

A close-up photograph of a vibrant green snake with yellow speckles, coiled around a large, textured green bell pepper. The snake's head is positioned at the top left, looking towards the center. The background is dark, making the green elements stand out.

# Python Básico

Programación Concurrente

Junio del 2017

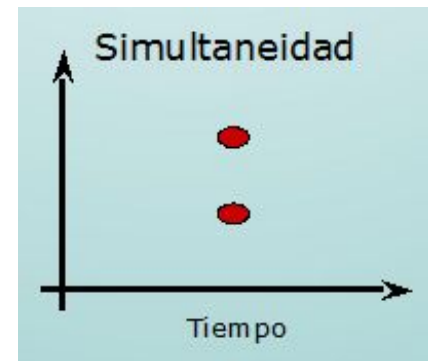
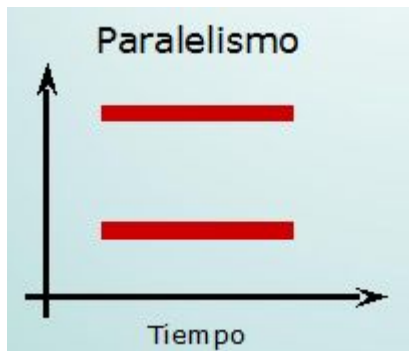
# Paradigma Concurrente

- Diseñado para tener dos o más contextos de ejecución activos. Se conocen como programas multi-red, o multihilos



# Paradigma concurrente

- **Paralelismo:** Sucede en un mismo intervalo de tiempo.
- **Solapamiento:** Intervalos superpuestos.
- **Simultaneidad:** Sucede en un mismo instante de tiempo.



# ¿Cuándo es proceso y cuándo es programa?

- **Programa:** Una lista de instrucciones a seguir, una entidad pasiva (nuestro código).
- **Proceso:** Entidad activa que: Emplea al programa. Típicamente opera sobre un conjunto de datos y tiene información de estado que indica, entre otras cosas, en qué punto va la ejecución.





# Procesos

- Un proceso puede informalmente entenderse como un programa en ejecución. Formalmente un proceso es "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".
- Un programa puede constar de varios procesos.



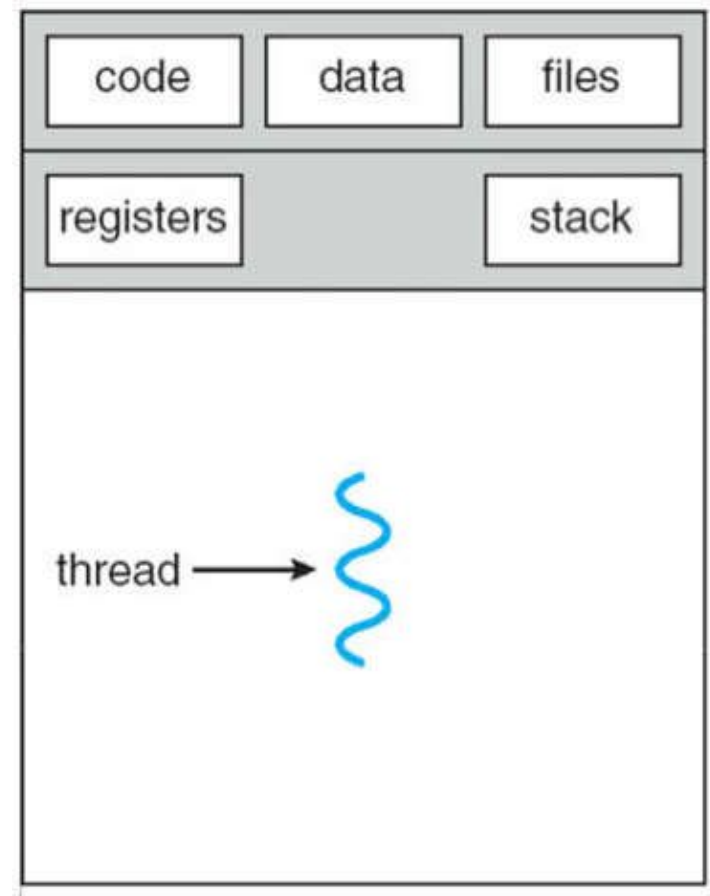
# Procesos

- Un proceso tiene un PCB(Process Control Block) que es la información que necesita el sistema para su ejecución:
- Estado del proceso, contador de programa, registros del CPU, planificación (scheduling), la prioridad del proceso, la cola en que está agendado, las tablas de mapeo de memoria, estado de E/S, lista de dispositivos y archivos asignados que el proceso tiene abiertos.

