EDUCBA
(https://www.educba.com/software-development/)

← (https://www.educba.com/best-c-compilers/)

→ (https://www.educba.com/data-types-in-c/)


# Introduction to clock() in C

Clock() plays a significant role in entire C programming as it helps in maintaining the timing record with proper hierarchy. The elapses or records based on timings with a number of ticks will help in making a 32-bit system clock on a per-second basis. The return value return function with a number of clock ticks that get elapsed with the start of each program bu case of failure, it returns a value of -1. Clock function is part of the library C function that mainly starts the flow with the inclusion of the ctime header file.

There is a proper syntax that represents the clock() function by returning some approximate processor time which further gets consumed by a program. Depending upon the clock time allocation of resources with each of the operating systems gets allocated. Syntax representation is as follows:

```
clock_k clock(void)
```

Where clock_k signifies the function with keyword clock and parameter to return void type.

## How clock() works in C?

- Clock function is mostly used to determine processor time within an operating system.
- It helps in allocating the time resources to the processor by counting the number of ticks per second for any manipulation.
- Clock function basically uses the following versions where it makes use of C language as ANSI/ISO 9899-1990
- There are other C functions as well which are similar to clock function and is represented within the standard library as:

```
Time_0 func (time.h)
```


- Clock() function follows some algorithmic code represented by the minutes and ti
example for :

(https://www.educba.com/software-development/)

- If it does not satisfy the condition, then it will in turn land into another scenario where delay or repeat will take place for less than 0 value.
- Then further it will call for power check whether it sustains or not then it will move into the increment of no-of minutes after that it will be counted in the done state for a completed operation as per flow chart depiction.

**Psuedocode**

```
# include <time.h>
clock_h  strt, end;
double cpu_time_usd;
start=clock();
// Perform Work
end = clock();
cpu_time_usd = ( (double) (end-start)) // This gives clock value
on per second basis…
```

- As per pseudocode initially, a start and end time exists with the clock() function then a variable for cpu_time_usd is taken where the clock function is then assigned to the start function.
- It will then perform all work for manipulation.


- Once the work comes to end the clock() function result will be assigned to the end function further giving the manipulation of CPU timings on the per-second basis for start and end at the time of resource allocation.

(https://www.educba.com/software-development/)

Here are the following examples mentioned below.

## Example #1

This program demonstrates how the time is consumed by function for its processing as shown in the output below.

**Code:**

```
#include <stdio.h>
#include <time.h>
void func_1()
{
printf("func_1() starts the flow... \n");
printf("Press enter_button to stop func_1 when entered.. \n");
while(1)
{
if (getchar())
break;
}
printf("func_1() gets an end here.. \n");
}
int main()

{
clock_t t_0;
t_0 = clock();
```

```
return 0;
}
```



**Output:**

```
func_1() starts the flow...
Press enter_button to stop func_1 when entered..
```

(https://www.educba.com/software-development/)

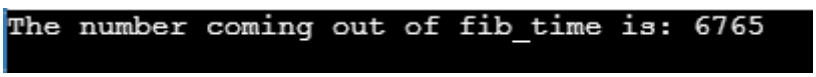
clock() function in C here is used for demonstrating the flow where func_1 consumes the flow of time with the execution of some value and its time as well. It takes some time for execution and allocates resources as per the operating system. If pressed enter then it will get halted with the required value as shown in the output.

## Example #2

This function demonstrates the time taken by fib_time(20) for consumption of time within the processor for any manipulation as shown in the output.

```
#include<stdio.h>
int fib_time(int a_0)
{
if (a_0 <= 1)
return a_0;
return fib_time(a_0-1) + fib_time(a_0-2);
}
int main ()
{
printf("The number coming out of fib_time is: %d", fib_time(20));
return 0;
}
```

**Output:**





**Explanation**: In this program, the number coming out of fib_time comes as the final compiled time of the fib_time() function that is used for the overview it is the final compilation time

(https://www.educba.com/software-development/)

```
#include <stdio.h>
#include <time.h>
int main () {
time_t currnt_time;
time(&currnt_time);
printf("Current_time comes_out to be: = %s", ctime(&currnt_time));
return(0);
}
```

**Output:**

```
Current_time comes_out to be: = Fri Jul  9 13:00:09 2021
```

## Example #4

Difftime is another c reference function that is also similar to clock() function but with some of the major differences which is depicted in the output below.

```
#include <stdio.h>
#include <time.h>
int main ()
{

time_t tm_1,tm_2;
char get_inpt [258];
double diff_sc;
```

(https://www.educba.com/software-development/)

```
printf ("It took almost %.2lf seconds_for typing the time...\n",
diff_sc );
return 0;
}
```

**Output:**

```
/tmp/ccvdn1vF.o: In function `main':
main.c:(.text+0x43): warning: the `gets' function is dangerous and should not be used.
Enter name of_fav food: burger
It took almost 15.00 seconds_for typing the time...
```

**Explanation**: This program demonstrates the function diff_sc with some of the time differences in comparison and it helps in understanding the instantaneous time for switching or any other functionality to work. Here one option is given with entering the name of fav food and to compare and fetch the instance of time with a difference with respect to the operating system as shown in the output.

Clock() function in C plays an important role as it helps developers gain insight into the timing constraints with respect to the current system or processor in use. It gives developers the ability to differentiate and troubleshoot even if some of the patches and releases are made with help of a check and run this clock() and ctime function as part of the same standard library for verification.

## Recommended Articles


This is a guide to clock() in C. Here we discuss How clock() works in C and examples al the codes and outputs. You may also have a look at the following articles to learn more

1. Java Clock (https://www.educba.com/java-clock/)

(https://www.educba.com/software-development/)

development/verifiable-certificate/?source=footer)

Reviews (https://www.educba.com/software-development/reviews/?source=footer)

Terms and Conditions (https://www.educba.com/terms-and-conditions/?source=footer)

Privacy Policy (https://www.educba.com/privacy-policy/?source=footer)

### Apps

iPhone & iPad (https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8)

Android (https://play.google.com/store/apps/details?id=com.educba.www)

### Resources

Free Courses (https://www.educba.com/software-development/free-courses/?source=footer)

Java Tutorials (https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer)

Python Tutorials (https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer)

All Tutorials (https://www.educba.com/software-development/software-development-tutorials/?source=footer)

### Certification Courses


All Courses (https://www.educba.com/software-development/courses/?source=footer)

Software Development Course - All in One Bundle (https://www.educba.com/software-development/courses/software-development-course/?source=footer)

(https://www.educba.com/software-development/)

Become an IoT Developer (https://www.educba.com/software-development/courses/iot-course/?source=footer)

ASP.NET Course (https://www.educba.com/software-development/courses/asp-net-course/?source=footer)

VB.NET Course (https://www.educba.com/software-development/courses/vb-net-course/?source=footer)

PHP Course (https://www.educba.com/software-development/courses/php-course/?source=footer)