



# FI UNAM

## UNA SITUACIÓN COTIDIANA PARALELIZABLE

*Nombre: De La Cruz López Carlos Odette*

*Grupo: 06*

*Profesor: Ing. Gunnar Eyal Wolf Iszaevich*

*Fecha: 29/Noviembre/2022*

# SISTEMAS OPERATIVOS



## Introducción

Sabemos que los sistemas operativos multitarea ejecutan más de una tarea a la vez. Esto es solo cierto si dispone de más de un procesador, si no, va alternando procesos en un solo procesador.

La ejecución de un proceso implica que el mismo en el momento de ejecución utiliza todo el procesador y con el espacio de memoria asignado. En sistemas multitarea, se van alternando procesos y cuando uno libera el procesador para que otro se ejecute, también libera la memoria asignada.

Esto lo realiza el planificador del sistema operativo lo cual implica la correcta gestión de la memoria para guardar los datos del proceso saliente, provocando cierta sobrecarga del sistema.

La programación multihilo se basa en que, dentro del desarrollo de la aplicación, hemos definido diversas tareas para que se ejecuten a la vez. Todas estas tareas forman parte de un mismo proceso.

Durante nuestro curso se nos explicó unos puntos importantes sobre la complejidad que tiene programar con hilos ya que:

No tenemos que preocuparnos cuando todos los datos que maneja un hilo son locales

Al utilizar variables globales o recursos externos, debemos recordar que el planificador puede interrumpir el flujo en cualquier momento

No tenemos garantía del ordenamiento que obtendremos

## Situación propuesta

En este proyecto se llegó a la idea de hacer un análisis de concurrencia en el apartado de videojuegos, o un caso claro sería, el utilizar en este proyecto sería Warzone, esto porque es un juego en el que se muestra un gran conflicto con el tema de paralelización, claro ejemplo serían dos. El acceso al juego mediante salas de espera, salas de juego y la llegada de una videollamada.

Provoca estos casos, concurrencia, estos casos son posibles, ya que, 2 o más personas. ¿Choquen o generen un por esta paralelización? En este proyecto planteamos el caso de acceso al juego en cuestión de menú, victoria.

Proponemos que cada hilo es un usuario diferente que accede al videojuego.

Primera cosa qué se analizó es el menú, en este menú se plantea el acceso a él desde el menú hacia una partida normal. En el cual podría ocurrir que 2 personas quieren acceder, al mismo tiempo, hay que realizar una propuesta de las variables globales en vez de considerar solo las variables locales, podríamos considerar

apuntadores y direccionamiento de memoria, al trabajar con hilos podríamos correr el riesgo de que no funcione, ya que conflicto harían 2 o más hilos a en un archivo que quisiese modificarse.

Segunda cosa qué se analizó sería el comportamiento del usuario, o sea, el jugador, al intentar acceder al juego desde su perspectiva, qué es lo que modifica en sus parámetros para poder llevar a cabo el acceso correcto al videojuego sin generar ningún tipo de concurrencias

Y un tercer caso, el cual es analizar cómo finaliza la partida, ya que al finalizar una partida se distribuye toda la calificación, puntaje y/o experiencia adquirida en el videojuego y se distribuye, pues en nuestro contador. A continuación, presentaremos qué tipo de estructura y primitiva se usaron para el desarrollo de este proyecto. ¿Cómo realmente se utilizó semáforos? Mute, multiplexores entre otros

### Mi solución

La solución que propongo en este proyecto consiste en descomponer lo que haría en un videojuego en 3 partes independientes interactuarán, posteriormente. En la estructura del Código al inicializarlo se declaran lo que serían nuestros primitivos, el cual nos ayudarán al desarrollo total del programa, consideramos a los primitivos. Esto gracias a las bibliotecas.

Segundo puesto, colocaremos todas las variables globales que utilizaremos, en nuestro caso sólo se utilizaron.

