same as return)
q		Quit editing the string, do not store
r		Replace single character
R		Replace multiple characters (overstrike)
s		Substitute for single character
t		Go to character before next typed character
T		Go to character before next typed character
u		Undo last change
U		Undo all changes to string
w		Go to next word
x		Erase single character
^X		Quit editing the string, do not store
0		Beginning of string
$		End of line
~		Switch case of current alphabetic character
.		Repeat last edit command
^		Beginning of string
?		Display online help information

ESC Enter ESC command mode

Insert Mode only:

^H Delete left of cursor (same as backspace key)

^V Add next raw character to string

ENVIRONMENT

The editor recognizes the environment variable FVINIT as a list of key commands to execute on startup as if entered by the keyboard. This is done before the display is brought active, so that preferred fv mode commands can be placed into the FVINIT and executed automatically at invocation. In the event of having both this environment variable set and specifying the -k parameter, the commands specified in the FVINIT environment variable will always be executed first.

EXAMPLES

The following command:

% fv -s -k '/_exit\ 015n\ 015cw_die\ 015qy' deathbin

will search out the keyword _exit in file deathbin and change it to be _die.

SEE ALSO

vi(1), **od**(1V).

NOTES

In the main editor and string editor, entering a number before a command repeats that command similar to vi.

fv is mainly a stable product, but segmentation faults should be expected at the least unexpected events.

This program is freeware meaning that it may not be sold for profit or included in a package which is sold for commercial gain. The author grants fitness of this product for NO purpose, nor is liable for damages incurred as the result of attempting to apply this product for a purpose.

fv is Copyright 1991-1996 Chris Hooper (cdh@mtu.edu)