****

**Examen #2**

Semana: 10

**Nombre del estudiante:**

Serlio Alejandro Girón Paz 12141146

**Sede de estudio:**

UNITEC Tegucigalpa

**Docente:**

Román Arturo Pineda Soto

**Sección:**

1741 SISTEMAS OPERATIVOS II 2024Q2

**Lugar y fecha de entrega:**

Comayagüela M.D.C. 18 de junio de 2024

Universidad Tecnológica Centroamericana  
Facultad de Ingeniería  
Sistemas Operativos 2 - 6:30 p.m.  
25 puntos oro

En el lenguaje de programación de su elección elabore un simulador de caché en disco.

El programa debe generar de forma aleatoria un espacio de direcciones y cargarlos con datos (también pueden ser aleatorios, pero con sentido).

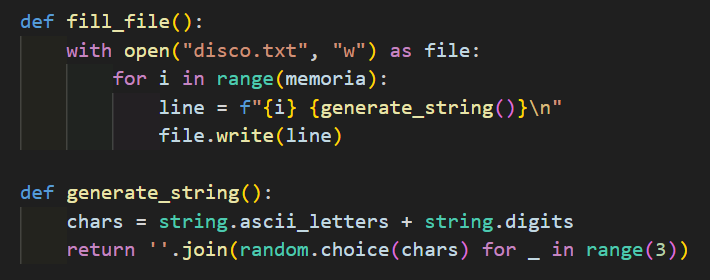
Debe seleccionar un espacio de al menos ⅛ del tamaño del espacio principal de  
memoria para el caché.

Debe simular dos procesos consumidores del espacio y generar lecturas aleatorias, estas lecturas deben llenar el caché y este se debe reciclar de forma circular.

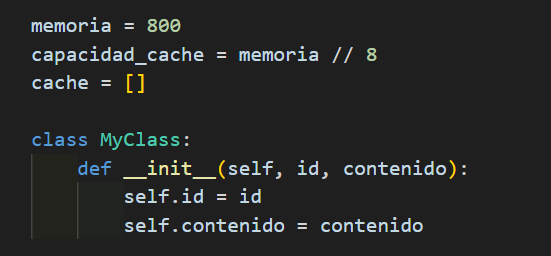
Si una lectura se encuentra en el caché debe escribir en pantalla "match de caché" y mostrar el valor que se obtuvo del mismo.

Debe considerar todas las buenas prácticas vistas en clase para el buen manejo del caché.

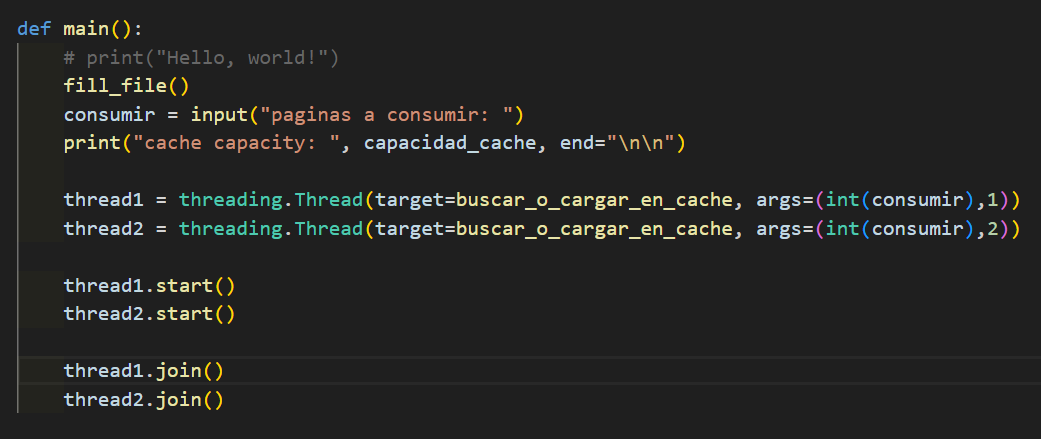
El primer paso es generar paginas de manera aleatoria, esto lo hice con las siguientes funciones.



