

Forum:[Hack Forums](#)

Thread: [C# \[TUT \] Sleep ur program without stopping the whole program! \[/TUT\]](#)

pimpie1995 - April 17, 2012, 9:45 am

Hey guys!

I have searched for this for a long time and after i tried somethings i figured out i found a way to Sleep ur program without it will stop the whole program

we all know the Thread.Sleep Method, but that sucks

i made this little count down;

Syntax / Code:

Quote: int counter = 0

```
private void timer1_Tick(object sender, EventArgs e)
{
    if (counter < Convert.ToInt32(tbSeconds.Text))
    {
        counter++;
        lblCounter.Text = Convert.ToString(counter);
    }
    else
    {
        tbWorked.Text = "WORKED!";
        timer1.Stop();
    }
}

private void bMakeittrain_Click(object sender, EventArgs e)
{
    timer1.Start();
}
```

I think if you already are advanced with coding in C#, you should know now how to "pause" your script with this method :)

Have fun with trying guys and good luck!

iVision - April 17, 2012, 9:49 am

I found the best way, multithreading!

pimpie1995 - April 17, 2012, 9:53 am

could you learn me that :x?, im sure there are still 100th more ways, but i just wanted to share with you guys :p

BaussHacker - April 17, 2012, 11:11 am

(Apr 17, 2012 09:49 AM)iVision Wrote: [\[CITE\]](#) I found the best way, multithreading!

A timer is also a thread lol. ?

Also saying the Thread.Sleep method suck...

It's great when working with multithreading, because you won't slip your cpu up by putting one Thread.Sleep(1) inside your while loop within your thread.

You're obviously new, I can see that and for that reason you shouldn't throw such statement and your program is not sleeping at all...

pimpie1995 - April 17, 2012, 12:23 pm

(Apr 17, 2012 11:11 AM)?ray Wrote: [\[CITE\]](#) A timer is also a thread lol. ?

Also saying the Thread.Sleep method suck...

It's great when working with multithreading, because you won't slip your cpu up by putting one Thread.Sleep(1) inside your while loop within your thread.

You're obviously new, I can see that and for that reason you shouldn't throw such statement and your program is not sleeping at all...

Yes i am new, i am studing C# on school now and i really like to work with it :) tnkx for the feedback

BaussHacker - April 17, 2012, 12:33 pm

That's just great :) If you have any questions or need help, then feel free to PM me :)

DragQueen - April 17, 2012, 1:10 pm

its call Multithreading.

and the best thing its that timer1_Tick
running in the main thread

SpriteZ - April 17, 2012, 2:38 pm

A while loop would be better, but multithreading is still a great option.

pimpie1995 - April 17, 2012, 11:44 pm

(Apr 17, 2012 02:38 PM)SpriteZ Wrote: [\[CITE\]](#) A while loop would be better, but multithreading is still a great option.

you cant pause ur action with a while loop... also when a while loop is to long, it will freeze ur program until its finished

BaussHacker - April 18, 2012, 12:10 am

(Apr 17, 2012 02:38 PM)SpriteZ Wrote: [\[CITE\]](#) A while loop would be better, but multithreading is still a great option.

Putting the loop in a new thread, would be better. ?

Volim te - April 18, 2012, 1:04 am

Bad coding sorry less use that Convert.ToInt32() in a try/catch block.

Maiki - April 18, 2012, 2:48 am

Uhm...why not use a backgroundworker ?

DragQueen - April 18, 2012, 4:20 am

(Apr 17, 2012 11:44 PM)pimpie1995 Wrote: [\[CITE\]](#) you cant pause ur action with a while loop... also when a while loop is to long, it will freeze ur program until its finished
if this while ... in diffrent thread.
it can be done.

blazzored - April 18, 2012, 10:37 pm

```
DateTime start = DateTime.Now;
while (DateTime.Now.Subtract(start).Seconds < 10)
{
    Application.DoEvents();
    System.Threading.Thread.Sleep(1);
}
```

I win.

BaussHacker - April 18, 2012, 10:41 pm

(Apr 18, 2012 10:37 PM)blazzored Wrote: [\[CITE\]](#) DateTime start = DateTime.Now;
while (DateTime.Now.Subtract(start).Seconds < 10)
{
 Application.DoEvents();
 System.Threading.Thread.Sleep(1);
}

I win.

No you don't. Yours will still sleep the app.

I win.

```
Code System.Threading.Thread newThread = new System.Threading.Thread(new
ThreadStart(delegate {
    int count = 10;
    while (++count < 10)
    {
        //lolwut
        System.Threading.Thread.Sleep(1);
    }
}));
newThread.Start();
```

0xDEAD10CC - April 18, 2012, 11:42 pm

(Apr 18, 2012 10:41 PM)?ray Wrote: [\[CITE\]](#) No you don't. Yours will still sleep the app.

I win.

```
Code System.Threading.Thread newThread = new System.Threading.Thread(new
ThreadStart(delegate {
    int count = 10;
    while (++count < 10)
    {
        //lolwut
        System.Threading.Thread.Sleep(1);
    }
}));
newThread.Start();
lol I don't think so...
Quote:int count = 10;
while (++count < 10)
```

? Are you sure?

And yours doesn't update anything on the UI thread in this example either.

