Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'fnmatch.3p' command

$ man fnmatch.3p

FNMATCH(3P) POSIX Programmer's Manual FNMATCH(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

fnmatch ? match a filename string or a pathname

SYNOPSIS

#include <fnmatch.h>

int fnmatch(const char *pattern, const char *string, int flags);

DESCRIPTION

The fnmatch() function shall match patterns as described in the Shell and Utilities volume of POSIX.1?2017, Section 2.13.1, Patterns Matching a Single Character and Section 2.13.2, Patterns Matching Multiple Char? acters. It checks the string specified by the string argument to see if it matches the pattern specified by the pattern argument.

The flags argument shall modify the interpretation of pattern and string. It is the bitwise-inclusive OR of zero or more of the flags defined in <fnmatch.h>. If the FNM_PATHNAME flag is set in flags, then a <slash> character ('/') in string shall be explicitly matched by a <slash> in pattern; it shall not be matched by either the <asterisk> or <question-mark> special characters, nor by a bracket expression. If the

FNM_PATHNAME flag is not set, the <slash> character shall be treated as an ordinary character.

If FNM_NOESCAPE is not set in flags, a <backslash> character in pattern followed by any other character shall match that second character in string. In particular, "\\" shall match a <backslash> in string. If pattern ends with an unescaped <backslash>, fnmatch() shall return a non-zero value (indicating either no match or an error). If FNM_NOESCAPE is set, a <backslash> character shall be treated as an or? dinary character.

If FNM_PERIOD is set in flags, then a leading <period> ('.') in string shall match a <period> in pattern; as described by rule 2 in the Shell and Utilities volume of POSIX.1?2017, Section 2.13.3, Patterns Used for Filename Expansion where the location of ``leading'' is indicated by the value of FNM_PATHNAME:

* If FNM_PATHNAME is set, a <period> is ``leading'' if it is the first character in string or if it immediately follows a <slash>.
* If FNM_PATHNAME is not set, a <period> is ``leading'' only if it is the first character of string.

If FNM_PERIOD is not set, then no special restrictions are placed on matching a period.

RETURN VALUE

If string matches the pattern specified by pattern, then fnmatch() shall return 0. If there is no match, fnmatch() shall return FNM_NO? MATCH, which is defined in <fnmatch.h>. If an error occurs, fnmatch() shall return another non-zero value.

ERRORS

No errors are defined.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The fnmatch() function has two major uses. It could be used by an ap? plication or utility that needs to read a directory and apply a pattern

against each entry. The find utility is an example of this. It can also be used by the pax utility to process its pattern operands, or by ap? plications that need to match strings in a similar manner.

The name fnmatch() is intended to imply filename match, rather than pathname match. The default action of this function is to match file? name strings, rather than pathnames, since it gives no special signifi? cance to the <slash> character. With the FNM_PATHNAME flag, fnmatch() does match pathnames, but without tilde expansion, parameter expansion, or special treatment for a <period> at the beginning of a filename.

RATIONALE

This function replaced the REG_FILENAME flag of regcomp() in early pro? posals of this volume of POSIX.1?2017. It provides virtually the same functionality as the regcomp() and regexec() functions using the REG_FILENAME and REG_FSLASH flags (the REG_FSLASH flag was proposed for regcomp(), and would have had the opposite effect from FNM_PATHNAME), but with a simpler function and less system overhead.

FUTURE DIRECTIONS

None.

SEE ALSO

glob(), Section 2.6, Word Expansions

The Base Definitions volume of POSIX.1?2017, <fnmatch.h>

files to man page format. To report such errors, see https://www.ker?

nel.org/doc/man-pages/reporting_bugs.html .