

Times, times and times in Unix (calendar or wall-clock view)

J. Magalhães Cruz Nov.2024

Disclaimer.....	1
Basics.....	1
Data structures.....	3
Functions.....	4
Functions - getting:.....	4
Functions - converting:.....	4

Disclaimer

1. some data structures and functions are simplified, in the sense that only the most common use is given
2. the C types of several `typedef` units can (and will) vary from system to system

Basics

Epoch:
1970-01-01 00:00:00 (UTC) (yyyy-mm-dd hh:mm:ss)

UTC:
Universal Time Coordinated , Universel Temps Coordonné (compromise initialism between English and French!)

Real time (elapsed time):
amount of time since some point (e.g. the Epoch). Units: seconds+microseconds+nanoseconds...

Broken-down time:
time value separated out into components (year, month, day, hour, minute, second, etc.). Existing functions consider time since 1900!

Process time:
amount of processor (CPU) time used by a process. Units: `clock_t` (clock ticks)
---> to be added!

```
$ man 7 time
$ man system_data_types
$ man ? *
```

(deliberately left blank)

Data structures

name	explanation	declaration
time_t	number of seconds elapsed since some point	<pre>#typedef time_t long // int, ...</pre>
suseconds_t	number of microseconds [type: long, int, ...]	<pre>#typedef suseconds_t long // int, ...</pre>
struct timeval	Describes time in seconds and microseconds (possibly elapsed since some point)	<pre>struct timeval { time_t tv_sec; // seconds suseconds_t tv_usec; // microseconds };</pre>
struct timezone	Additional to struct timeval for time zone information	<pre>struct timezone { int tz_minuteswest; // minutes west of Greenwich int tz_dsttime; // type of Daylight Saving Time correction };</pre>
struct timespec	Describes time in seconds and nanoseconds (possibly elapsed since some point)	<pre>struct timespec { time_t tv_sec; // seconds long tv_nsec; // nanoseconds };</pre>
struct tm	Presents "broken-down" time, elapsed since 1900	<pre>struct tm { int tm_sec; // Seconds [0, 60] int tm_min; // Minutes [0, 59] int tm_hour; // Hour [0, 23] int tm_mday; // Day of the month [1, 31] int tm_mon; // Month [0, 11] (January = 0) int tm_year; // Year minus 1900 int tm_wday; // Day of the week [0, 6] (Sunday = 0) int tm_yday; // Day of the year [0, 365] (Jan/01 = 0) int tm_isdst; // Daylight Saving Time flag long tm_gmtoff; // Seconds East of UTC char *tm_zone; // Timezone abbreviation };</pre>
struct timeb	Describes time in seconds and milliseconds (possibly elapsed since some point)	<pre>struct timeb { time_t time; // seconds unsigned short millitm; // milliseconds short timezone; // minutes west of Greenwich short dstflag; // Daylight Saving Time flag };</pre>

Functions

Functions - getting:

get function	result	usage
time()	current seconds since the Epoch	<pre>time_t t = time(NULL); printf("\nCurrent seconds since Epoch: %ld.\n", t);</pre>
gettimeofday()	current seconds and microseconds since the Epoch	<pre>struct timeval tv; gettimeofday(&tv, NULL); printf("\nCurrent seconds+microsecs since Epoch: %ld+%ld.\n", tv.tv_sec, tv.tv_usec);</pre>
clock_gettime()	current seconds and nanoseconds since the Epoch	<pre>struct timespec ts; clock_gettime(CLOCK_REALTIME, &ts); printf("\nCurrent seconds+nanosecs since Epoch: %ld+%ld.\n", ts.tv_sec, ts.tv_nsec);</pre>
ftime()	current seconds and milliseconds since the Epoch	<pre>struct timeb tb; ftime(&tb); printf("\nCurrent seconds+millisecs since Epoch: %ld+%hu.\n", tb.time, tb.millitm);</pre>

Functions - converting:

from	convert function	to	usage
seconds since Epoch	ctime()	string with current time format: "Fri Nov 15 17:54:32 2024\n"	<pre>time_t t = time(NULL); printf("\nCurrent date/time: %s.\n", ctime(&t));</pre>
seconds since Epoch	localtime()	broken-down time representation since 1900	<pre>time_t t = time(NULL); struct tm *ptm = localtime(&t); printf("\nCurrent years+months+days+hours+mins+secs since 1900: %d+%d+%d+%d+%d.\n", ptm->tm_year, ptm->tm_mon, ptm->tm_mday, ptm->tm_hour, ptm->tm_min, ptm->tm_sec);</pre>
broken-down time representation since 1900	asctime()	string with current time format: "Fri Nov 15 17:54:32 2024\n"	<pre>time_t t = time(NULL); struct tm *ptm = localtime(&t); printf("\nCurrent date/time: %s\n", asctime(ptm));</pre>
broken-down time representation since 1900	strftime()	string with current time in chosen format	<pre>char str[256]; time_t t = time(NULL); struct tm *ptm = localtime(&t); strftime(str, 256, "%Y+%m+%d+%H+%M+%S", ptm); printf("\nCurrent years+months+days+hours+mins+secs: %s.\n", str);</pre>
broken-down time representation since 1900	mktime()	seconds since Epoch	<pre>time_t t = time(NULL); struct tm *ptm = localtime(&t); printf("\nCurrent seconds since Epoch: %l\n", mktime(ptm)); // Note: this function changes its argument!</pre>