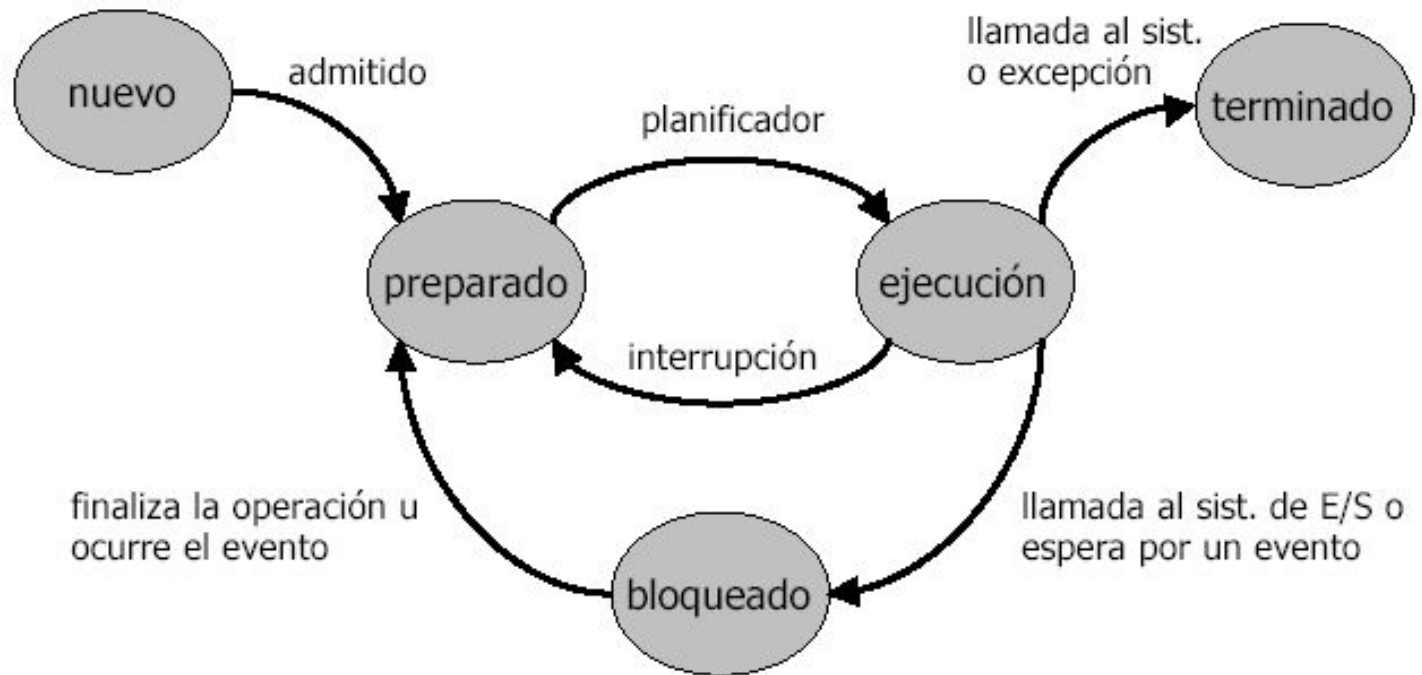


# Procesos

- Todo proceso que no usa hilos cuenta con al menos un hilo de ejecución, llamado el hilo main.



# Ciclo de vida de un proceso





# Hilos

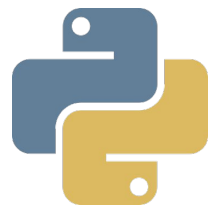
- Un hilo es una unidad básica de utilización de CPU, la cual contiene un id de hilo, su propio program counter, un conjunto de registros, y una pila
- Se representa a nivel del sistema operativo con una estructura llamada TCB (thread control block).



