Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'pinfo.1' command

$ man pinfo.1

PINFO(1) General Commands Manual PINFO(1)

NAME

pinfo - curses based lynx-style info browser

SYNTAX

pinfo [options] [infopage]

DESCRIPTION

This is a program for viewing info files. You specify which page you want to read by passing it an infopage argument. This argument contains the name of an info page (i.e. 'bash'). The program will then (by de? fault) search for it in the current directory, /usr/share/info, /usr/info, /usr/local/share/info, /usr/local/info. and /opt/info. The search path can be adjusted by INFOPATH environment variable or in the configuration file. Pinfo will also automatically add the suffix '-info', '-info.Z', '-info.gz', or '-info.bz2'. At present other suf? fixes are not recognized, but you can easily add them to the function openinfo() in filehandling_functions.c.

When the search for info pages fails, man is called with the infopage argument, and it's output is parsed by pinfo. This means that when you don't have the appropriate info page, but have a man page instead; the man page will be viewed.

When no infopage is specified, the default `dir' page is shown.

Supported options are

-h, --help - print help information and exit.

-v, --version - print version information and exit.

-m, --manual - uses manual page instead of info by default. (pinfo -m could be used as a manual pager). Warning: Everything what follows this option is passed to the `man' program. Don't be confused if pinfo op? tions, which followed `-m' don't work. When using this option, pinfo does not parse the info options as usual! It invokes the man part of program.

You can also call the man function of pinfo in another way. When pinfo is called with an argv[0] (the program file name), which contains the word 'man' in it's name, the man functions are enabled automatically. Previously there was a symlink to pinfo, called pman, but I had to re? move it from the distribution, since it's name was in conflict with some other utility. Anyway, you can feel free to create such a link if you wish.

-r, --raw-filename - uses a raw filename first (i.e. the name which you specified as infopage is considered to be a real file in the specified location).

-f, --file synonym for -r.

-a, --apropos - if this is set, apropos is called when no man or info page could be found.

-p, --plain-apropos - if this is set, call only apropos.

-c, --cut-man-headers - if this is set, man parsing code will try to cut out the repeated man headers. Use with care. ;)

-s, --squeeze-lines- cut empty lines from manual pages. This option en? ables auto cutting of every repeated newline in a manual page.

-d, --dont-handle-without-tag-table - don't display texinfo pages with? out tag table.

-t, --force-manual-tag-table- forces manual detection of tag table. This allows you to view info pages, which may be corrupted. (as i.e. version of jed's pages, shipped with RH5.0). The tag table corruption usually appears in that the info links, which you follow, move you to quite unexpected nodes.

--node=nodename, --node nodename- Go to the node `nodename' of info

file. Since 0.6.7 it is also possible to specify nodes as in standalone info via file names, like `(gcc)Introduction'.

--rcfile=filename, --rcfile filename- Use alternate configuration file.

--long-manual-links, -l- Use long link names in manuals. On some sys? tems the manual hierarchy is divided into subsections like `3ncurses', etc, while on other systems all belongs to section `3'. If this option is what your system is like, feel free to use it.

--clear-at-exit, -x- Clear screen at exit.

The options are handled by GNU getopt, so you can here (as in other programs) abbreviate the option names to the minimal number of charac? ters by which the options differ.

Warning! If you do not have getopt, these options will not work!

DEFAULT KEYS WHEN BROWSING INFO FILE

Just take a look at the example configuration file (below), and at the key descriptions. Keys available in manual viewer differ a bit from the keys available in info viewer.

ENVIRONMENT

There is a variable $INFOPATH, which can specify the paths to be searched for info files. It's format is similar to that of the $PATH variable. An example setting could look like:

/usr/info:/usr/somewhere/info:/not/even/in/usr/info

etc. Directories are separated by colons.

COLOR AND KEY DEFINITIONS

There are configuration files called ~/.pinforc and [prefix]/etc/pin? forc, for local and global configuration (where prefix is the prefix of the directory, where pinfo is installed, i.e. /usr/local, or /).

Here's an example of such a file; we'll discuss the contents below:

Here are some color setting.

Whitespace between the entries is optional.

