

Universidad de Carabobo  
Facultad Experimental de Ciencias y Tecnología  
Departamento de Computación  
Asignatura: Sistemas Operativos – CAO503  
Profesora: Mirella Herrera

**PROYECTO: Sincronización entre Procesos / Semestre 1-2023**

**Objetivo**

Poner en práctica, los conocimientos adquiridos sobre el manejo de la concurrencia entre procesos, utilizando herramientas para la sincronización y comunicación, aplicadas en la resolución de problemas.

**Condiciones**

1. El proyecto se desarrollará en grupos de máximo tres (03) estudiantes.
2. Cada grupo deberá desarrollar el ejercicio 1, y escogerá otro de la lista de problemas restantes.
3. El código deberá ser desarrollado en Lenguaje C, intradocumentado y la herramienta de sincronización entre procesos es semáforos.
4. Copia de soluciones no serán revisadas y el grupo perderá la calificación del proyecto.
5. El proyecto deberá ser entregado en fecha tope: **domingo 26 de noviembre de 2023 a las 11: 59 pm**

**Pautas para la Entrega**

1. Archivo .zip con el proyecto: código en lenguaje C, makefile y README.md.
2. Video de la ejecución con los distintos casos de prueba y sus resultados, contemplando datos de prueba para ejecuciones correctas y casos de borde. Debe colocar banderas en la ejecución para reconocer los caminos de las distintas pruebas.
3. Documentación (en formato PDF, letra tipo Calibri tamaño 12 e interlineado sencillo), explicando su análisis, procesos, secciones críticas, recursos críticos, escenarios de concurrencia, así como la utilidad y aprendizaje con el proyecto. **Es importante señalar que todas las asunciones que realicen en la interpretación de las diferentes problemáticas, deben ser expresadas en el documento y recuerden que el paradigma de programación es concurrente, por lo tanto, no se puede restringir el orden en que los procesos se entrelazan o ejecutan.**
4. El archivo .zip deberá ser enviado al correo electrónico [sistemasoperativosfacytuc@gmail.com](mailto:sistemasoperativosfacytuc@gmail.com), indicando en el asunto: Proyecto SO-1-2023: nombres, apellidos y número de cédula; de cada integrante del grupo y en el texto señalar los problemas seleccionados.

### **Problema 1 para todos los grupos**

1. Se dispone de una base de datos a la que se puede acceder mediante múltiples hilos concurrentes. Sobre la base de datos pueden ejecutarse tres operaciones: leer, escribir y administrar.

Las operaciones cumplirán las siguientes reglas de coordinación:

- Se permiten lecturas concurrentes.
- En cada momento sólo puede haber activa una operación de escritura.
- Mientras se ejecuta una operación de escritura, no se puede realizar ninguna lectura.
- Mientras se ejecutan operaciones de lectura, no se puede realizar ninguna escritura.
- La operación de administrar se puede ejecutar en cualquier momento y puede solaparse con las operaciones de lectura o de escritura que estén realizándose en ese momento.
- Pueden realizarse varias operaciones de administración concurrentes.
- Mientras se está realizando alguna operación de administración, las nuevas solicitudes de lectura o de escritura deben mantenerse en espera.

#### **Se pide contabilizar:**

- a) Número de operaciones realizadas, por tipo y por hora
- b) Número total de operaciones por tipo, en el día
- c) Número de veces que se realizaron operaciones concurrentes permitidas, por hora; y número y tipo de operaciones por cada vez. Por ejemplo, en el día x hubo 15 veces en las que se realizaron operaciones concurrentes permitidas y sus tipos fueron:  
Vez 1: n lecturas, m escrituras y p administración.  
Vez q: n lecturas, m escrituras y p administración
- d) Número e identificación de Hora(s) pico por tipo de operación durante el día

La simulación deberá prestar atención para evitar las situaciones de interbloqueo, inanición y espera activa. Se debe procurar un elevado grado de concurrencia entre los procesos. Asimismo, para que la simulación sea inteligible, cada hilo deberá emitir mensajes informativos por pantalla cada vez que su situación se modifique. De esta forma, la salida permitirá tener acceso a toda la información, para que se pueda verificar el comportamiento.

### **Problemas restantes, para escoger uno (01) por grupo**

2. Se desea modelar el siguiente juego de mesa:  
Del juego forman parte N jugadores y un jefe de mesa. Los jugadores pueden mover sus fichas todos al mismo tiempo, siempre y cuando el jefe de mesa no esté reordenando el tablero, lo cual hará cada vez que termine de pensar sus cambios. Esto implica, que el jefe de mesa debe esperar a que la mesa quede libre de jugadores antes de reordenar el tablero.