# Hilos

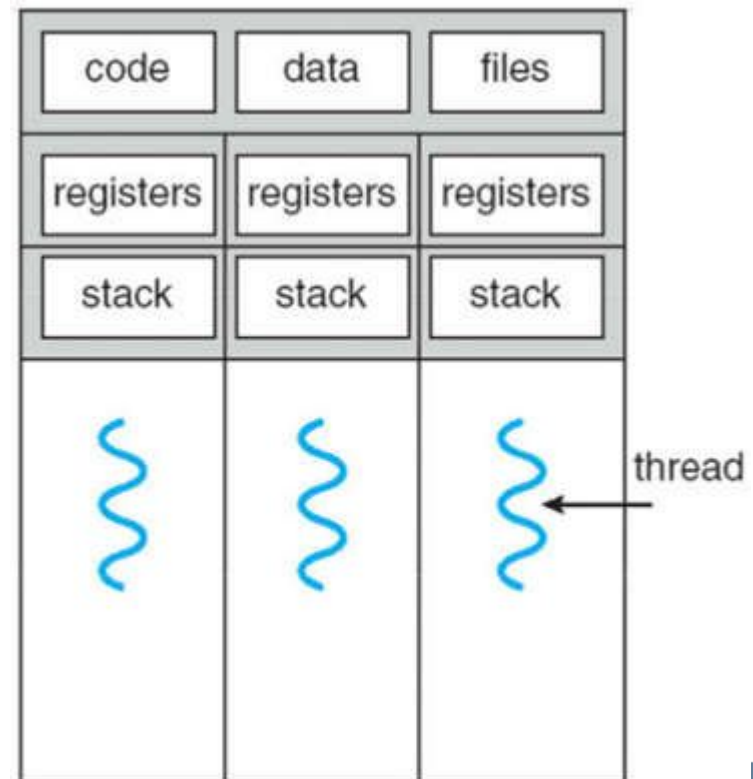
- Los hilos comparten con otros hilos que pertenecen al mismo proceso la sección de código, la sección de datos, entre otras cosas.
- Si un proceso tiene múltiples hilos, puede realizar más de una tarea a la vez (esto es real cuando se posee más de un CPU).

