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 'localtime.3p' command

$ man localtime.3p

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

localtime, localtime_r ? convert a time value to a broken-down local time

SYNOPSIS

```
#include <time.h>
struct tm *localtime(const time_t *timer);
struct tm *localtime_r(const time_t *restrict timer,
    struct tm *restrict result);
```

DESCRIPTION

For localtime(): The functionality described on this reference page is aligned with the ISO C standard. Any conflict between the requirements described here and the ISO C standard is unintentional. This volume of POSIX.1?2017 defers to the ISO C standard.

The localtime() function shall convert the time in seconds since the Epoch pointed to by timer into a broken-down time, expressed as a local time. The function corrects for the timezone and any seasonal time ad? justments. Local timezone information is used as though localtime()

calls tzset().

The relationship between a time in seconds since the Epoch used as an argument to localtime() and the tm structure (defined in the <time.h> header) is that the result shall be as specified in the expression given in the definition of seconds since the Epoch (see the Base Defi? nitions volume of POSIX.1?2017, Section 4.16, Seconds Since the Epoch) corrected for timezone and any seasonal time adjustments, where the names in the structure and in the expression correspond.

The same relationship shall apply for localtime_r().

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

The asctime(), ctime(), gmtime(), and localtime() functions shall re? turn values in one of two static objects: a broken-down time structure and an array of type char. Execution of any of the functions may over? write the information returned in either of these objects by any of the other functions.

The localtime_r() function shall convert the time in seconds since the Epoch pointed to by timer into a broken-down time stored in the struc? ture to which result points. The localtime_r() function shall also re? turn a pointer to that same structure.

Unlike localtime(), the localtime_r() function is not required to set tzname. If localtime_r() sets tzname, it shall also set daylight and timezone. If localtime_r() does not set tzname, it shall not set day? light and shall not set timezone.

RETURN VALUE

Upon successful completion, the localtime() function shall return a pointer to the broken-down time structure. If an error is detected, localtime() shall return a null pointer and set errno to indicate the error.

Upon successful completion, localtime_r() shall return a pointer to the structure pointed to by the argument result. If an error is detected, localtime_r() shall return a null pointer and set errno to indicate the error.

ERRORS

The localtime() and localtime_r() functions shall fail if:

EOVERFLOW

The result cannot be represented.

The following sections are informative.

EXAMPLES

Getting the Local Date and Time

The following example uses the time() function to calculate the time elapsed, in seconds, since January 1, 1970 0:00 UTC (the Epoch), local? time() to convert that value to a broken-down time, and asctime() to convert the broken-down time values into a printable string.

```
#include <stdio.h>
#include <time.h>
int main(void)
{
    time_t result;
    result = time(NULL);
    printf("%s%ju secs since the Epoch\n",
        asctime(localtime(&result)),
            (uintmax_t)result);
    return(0);
}
```

This example writes the current time to stdout in a form like this:

```
Wed Jun 26 10:32:15 1996
835810335 secs since the Epoch
```

Getting the Modification Time for a File

The following example prints the last data modification timestamp in the local timezone for a given file.

```
#include <stdio.h>
#include <time.h>
#include <sys/stat.h>
int
print_file_time(const char *pathname)
{
```

```
    struct stat statbuf;
    struct tm *tm;
    char timestr[BUFSIZ];
    if(stat(pathname, &statbuf) == -1)
        return -1;
    if((tm = localtime(&statbuf.st_mtime)) == NULL)
        return -1;
    if(strftime(timestr, sizeof(timestr), "%Y-%m-%d %H:%M:%S", tm) == 0)
        return -1;
    printf("%s: %s.%09ld\n", pathname, timestr, statbuf.st_mtim.tv_nsec);
    return 0;
}
```

Timing an Event

The following example gets the current time, converts it to a string using localtime() and asctime(), and prints it to standard output using fputs(). It then prints the number of minutes to an event being timed.

```
#include <time.h>
#include <stdio.h>
...
time_t now;
int minutes_to_event;
...
time(&now);
printf("The time is ");
fputs(asctime(localtime(&now)), stdout);
printf("There are still %d minutes to the event.\n",
    minutes_to_event);
...
```

APPLICATION USAGE

The localtime_r() function is thread-safe and returns values in a user-supplied buffer instead of possibly using a static data area that may be overwritten by each call.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

asctime(), clock(), ctime(), difftime(), getdate(), gmtime(), mktime(), strftime(), strptime(), time(), tzset(), utime()

The Base Definitions volume of POSIX.1?2017, Section 4.16, Seconds Since the Epoch, <time.h>

COPYRIGHT