-- --

IEEE P1003.2 Draft 11.2 − September 1991

do not permit it to be overwritten.

4.49 `pr` — Print files

4.49.1 Synopsis

`pr` [`+`*page*] [`−`*column*] [`−adFmrt`] [`−e`[*char*][*gap*]] [`−h` *header*] [`−i`[*char*][*gap*]]
[`−l` *lines*] [`−n`[*char*][*width*]] [`−o` *offset*] [`−s`[*char*]] [`−w` *width*] [*file* . . .]

4.49.2 Description

The `pr` utility is a printing and pagination filter. If multiple input files are specified, each shall be read, formatted, and written to standard output. By default, the input shall be separated into 66-line pages, each with:

— A 5-line header that includes the page number, date, time, and the pathname of the file.

— A 5-line trailer consisting of blank lines.

If standard output is associated with a terminal, diagnostic messages shall be deferred until the `pr` utility has completed processing.

When options specifying multicolumn output are specified, output text columns shall be of equal width; input lines that do not fit into a text column shall be truncated. By default, text columns shall be separated with at least one `<blank>`.

4.49.3 Options

The `pr` utility shall conform to the utility argument syntax guidelines described in 2.10.2, except that: the *page* option has a `'+'` delimiter; *page* and *column* can be multidigit numbers; some of the option-arguments are optional; and some of the option-arguments cannot be specified as separate arguments from the preceding option letter. In particular, the `−s` option does not allow the option letter to be separated from its argument, and the options `−e`, `−i`, and `−n` require that both arguments, if present, not be separated from the option letter.

The following options shall be supported by the implementation. In the following option descriptions, *column*, *lines*, *offset*, *page*, and *width* are positive decimal integers; *gap* is a nonnegative decimal integer.

`+`*page* Begin output at page number *page* of the formatted input.

`−`*column* Produce output that is *columns* wide (default shall be 1) and is written down each column in the order in which the text is received from the input file. This option should not be used with `−m`. The options `−e` and `−i` shall be assumed for multiple text-column output. Whether or not text columns are balanced is unspecified, but a text column shall never exceed the length of the page (see the `−l` option). When used with `−t`, use the minimum number of lines to write the output.

`−a` Modify the effect of the `-`*column* option so that the columns are filled across the page in a round-robin order (e.g., when *column* is 2, the first input line heads column 1, the second heads column 2, the third is the second line in column 1, etc.).

`−d` Produce output that is double-spaced; append an extra `<newline>` following every `<newline>` found in the input.

`−e`[*char*][*gap*]
Expand each input `<tab>` to the next greater column position specified by the formula $n*gap+1$, where n is an integer > 0. If *gap* is zero or is omitted, it shall default to 8. All `<tab>` characters in the input shall be expanded into the appropriate number of `<space>`s. If any nondigit character, *char*, is specified, it shall be used as the input tab character.

`−F` Use a `<form-feed>` character for new pages, instead of the default behavior that uses a sequence of `<newline>` characters.

`−h` *header* Use the string *header* to replace the contents of the *file* operand in the page header. See 4.49.6.1.

`−i`[*char*][*gap*]
In output, replace multiple `<space>`s with `<tab>`s wherever two or more adjacent `<space>`s reach column positions $gap+1$, $2*gap+1$, $3*gap+1$, etc. If *gap* is zero or is omitted, default `<tab>` settings at every eighth column position shall be assumed. If any nondigit character, *char*, is specified, it shall be used as the output `<tab>` character.

`−l` *lines* Override the 66-line default and reset the page length to *lines*. If *lines* is not greater than the sum of both the header and trailer depths (in lines), the `pr` utility shall suppress both the header and trailer, as if the `−t` option were in effect.

`−m` Merge files. Standard output shall be formatted so the `pr` utility writes one line from each file specified by a *file* operand, side by side into text columns of equal fixed widths, in terms of the number of column positions. Implementations shall support merging of at least nine *file* operands.

`−n`[*char*][*width*]
Provide *width*-digit line numbering (default for *width* shall be 5). The number shall occupy the first *width* column positions of each text column of default output or each line of `−m` output. If *char* (any nondigit character) is given, it shall be appended to the line number to separate it from whatever follows (default for *char* shall be a `<tab>`).

`−o` *offset* Each line of output shall be preceded by offset <space>s. If the `−o` option is not specified, the default offset shall be zero. The space taken shall be in addition to the output line width (see `−w` option below).

`−r` Write no diagnostic reports on failure to open files.

`−s`[*char*] Separate text columns by the single character *char* instead of by the appropriate number of `<space>`s (default for *char* shall be the `<tab>` character).

`−t` Write neither the five-line identifying header nor the five-line trailer usually supplied for each page. Quit writing after the last line of each file without spacing to the end of the page.

`−w` *width* Set the width of the line to *width* column positions for multiple text-column output only. If the `−w` option is not specified and the `−s` option is not specified, the default width shall be 72. If the `−w` option is not specified and the `−s` option is specified, the default width shall be 512.

For single column output, input lines shall not be truncated.

4.49.4 Operands

The following operand shall be supported by the implementation:

file A pathname of a file to be written. If no *file* operands are specified, or if a *file* operand is −, the standard input shall be used.

