Implementing “Time Function”

Struct Timespec

The *<time.h>* header declares the structure **timespec**, which has at least the following members:

time\_t tv\_sec seconds

long tv\_nsec nanoseconds

The following manifest constants are defined:

CLOCK\_REALTIME

The identifier of the systemwide realtime clock.

TIMER\_ABSTIME

Flag indicating time is absolute with respect to the clock associated with a timer.

The **clock\_t**, **clockid\_t**, **size\_t**, **time\_t** and **timer\_t** types are defined as described in [*<sys/types.h>*](http://pubs.opengroup.org/onlinepubs/7908799/xsh/systypes.h.html).

Although the value of CLOCKS\_PER\_SEC is required to be 1 million on all XSI-conformant systems, it may be variable on other systems and it should not be assumed that CLOCKS\_PER\_SEC is a compile-time constant.

The value of CLK\_TCK is currently the same as the value of *sysconf*(\_SC\_CLK\_TCK); however, new applications should call *sysconf*() because the CLK\_TCK macro may be withdrawn in a future issue.

The *<time.h>* header provides a declaration for *getdate\_err*.

The following are declared as functions and may also be defined as macros. Function prototypes must be provided for use with an ISO C compiler.

char \*[asctime](http://pubs.opengroup.org/onlinepubs/7908799/xsh/asctime.html)(const struct tm \*);

char \*[asctime\_r](http://pubs.opengroup.org/onlinepubs/7908799/xsh/asctime_r.html)(const struct tm \*, char \*);

clock\_t [clock](http://pubs.opengroup.org/onlinepubs/7908799/xsh/clock.html)(void);

int [clock\_getres](http://pubs.opengroup.org/onlinepubs/7908799/xsh/clock_getres.html)(clockid\_t, struct timespec \*);

int [clock\_gettime](http://pubs.opengroup.org/onlinepubs/7908799/xsh/clock_gettime.html)(clockid\_t, struct timespec \*);

int [clock\_settime](http://pubs.opengroup.org/onlinepubs/7908799/xsh/clock_settime.html)(clockid\_t, const struct timespec \*);

char \*[ctime](http://pubs.opengroup.org/onlinepubs/7908799/xsh/ctime.html)(const time\_t \*);

char \*[ctime\_r](http://pubs.opengroup.org/onlinepubs/7908799/xsh/ctime_r.html)(const time\_t \*, char \*);

double [difftime](http://pubs.opengroup.org/onlinepubs/7908799/xsh/difftime.html)(time\_t, time\_t);

struct tm \*[getdate](http://pubs.opengroup.org/onlinepubs/7908799/xsh/getdate.html)(const char \*);

struct tm \*[gmtime](http://pubs.opengroup.org/onlinepubs/7908799/xsh/gmtime.html)(const time\_t \*);

struct tm \*[gmtime\_r](http://pubs.opengroup.org/onlinepubs/7908799/xsh/gmtime_r.html)(const time\_t \*, struct tm \*);

struct tm \*[localtime](http://pubs.opengroup.org/onlinepubs/7908799/xsh/localtime.html)(const time\_t \*);

struct tm \*[localtime\_r](http://pubs.opengroup.org/onlinepubs/7908799/xsh/localtime_r.html)(const time\_t \*, struct tm \*);

time\_t [mktime](http://pubs.opengroup.org/onlinepubs/7908799/xsh/mktime.html)(struct tm \*);

int [nanosleep](http://pubs.opengroup.org/onlinepubs/7908799/xsh/nanosleep.html)(const struct timespec \*, struct timespec \*);

size\_t [strftime](http://pubs.opengroup.org/onlinepubs/7908799/xsh/strftime.html)(char \*, size\_t, const char \*, const struct tm \*);

char \*[strptime](http://pubs.opengroup.org/onlinepubs/7908799/xsh/strptime.html)(const char \*, const char \*, struct tm \*);

time\_t [time](http://pubs.opengroup.org/onlinepubs/7908799/xsh/time.html)(time\_t \*);

int [timer\_create](http://pubs.opengroup.org/onlinepubs/7908799/xsh/timer_create.html)(clockid\_t, struct sigevent \*, timer\_t \*);

int [timer\_delete](http://pubs.opengroup.org/onlinepubs/7908799/xsh/timer_delete.html)(timer\_t);

int [timer\_gettime](http://pubs.opengroup.org/onlinepubs/7908799/xsh/timer_gettime.html)(timer\_t, struct itimerspec \*);

int [timer\_getoverrun](http://pubs.opengroup.org/onlinepubs/7908799/xsh/timer_getoverrun.html)(timer\_t);

int [timer\_settime](http://pubs.opengroup.org/onlinepubs/7908799/xsh/timer_settime.html)(timer\_t, int, const struct itimerspec \*,

struct itimerspec \*);

void [tzset](http://pubs.opengroup.org/onlinepubs/7908799/xsh/tzset.html)(void);

Sample Usage

Strcut timespec start,end;

clock\_gettime(CLOCK\_REALTIME, &start);

Function()

clock\_gettime(CLOCK\_REALTIME, &end);

timeinseconds= end->tv\_sec - start->tv\_sec