

Forum:[Hack Forums](#)

Thread: [\[TUTORIAL\]Basic Multithreading\[TUTORIAL\]](#)

yungdsl - October 7, 2012, 7:47 pm

Basic Multithreading Tutorial

For this tutorial we will be using Python 2.7.3

Hello python section, I am back with another tutorial. Today, we are going to learn some basic multithreading.

What is multithreading?

Multithreading is more or less having two parts of your program execute at the same time, while still sharing the same data. For example, you could have two while loops going at the same time.

Now, on to programming. First we will need to import the module for threading. Be sure to include time, we will be using it later.

Code import threading, time

Now, we are going to have to have a function for the thread to use. Later on we can also build classes to use, but lets just do this for now. Lets make our function like this:

Code import threading, time

```
def workerFunc(id):  
    while True:  
        time.sleep(0.5)  
        print "Hello from worker", str(id)
```

This is just a simple function, that will wait half of a second and then print the phrase Hello from worker (id). Now we will set up the threading like this:

Code import threading, time

```
def workerFunc(id):  
    while True:  
        time.sleep(1)  
        print "Hello from worker", str(id)
```

```
thread01 = threading.Thread(target=workerFunc, args=(1,))  
thread02 = threading.Thread(target=workerFunc, args=(2,))  
thread01.start()  
time.sleep(3.7)  
thread02.start()
```

Now, when you run that you should get output something like this:

[img]<http://img692.imageshack.us/img692/7752/14c20dc026114f9e8ea85ce.png>[/img]

What this does so far, is makes two threads which use the function workerFunc. The id argument is passed to them. Then, the program starts thread01 and then waits 3.7 seconds, and then starts

thread02.

Thats the end of this tutorial! Part 2 coming soon with object orientation and more advanced design.

D3vildog - October 7, 2012, 7:49 pm

Daym Son, that is some sexy multi-threading, pretty soon we will have Crysis 3 in Python.

.php - October 7, 2012, 7:52 pm

Awesome :D Helped me gain a better understanding!

yungdsl - October 7, 2012, 7:57 pm

(Oct 7, 2012 07:52 PM).php Wrote: [\[CITE\]](#) Awesome :D Helped me gain a better understanding! Your welcome. Be sure to look at my next tutorial on threading, whenever I make it. Probably tonight.

Integral - October 7, 2012, 8:02 pm

simple tutorial, but gets the point across very nicely - nice job.

tulamba - October 7, 2012, 8:29 pm

Good job my friend.

I have a question

dude they are always repeating the pattern 2,1,2,1

if i have multi-core machine, which is the case with today's CPUs, then i can run them concurrently. It means CPU1 running thread1 and CPU2 running thread2.

What are your thoughts on this?

Asp1r3-At0m - October 7, 2012, 9:16 pm

you are a genius, i was trying to understand what the hell was threading and how do do it. thanks !

yungdsl - October 7, 2012, 10:19 pm

(Oct 7, 2012 08:29 PM)tulamba Wrote: [\[CITE\]](#) Good job my friend.

I have a question

dude they are always repeating the pattern 2,1,2,1

if i have multi-core machine, which is the case with today's CPUs, then i can run them concurrently. It means CPU1 running thread1 and CPU2 running thread2.

What are your thoughts on this?If you are wondering why they follow that pattern, its because thread 2 starts 3.7 seconds after thread1. And this should work the same on pcs with any amount of cores.

(Oct 7, 2012 09:16 PM)Asp1r3-At0m Wrote: [\[CITE\]](#) you are a genius, i was trying to understand what the hell was threading and how do do it. thanks !

Yes, I had the same trouble with it at first but the threading module was the easiest way to do it and it really helped me.

Curator - October 7, 2012, 11:51 pm

(Oct 7, 2012 08:29 PM)tulamba Wrote: [\[CITE\]](#) Good job my friend.

I have a question

dude they are always repeating the pattern 2,1,2,1

if i have multi-core machine, which is the case with today's CPUs, then i can run them concurrently. It means CPU1 running thread1 and CPU2 running thread2.

What are your thoughts on this?Python's default implementation (called CPython) does not support concurrent multithreading on multicore systems.

However, you can use the multiprocessing module and similar library calls to achieve true concurrency. Each worker is its own process instead of its own thread.

FaggotFlounder - October 8, 2012, 8:22 am

Informative.

Thanks for sharing it with us :)

Jaket25 - October 8, 2012, 8:39 am

Great tutorial, little examples like this really do help explain new modules to me.

Just trying to think of a practical use to implement this module into now! Haha

Thanks a lot, we need more tutorials like this ?

tulamba - October 8, 2012, 9:31 am

Thanks Curator and Kaonashi for your comments

(Oct 7, 2012 11:51 PM)Curator Wrote: [\[CITE\]](#) Python's default implementation (called CPython) does not support concurrent multithreading on multicore systems.

However, you can use the multiprocessing module and similar library calls to achieve true concurrency. Each worker is its own process instead of its own thread.

dberube - October 8, 2012, 11:14 am

The multiprocessing module works the same way:

Code from multiprocessing import Process

```
def f(name):  
    print 'hello', name
```

```
if __name__ == '__main__':  
    p = Process(target=f, args=('bob',))  
    p2 = Process(target=f, args=('john',))  
    p.start()  
    p2.start()  
    p.join()  
    p2.join()
```

zapfen - October 23, 2012, 12:41 pm

Thats a nice tutorial :) Thank you!

yungdsl - October 23, 2012, 2:15 pm

(Oct 23, 2012 12:41 PM)zapfen Wrote: [CITE](#) Thats a nice tutorial :) Thank you!

No problem, I'm going to do a tutorial on building classes for multithreading tonight.

3SidedSquare - October 23, 2012, 6:48 pm

Threading seemed so intimidating, just looked over your tutorial and did the example, I feel like I AM A MARINE BIOLOGIST, I GOT THIS.