

Ayudantía 09

Threading

Felipe Garrido Vicente Domínguez

Departamento de Ciencia de la Computación
Pontificia Universidad Católica de Chile

IIC2233, 2015-1

Tabla de contenidos

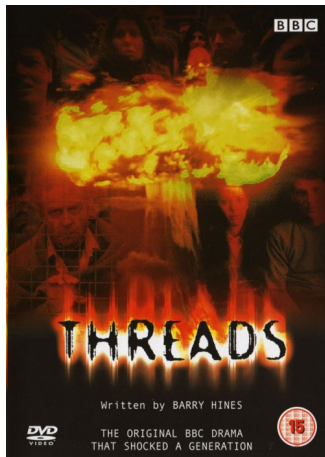
- 1 ¿Qué son los threads?
 - Definición y usos
- 2 Threads en Python
 - Sintaxis
- 3 Multithreading
 - Mecanismos de sincronización

¿Qué son los threads?



¿Hilos?.

¿Qué son los threads?



¿Una película?.

¿Qué son los threads?

You are here: Forum > News & Announcements

Forum: News & Announcements

Read the latest news and announcements from the Wakanda team.

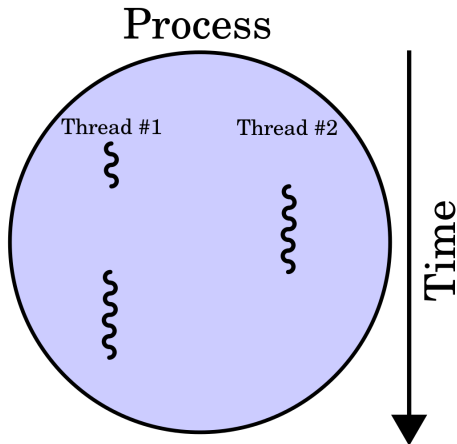
Threads 1 to 20 of 43 Page 1 of 3 1 2 3 Last >>

Forum Tools Search Forum

Title / Thread Starter	Thread / Post Score	Replies / Views	Last Post By
Our community at JS.everywhere(2012) • Wakanday Started by jmbiraghi, 09-13-2012 08:47 AM	Thread: 1 Best Post: 0	Replies: 0 Views: 95	jmbiraghi 09-13-2012, 08:47 AM
Important information about Mac OS X 10.8 "Mountain Lion" Started by Omid, 07-25-2012 12:52 PM	Thread: 0 Best Post: -1	Replies: 1 Views: 534	Kentuckyguy 07-31-2012, 02:24 AM
Demo script available for Webinar: Top 10 Newest Features Started by chinshr, 07-19-2012 05:48 PM	Thread: 0 Best Post: 0	Replies: 5 Views: 655	Olivia Halimi 07-25-2012, 11:32 AM
New build for Wakanda Production Release Started by Olivia Halimi, 07-25-2012 09:41 AM	Thread: 0 Best Post: 0	Replies: 1 Views: 239	pkesch 07-25-2012, 09:45 AM
Wakanda First Production Release... Long lives Wakanda! Started by Olivia Halimi, 06-28-2012 06:38 AM 1 2 3	Thread: 4 Best Post: 3	Replies: 20 Views: 1,599	nw2206 07-06-2012, 09:52 PM
[SOLVED] New Wakanda Beta awaits you in our Stabilization Channel! Started by Olivia Halimi, 05-30-2012 12:45 PM 1 2	Thread: 1 Best Post: 1	Replies: 13 Views: 1,923	Michael5 06-09-2012, 03:44 PM
Missed our introduction to Data Modeling webinar last week? You can still catch up! Started by Olivia Halimi, 05-23-2012 05:39 AM	Thread: 0 Best Post: 2	Replies: 3 Views: 869	Olivia Halimi 05-23-2012, 12:25 PM
WAK2 is here, check out our Stabilization Channel! Started by Olivia Halimi, 04-27-2012 10:38 AM	Thread: 0 Best Post: 1	Replies: 9 Views: 1,203	newmams 05-16-2012, 05:04 PM

¿Conversaciones? Esto se asemeja un poco más.

¿Qué son los threads?



En computación, esto es lo que entenderemos como threads.

¿Qué son los threads?