4.49.5 External Influences

4.49.5.1 Standard Input

The standard input shall be used only if no *file* operands are specified, or if a *file* operand is −. See Input Files.

4.49.5.2 Input Files

The input files shall be text files.

4.49.5.3 Environment Variables

The following environment variables shall affect the execution of `pr`:

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

LC_ALL This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**.

LC_CTYPE This variable shall determine the locale for the interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments and input files) and which characters are defined as printable (character class `print`). Nonprintable characters still shall be written to standard output, but shall be not counted for the purpose for column-width and line-length calculations.

LC_MESSAGES This variable shall determine the language in which messages should be written.

LC_TIME This variable shall determine the format of the date and time for use in writing header lines.

TZ This variable shall determine the time zone for use in writing header lines.

4.49.5.4 Asynchronous Events

If `pr` receives an interrupt while writing to a terminal, it shall flush all accumulated error messages to the screen before terminating.

4.49.6 External Effects

4.49.6.1 Standard Output

The `pr` utility output shall be a paginated version of the original file (or files). This pagination shall be accomplished using either `<form-feed>`s or a sequence of `<newline>`s, as controlled by the `−F` option. Page headers shall be generated unless the `−t` option is specified. The page headers shall be of the form:

`"\n\n%s %s Page %d\n\n\n"`, <*output of* `date`>, <*file*>, <*page number*>

In the POSIX Locale, the <*output of* `date`> field, representing the date and time of last modification of the input file (or the current date and time if the input file is standard input), shall be equivalent to the output of the following command as it would appear if executed at the given time:

```
date "+%b %e %H:%M %Y"
```

without the trailing `<newline>`, if the page being written is from standard input. If the page being written is not from standard input, in the POSIX Locale, the same format shall be used, but the time used shall be the modification time of the file corresponding to *file* instead of the current time. When the LC_TIME locale category is not set to the POSIX Locale, a different format and order of presentation of this field may be used.

If the standard input is used instead of a *file* operand, the <*file*> field shall be replaced by a null string.

If the `−h` option is specified, the *file* field shall be replaced by the *header* argument.

4.49.6.2 Standard Error

Used only for diagnostic messages.

4.49.6.3 Output Files

None.

4.49.7 Extended Description

None.

4.49.8 Exit Status

The `pr` utility shall exit with one of the following values:

0	All files were written successfully.
>0	An error occurred.

4.49.9 Consequences of Errors

Default.

4.49.10 Rationale. *(This subclause is not a part of P1003.2)*

Examples, Usage

To print a numbered list of all files in the current directory:

```
ls -a | pr -n -h "Files in $(pwd)."
```

History of Decisions Made

This utility is one of those that does not follow the Utility Syntax Guidelines because of its historical origins. The working group could have added new options that obeyed the guidelines (and marked the old options *obsolescent*) or devised an entirely new utility; there are examples of both actions in this standard. For this utility, it chose to leave some of the options as they are because of their heavy 1 usage by existing applications. However, due to interest in the international community, the developers of the standard have agreed to provide an alternative syntax for the next version of this standard that conforms to the spirit of the Utility Syntax Guidelines. This new syntax will be accompanied by the existing syntax, marked as obsolescent. System implementors are encouraged to develop and promulgate a new syntax for `pr`, perhaps using a different utility name, that can be adopted for the next version of this standard.

Implementations are required to accept option arguments to the `−h`, `−l`, `−o`, and `−w` options whether presented as part of the same argument or as a separate argument to `pr`, as suggested by the utility syntax guidelines. The `−n` and `−s` options, however, are specified as in historical practice because they are frequently specified without their optional arguments. If a `<blank>` were allowed before the option-argument in these cases, a file operand could mistakenly be interpreted as an option-argument in historical applications.

Historical implementations of the `pr` utility have differed in the action taken for the `−f` option. BSD uses it as described here for the `−F` option; System V uses it to change trailing `<newline>`s on each page to a `<form-feed>` and, if standard output is a TTY device, sends an `<alert>` to standard error and reads a line from `/dev/tty` before the first page. Draft 9 incorrectly specified part of the System V behavior, raising several ballot objections. There were strong arguments from both sides of this issue concerning existing practice and additional arguments against the System V `−f` behavior, on the grounds that it was not a modular design to have the behavior of an option change depending on where output is directed. Therefore, the `−f` option is not specified and the `−F` option has been added.

The `−p` option was omitted since it represents a purely interactive usage. 1

The *<output of date>* field in the `−l` format is specified only for the POSIX Locale. As noted, the format can be different in other locales. No mechanism for defining this is present in this standard, as the appropriate vehicle is a messaging system; i.e., the format should be specified as a ''message.''

4.50 `printf` — Write formatted output

4.50.1 Synopsis

`printf` *format* [*argument* ...]

4.50.2 Description

The `printf` utility shall write formatted operands to the standard output. The *argument* operands shall be formatted under control of the *format* operand.

4.50.3 Options

None.

4.50.4 Operands

The following operands shall be supported by the implementation:

format A string describing the format to use to write the remaining operands; see 4.50.7.

argument The strings to be written to standard output, under the control of *format* ; see 4.50.7.

4.50.5 External Influences

4.50.5.1 Standard Input

None.

4.50.5.2 Input Files

None.

4.50.5.3 Environment Variables

The following environment variables shall affect the execution of `printf`:

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.