IEEE P1003.2 Draft 11.2 – September 1991

2.7 Required Files

The following directories shall exist on conforming systems and shall be used as described. Strictly Conforming POSIX.2 Applications shall not assume the ability to create files in any of these directories.

`/` The root directory.

`/dev` Contains `/dev/null` and `/dev/tty`, described below.

The following directory shall exist on conforming systems and shall be used as described.

`/tmp` A directory made available for programs that need a place to create temporary files. Applications shall be allowed to create files in this directory, but shall not assume that such files are preserved between invocations of the application.

The following files shall exist on conforming systems and shall be both readable and writable.

`/dev/null` An infinite data source/sink. Data written to `/dev/null` is discarded. Reads from `/dev/null` always return end-of-file (EOF).

`/dev/tty` In each process, a synonym for the controlling terminal associated with the process group of that process, if any. It is useful for programs or shell procedures that wish to be sure of writing messages to or reading data from the terminal no matter how output has been redirected.

2.7.1 Required Files Rationale. *(This subclause is not a part of P1003.2)*

A description of the historical `/usr/tmp` was omitted, removing any concept of differences in emphasis between the `/` and `/usr` versions. The descriptions of `/bin`, `/usr/bin`, `/lib`, and `/usr/lib` were omitted because they are not useful for applications. In an early draft, a distinction was made between *system* and *application* directory usage, but this was not found to be useful.

In Draft 8, `/`, `/dev`, `/local`, `/usr/local`, and `/usr/man` were removed. The directories `/` and `/dev` were restored in Draft 9. It was pointed out by several balloters that the notion of a hierarchical directory structure is key to other information presented in later sections of the standard. (Previously, some had argued that special devices and temporary files could conceivably be handled without a directory structure on some implementations. For example, the system could treat the characters ‘‘`/tmp`’’ as a special token that would store files using some non-POSIX file system structure. This notion was rejected by the working group, which requires that all the files in this clause be implemented via POSIX file systems.)

The `/tmp` directory is retained in the standard to accommodate historical applications that assume its availability. Future implementations are encouraged to provide suitable directory names in **TMPDIR** and future applications are encouraged to use the contents of **TMPDIR** for creating temporary files.

The standard files `/dev/null` and `/dev/tty` are required to be both readable and writable to allow applications to have the intended historical access to these files.

2.8 Regular Expression Notation

Editor's Note: The entire rationale for this clause appears at the end of the clause.

Regular Expressions (REs) provide a mechanism to select specific strings from a set of character strings.

Regular expressions are a context-independent syntax that can represent a wide variety of character sets and character set orderings, where these character sets are interpreted according to the current locale. While many regular expressions can be interpreted differently depending on the current locale, many features, such as character class expressions, provide for contextual invariance across locales.

The Basic Regular Expression (BRE) notation and construction rules in 2.8.3 shall apply to most utilities supporting regular expressions. Some utilities, instead, support the Extended Regular Expressions (ERE) described in 2.8.4; any exceptions for both cases are noted in the descriptions of the specific utilities using regular expressions. Both BREs and EREs are supported by the Regular Expression Matching interface in 7.3.

2.8.1 Regular Expression Definitions

For the purposes of this clause, the following definitions apply.

2.8.1.1 entire regular expression: The concatenated set of one or more BREs or EREs that make up the pattern specified for string selection.

2.8.1.2 matched: A sequence of zero or more characters is said to be matched by a BRE or ERE when the characters in the sequence corresponds to a sequence of characters defined by the pattern.

Matching shall be based on the bit pattern used for encoding the character, not on 1
the graphic representation of the character. 1

The search for a matching sequence shall start at the beginning of a string and
stop when the first sequence matching the expression is found, where ''first'' is
defined to mean ''begins earliest in the string.'' If the pattern permits a variable
number of matching characters and thus there is more than one such sequence
starting at that point, the longest such sequence shall be matched. For example: 1
the BRE `bb*` matches the second through fourth characters of `abbbc`, and the ERE 1
`(wee|week)(knights|night)` matches all ten characters of `weeknights`. 1