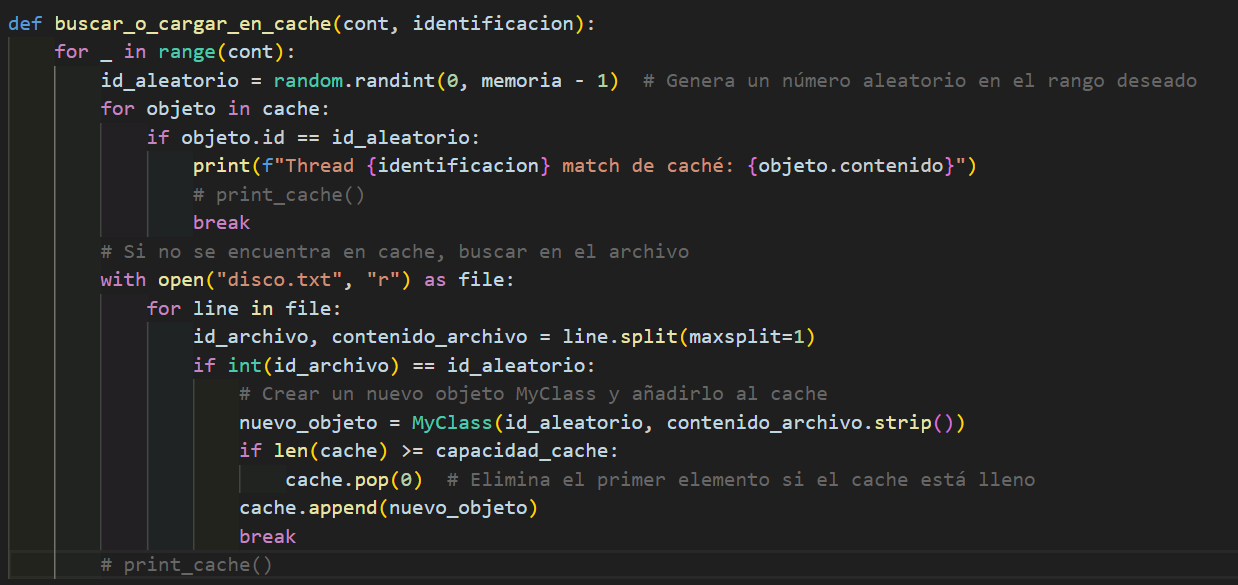
Con la función fill\_file llenamos un archivo disco donde guardamos páginas. Guardamos el id de la página y guardamos su contenido. El contenido lo generamos con generate\_string. Con la función generate\_string genero cadenas aleatorias de tres caracteres. La cadena es una cadena alfanumérica. Después de tener el disco con todas las paginas comenzamos con el trabajo de la cache.



En este proyecto maneje los dos procesos de manera concurrente con la ayuda de threads. Desde el main creo dos threads y les asignó la función a realizar. De igual manera le envió a los threads cuantas lecturas deben generar con la variable ‘consumir’.



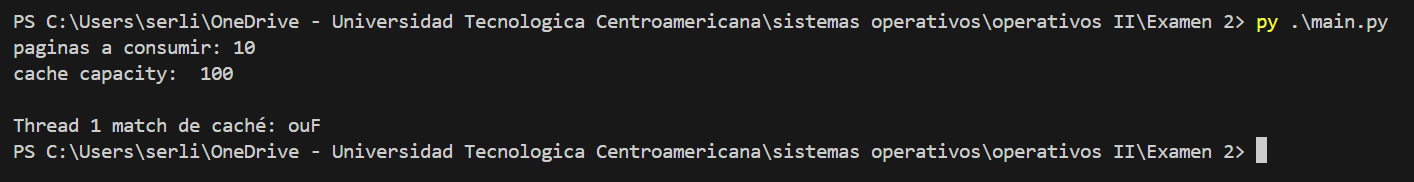
Cada uno de los threads realiza la siguiente tarea.

