C++
Information
Tutorials
Reference
Articles
Forum

Reference
C library:
<cassert> (assert.h)
<cctype> (ctype.h)
<cerrno> (errno.h)
<cfenv> (fenv.h)
<cfloat> (float.h)
<cinttypes> (inttypes.h)
<ciso646> (iso646.h)
<climits> (limits.h)
<clocale> (locale.h)
<cmath> (math.h)
<csetjmp> (setjmp.h)
<csignal> (signal.h)
<cstdarg> (stdarg.h)
<cstdbool> (stdbool.h)
<cstddef> (stddef.h)
<cstdint> (stdint.h)
<cstdio> (stdio.h)
<cstdlib> (stdlib.h)
<cstring> (string.h)
<ctgmath> (tgmath.h)
<ctime> (time.h)
<cuchar> (uchar.h)
<cwchar> (wchar.h)
<cwctype> (wctype.h)
Containers:
Input/Output:
Multi-threading:
Other:

<ctime> (time.h)
functions:
asctime
clock
ctime
difftime
gmtime
localtime
mktime
strftime
time
macros:
CLOCKS_PER_SEC
NULL
types:
clock_t
size_t
time_t
struct tm

function

clock

<ctime>

```
clock_t clock (void);
```

Clock program

Returns the processor time consumed by the program.

The value returned is expressed in *clock ticks*, which are units of time of a constant but system-specific length (with a relation of `CLOCKS_PER_SEC` *clock ticks* per second).

The epoch used as reference by `clock` varies between systems, but it is related to the program execution (generally its launch). To calculate the actual processing time of a program, the value returned by `clock` shall be compared to a value returned by a previous call to the same function.

Parameters

none

Return Value

The number of clock ticks elapsed since an epoch related to the particular program execution.

On failure, the function returns a value of -1.

`clock_t` is a type defined in `<ctime>` as an alias of a fundamental arithmetic type.

Example

```
/* clock example: frequency of primes */
#include <stdio.h>      /* printf */
#include <time.h>       /* clock_t, clock, CLOCKS_PER_SEC */
#include <math.h>       /* sqrt */

int frequency_of_primes (int n) {
  int i,j;
  int freq=n-1;
  for (i=2; i<=n; ++i) for (j=sqrt(i);j>1;--j) if (i%j==0) {--freq; break;}
  return freq;
}

int main ()
{
  clock_t t;
  int f;
  t = clock();
  printf ("Calculating...\n");
  f = frequency_of_primes (99999);
  printf ("The number of primes lower than 100,000 is: %d\n",f);
  t = clock() - t;
  printf ("It took me %d clicks (%f seconds).\n",t,((float)t)/CLOCKS_PER_SEC);
  return 0;
}
```

Output:

```
Calculating...
The number of primes lower than 100,000 is: 9592
It took me 143 clicks (0.143000 seconds).
```

Data races

Concurrently calling this function is safe, causing no data races.

Exceptions (C++)

No-throw guarantee: this function never throws exceptions.

See also

time	Get current time (function)
difftime	Return difference between two times (function)