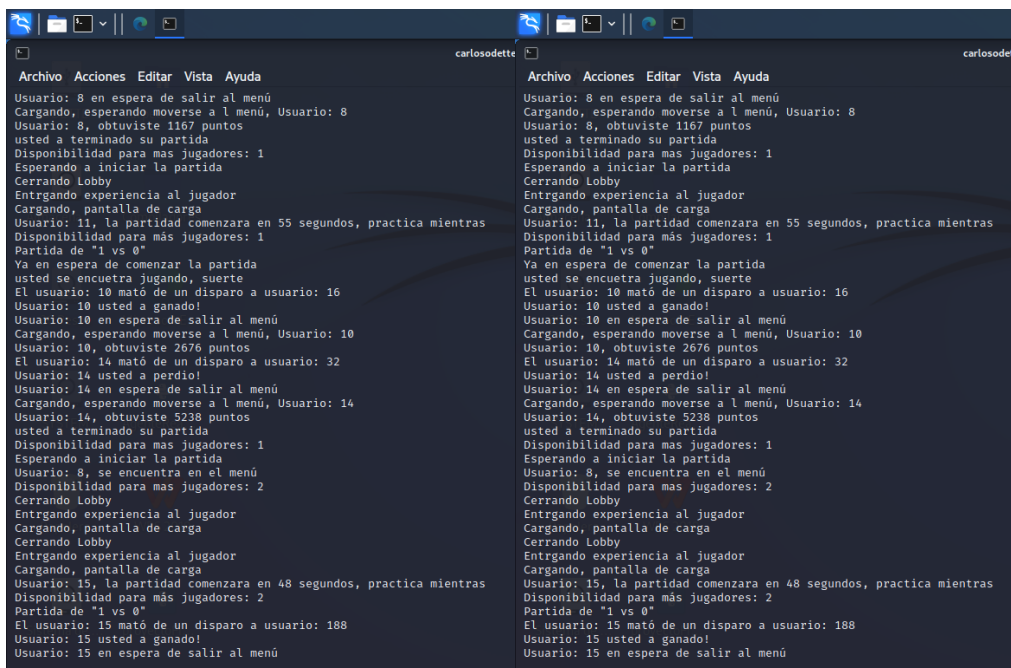
Tercer caso: El cual será estarán compartiendo entre ellas mismas y evitaremos la concurrencia.

Creamos la función desarrollo del lobby. En esta planteamos desde que se ejecuta el juego y cómo va accediendo poco a poco, ya que en la vida real iniciar sesión se requiere extraer toda la información de cada usuario individualmente. Proyectamos como fue el acceso a estos servidores. Los cuales tienen un acceso limitado para los usuarios. Además de que también existen cantidades finitas de salas. Para llegar al enfrentamiento contra los contrincantes. Otro caso que se va a haber mucho durante la codificación es que se programó algo llamado mensaje consola. El cual antes tiene una función. De reservar un hilo para que éste no genere conflicto con los demás hilos al intentar imprimir su estado actual. Durante el desarrollo de la práctica y del Código se comprende que. Intenta acceder al juego sin generar ningún tipo de conflicto entre sí. ¿Para que el juego o la imitación fuera algo? Medianamente real. ¿Se utilizaron métodos aleatorios para llegar casos en los que se puede ganar y/o se pueden perder. Además de que en la consola se nos plantea el tiempo que restante que. Para poder entrar a una sala y comenzar a jugar con los demás. Durante la secuencia de código se visualiza que al terminar una partida se nos envía una pantalla de carga, el cual también requiere trabajarse en hilos. Se

nos manda una solicitud, sobre nuestro puntaje obtenido durante la partida. Segundo apartado sería nosotros viendo el juego desde nuestra perspectiva personal ya que nosotros requerimos el acceso en el momento que entramos, ver en qué momento se activa o desactiva la secuencia de los hilos. Vas siguiendo los mismos casos planteados de iniciar el juego desde cero hasta finalizar una partida, el cual también veremos cómo estos estados se modifican desde nuestro lado. Y en la última función es la más pequeña, el cual intenta calcular todo lo que es la pantalla de carga negra, o sea, salir de la partida. La consola nos envía esos estados en la cual se va ajustando los hilos de mayor o menor nivel para que estos se pongan en pausa y continuó.