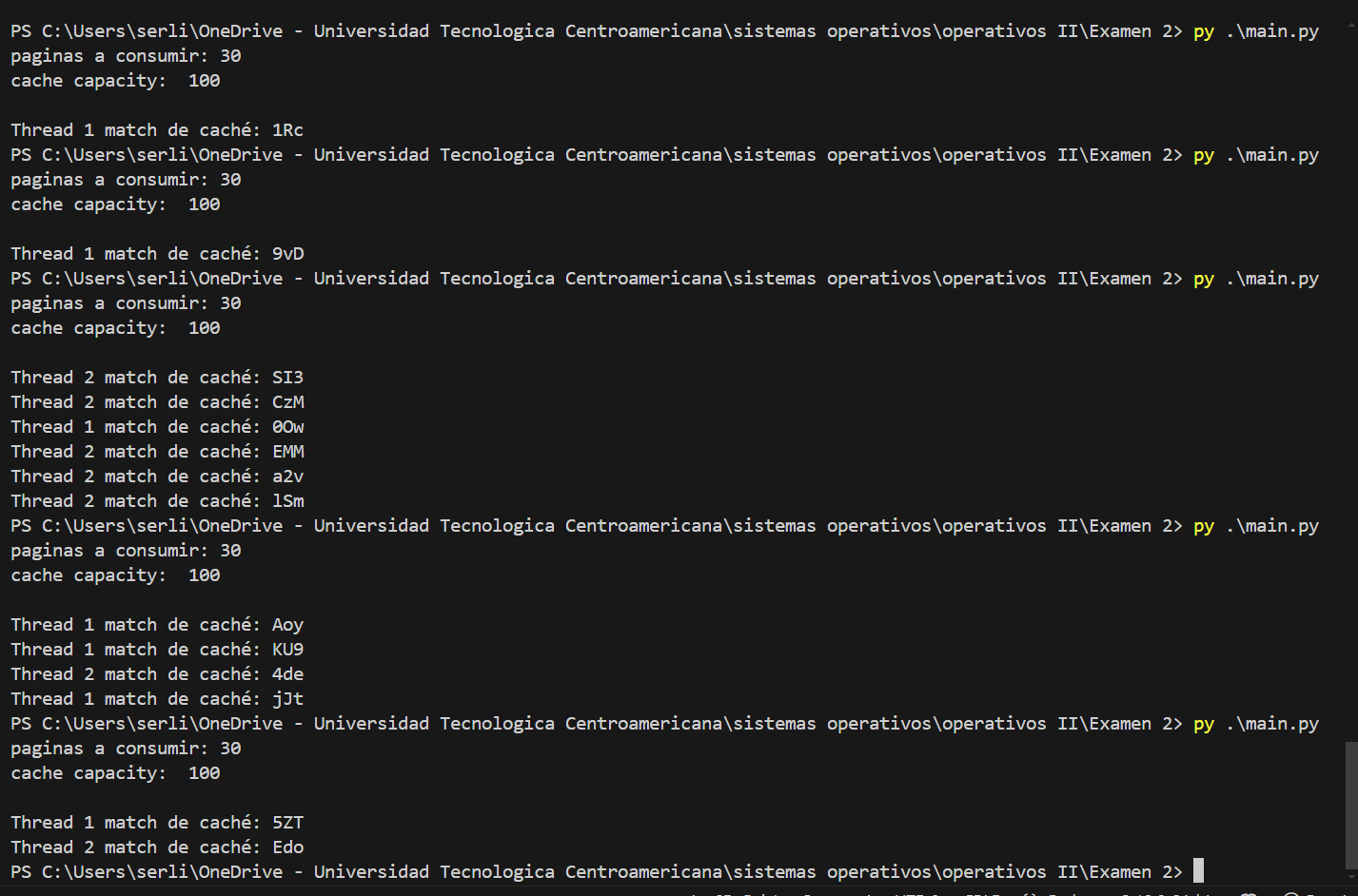


Esta tarea recibe el numero de lecturas a generar. Primero toma un numero aleatorio dentro del rango de ids que se encuentran en el disco duro. Luego va a la cache y busca si hay se hace match. Si no se hace match añade la página a la cache.

Salidas

Dependiendo del numero de lecturas que le pedimos a los procesos generar se determina el numero de matches, mientras más lecturas pidamos generar. Mas matches tendremos.





Para cada corrida se hacen diferentes matches y diferentes números de matches.

Para tener un numero considerable de matches necesitamos un numero considerable de lecturas generadas.