COL_NORMAL = COLOR_WHITE, COLOR_BLACK, NO_BOLD, NO_BLINK

COL_MENUSELECTED = COLOR_RED, COLOR_BLACK, BOLD, NO_BLINK

COL_MENU=COLOR_BLUE,COLOR_BLACK,BOLD, NO_BLINK

COL_NOTESELECTED=COLOR_RED,COLOR_BLACK,BOLD, NO_BLINK

COL_NOTE=COLOR_GREEN,COLOR_BLACK,BOLD, NO_BLINK

COL_TOPLINE=COLOR_YELLOW,COLOR_BLUE,BOLD, NO_BLINK

COL_BOTTOMLINE=COLOR_YELLOW,COLOR_BLUE,BOLD, NO_BLINK

COL_MANUALBOLD=COLOR_WHITE,COLOR_BLACK,BOLD, NO_BLINK

COL_MANUALITALIC=COLOR_WHITE,COLOR_BLACK,BOLD, NO_BLINK

COL_URL=COLOR_MAGENTA,COLOR_BLACK,BOLD, NO_BLINK

COL_URLSELECTED=COLOR_RED,COLOR_BLACK,NO_BOLD, NO_BLINK

COL_INFOHIGHLIGHT=COLOR_WHITE,COLOR_BLACK,BOLD, NO_BLINK

#

Here are some keybindings as well...

#

KEY_TOTALSEARCH_1 = 's'

KEY_TOTALSEARCH_2 = 'S'

KEY_SEARCH_1 = '/'

KEY_SEARCH_2 = '.'

KEY_GOTO_1='g'

KEY_GOTO_2='m'

KEY_HOME_1='h'

KEY_HOME_2='H'

KEY_PREVNODE_1='p'

KEY_PREVNODE_2='P'

KEY_NEXTNODE_1='n'

KEY_NEXTNODE_2='N'

KEY_UP_1=KEY_UP

KEY_UP_2='u'

KEY_END_1=

KEY_END_2='e'

KEY_PGDN_1=KEY_NPAGE

KEY_PGDN_2=' '

KEY_PGDN_AUTO_1=0

KEY_PGDN_AUTO_2=' '

KEY_PGUP_1=KEY_PPAGE

KEY_PGUP_2='b'

KEY_PGUP_AUTO_1=0

KEY_PGUP_AUTO_2='b'

KEY_DOWN_1=KEY_DOWN

KEY_DOWN_2='d'

KEY_TOP_1=KEY_HOME

KEY_TOP_2='t'

KEY_BACK_1=KEY_LEFT

KEY_BACK_2='l'

KEY_FOLLOWLINK_1=KEY_RIGHT

KEY_FOLLOWLINK_2='0

12 is a code for ctrl+l

KEY_REFRESH_1=12

KEY_REFRESH_2='~'

KEY_SHELLFEED_1='!'

KEY_SHELLFEED_2='1'

KEY_QUIT_1='q'

KEY_QUIT_2='Q'

KEY_DIRPAGE_1='d'

KEY_DIRPAGE_2='D'

KEY_GOLINE_1='l'

KEY_GOLINE_2=0

KEY_PRINT_1=']'

KEY_PRINT_2=0

#

Some options, explained in the man page

#

MANUAL=false

CUT-MAN-HEADERS=true

CUT-EMPTY-MAN-LINES=true

RAW-FILENAME=false

APROPOS=false

DONT-HANDLE-WITHOUT-TAG-TABLE=false

LONG-MANUAL-LINKS=false

FILTER-0xB7=true

QUIT-CONFIRMATION=false

QUIT-CONFIRM-DEFAULT=no

CLEAR-SCREEN-AT-EXIT=true

STDERR-REDIRECTION="2> /dev/null"

HTTPVIEWER=lynx

FTPVIEWER=lynx

MAILEDITOR=pine

MANLINKS=1:8:2:3:4:5:6:7:9:n:p:o:3X11:3Xt

INFOPATH=/usr/info:/usr/share/info:/usr/local/info

HIGHLIGHTREGEXP=Bash.*has

SAFE-USER=nobody

SAFE-GROUP=nobody

As you can see, the format is simple. First I'll explain the color def?
initions. First you must enter a color name (all available color names
are present in the example, and they're self explanatory, I think.
There is also a special color COLOR_DEFAULT, which stands for transpar?
ency). Then you enter the foreground color, and the background color.
The BOLD attribute means that we want the foreground color to be high?
lighted. (i.e. light blue, light green). BLINK attribute is the blink?
ing attribute, or highlighted background in some other configurations.
Now let's move to the key definitions. Here we first put a key name
(again all keys are present in the example); then we enter it's value
-- either surrounded by apostrophes, or a keycode number (like in
KEY_REFRESH_1), or its mnemonic code name if it's a special key (like
i.e. in KEY_FOLLOWLINK_1).
If you wish to specify key by code value, use the supplied program
'testkey' to obtain the needed value. It mainly is a feature, when you
want to add some CTRL+letter keybindings, and similar.
For each function you can bind two keys, i.e. you could bind both Enter
and Cursor Right to the FollowLink-function. As you can see in the ex?
ample above, the two key names are KEY_FOLLOWLINK_1 and KEY_FOL?
LOWLINK_2.

