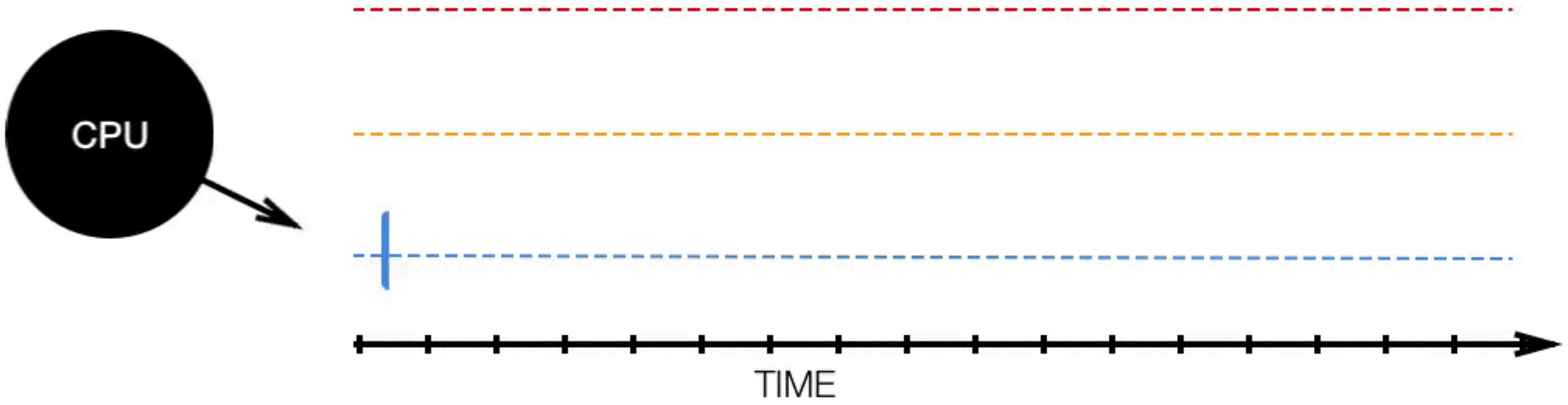
Threads

Ayudantía 08

¿Qué son los Threads?

Secuencias de instrucciones que pueden ser ejecutadas de forma “paralela”.
Permiten a una aplicación realizar varias tareas “a la vez”.

¿Cómo funcionan?



¿Cuándo usar Threads?

Creación y ejecución de Threads

Como son muy dedicados, ya saben porque leyeron el material de clases :)

Acá un BREVE repaso

Usando una función

```
mythread = threading.Thread(target=my_func, args=(*args), daemon=True)
mythread.start()
mythread.join()
lock.acquire()
lock.release()
```

Usando una clase

```
class MyThread(threading.Thread):

        def __init__(self, *args):
                super().__init__()

        def run(self):
                # do some stuff...

t1 = MyThread()
t1.start()
```

¿Cómo alcanzo la felicidad plena?

¿Qué es target?

¿Qué es join?

¿Qué es lock.release?

¿Qué es daemon?

¿Qué es args?

¿Qué es lock.acquire?

¿Por qué existimos?

Lo más importante...

- Mantener un poco de orden en la incertidumbre de los threads para que no sea puro caos!

- Mantener variables compartidas consistentes



Let the lloratón begin...

Es fin de semestre, y los alumnos llegan corriendo apurados a la lloratón para intentar aprobar el curso. Lamentablemente, hay solo 5 ayudantes para todos los alumnos, y el tiempo es corto, por lo que todos se empujan y pelean para lograr llegar a tan anhelado 4. Todo es un desorden, ya no existen las colas y nadie respeta las normas sociales que inducen al orden y buen funcionamiento de la sociedad. Para hacerla corta... tienes que modelar esto.

Requisitos

Debemos implementar las clases:

Lloratón: Thread principal, es el encargado de llevar el tiempo de la simulación. Una vez que este thread termine, deben terminar todos los otros. Es el encargado de inicializar el spawner de alumnos

Alumno: Thread que busca subir su nota a toda costa. Posee un id, y una nota de examen. Una vez terminada la lloraton, tendrá una nota final.

Ayudante: Tiene un nombre, y una piedad que indica cuantas décimas esta disponible a dar a un alumno.

Requisitos

La llegada de alumnos a la lloratón, será procesada por el spawner, thread encargado de crear e iniciar a los alumnos nuevos que vayan llegando a la lloratón.

Los ayudantes solo podrán atender a 1 alumno a la vez, y se demora un tiempo entre 2 y 5 segundos en decidir cuántas décimas darle al alumno.

Como último requisito, se te pide que una vez que la lloratón termine, indiques cuántos alumnos aprobaron y cuántos reprobaron, y luego todos los threads terminen de ejecutar (nadie puede seguir reclamando una vez que la lloratón terminó)

Vamos a python...