IEEE P1003.2a Draft 8 – December 1991

5.17 `mesg` — Permit or deny messages

5.17.1 Synopsis

```
mesg [ y | n ]
```

5.17.2 Description

The `mesg` utility shall control whether other users are allowed to send messages
via `write`, `talk`, or other utilities to a terminal device. The terminal device 8
affected shall be determined by searching for the first terminal in the sequence of 8
devices associated with standard input, standard output, and standard error, 8
respectively. With no arguments, `mesg` shall report the current state without 8
changing it. Processes with appropriate privileges may be able to send messages
to the terminal independent of the current state.

5.17.3 Options

None.

5.17.4 Operands

The following operands shall be supported by the implementation in the POSIX Locale:

`y` Grant permission to other users to send messages to the terminal device. 8

`n` Deny permission to other users to send messages to the terminal device. 8

5.17.5 External Influences

5.17.5.1 Standard Input

None.

5.17.5.2 Input Files

None.

5.17.5.3 Environment Variables

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

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).

LC_MESSAGES This variable shall determine the language in which messages (from the `mesg` utility) are displayed.

5.17.5.4 Asynchronous Events

Default.

5.17.6 External Effects

5.17.6.1 Standard Output

If no operand is specified, `mesg` shall display the current terminal state in an unspecified format.

5.17.6.2 Standard Error

Used only for diagnostic messages.

5.17.6.3 Output Files

None.

5.17.7 Extended Description

None.

5.17.8 Exit Status

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

0 Receiving messages is allowed.

1 Receiving messages is not allowed.

>1 An error occurred.

5.17.9 Consequences of Errors

Default.

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

Usage, Examples

The terminal changed by `mesg` is that associated with the standard input, output, 8
or error, rather than the controlling terminal for the session. This is because 8
users logged in more than once should be able to change any of their login terminals without having to stop the job running in those sessions. This is not a security problem involving the terminals of other users because appropriate privileges would be required to affect another user's terminals.

History of Decisions Made

The method of checking each of the first three file descriptors in sequence until a 8
terminal is found was adopted from System V. 8

The file `/dev/tty` is not specified for the terminal device because it was thought to be too restrictive. Typical environment changes for the `n` operand are that write permissions are removed for *others* and *group* from the appropriate device. It was decided to leave the actual description of what is done as unspecified because of potential differences between implementations.

The format for standard output is unspecified because of differences between historical implementations. This output is generally not useful to shell scripts (they can use the exit status), so exact parsing of the output is unnecessary.

The mechanism by which the message status of the terminal is changed is unspecified. Therefore, unspecified actions may cause the status of the terminal to change after `mesg` has successfully completed. These actions may include, but are not limited to: another invocation of the `mesg` utility; login procedures; invocation of the `stty` utility; invocation of the `chmod` utility; etc.

5.18 `more` — Display files on a page-by-page basis

5.18.1 Synopsis

`more` [−ceisu] [−n *number*] [−t *tagstring*] [−p *command*] [*file* ...] 8

Obsolescent Version:

`more` [−ceisu] [−n *number*] [+/*command*] [−t *tagstring*] [*file* ...] 8

5.18.2 Description

The `more` utility shall read files and either write them to the terminal on a page-by-page basis or filter them to standard output. If standard output is not a terminal device, all input files shall be copied to standard output in their entirety, 8 without modification. If standard output is a terminal device, the files shall be 8 written a number of lines (one ‘‘screenful’’) at a time under the control of user commands; see 5.18.7.

Certain block-mode terminals do not have all the capabilities necessary to support the complete `more` definition; they are incapable of accepting commands that are not terminated with a <newline>. Implementations that support such terminals shall provide an operating mode to `more` in which all commands can be terminated with a <newline> on those terminals. This mode shall:

— Be documented in the system documentation. 7

— At invocation, inform the user of the terminal deficiency that requires the <newline> usage and provide instructions on how this warning can be suppressed in future invocations.

— Not be required for implementations supporting only fully capable terminals.

— Not affect commands already requiring <newline>s.

— Not affect users on the capable terminals from using `more` as described in this standard.

5.18.3 Options

The `more` utility shall conform to the utility argument syntax guidelines described in 2.10.2, except that +/*command* of the obsolescent version uses a non-standard syntax.

The following options shall be supported by the implementation:

−c If a screen is to be written that has no lines in common with the current screen, or `more` is writing its first screen, `more` shall not scroll the screen, but instead shall redraw each line of the screen in turn, from the top of the screen to the bottom. In addition, if