- 

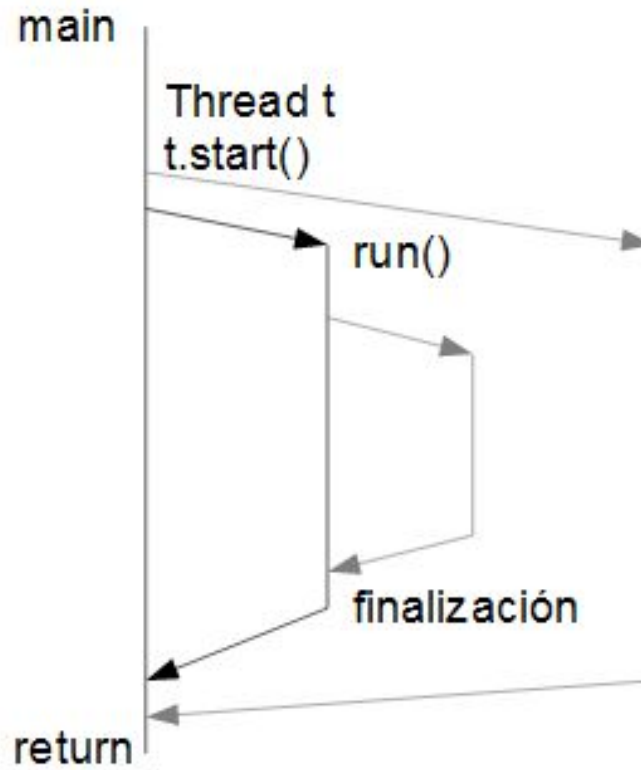


# Hilos

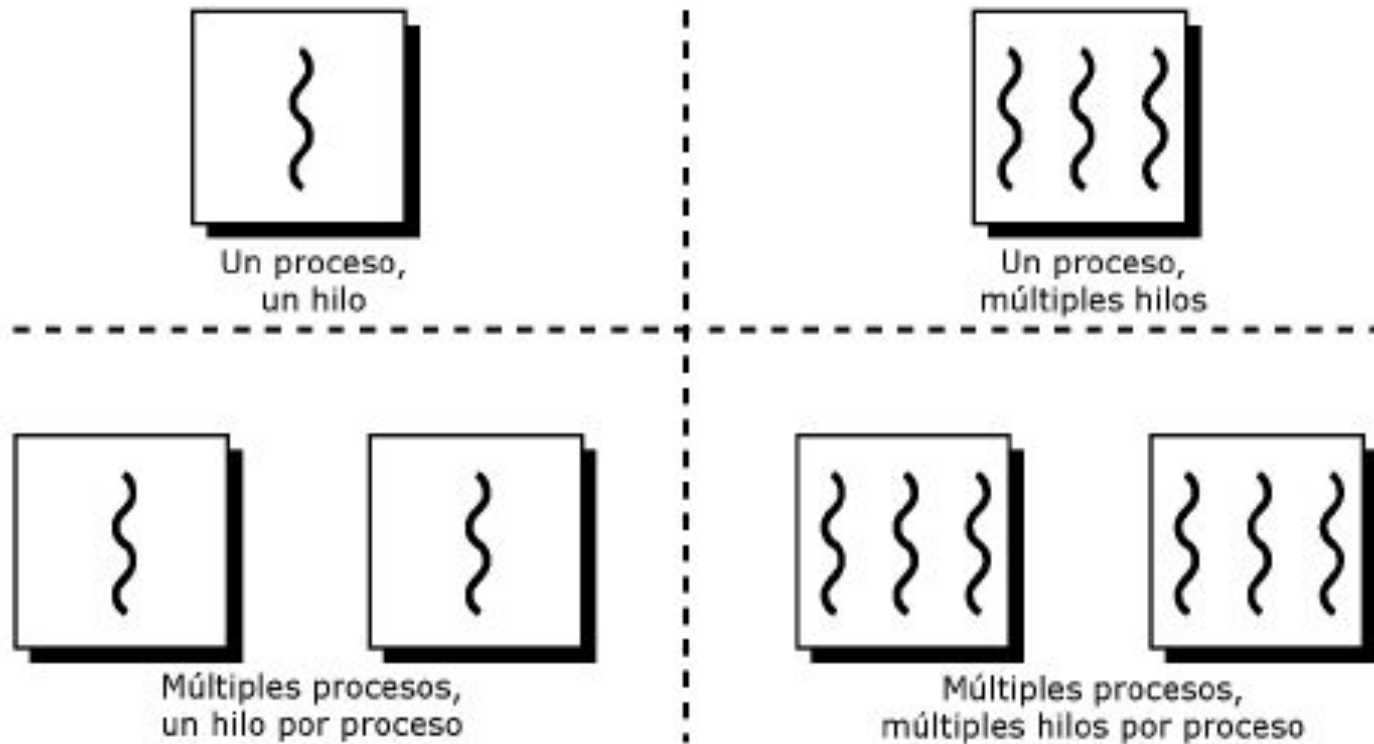
- En un proceso con varios hilos cada uno tiene su registro y todos pueden acceder a los mismos datos.



# Ciclo de vida de un hilo



# En general



## Precauciones

- En python para utilizar hilos se debe importar el módulo `threading` para python 3.x y en python 2.x se utilizaba `thread`.
- `thread` ha cambiado de nombre a `_thread`. Es el código de infraestructura que se utiliza para implementar `threading`, y no se recomienda usar ya que es a muy bajo nivel.





# Los hilos también son objetos

- Para crear un hilo nuevo, importe la **threading** y defina la función que realiza el hilo. Para ejecutar dichos hilos, cree una o más instancias de la clase **Thread** pasando los **parámetros** como una tupla y llame al método start.

```
import threading
```

```
def funcion(*parametros):
```

```
    # código que ejecutará el hilo
```

```
hilo = threading.Thread(target= funcion,args=(parámetros,))
```

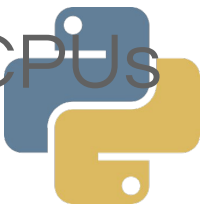
```
hilo.start()
```

TUPLA



# Ventajas

- **Respuesta:** el tiempo de respuesta mejora, ya que el programa puede continuar ejecutándose, aunque parte de él esté bloqueado.
- **Compartir recursos:** los hilos comparten la memoria y los recursos del proceso al que pertenecen.
- **Economía:** Es más fácil la creación, cambio de contexto y gestión de hilos que de procesos.
- **Utilización múltiples CPUs:** permite que hilos de un mismo proceso ejecuten en diferentes CPUs a la vez.



# Desventajas

- **Indeterminismo:** No sabemos a ciencia cierta que hilo llega antes al procesador.
- **Necesidad de sincronización:** Debemos evitar que los hilos se bloqueen entre ellos.



# Enlaces de ayuda

- Teoría de Sistemas Operativos  
<http://gwolf.sistop.org/laminas/05-procesos-e-hilos.pdf>
- Procesos e Hilos  
<http://effbot.org/media/downloads/librarybook-threads-and-processes.pdf>

