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 'basename.1p' command

$ man basename.1p

BASENAME(1P) POSIX Programmer's Manual BASENAME(1P)

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

basename ? return non-directory portion of a pathname

SYNOPSIS

basename string [suffix]

DESCRIPTION

The string operand shall be treated as a pathname, as defined in the Base Definitions volume of POSIX.1?2017, Section 3.271, Pathname. The string string shall be converted to the filename corresponding to the last pathname component in string and then the suffix string suffix, if present, shall be removed. This shall be done by performing actions equivalent to the following steps in order:

1. If string is a null string, it is unspecified whether the resulting string is '.' or a null string. In either case, skip steps 2 through 6.
2. If string is "//", it is implementation-defined whether steps 3 to 6 are skipped or processed.
3. If string consists entirely of <slash> characters, string shall be

set to a single <slash> character. In this case, skip steps 4 to 6.

4. If there are any trailing <slash> characters in string, they shall be removed.

5. If there are any <slash> characters remaining in string, the prefix of string up to and including the last <slash> character in string shall be removed.

6. If the suffix operand is present, is not identical to the charac? ters remaining in string, and is identical to a suffix of the char? acters remaining in string, the suffix suffix shall be removed from string. Otherwise, string is not modified by this step. It shall not be considered an error if suffix is not found in string.

The resulting string shall be written to standard output.

OPTIONS

None.

OPERANDS

The following operands shall be supported:

string A string.

suffix A string.

STDIN

Not used.

INPUT FILES

None.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of base? name:

LANG Provide a default value for the internationalization vari? ables that are unset or null. (See the Base Definitions vol? ume of POSIX.1?2017, Section 8.2, Internationalization Vari? ables for the precedence of internationalization variables used to determine the values of locale categories.)

LC_ALL If set to a non-empty string value, override the values of all the other internationalization variables.

LC_CTYPE Determine the locale for the interpretation of sequences of

bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

LC_MESSAGES

Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard er? ror.

NLSPATH Determine the location of message catalogs for the processing of LC_MESSAGES.

ASYNCHRONOUS EVENTS

Default.

STDOUT

The basename utility shall write a line to the standard output in the following format:

"%s\n", <resulting string>

STDERR

The standard error shall be used only for diagnostic messages.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

0 Successful completion.

>0 An error occurred.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

The definition of pathname specifies implementation-defined behavior for pathnames starting with two <slash> characters. Therefore, applica? tions shall not arbitrarily add <slash> characters to the beginning of a pathname unless they can ensure that there are more or less than two or are prepared to deal with the implementation-defined consequences.

EXAMPLES

If the string string is a valid pathname:

$(basename -- "string")

produces a filename that could be used to open the file named by string in the directory returned by:

$(dirname -- "string")

If the string string is not a valid pathname, the same algorithm is used, but the result need not be a valid filename. The basename utility is not expected to make any judgements about the validity of string as a pathname; it just follows the specified algorithm to produce a result string.

The following shell script compiles /usr/src/cmd/cat.c and moves the output to a file named cat in the current directory when invoked with the argument /usr/src/cmd/cat or with the argument /usr/src/cmd/cat.c:

```
c99 -- "$(dirname -- "$1")/$(basename -- "$1" .c).c" &&
mv a.out "$(basename -- "$1" .c)"
```

The EXAMPLES section of the basename() function (see the System Inter?faces volume of POSIX.1?2017, basename()) includes a table showing ex?amples of the results of processing several sample pathnames by the basename() and dirname() functions and by the basename and dirname utilities.

RATIONALE

The behaviors of basename and dirname have been coordinated so that when string is a valid pathname:

$(basename -- "string")

would be a valid filename for the file in the directory:

$(dirname -- "string")

This would not work for the early proposal versions of these utilities due to the way it specified handling of trailing <slash> characters.

Since the definition of pathname specifies implementation-defined be?havior for pathnames starting with two <slash> characters, this volume of POSIX.1?2017 specifies similar implementation-defined behavior for the basename and dirname utilities.

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.5, Parameters and Variables, dirname

The Base Definitions volume of POSIX.1?2017, Section 3.271, Pathname, Chapter 8, Environment Variables

The System Interfaces volume of POSIX.1?2017, basename(), dirname()

COPYRIGHT