NAME

pg – files perusal filter for CRTs

SYNOPSIS

pg [–*number*] [–**p** *string*] [–**cefnrs**] [**+***linenumber*] [**+/***pattern***/**] [*filename* ...]

AVAILABILITY

SUNWcsu

DESCRIPTION

The **pg** command is a filter that allows the examination of *filenames* one screenful at a time on a CRT. If the user types a RETURN, another page is displayed; other possibilities are listed below.

This command is different from previous paginators in that it allows you to back up and review something that has already passed. The method for doing this is explained below.

To determine terminal attributes, **pg** scans the **terminfo**(4) data base for the terminal type specified by the environment variable **TERM**. If **TERM** is not defined, the terminal type **dumb** is assumed.

OPTIONS

–*number* An integer specifying the size (in lines) of the window that **pg** is to use instead of the default. (On a terminal containing 24 lines, the default window size is 23).

–**p** *string* **pg** uses *string* as the prompt. If the prompt string contains a **%d**, the first occurrence of **%d** in the prompt will be replaced by the current page number when the prompt is issued. The default prompt string is ‘‘**:**’’.

–**c** Home the cursor and clear the screen before displaying each page. This option is ignored if **clear_screen** is not defined for this terminal type in the **terminfo**(4) data base.

–**e** **pg** does *not* pause at the end of each file.

–**f** Normally, **pg** splits lines longer than the screen width, but some sequences of characters in the text being displayed (for instance, escape sequences for underlining) generate undesirable results. The –**f** option inhibits **pg** from splitting lines.

–**n** Normally, commands must be terminated by a **<***newline***>** character. This option causes an automatic end of command as soon as a command letter is entered.

–**r** Restricted mode. The shell escape is disallowed. **pg** prints an error message but does not exit.

–**s** **pg** prints all messages and prompts in the standard output mode (usually inverse video).

+*linenumber* Start up at *linenumber*.

+/*pattern***/** Start up at the first line containing the regular expression pattern.

OPERANDS

The following operands are supported:

filename A path name of a text file to be displayed. If no *filename* is given, or if it is –, the standard input is read.

USAGE

Commands

The responses that may be typed when **pg** pauses can be divided into three categories: those causing further perusal, those that search, and those that modify the perusal environment.

Commands that cause further perusal normally take a preceding *address*, an optionally signed number indicating the point from which further text should be displayed. This *address* is interpreted in either pages or lines depending on the command. A signed *address* specifies a point relative to the current page or line, and an unsigned *address* specifies an address relative to the beginning of the file. Each command has a default address that is used if none is provided.

The perusal commands and their defaults are as follows:

(+1)<*newline*> or <*blank*>
This causes one page to be displayed. The address is specified in pages.

(+1) **l** With a relative address this causes **pg** to simulate scrolling the screen, forward or backward, the number of lines specified. With an absolute address this command prints a screenful beginning at the specified line.

(+1) **d** or **^D** Simulates scrolling half a screen forward or backward.

*i***f** Skip *i* screens of text.

*i***z** Same as <*newline*> except that *i*, if present, becomes the new default number of lines per screenful.

The following perusal commands take no *address*.

. or **^L** Typing a single period causes the current page of text to be redisplayed.

$ Displays the last windowful in the file. Use with caution when the input is a pipe.

The following commands are available for searching for text patterns in the text. The regular expressions are described on the **regexp**(5) manual page. They must always be terminated by a <*newline*>, even if the *–n* option is specified.

*i***/***pattern***/** Search forward for the *i*th (default *i*=1) occurrence of *pattern*. Searching begins immediately after the current page and continues to the end of the current file, without wrap-around.

i^*pattern*^
*i***?***pattern***?** Search backwards for the *i*th (default *i*=1) occurrence of *pattern*. Searching begins immediately before the current page and continues to the beginning of the current file, without wrap-around. The ^ notation is useful for Adds 100 terminals which will not properly handle the ?.

After searching, **pg** will normally display the line found at the top of the screen. This can be modified by appending **m** or **b** to the search command to leave the line found in the middle or at the bottom of the window from now on. The suffix **t** can be used to restore the original situation.

The user of **pg** can modify the environment of perusal with the following commands:

*i***n** Begin perusing the *i*th next file in the command line. The *i* is an unsigned number, default value is 1.

*i***p** Begin perusing the *i*th previous file in the command line. *i* is an unsigned number, default is 1.

*i***w** Display another window of text. If *i* is present, set the window size to *i*.

s *filename* Save the input in the named file. Only the current file being perused is saved. The white space between the **s** and *filename* is optional. This command must always be terminated by a <*newline*>, even if the –*n* option is specified.

h Help by displaying an abbreviated summary of available commands.

q or **Q** Quit **pg**.

!*command* *Command* is passed to the shell, whose name is taken from the **SHELL** environment variable. If this is not available, the default shell is used. This command must always be terminated by a <*newline*>, even if the –*n* option is specified.

At any time when output is being sent to the terminal, the user can hit the quit key (normally CTRL-\) or the interrupt (break) key. This causes **pg** to stop sending output, and display the prompt. The user may then enter one of the above commands in the normal manner. Unfortunately, some output is lost when this is done, because any characters waiting in the terminal's output queue are flushed when the quit signal occurs.

If the standard output is not a terminal, then **pg** acts just like **cat**(1), except that a header is printed before each file (if there is more than one).

EXAMPLES

The following command line uses **pg** to read the system news:

```
example% news | pg –p "(Page %d):"
```

ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **pg**: **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

The following environment variables affect the execution of **pg**:

COLUMNS Determine the horizontal screen size. If unset or **NULL**, use the value of **TERM**, the window size, baud rate, or some combination of these, to indicate the terminal type for the screen size calculation.

LINES Determine the number of lines to be displayed on the screen. If unset or **NULL**, use the value of **TERM**, the window size, baud rate, or some combination of these, to indicate the terminal type for the screen size calculation.

SHELL Determine the name of the command interpreter executed for a !command.

TERM Determine terminal attributes. Optionally attempt to search a system-dependent database, keyed on the value of the **TERM** environment variable. If no information is available, a terminal incapable of cursor-addressable movement is assumed.

EXIT STATUS

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

FILES

/tmp/pg* temporary file when input is from a pipe
/usr/share/lib/terminfo/?/* terminal information database

SEE ALSO

cat(1), **grep**(1), **more**(1), **terminfo**(4), **environ**(5), **regexp**(5)

NOTES

While waiting for terminal input, **pg** responds to BREAK, CTRL-C, and CTRL–\ by terminating execution. Between prompts, however, these signals interrupt **pg**'s current task and place the user in prompt mode. These should be used with caution when input is being read from a pipe, since an interrupt is likely to terminate the other commands in the pipeline.

The terminal **/**, ^, or **?** may be omitted from the searching commands.

If terminal tabs are not set every eight positions, undesirable results may occur.

When using **pg** as a filter with another command that changes the terminal I/O options, terminal settings may not be restored correctly.