El jefe de mesa además de la función de reordenar el tablero, dejará cartas con instrucciones respecto al comportamiento que debe tomar el jugador en caso que el jefe de mesa esté esperando para reordenar el tablero. Hay 2 tipos de cartas, una que indica que se debe esperar a que el jefe de mesa reordene el tablero para poder jugar, y otra que le permite jugar antes del reordenamiento de fichas.

El jugador debe sacar una carta cada vez que quiera jugar y el jefe de mesa esté esperando para reordenar el tablero. Solamente una persona (jugador/jefe de mesa) podrá tener acceso al mazo de cartas en cada momento.

En caso de que no queden cartas en la mesa, el jugador deberá esperar a que el jefe de mesa reordene el tablero. Asimismo, el mazo nunca puede tener más de 10 cartas apiladas.

El jefe de mesa deberá poder hacer las 2 funciones a la vez (pensar reordenamiento y reordenar con elegir carta y colocarla en mazo).

Se dispone de las siguientes funciones que son ejecutadas por los jugadores:

- pensar\_jugada()
- sacar\_carta\_de\_mazo():Carta
- jugar()

Se dispone de las siguientes funciones que serán ejecutadas por el Jefe de Mesa:

- pensar\_reordenamiento()
- reordenar\_tablero()
- elegir\_proxima\_carta():Carta
- colocar\_carta\_en\_mazo(Carta)

**Se pide contabilizar:**

- a) Número de veces que el jefe de mesa reordenó el tablero durante una partida
- b) Número de jugadas por jugador durante la partida
- c) Número total de cartas de jugar durante la partida, por jugador
- d) Número total de cartas de esperar durante la partida, por jugador
- e) Número total de cartas de jugar durante la partida
- f) Número total de cartas de esperar durante la partida

La simulación deberá prestar atención para evitar las situaciones de interbloqueo, inanición y espera activa. Se debe procurar un elevado grado de concurrencia entre los procesos. Asimismo, para que la simulación sea inteligible, jugadores y el jefe de mesa deberán emitir mensajes informativos por pantalla cada vez que su situación se modifique. De esta forma, la salida permitirá tener acceso a toda la información, para que se pueda verificar el comportamiento.

3. El bingo **Dos de Oro** cuenta con un presentador y varios jugadores. Antes de comenzar el juego, cada jugador genera aleatoriamente una cantidad de cartones entre 1 y 10 con los que comenzará el juego. Obviamente, todos los números de un cartón son distintos. Cada cartón contiene 15 números, de los 90 posibles. Cuando todos los jugadores han generado sus cartones, el presentador empieza a cantar los números y si algún jugador tiene bingo, se lo comunica al presentador. El presentador confía en que los jugadores no le engañan con respecto al bingo, pero sólo el primero en gritar bingo es el ganador, de forma que el presentador debe confirmar al ganador y entonces la partida termina. Cada jugador empieza con 100 fichas. Cada cartón cuesta 5 fichas y por cada bingo ganado recibe el 90 % de lo recaudado. Un jugador no puede generar más cartones de los que puede pagar. Cuando la partida termina, se comienza una nueva entre los jugadores que todavía

tengan suficientes fichas. Si no hay al menos dos jugadores que puedan pagar un cartón, el presentador cierra el bingo y todo acaba.

La simulación deberá prestar atención para evitar las situaciones de interbloqueo, inanición y espera activa. Se debe procurar un elevado grado de concurrencia entre los procesos. Asimismo, para que la simulación sea inteligible, el presentador y los jugadores deberán emitir mensajes informativos por pantalla cada vez que su situación se modifique. De esta forma, la salida permitirá tener acceso a toda la información, lo cual quiere decir que cuando un jugador está listo para jugar, procesa un número, espera a que salgan números nuevos o tiene bingo, esto debe indicarlo en un mensaje junto con toda la información complementaria necesaria (cuáles son los cartones que ha generado, cuál es el número que va a procesar, cuánto dinero le queda, entre otros). Análogamente, el presentador debe emitir mensajes cuando empieza una nueva partida, cuando emite números, cuando espera a que alguien anuncie un bingo y así sucesivamente, de forma tal que pueda verificarse el comportamiento.