I win:

```
Code new System.Threading.Thread(new ThreadStart(delegate
{
    int count = 0;
    while (count++ < 10)
    {
        System.Threading.Thread.Sleep(500);
        Invoke((MethodInvoker)delegate { textBox1.Text = count.ToString(); });
    }
})).Start();
```

BaussHacker - April 19, 2012, 12:59 am

Oh it was a mistake. I just woke up when I replied haha, so didn't even notice it :D Also I felt lazy to make the update, so just made the comment ?

0xDEAD10CC - April 19, 2012, 1:02 am

(Apr 19, 2012 12:59 AM)?ray Wrote: [\[CITE\]](#) Oh it was a mistake. I just woke up when I replied haha, so didn't even notice it :D Also I felt lazy to make the update, so just made the comment ?
? Now we're even haha ?

Happens to everybody

Cheers :)

BaussHacker - April 19, 2012, 1:30 am

(Apr 19, 2012 01:02 AM)PhazeShift Wrote: [\[CITE\]](#) ? Now we're even haha ?

Happens to everybody

Cheers :)

Indeed buddy ? It's impossible to be perfect ?

blazzored - April 19, 2012, 12:35 pm

(Apr 18, 2012 10:41 PM)?ray Wrote: [\[CITE\]](#) No you don't. Yours will still sleep the app.

I win.

```
Code System.Threading.Thread newThread = new System.Threading.Thread(new
ThreadStart(delegate {
    int count = 10;
    while (++count < 10)
    {
        //lolwut
        System.Threading.Thread.Sleep(1);
    }
}));
newThread.Start();
No it doesn't. How about you try. Doesn't hang the application.
```

BaussHacker - April 19, 2012, 2:03 pm

(Apr 19, 2012 12:35 PM)blazzored Wrote: [\[CITE\]](#) No it doesn't. How about you try. Doesn't hang the application.

Because it goes so fast, try change the value to 1 million and you will see. ???

0xDEAD10CC - April 19, 2012, 2:07 pm

(Apr 19, 2012 12:35 PM)blazzored Wrote: [\[CITE\]](#) No it doesn't. How about you try. Doesn't hang the application.

You don't even understand your own code lol. The reason why it might not appear that it does is because that code you posted takes a fraction of a second to finish. Yet it frequents freezing the UI thread until an update in the loop happens.

I recommend that you try your code with a higher sleep input, and then come back and say it doesn't keep the UI unresponsive to updating.

```
Code DateTime start = DateTime.Now;
while (DateTime.Now.Subtract(start).Seconds < 10)
{
    Application.DoEvents();
    System.Threading.Thread.Sleep(1000);
}
```

```
MessageBox.Show("You're Done");
```

Try to move the form without having any intervals where you can't move the form until you see the messagebox :)

Xenon - April 19, 2012, 3:19 pm

You could run a separate thread to increment the time and have that thread sleep instead of the main execution thread. But I guess a timer works just as well.

BaussHacker - April 19, 2012, 3:27 pm

(Apr 19, 2012 03:19 PM)Xenon Wrote: [\[CITE\]](#) You could run a separate thread to increment the time and have that thread sleep instead of the main execution thread. But I guess a timer works just as well.

Did you come here to repeat what has been said on the last 2 pages?

TSO - April 23, 2012, 12:03 pm

This is what I use and it doesn't require making the thread sleep or any timers.

```
Code public static DateTime delay(int MilliSeconds)
{
    System.DateTime ThisMoment = System.DateTime.Now;
    System.TimeSpan duration = new System.TimeSpan(0, 0, 0, 0, MilliSeconds);
    System.DateTime Afterwards = ThisMoment.Add(duration);
    while (Afterwards >= ThisMoment)
    {
        System.Windows.Forms.Application.DoEvents();
        ThisMoment = System.DateTime.Now;
    }
    return System.DateTime.Now;
}
```

pimpie1995 - April 24, 2012, 2:37 am

(Apr 23, 2012 12:03 PM)TSO Wrote: [\[CITE\]](#) This is what I use and it doesn't require making the

thread sleep or any timers.

```
Code public static DateTime delay(int MilliSeconds)
{
    System.DateTime ThisMoment = System.DateTime.Now;
    System.TimeSpan duration = new System.TimeSpan(0, 0, 0, 0, MilliSeconds);
    System.DateTime AfterWards = ThisMoment.Add(duration);
    while (AfterWards >= ThisMoment)
    {
        System.Windows.Forms.Application.DoEvents();
        ThisMoment = System.DateTime.Now;
    }
    return System.DateTime.Now;
}
```

Thats really helpfull :)! good post
353535

iVision - April 24, 2012, 3:59 am

(Apr 23, 2012 12:03 PM)TSO Wrote: [\[CITE\]](#) This is what I use and it doesn't require making the thread sleep or any timers.

```
Code public static DateTime delay(int MilliSeconds)
{
    System.DateTime ThisMoment = System.DateTime.Now;
    System.TimeSpan duration = new System.TimeSpan(0, 0, 0, 0, MilliSeconds);
    System.DateTime AfterWards = ThisMoment.Add(duration);
    while (AfterWards >= ThisMoment)
    {
        System.Windows.Forms.Application.DoEvents();
        ThisMoment = System.DateTime.Now;
    }
    return System.DateTime.Now;
}
```

}This is not freezing the UI but freeze the computer, this will overload the cpu on a longer time.

Vuln - April 25, 2012, 1:19 am

Application.DoEvents(); before sleeping prevents freezing too, I'm not a big fan of it though.