Sleep function in C++

I need a function like `Sleep(time);` that pauses the program for X milliseconds, but in C++.

Please write which header to add and the function's signature. Thank you!

c++

edited Nov 1 '09 at 21:29 — strager — 63k 14 102 154

asked Nov 1 '09 at 21:25 — Tal Stol

5 This is platform specific, so you're gonna need specific OS includes, or Boost (which will do the previous for you). – GManNickG Nov 1 '09 at 21:27

6 Answers

Use `std::this_thread::sleep_for` :

```
std::chrono::milliseconds timespan(111605); // or whatever

std::this_thread::sleep_for(timespan);
```

There is also the complimentary `std::this_thread::sleep_until` .

Prior to C++11, C++ had no thread concept and no sleep capability, so your solution was necessarily platform dependent. Here's a snippet that defines a `sleep` function for Windows or Unix:

```
#ifdef _WIN32
    #include <windows.h>

    void sleep(unsigned milliseconds)
    {
        Sleep(milliseconds);
    }
#else
    #include <unistd.h>

    void sleep(unsigned milliseconds)
    {
        usleep(milliseconds * 1000); // takes microseconds
    }
#endif
```

But a much simpler pre-C++11 method is to use `boost::this_thread::sleep` .

edited May 17 '13 at 3:10

answered Nov 1 '09 at 21:39 — GManNickG — 251k 32 371 484

On unix, include `#include <unistd.h>` ... The call your interested in is usleep()... Which takes microseconds, so you should multiply your millisecond value by 1000 and pass the result to usleep()...

edited Nov 1 '09 at 21:49
sbi
143k 36 188 361

answered Nov 1 '09 at 21:31
dicroce
17.9k 17 78 121

There is no portable way to do this.

A portable way is to use Boost or Ace library. There is ACE_OS::sleep(); in ACE.

answered Nov 1 '09 at 21:29
alexkr
3,664 1 16 20

1 ... and the (supposed) reason there is no portable way to do it in the standard is because the a clock's precision (smallest unit of time) is hardware-dependent or OS-dependent. No, I don't find it a convincing reason either, but there we are. – wilhelmtell Nov 1 '09 at 21:34

There is no such thing as thread defined by standard... and you want sleep. Sleep is a OS provided functionality. I can have environment which does not provide me such feature. – alexkr Nov 1 '09 at 21:41

@wilhelmtell: That is not the reason at all. Who is it making this supposition other than yourself? There is no standard for thread support (yest), and if there are no threads (or rather only one thread), there is no need for a thread sleep rather than a simple 'busy-wait', which can be implemented with <time.h>/<ctime>. The support must be provided by the thread library or OS. – Clifford Nov 1 '09 at 23:06

@alexkr - Two C++ standards later, this answer seems a bit outdated. You might want to revise it? – Bo Persson Sep 6 '15 at 13:33

you might edit your answer since there is a portable way to do this now. – Abhinav Gauniyal Sep 18 '16 at 3:19

You'll need at least C++11.

```
#include <thread>
#include <chrono>

...

std::this_thread::sleep_for(std::chrono::milliseconds(200));
```

answered Nov 24 '16 at 10:13
Jostein Topland
89 4

```
#include "windows.h"
Sleep(10);
```

for unix probably `#include <unistd.h>`

just google it...

edited Apr 26 '15 at 7:10

answered Nov 1 '09 at 21:27
Dani
9,488 8 40 82

7 On Windows, Sleep() is capitalized. On Unix, there is usleep() in unistd.h – asveikau Nov 1 '09 at 21:35

Thanks for the remark. I wrote this as a hint. – Dani Apr 26 '15 at 7:11

The simplest way I found for C++ 11 was this:

Your includes:

```
#include <windows.h>
#include <chrono>
#include <thread>
```

Your code (this is an example for sleep 1000 millisecond):

```
std::chrono::duration<int, std::milli> timespan(1000);
std::this_thread::sleep_for(timespan);
```

The duration could be configured to any of the following:

```
 std::chrono::nanoseconds  duration</*signed integer type of at least 64 bits*/,
std::nano>
 std::chrono::microseconds  duration</*signed integer type of at least 55 bits*/,
std::micro>
 std::chrono::milliseconds  duration</*signed integer type of at least 45 bits*/,
std::milli>
 std::chrono::seconds  duration</*signed integer type of at least 35 bits*/>
 std::chrono::minutes  duration</*signed integer type of at least 29 bits*/,
std::ratio<60>>
 std::chrono::hours  duration</*signed integer type of at least 23 bits*/,
std::ratio<3600>>
```

answered Apr 19 '16 at 6:54

Merav Kochavi
1,120 10 24