Algunas situaciones importantes que deben ser consideradas, son las siguientes:

- **Pérdida de números:** Algunos jugadores leen los números que ha leído el presentador más rápido que otros, pero hasta que todos los jugadores no hayan leído un número salido del bombo, este no debe ser borrado de la pizarra de números.
- **Violación de la exclusión mutua:** Si, por ejemplo, el presentador quiere introducir un nuevo número en la pizarra de números que han salido del bombo, mientras otro proceso está leyendo de la pizarra.
- **Inanición:** Si, por ejemplo, un jugador se queda esperando a que el presentador lea nuevos números y mientras tanto, otro jugador consigue bingo y el primer jugador nunca leyó el número de la pizarra.
- **Espera activa:** Por ejemplo, que un jugador compruebe constantemente si el presentador ha anunciado números nuevos. Lo correcto es esperar a que el presentador lo despierte autorice en su momento.
- **Establecimiento de turnos:** No se debe hacer suposiciones sobre el orden en el que se pasan a ejecutar los procesos. Por ejemplo, que el presentador espere a que todos los jugadores procesen un número antes de anunciar uno nuevo. El presentador debe leer los números tan rápido como sea posible hasta que alguien obtenga bingo o bien se acaben los números. Otro caso de establecimiento de turnos se produciría si los jugadores gestionaran sus cartones en forma exclusiva, uno cada vez.
- **Ausencia de concurrencia o baja concurrencia:** Por ejemplo, si la gestión de los cartones se hiciera en forma serial. Se supone que los cartones son locales a cada proceso, por lo que cada jugador puede marcar los números que ya han salido a su ritmo. Lógicamente, los jugadores que juegan más cartones tenderán a tardar más en hacerlo que los que tienen menos, por lo que es de esperar que acumularán cierto retraso en el procesamiento de los números con respecto a los otros jugadores.

**Se pide:**

- Contabilizar la cantidad de dinero que ha ganado el presentador para el bingo (Cada ficha tiene el valor de 10 monedas)
- Número de cartones ganadores por jugador y en total
- Jugador(es) con más bingos ganados
- Jugador(es) con menos bingos ganados

4. En una tienda de computadores trabajan X vendedores que atienden a Y clientes en cada una de las secciones de la tienda. Estos vendedores ayudan a los clientes a armar una configuración que será registrada en una orden para solicitar un desktop o una laptop, y luego entregan la orden a los receptores del almacén. El personal del almacén está conformado por tres receptores, el receptor 1 clasifica el tipo de orden de acuerdo al pedido entre desktop y laptops. Las órdenes para computadores de escritorio son transferidas al receptor 2 y los portátiles al receptor 3.

Los receptores 2 y 3 se encargan de armar las configuraciones colocando las partes del computador (pantallas, cpu, teclados, ratones, cámaras, etc.) seleccionadas por el cliente, de acuerdo al stock existente en el mismo.

A medida que se arman las configuraciones y son completadas, los receptores del almacén avisan al vendedor que consignó la orden que la misma está lista, para que trasladen el computador a los cajeros de la tienda. Asimismo, el receptor 1 debe estar pendiente de la disponibilidad en stock de los productos y solicitar la reposición al almacén central en caso de llegar a un nivel mínimo de productos en el almacén de la tienda.

La tienda cuenta con M gerentes que deben ser informados en todo momento, sobre lo que está ocurriendo con las órdenes, productos y servicios.

Es así como se identifican dos elementos claves para el uso de las herramientas de sincronización y comunicación entre procesos: concurrencia y competencia. Los clientes hacen solicitudes, los vendedores preparan las órdenes de equipos, el receptor 1 canaliza las órdenes de acuerdo a desktops y laptops, los receptores 2 y 3 configuran los equipos de acuerdo a los componentes solicitados, el receptor 1 conoce la cantidad de componentes disponibles en el almacén local y central; y en algún lado debe llevarse registro de las configuraciones vendidas, montos, cantidad disponible, etc.

**Se pide,**

a) **Desarrollar los módulos:**

- Vendedor
- Cliente
- Receptor 1
- Receptores 2 y 3
- Cajero
- Almacén de la tienda y central
- Mostrador de receptores
- Mostrador de cajeros
- Gerentes
- El Sistema

b) **Al final del día contabilizar lo siguiente:**

- Número de laptops y desktops vendidos
- Número de órdenes clasificadas
- Número de veces que se alcanzó el mínimo en el stock para reposición
- Cantidad de productos solicitados al almacén central
- Cantidad disponible en el almacén de la tienda luego de cada venta
- Monto total facturado

La simulación deberá prestar atención para evitar las situaciones de interbloqueo, inanición y espera activa.

Se debe procurar un elevado grado de concurrencia entre los procesos. Asimismo, para que la simulación sea inteligible, los vendedores, cajeros, clientes, receptores, mostradores, gerentes, almacén y tienda central; deberán emitir mensajes informativos por pantalla cada vez que su situación se modifique. De esta forma, la salida permitirá tener acceso a toda la información, para que se pueda verificar el comportamiento.