Definición y usos

- En ciencia de la computación entenderemos un **thread** como la unidad básica de procesamiento que puede ser ejecutada o planificada por un sistema operativo.
- Los threads son parte de un *proceso*. En palabras simples este puede entenderse como un programa en ejecución, o formalmente como *Una unidad de actividad que se caracteriza por la ejecución de una secuencia de instrucciones, un estado actual, y un conjunto de recursos del sistema asociados* (Stallings 5ª edición pag. 109).

¿Qué son los threads?

Definición y usos

- En general cuando se ejecuta un programa en un computador este no se corre de forma lineal, más bien hay una serie de procesos funcionando a la vez de forma "*pseudoparalela*".
- Los **threads** nos permiten generar este *pseudoparalelismo* en nuestro programa y nos dan la capacidad de ejecutar multiples tareas de forma simultanea.
- En **Python**, y en la mayoría de los lenguajes de programación, la forma de usar los **threads** es creando a una variable del tipo **thread**, a esta se le asigna una función o método a ejecutar y luego se inicializa este. De esta forma el código principal sigue ejecutandose mientras el **thread** también sigue de forma *pseudoparalela*.

Threads en Python

Sintaxis

Código ejemplo

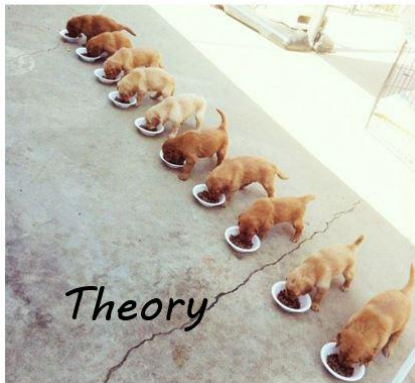
```
1 import threading
2
3 def sumainfinita(a,b):
4     while True:
5         c = a + b
6         print(c)
7
8 t_1 = threading.Thread(target=sumainfinita, args=(a,b))
9 t_1.daemon = True
10 t_1.start()
```

Multithreading

- El uso de **multithreading** puede resultar muy útil para acelerar la ejecución de un programa o para lograr de que cumpla lo requerido.
- Aunque tenga muchos beneficios esta práctica, también acarrea problemas.
- Los principales son la coordinación de los **threads** (lograr de que uno ejecute en ciertos momentos del otro) y las colisiones entre ellos (cuando más de uno intenta acceder a una sección de código simultáneamente).

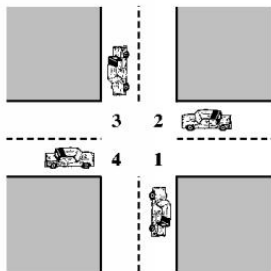
Multithreading

Multithreaded programming

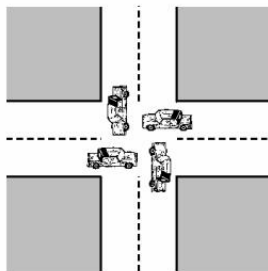


En la práctica...

Multithreading



(a) Deadlock possible



(b) Deadlock

Deadlock en la teoría.

Multithreading



Deadlock en la práctica.

Multithreading



Multithreading



¿Qué podemos hacer?

Multithreading

Mecanismos de sincronización

- Como estos son problemas recurrentes, existen mecanismos de sincronización de **threads**.
- Estos mecanismos nos permiten tener un mayor control sobre los **threads** y así poder evitar las colisiones u otros problemas similares.
- Uno muy recurrente en los lenguajes de programación es el **lock**. Este consiste en encapsular una sección de código (generalmente una sección que ocasione problemas entre los **threads**) y solo permitir el acceso de un **thread** a la vez a esa parte del programa.

Multithreading

Sintaxis

Código ejemplo:

```
1 import threading
2
3 lock = threading.Lock()
4
5 def funcioninteresante():
6     ...
7     lock.acquire()
8     ...
9     # zona de conflicto
10    ...
11    lock.release()
12    ...
```

Multithreading

Sintaxis

Código ejemplo:

```
1 import threading
2
3 lock = threading.Lock()
4
5 def funcioninteresante():
6     ...
7     with lock:
8         # zona de conflicto
9     ...
```