IEEE P1003.2a Draft 8 – December 1991

worded to allow this implementation. Consideration was given to invalidating this approach because of the historical model from the C-shell and System V shell. However, since the System V `time` historically has not produced accurate results in pipeline timing (because the constituent processes are not all owned by the same parent process, as allowed by POSIX), it did not seem worthwhile to break existing KornShell usage.

The term *utility* is used, rather than *command*, to highlight the fact that shell compound commands, pipelines, special built-ins, etc., cannot be used directly. However, *utility* includes user application programs and shell scripts, not just the standard utilities.

5.30 `tput` — Change terminal characteristics

5.30.1 Synopsis

`tput [−T` *type* `]` *operand* . . .

5.30.2 Description

The `tput` utility shall display terminal-dependent information. The manner in which this information is retrieved is unspecified. The information displayed shall clear the terminal screen, initialize the user's terminal, or reset the user's terminal, depending on the operand given. The exact consequences of displaying this information are unspecified.

5.30.3 Options

The `tput` utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following option shall be supported by the implementation:

`−T` *type* Indicate the type of terminal. If this option is not supplied and the **TERM** variable is unset or null, an unspecified default terminal type shall be used. The setting of *type* shall take precedence over the value in **TERM**.

5.30.4 Operands

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

`clear` Display the clear-screen sequence.

`init` Display the sequence that will initialize the user's terminal in an implementation-defined manner.

`reset` Display the sequence that will reset the user's terminal in an implementation-defined manner.

If a terminal does not support any of the operations described by these operands, this shall not be considered an error condition.

5.30.5 External Influences

5.30.5.1 Standard Input

None.

5.30.5.2 Input Files

None.

5.30.5.3 Environment Variables

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

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 should be displayed.

TERM This variable shall be interpreted as the name of the terminal type. If this variable is unset or null, and if the `-T` 7
option is not specified, an unspecified default terminal 7
type shall be used. 7

5.30.5.4 Asynchronous Events

Default.

5.30.6 External Effects

5.30.6.1 Standard Output

If standard output is a terminal device, it may be used for writing the appropriate sequence to clear the screen or reset or initialize the terminal. If standard output is not a terminal device, undefined results occur.

5.30.6.2 Standard Error

Used only for diagnostic messages.

5.30.6.3 Output Files

None.

5.30.7 Extended Description

None.

5.30.8 Exit Status

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

0 The requested string was written successfully.

1 Unspecified.

2 Usage error.

3 No information is available about the specified terminal type.

4 The specified operand is invalid.

>4 An error occurred.

5.30.9 Consequences of Errors

If one of the operands is not available for the terminal, `tput` continues processing the remaining operands.

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

Usage, Examples

Initialize the terminal according to the type of terminal in the environmental variable **TERM**. This command can be included in a `.profile` file.

```
tput init
```

Reset a 450 terminal.

```
tput -T 450 reset
```

Some terminals use control characters to perform the stated functions, and on such terminals it might make sense to use `tput` to store the initialization strings in a file or environment variable for later use. However, because other terminals might rely on system calls to do this work, the standard output cannot be used in a portable manner, such as the following nonportable constructs:

```
ClearVar=‘tput clear‘
tput reset | mailx -s "Wake Up" ddg
```

History of Decisions Made

The list of operands was pared down to a bare minimum for the following reasons:

(1) The only features chosen were those that were likely to be used by human users interacting with a terminal.

(2) Specifying the full *terminfo* set was not considered desirable, but the working group did not want to get into a ‘‘my favorite operand is . . .’’ dispute with balloters.

(3) This standard is not attempting to provide applications with fancy terminal handling capabilities, as that falls outside of its assigned scope, and intersects with the responsibilities of other standards bodies.

The difference between resetting and initializing a terminal is left unspecified, as they vary greatly based on hardware types. In general, resetting is a more severe action.

Exit status of 1 is historically reserved for finding out if a Boolean operand is not set. Although the operands were pared down to a minimum set, the exit status of 1 should still be reserved for the Boolean operands, for those sites that wish to support them.

5.31 unalias — Remove alias definitions

5.31.1 Synopsis

```
unalias alias-name ...
unalias –a
```

5.31.2 Description

The `unalias` utility shall remove the definition for each alias name specified. See 3.3.1. The aliases shall be removed from the current shell execution environment; see 3.12.

5.31.3 Options

The `unalias` utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following option shall be supported by the implementation:

`–a` Remove all alias definitions from the current shell execution environment.

5.31.4 Operands

The following operand shall be supported by the implementation:

alias-name The name of an alias to be removed.

5.31.5 External Influences

5.31.5.1 Standard Input

None.

5.31.5.2 Input Files

None.

5.31.5.3 Environment Variables

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