-- --

IEEE P1003.2 Draft 11.2 – September 1991

4.66 `tty` — Return user's terminal name

4.66.1 Synopsis

```
tty
```

Obsolescent Version :

```
tty −s
```

4.66.2 Description

The `tty` utility shall write to the standard output the name of the terminal that is open as standard input. The name that is used shall be equivalent to the string that would be returned by the POSIX.1 {8} *ttyname* () function.

4.66.3 Options

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

The following option shall be supported by the implementation:

`−s` (Obsolescent.) Do not write the terminal name. Only the exit status shall be affected by this option. The terminal status shall be determined as if the POSIX.1 {8} *isatty* () function were used.

4.66.4 Operands

None.

4.66.5 External Influences

4.66.5.1 Standard Input

While no input is read from standard input, standard input shall be examined to determine whether or not it is a terminal, and/or to determine the name of the terminal.

4.66.5.2 Input Files

None.

4.66.5.3 Environment Variables

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

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 For the obsolescent version, 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 written.

4.66.5.4 Asynchronous Events

Default.

4.66.6 External Effects

4.66.6.1 Standard Output

If the `-s` option is specified, standard output shall not be used. If the `-s` option is not specified and standard input is a terminal device, a pathname of the terminal as specified by POSIX.1 {8} *ttyname*() shall be written in the following format:

`"%s\n"`, <*terminal name*>

Otherwise, a message shall be written indicating that standard input is not connected to a terminal. In the POSIX Locale, the `tty` utility shall use the format:

```
"not a tty\n"
```

4.66.6.2 Standard Error

Used only for diagnostic messages.

4.66.6.3 Output Files

None.

4.66.7 Extended Description

None.

4.66.8 Exit Status

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

0	Standard input is a terminal.
1	Standard input is not a terminal.
>1	An error occurred.

4.66.9 Consequences of Errors

Default.

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

Examples, Usage

This utility checks the status of the file open as standard input against that of a system-defined set of files. It is possible that no match can be found, or that the match found need not be the same file as that which was opened for standard input (although they are the same device).

The `−s` option is useful only if the exit code is wanted. It does not rely on the ability to form a valid pathname. The `−s` option was made obsolescent because the same functionality is provided by `test -t 0`, but not dropped completely because historical scripts depend on this form.

History of Decisions Made

The definition of `tty` was made more explicit to explain the difference between a tty and a pathname of a tty.

4.67 `umask` — Get or set the file mode creation mask

4.67.1 Synopsis

`umask [–S] [`*mask*`]`

4.67.2 Description

The `umask` utility shall set the file mode creation mask of the current shell execution environment (see 3.12) to the value specified by the *mask* operand. This mask shall affect the initial value of the file permission bits of subsequently created files.

If the *mask* operand is not specified, the `umask` utility shall write to standard output the value of the invoking process's file mode creation mask.

4.67.3 Options

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

The following option shall be supported by the implementation:

`–S` Produce symbolic output.

The default output style is unspecified, but shall be recognized on a subsequent invocation of `umask` on the same system as a *mask* operand to restore the previous file mode creation mask.

4.67.4 Operands

The following operand shall be supported by the implementation:

mask A string specifying the new file mode creation mask. The string is treated in the same way as the *mode* operand described in 4.7.7 (`chmod` Extended Description).

For a *symbolic_mode* value, the new value of the file mode creation mask shall be the logical complement of the file permission bits portion of the file mode specified by the *symbolic_mode* string.

In a *symbolic_mode* value, the permissions *op* characters `+` and `-` shall be interpreted relative to the current file mode creation mask; `+` shall cause the bits for the indicated permissions to be cleared in the mask; `-` shall cause the bits for the indicated permissions to be set in the mask.