

**CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E  
INGENIERÍAS (CUCEI)**

**DIVISIÓN DE ELECTRÓNICA Y COMPUTACIÓN  
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES**

**Carrera:** Ingeniería en Computación

**Nombre Materia:** Seminario de Solución de Problemas de  
Sistemas Operativos

**Profesor:** Valdés López Julio Esteban

**SECCIÓN:** Do8

**Nombre alumno:** López Arellano Ricardo David

**CODIGO:** 217136143



**Actividad 5**

**Fecha de entrega: 18/03/2023**

## **Investigar que son los algoritmos de planificación:**

Los algoritmos de planificación son una parte importante de los sistemas operativos y se utilizan para determinar el orden en que los procesos deben ejecutarse en un sistema informático. Estos algoritmos se encargan de asignar recursos del sistema, como la CPU, memoria y dispositivos de entrada/salida, a los diferentes procesos que se están ejecutando en el sistema.

## **Investigue en que consiste y las soluciones de:**

- **La cena de los filósofos**

No existe una única solución a "La cena de los filósofos" ya que se trata de una obra de arte que permite diferentes interpretaciones y lecturas. Sin embargo, se puede decir que el cuadro representa una alegoría de la filosofía y la búsqueda de la sabiduría, y que cada uno de los filósofos representados simboliza una corriente de pensamiento.

Además, la pintura muestra la importancia del diálogo y la discusión en la filosofía, ya que los filósofos están inmersos en una profunda conversación. También se puede observar la diversidad de pensamiento y enfoques representados en la mesa, lo que sugiere que la verdad y la sabiduría pueden encontrarse en diferentes perspectivas y enfoques.

- **Productor-Consumidor**

El problema del Productor-Consumidor es un clásico en la programación concurrente que involucra la coordinación de múltiples procesos o hilos de ejecución.

En este problema, existen dos tipos de procesos: productores y consumidores. Los productores crean elementos y los colocan en un buffer compartido, mientras que los consumidores toman elementos del buffer y los procesan. El objetivo es garantizar que el buffer no se sobrecargue con elementos no procesados, ni que los consumidores intenten procesar elementos inexistentes en el buffer.

La solución más común para este problema es utilizar una estructura de datos llamada semáforo, que se encarga de controlar el acceso al buffer compartido. El semáforo mantiene un contador que indica el número de elementos disponibles en el buffer, y los productores y consumidores deben solicitar permiso al semáforo antes de acceder al buffer.

- **Lectores-Escritores**

El problema de los Lectores-Escritores es otro clásico de la programación concurrente que involucra la coordinación de múltiples procesos o hilos de ejecución.

En este problema, existen dos tipos de procesos: lectores y escritores. Los lectores pueden acceder a una región compartida de datos para leerlos, mientras que los escritores pueden acceder a la misma región compartida para escribir nuevos datos. El objetivo es garantizar que los datos compartidos sean accesibles para los procesos lectores y escritores de manera segura y sin generar conflictos.

La solución más común para este problema es utilizar una estructura de datos llamada semáforo, que se encarga de controlar el acceso a la región compartida de datos. Se utilizan dos semáforos: uno para controlar el acceso de los escritores y otro para controlar el acceso de los lectores.

Cuando un escritor quiere acceder a la región compartida, solicita permiso al semáforo de escritores. Si hay otros escritores o lectores accediendo a la región compartida, el semáforo detiene al escritor hasta que se haya liberado el acceso.

## **Preguntas.**

### **1. ¿En qué consiste el problema de la concurrencia?**

El problema de la concurrencia es un problema común en la programación que surge cuando varios procesos o hilos de ejecución compiten por los mismos recursos o datos compartidos. Cuando varios procesos intentan acceder a los mismos recursos simultáneamente, puede ocurrir que los resultados de la ejecución no sean los esperados o incluso que se produzcan errores o bloqueos.

El problema de la concurrencia puede ser especialmente complejo cuando se trabaja con sistemas distribuidos o en tiempo real, donde los procesos pueden estar ubicados en diferentes nodos y los recursos compartidos pueden incluir bases de datos, archivos, colas de mensajes, entre otros.

### **2. ¿Cuáles son los procesos concurrentes cooperantes?**

Los procesos concurrentes cooperantes son aquellos procesos que trabajan juntos para lograr un objetivo común, compartiendo recursos y comunicándose entre sí para coordinar su actividad. Estos procesos son comunes en sistemas distribuidos, donde se requiere la colaboración de múltiples procesos para realizar tareas complejas.

Entre los procesos concurrentes cooperantes más comunes se incluyen:

1. **Productor-Consumidor:** proceso productor que genera datos y los almacena en una estructura compartida, y proceso consumidor que extrae los datos de la estructura compartida y los utiliza para realizar otras tareas.
2. **Lectores-Escritores:** proceso lector que accede a datos compartidos para leerlos, y proceso escritor que accede a los datos compartidos para modificarlos.
3. **Filósofos cenando:** proceso que representa a un filósofo que puede estar pensando o comiendo. Los filósofos comparten recursos (los palillos) y

deben coordinar su actividad para evitar conflictos y asegurar que todos puedan comer.

4. Trabajadores en una fábrica: proceso que representa a un trabajador que realiza una tarea en una fábrica. Los trabajadores deben coordinarse para asegurar que la producción se realice de manera eficiente y sin errores.

### **3. ¿En qué consiste la Exclusión mutua?**

La exclusión mutua es un problema de concurrencia que se refiere a la necesidad de garantizar que varios procesos no accedan simultáneamente a un recurso compartido, ya que esto podría provocar errores en la ejecución o resultados inconsistentes.

