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 'nl_langinfo.3p' command

$ man nl_langinfo.3p

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

nl_langinfo, nl_langinfo_l ? language information

SYNOPSIS

```
#include <langinfo.h>
char *nl_langinfo(nl_item item);
char *nl_langinfo_l(nl_item item, locale_t locale);
```

DESCRIPTION

The nl_langinfo() and nl_langinfo_l() functions shall return a pointer to a string containing information relevant to the particular language or cultural area defined in the current locale, or in the locale repre? sented by locale, respectively (see <langinfo.h>). The manifest con? stant names and values of item are defined in <langinfo.h>. For exam? ple:

nl_langinfo(ABDAY_1)

would return a pointer to the string "Dom" if the identified language was Portuguese, and "Sun" if the identified language was English.

nl_langinfo_l(ABDAY_1, loc)

would return a pointer to the string "Dom" if the identified language of the locale represented by loc was Portuguese, and "Sun" if the iden? tified language of the locale represented by loc was English.

The nl_langinfo() function need not be thread-safe.

The behavior is undefined if the locale argument to nl_langinfo_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale ob? ject handle.

RETURN VALUE

In a locale where langinfo data is not defined, these functions shall return a pointer to the corresponding string in the POSIX locale. In all locales, these functions shall return a pointer to an empty string if item contains an invalid setting.

The application shall not modify the string returned. The pointer re? turned by nl_langinfo() might be invalidated or the string content might be overwritten by a subsequent call to nl_langinfo() in any thread or to nl_langinfo_l() in the same thread or the initial thread, by subsequent calls to setlocale() with a category corresponding to the category of item (see <langinfo.h>) or the category LC_ALL, or by sub? sequent calls to uselocale() which change the category corresponding to the category of item. The pointer returned by nl_langinfo_l() might be invalidated or the string content might be overwritten by a subsequent call to nl_langinfo_l() in the same thread or to nl_langinfo() in any thread, or by subsequent calls to freelocale() or newlocale() which free or modify the locale object that was passed to nl_langinfo_l().

The returned pointer and the string content might also be invalidated if the calling thread is terminated.

ERRORS

No errors are defined.

The following sections are informative.

EXAMPLES

Getting Date and Time Formatting Information

The following example returns a pointer to a string containing date and time formatting information, as defined in the LC_TIME category of the

current locale.

```
#include <time.h>
#include <langinfo.h>
...
strftime(datestring, sizeof(datestring), nl_langinfo(D_T_FMT), tm);
...
```

APPLICATION USAGE

The array pointed to by the return value should not be modified by the program, but may be modified by further calls to these functions.

RATIONALE

The possible interactions between internal data used by nl_langinfo() and nl_langinfo_l() are complicated by the fact that nl_langinfo_l() must be thread-safe but nl_langinfo() need not be. The various imple? mentation choices are:

1. nl_langinfo_l() and nl_langinfo() use separate buffers, or at least one of them does not use an internal string buffer. In this case there are no interactions.
2. nl_langinfo_l() and nl_langinfo() share an internal per-thread buf? fer. There can be interactions, but only in the same thread.
3. nl_langinfo_l() uses an internal per-thread buffer, and nl_lang? info() uses (in all threads) the same buffer that nl_langinfo_l() uses in the initial thread. There can be interactions, but only when nl_langinfo_l() is called in the initial thread.

FUTURE DIRECTIONS

None.

SEE ALSO

setlocale(), uselocale()

The Base Definitions volume of POSIX.1?2017, Chapter 7, Locale, <lang? info.h>, <locale.h>, <nl_types.h>