Here's an explanation of the key names:

KEY_TOTALSEARCH_1

Key for searching through all nodes of info file.

KEY_TOTALSEARCH_2

Alternate key for searching through all nodes of info file.

KEY_SEARCH_1

Key for searching through current node (or manual).

KEY_SEARCH_2

Alternate key for searching through current node (or manual).

KEY_SEARCH_AGAIN_1

Key for repeating the last search.

KEY_SEARCH_AGAIN_2

Alternate key for repeating the last search.

KEY_GOTO_1

Key for explicitly going to a node (by specifying it's name).

KEY_GOTO_2

Alternate key for explicitly going to a node (by spec? ifying it's name).

KEY_PREVNODE_1

Key for going to a node marked as 'Prev' in the header. In man page viewer this goes to the previous man section.

KEY_PREVNODE_2

Alternate key for going to a node marked as 'Prev' in the header. In man page viewer this goes to the previ? ous man section.

KEY_NEXTNODE_1

Key for going to a node marked as 'Next' in the header. In man page viewer this goes to the next man section.

KEY_NEXTNODE_2

Alternate key for going to a node marked as 'Next' in the header. In man page viewer this goes to the next man section.

KEY_UP_1 Key for scrolling text one line up.

KEY_UP_2 Alternate key for scrolling text one line up.

KEY_END_1 Key for going to the end of the node.

KEY_END_2 Alternate key for going to the end of the node.

KEY_PGDN_1

Key for going one page down in the viewed node.

KEY_PGDN_2

Alternate key for going one page down in the viewed node.

KEY_PGDN_AUTO_1

Key for going to the next node when you're at the end of node (default is zero -- turned off).

KEY_PGDN_AUTO_2

Alternate key for going to the next node when you're at the end of node (default is space, as for pgdn_2).

KEY_HOME_1

Key for going to the beginning of the node.

KEY_HOME_2

Alternate key for going to the beginning of the node.

KEY_PGUP_1

Key for going one page up in the viewed node.

KEY_PGUP_2

Alternate key for going one page up in the viewed node.

KEY_PGUP_AUTO_1

Key for going to the `up' node, when being at the top of node. (Default value is zero -- turned off).

KEY_PGUP_AUTO_2

Alternate key for going to the `up' node, when being

at the top of node. (Default value is `-', as for pgup_2).

KEY_DOWN_1

Key for scrolling the text down one line.

KEY_DOWN_2

Alternate key for scrolling the text down one line.

KEY_TOP_1 Key for going to the top (first) node.

KEY_TOP_2 Alternate key for going to the top (first) node.

KEY_BACK_1

Key for going back (in the history of viewed nodes).

KEY_BACK_2

Alternate key for going back (in the history of viewed nodes).

KEY_FOLLOWLINK_1

Key for following a hypertext link.

KEY_FOLLOWLINK_2

Alternate key for following a hypertext link.

KEY_REFRESH_1

Key for refreshing the screen (hard coded is the ^L value).

KEY_REFRESH_2

Alternate key for refreshing the screen.

KEY_SHELLFEED_1

Key for calling a shell command, and passing the viewed node to the stdin of that command.

KEY_SHELLFEED_2

Alternate key for calling a shell command, and passing the viewed node to the stdin of that command.

KEY_QUIT_1

Key for exiting the program.

KEY_QUIT_2

Alternate key for exiting the program.

KEY_GOLINE_1

Key for going to a specified line in file.

KEY_GOLINE_2

Alternate key for going to a specified line in file.

KEY_PRINT_1

Key for printing viewed node or man page.

KEY_PRINT_2

Alternate key for printing viewed node or man page.

The special mnemonics for keys (which are defined at present) are:

KEY_BREAK

KEY_DOWN

KEY_UP

KEY_LEFT

KEY_RIGHT

KEY_DOWN

KEY_HOME

KEY_BACKSPACE

KEY_NPAGE

KEY_PPAGE

KEY_END [Note: this works probably ONLY with Linux ncurses]

KEY_F(x)

KEY_CTRL('c')

- this assigns the key value to a ctrl+c combination.

c may be any letter you wish.

