How to find the execution time of a C program

In this post, we will discuss how to find the execution time of a C program in windows and linux environment.

There are four commonly used methods to find the execution time of a C program –

1. clock()

We can use `clock()` function provided by `<time.h>` header file to calculate the CPU time consumed by a task within a C application. It returns `clock_t` type which stores the number of clock ticks.

In order to compute the number of seconds elapsed, we need to divide the number of clock ticks elapsed by `CLOCKS_PER_SEC` macro (also present in `<time.h>`) as shown below:

```c
#include <stdio.h>
#include <time.h>       // for clock_t, clock(), CLOCKS_PER_SEC
#include <unistd.h>     // for sleep()

// main function to find the execution time of a C program
int main()
{
    // to store execution time of code
    double time_spent = 0.0;

    clock_t begin = clock();

    // do some stuff here
```

Browse

Adobe Algorithm Amazon BFS Binary Search Bit Hacks DFS FIFO Google Greedy Hashing Intro JSON LCS LIFO Maze Memoized Microsoft Must Know Priority Queue Probability Recursive Searching Sliding Window Tabulation Tricky Trie

```c

    // calculate elapsed time by finding difference (end - begin) and
    // dividing the difference by CLOCKS_PER_SEC to convert to seconds
    time_spent += (double)(end - begin) / CLOCKS_PER_SEC;

    printf("Time elpased is %f seconds", time_spent);

    return 0;
}
```

Download   Run Code

Output (may vary):

```
Time elpased is 0.000014 seconds
```

Please note that `clock()` function doesn't return the actual amount of time elapsed but returns the amount of time taken by the underlying operating system to run the process. In other words, the actual wall clock time might actually be much greater.

2. time()

The `<time.h>` header also provides `time()` function that returns the number of seconds elapsed since the Epoch (00:00:00 UTC, January 1, 1970). It takes pointer to `time_t` as an argument which is usually passed as `NULL` and returns `time_t` type. If the argument is not `NULL`, then the return value is also stored in the memory pointed by the argument.

It's usage is similar to `clock()` function as shown below:

```c
#include <stdio.h>
#include <time.h>       // for time()
#include <unistd.h>     // for sleep()

// main function to find the execution time of a C program
int main()
{
    time_t begin = time(NULL);

    // do some stuff here
    sleep(3);

    time_t end = time(NULL);

    // calculate elapsed time by finding difference (end - begin)
    printf("Time elpased is %d seconds", (end - begin));
```

Subscribe to posts

Enter your email address to subscribe to new posts and receive notifications of new posts

```

    return 0;
}
```

Download Run Code

Output :

```
Time elpased is 3 seconds
```

3. gettimeofday()

The `gettimeofday()` function returns the wall clock time elapsed since the Epoch and store it in the `timeval` structure, expressed as seconds and microseconds.

It is defined in `<sys/time.h>` header file and takes two arguments – the first arugment is reference to the `timeval` structure and the second argument is a null pointer. The `timeval` structure is declared as below by the `<time.h>` header:

```
struct timeval {
    long tv_sec;  /* seconds */
    long tv_usec; /* microseconds */
};
```

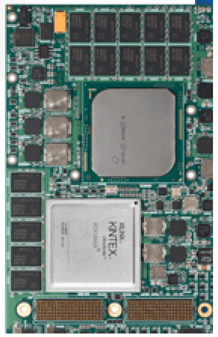
Below code demonstates the usage of `gettimeofday()` by measuring the wall clock time:

```
#include <stdio.h>
#include <sys/time.h>   // for gettimeofday()
#include <unistd.h>     // for sleep()

// main function to find the execution time of a C program
int main()
{
    struct timeval start, end;

    gettimeofday(&start, NULL);

    // do some stuff here
    sleep(5);

    gettimeofday(&end, NULL);

    long seconds = (end.tv_sec - start.tv_sec);
    long micros = ((seconds * 1000000) + end.tv_usec) - (start.tv_usec);

```

by email.

Email Address

Subscribe



```
    printf("Time elpased is %d seconds and %d micros\n", seconds, micros);

    return 0;
}
```

Download Run Code

Output (may vary):

```
Time elpased is 5 seconds and 5000147 micros
```

This function is supported by GCC compilers and might not work on Windows.

4. clock_gettime()

We can also use `clock_gettime()` function defined in `<time.h>` header file which supports upto nanosecond accuracy. It takes two arguments – the first arugment is clock type and the second argument is a pointer to `timespec` structure. The `timespec` structure is provided by the `<time.h>` header and is declared as:

```
struct timespec {
    time_t tv_sec;  /* seconds */
    long   tv_nsec; /* nanoseconds */
};
```

Below code calculates elapsed time using system-wide realtime clock, identified by `CLOCK_REALTIME` whose time represents seconds and nanoseconds since the Epoch.

```
#include <stdio.h>
#include <time.h>    // for clock_t, clock()
#include <unistd.h>    // for sleep()

#define BILLION  1000000000.0;

// main function to find the execution time of a C program
int main()
{
    struct timespec start, end;

    clock_gettime(CLOCK_REALTIME, &start);

    // do some stuff here
    sleep(3);

    clock_gettime(CLOCK_REALTIME, &end);

    // time_spent = end - start
    double time_spent = (end.tv_sec - start.tv_sec) +
                        (end.tv_nsec - start.tv_nsec) / BILLION;

    printf("Time elapsed is %f seconds", time_spent);

    return 0;
}
```

Download

Please note that the `clock_gettime()` function will work only on very few UNIX machines.

Related Post:

Measure elapsed time of a C++ program using chrono library

In this post, we will discuss how to measure elapsed time of a C++ program in seconds, milliseconds, microseconds and nanoseconds using chrono library.   Since C++11, the best way to measure elapsed time in C++ is by using the chrono library which deal with time.   Below C++ program calculates the time elapsed for … Continue reading

td Techie Delight 1

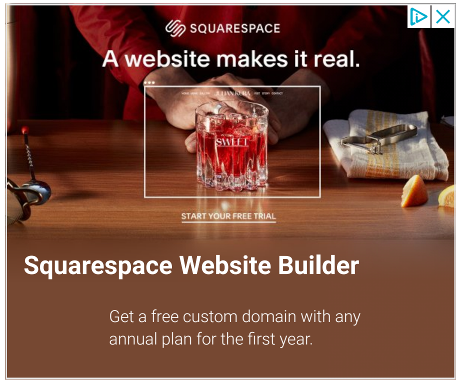
★★★★★ (**4** votes, average: **5.00** out of 5)

Thanks for reading.

Please use our online compiler to post code in comments. To contribute, get in touch with us.
Like us? Please spread the word and help us grow. Happy coding 🙂

Sharing is caring:

Tweet Share Pocket 4 More



Squarespace Website Builder

Get a free custom domain with any annual plan for the first year.

Misc

← Find minimum and maximum element in an array in Java

Sort a String in Java →

Leave a Reply

b i link b-quote u ul ol li code spoiler

Join the discussion (Set your avatar at https://gravatar.com/)

Subscribe ▾ ▴ newest ▴ oldest ▴ most voted

kkk

Guest

Says “Coding made easy” but uses a blocker in the page so people can’t select and copy content. Nice.

0 Reply 3 months ago

Techie Delight

Author

It was done for a reason. But you can copy, download code and even run it online using ‘Run code’ option.

0 Reply 3 months ago