Para finalizar, se plantea una lista en la cual se manejarán todos los hilos, tanto para el usuario como para la simulación de warzone. ¿Podemos modificar en la sección de variables locales, que se encuentra en la función main, nosotros planteamos casos pequeños como 20 o 24? Para que sea más fácil de analizar, sin embargo, en un caso real se utiliza números extremadamente gigantes, el Código genera algún tipo de mensaje, el cual nos plantea un error que no genera un resultado diferente o que corrompa el Código solo es común mensaje de advertencia.

## Ejemplos de invocación



The image shows two side-by-side screenshots of a terminal window. Both windows have a title bar with the name 'carlosodette' and standard window controls. The terminal displays a log of game events in Spanish. The logs are identical in both windows, showing a sequence of events for multiple users (8, 10, 11, 14, 15) including waiting for the menu, loading, playing, and finishing the game. The text is as follows:

```
Archivo Acciones Editar Vista Ayuda
Usuario: 8 en espera de salir al menú
Cargando, esperando moverse a l menú, Usuario: 8
Usuario: 8, obtuviste 1167 puntos
usted a terminado su partida
Disponibilidad para mas jugadores: 1
Esperando a iniciar la partida
Cerrando Lobby
Entrgando experiencia al jugador
Cargando, pantalla de carga
Usuario: 11, la partidad comenzara en 55 segundos, practica mientras
Disponibilidad para más jugadores: 1
Partida de "1 vs 0"
Ya en espera de comenzar la partida
usted se encuentra jugando, suerte
El usuario: 10 mató de un disparo a usuario: 16
Usuario: 10 usted a ganado!
Usuario: 10 en espera de salir al menú
Cargando, esperando moverse a l menú, Usuario: 10
Usuario: 10, obtuviste 2676 puntos
El usuario: 14 mató de un disparo a usuario: 32
Usuario: 14 usted a perdió!
Usuario: 14 en espera de salir al menú
Cargando, esperando moverse a l menú, Usuario: 14
Usuario: 14, obtuviste 5238 puntos
usted a terminado su partida
Disponibilidad para mas jugadores: 1
Esperando a iniciar la partida
Usuario: 8, se encuentra en el menú
Disponibilidad para mas jugadores: 2
Cerrando Lobby
Entrgando experiencia al jugador
Cargando, pantalla de carga
Cerrando Lobby
Entrgando experiencia al jugador
Cargando, pantalla de carga
Usuario: 15, la partidad comenzara en 48 segundos, practica mientras
Disponibilidad para más jugadores: 2
Partida de "1 vs 0"
El usuario: 15 mató de un disparo a usuario: 188
Usuario: 15 usted a ganado!
Usuario: 15 en espera de salir al menú
```

## Entorno trabajado

Este código fue desarrollado en el lenguaje de programación Python 3.10 desde Visual Studio Code y la terminal de Kali. Se requirió de la ayuda de las bibliotecas. Thread, random y Time

Este código fue desarrollado en el lenguaje de programación. Python 3.10. Se requirió de la ayuda de las bibliotecas. Para ejecutar el programa solo requerimos ejecutar el python3.10 y el nombre de nuestro archivo, en nuestro caso, proyectofinal.py

## **Conclusiones.**

Este tipo de programación suele ser algo compleja, sin embargo, no tiene el beneficio de que tiene mejores tiempos de trabajo, ya que no trabaja el procesador individualmente. Sino que gracias a sus hilos. Divide las labores de nuestro programa generando que sea más rápido, el dilema es que genera algún tipo de conflicto en la memoria variable o función, ya que 2 hilos intentarán ejecutar en la misma variable o dirección de memoria. En el mismo tiempo, generando así un conflicto y provocando que el programa sea erróneo. Tiene una complejidad mayor este tipo de programación, ya que no se programa como se programarían POO Y estructuras de datos. En nuestro caso no pudimos plantear el uso de una interfaz gráfica, por lo complicado que nos resultó. Sin embargo, revisando ejemplos en Internet. Tiene una mayor funcionalidad este tipo de. Actividades que cuando se le tiene un manejo más amplio. Fue importante aprender esto durante nuestro curso.