KEY_ALT('c')

- this assigns the key value to a alt+c combination. c may be any letter you wish. If alt key won't work, you may use ESC+key combination.

'c' - this means a printable character c. The syntax is just like in C/C++ ;).

[number] - you can also specify key as it's code number. It is useful e.g. when specifying control keys, and some nonstandard keys. A numerical value of zero turns given key binding off.

See manual page for curs_getch (3x) for description of their meaning.

Warning! Try not to create some serious key binding conflicts!

The options in the last part of the example configuration file should be fairly self-explanatory. The variables that can be set to true or false do the same things as the command line arguments with the same names.

MANUAL If this is set to true the default is to first check for a man page, instead of a texinfo file.

CUT-MAN-HEADERS

If set to true, then pinfo tries to cut off the re? peated headers throughout man pages.

CUT-EMPTY-MAN-LINES

If set to true, then pinfo tries to cut off the re? peated newlines (i.e. it will shorten each set of con? secutive newlines to one newline).

RAW-FILENAME

If set to true, the file argument is taken to be the name of a file in the current working directory, i.e. the directories in INFOPATH will only be searched if a file with this name is not in the working directory.

APROPOS If set to true, apropos is called if no info or man page is found.

DONT-HANDLE-WITHOUT-TAG-TABLE

If set to true , pinfo will not attempt to display texinfo pages without tag tables.

HTTPVIEWER

Set this to the program you want to use to follow http links in documents.

FTPVIEWER Set this to the program you want to use to follow ftp links in documents.

MAILEDITOR

Set this to your favorite email program, and it will be started if you follow an email link in a document.

PRINTUTILITY

Utility, which you use for printing. I.e. `lpr'. If you don't use any, you may also try something like `cat >/dev/lp1', or sth. ;)

MANLINKS This specifies the section names, which may be refer? enced in your man pages (i.e. Xtoolkit man pages match the section 3Xt (see for example XtVaCreateWidget) manpage), Xlib function pages match section 3X11, etc. Such extensions may not be recognized by default, so it is a good idea to add them).

INFOPATH This allows you to override the default search path for info pages. The paths should be separated by colons.

MAN-OPTIONS

This specifies the options, which should be passed to the `man' program. (see man(1) for description of what they're like).

STDERR-REDIRECTION

Pinfo allows you to redirect the stderr output of called programs. For example if you don't want to see man's error messages about manual page formatting, you can use STDER-REDIRECTION="2> /dev/null". This is the default.

LONG-MANUAL-LINKS

This is another true/false option, which decides whether your system supports long manual section names, or not. (i.e. "3ncurses" instead of "3").

FILTER-0xB7

This decides, whether you want to convert 0xb7 chars to `o', or not. For example for iso-8859-2 fonts this makes man's list marks a bit nicer ;) (look for exam? ple at perl's man page, to see how those marks look like).

QUIT-CONFIRMATION

This decides whether you want to use quit confirmation on exit, or not.

QUIT-CONFIRM-DEFAULT

This yes/no option determines the default answer to the QUIT-CONFIRMATION dialog. (default answer is when you press a key, that does not match the asked ques? tion).

CLEAR-SCREEN-AT-EXIT

This true/false option determines if you want to have your screen cleared at exit, or no.

CALL-READLINE-HISTORY

This true/false option determines if you want to have a prompt of last history entry whenever calling read? line wrapper, eg. in subsequent searches.

HIGHLIGHTREGEXP

This is an option, through which you may pass to pinfo regexps, which should be highlighted when working with document. Warning! This may turn very slow if you use it without care!

SAFE-USER This option is used to pass the name of user, to which suid when pinfo is run with root privileges.

SAFE-GROUP

This option is used to pass the name of group, to which suid when pinfo is run with root privileges.

INTERNATIONALIZATION SUPPORT

Pinfo implements general features of gnu gettext library (the thing, which you need to see national messages ;). But it is not the end. Pinfo allows you to use national info pages! You only need to put them to your info directory, into a subdirectory, which is called `$LANG'.

LICENSE

This program is distributed under the terms of GPL.

BUGS

Please send bug reports to the author.

AUTHOR

Przemek Borys <pborys@dione.ids.pl>

If that E-mail address wont work (since the machine where it is being handled is a bit damaged lately), you can try pborys@zeus.polsl.gli?wice.pl, or pborys@p-soft.silesia.linux.org.pl.

There was also a lot of other people, who contributed to this code. See the AUTHORS file.

COMMENTS

The author would like to read some comments and suggestions from you, if any.