La exclusión mutua se basa en la idea de que, cuando varios procesos compiten por un recurso compartido, es necesario asegurar que sólo uno de ellos tenga acceso al recurso en un momento dado. Esto significa que, mientras un proceso está utilizando el recurso, los demás procesos deben esperar hasta que el recurso esté disponible de nuevo.

Para lograr la exclusión mutua, se utilizan mecanismos de sincronización como los semáforos, los mutexes y los monitores. Estos mecanismos permiten bloquear el acceso al recurso compartido para todos los procesos excepto para uno solo, que es el que tiene el acceso exclusivo. Cuando el proceso que está utilizando el recurso termina, libera el recurso para que otro proceso pueda acceder a él.

### **4. Defina Interbloqueo.**

El interbloqueo, también conocido como deadlock en inglés, es una situación en la que dos o más procesos o hilos de ejecución quedan bloqueados indefinidamente mientras esperan por recursos que están siendo utilizados por otros procesos. En otras palabras, el interbloqueo es un problema de concurrencia que puede ocurrir cuando varios procesos compiten por recursos compartidos y cada uno de ellos se bloquea esperando a que otro proceso libere un recurso que necesita para continuar su ejecución.

El interbloqueo es un problema crítico en sistemas concurrentes, ya que puede provocar la pérdida de recursos y la inestabilidad del sistema. Para evitar el interbloqueo, se pueden utilizar técnicas como la prevención, la detección y la recuperación. La prevención implica evitar la formación de ciclos de espera mutua, mientras que la detección implica la identificación temprana del interbloqueo para tomar medidas y recuperarse del mismo. En la recuperación, se toman medidas para liberar los recursos y desbloquear los procesos involucrados en el interbloqueo.

### **5. Defina Inanición.**

La inanición, también conocida como starvation en inglés, es un problema que ocurre en sistemas concurrentes cuando un proceso o hilo de ejecución no puede obtener los recursos necesarios para continuar su ejecución debido a la

competencia por dichos recursos. En otras palabras, la inanición se refiere a una situación en la que un proceso se queda esperando indefinidamente a que se le asigne un recurso que necesita para continuar, pero no lo obtiene debido a la priorización de otros procesos que compiten por el mismo recurso.

La inanición puede ocurrir en sistemas concurrentes cuando los recursos compartidos no están siendo gestionados adecuadamente, ya que algunos procesos pueden tener prioridad sobre otros, o los recursos pueden ser asignados de forma injusta o ineficiente. Por ejemplo, si varios procesos compiten por un recurso y siempre se le asigna a los mismos procesos, otros procesos pueden quedarse esperando indefinidamente sin poder acceder a dicho recurso, lo que puede causar inanición.

## **6. Defina Excesiva Cortesía.**

La excesiva cortesía, también conocida como over-politeness en inglés, se refiere a una forma de comportamiento en la que una persona muestra un nivel de cortesía o formalidad exagerado o innecesario, en situaciones donde no es realmente requerido. Esto puede implicar el uso excesivo de saludos, agradecimientos, disculpas o expresiones de gratitud, en contextos donde tales expresiones pueden ser consideradas innecesarias o incluso incómodas para las personas involucradas.

## **7. ¿Qué son los Hilos?**

En informática, un hilo (también conocido como thread en inglés) es una unidad de ejecución ligera que puede ser ejecutada simultáneamente con otras unidades de ejecución dentro de un proceso. En otras palabras, un hilo es un subproceso que puede ser programado para llevar a cabo tareas específicas de forma paralela a otros hilos que se ejecutan en el mismo proceso.

Los hilos son útiles en situaciones en las que un programa necesita realizar varias tareas al mismo tiempo, como por ejemplo, en aplicaciones que requieren una actualización constante de la interfaz de usuario mientras se realizan tareas de procesamiento en segundo plano. Al usar hilos, estas tareas pueden ser ejecutadas simultáneamente, mejorando así la eficiencia y el rendimiento del sistema.

## **8. ¿Qué son los Semáforos?**

Los semáforos son un mecanismo de sincronización de procesos que se utilizan en sistemas operativos y programación concurrente para controlar el acceso a recursos compartidos y prevenir problemas de exclusión mutua y condiciones de carrera.

Un semáforo es una variable entera que puede tomar valores positivos o cero y que es manipulada únicamente a través de dos operaciones atómicas: "wait" (esperar) y "signal" (señal). El "wait" reduce el valor del semáforo en uno si es

positivo, y si el valor es cero, el proceso que intenta realizar el "wait" se bloquea hasta que el valor del semáforo se vuelva positivo. El "signal" incrementa el valor del semáforo en uno, y si hay procesos bloqueados esperando en el semáforo, uno de ellos es desbloqueado y puede continuar su ejecución.

## **9. ¿Qué es lo que mejora el tener más de un núcleo?**

Tener más de un núcleo en un procesador mejora significativamente el rendimiento y la capacidad de procesamiento de un sistema. Los procesadores multinúcleo tienen varios núcleos de procesamiento en un solo chip, lo que permite a la computadora realizar múltiples tareas al mismo tiempo. Esto es posible gracias al paralelismo, que es la capacidad de realizar varias tareas en paralelo.

Cuando un procesador tiene varios núcleos, cada núcleo puede realizar una tarea diferente al mismo tiempo, lo que significa que se pueden ejecutar más tareas simultáneamente. Esto se traduce en una mejora del rendimiento y de la velocidad de procesamiento en tareas que pueden ser paralelizadas, como el procesamiento de datos, la edición de video, el renderizado de gráficos 3D, entre otros.