****

**Guía y Rúbrica de Evaluación**

**Segundo Previo**

| **Fecha:** |  |  |  |
| --- | --- | --- | --- |

| **INFORMACIÓN GENERAL** | |
| --- | --- |
| **Información del Curso** | |
| **Programa Académico:** | Ingeniería de Sistemas |
| **Asignatura:** | Sistemas operativos I |
| **Docente:** | Johan Smith Rueda Rueda |
| **Información del Estudiante** | |
| **Estudiante (s):** |  |
| **Código (s):** |  |
| **Nota:** |  |

**Estrategia de Evaluación desarrollada:**

| Examen | X |  | Trabajo en equipo |  |  | Exposición |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Laboratorio |  |  | Extensión |  |  | Revisión documental |  |
| Ensayo |  |  | Proyecto |  |  | Análisis casos de estudio |  |
| Artículo |  |  | Informe |  |  | Pitch |  |
| Taller |  |  | Mapas mentales y conceptuales |  |  | Modelado de negocio |  |
| Tarea Univirtual |  |  | Aprendizaje basado en problemas |  |  | Simulaciones |  |

Si su elección fue otra, describa a continuación la Estrategia de Evaluación propuesta:

| Nombre: |  |
| --- | --- |
| Descripción: |  |
|  |  |
|  |  |
|  |  |

**Información Rúbrica de Evaluación Resultados de Aprendizaje:**

| **Competencia asociada al perfil de egreso** | **Resultado de**  **Aprendizaje del Programa** | **Ítems relacionados a la evaluación de los RA** | **Criterio de**  **Evaluación**  **(RA de asignatura)** | **Porcentaje del avance de la evaluación** |
| --- | --- | --- | --- | --- |
| PE04: Evalúa desde una perspectiva ética, crítica y analítica las soluciones de innovación propuestas para dar respuesta a problemas del entorno. | RAP4: Reconocer responsabilidades éticas y profesionales en situaciones de ingeniería y hacer juicios informados, que deben considerar el impacto de las soluciones de ingeniería en contextos globales, económicos, ambientales y sociales. | Todos | RA1: Conecta los fundamentos teóricos y prácticos con su perfil profesional para realizar juicios informados mediante el análisis de casos de uso.  RA2: Examina el impacto que tienen los sistemas operativos con las diversas áreas de actuación del ingeniero de sistemas a través del relacionamiento de lo aprendido con diversos escenarios de aplicación. | 100% |
| PE07: Integra nuevos conocimientos para adaptarse y transformar el entorno, de acuerdo con las exigencias dinámicas de la profesión. | RAP7: Adquirir y aplicar nuevos conocimientos según sea necesario, utilizando estrategias de aprendizaje apropiadas. | Todos | 100% |

| **Indicadores** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Criterio de evaluación** | **Nivel Inferior**  **0≤58%** | **Nivel Bajo**  **59%≤70%** | **Nivel Medio**  **71%≤82%** | **Nivel Alto**  **83%≤92%** | **Nivel Superior**  **93%≤100%** |
| Identifica las funciones y estructura de un sistema operativo, y el impacto que estos tienen en la generación de soluciones de ingeniería. | No aplica los conocimientos adquiridos sobre la gestión de procesos que realiza un sistema operativo y su impacto en los sistemas informáticos.  No realiza juicios informados sobre el impacto de los sistemas operativos y las tecnologías relacionadas en las soluciones de ingeniería.  No expresa sus ideas de forma clara y coherente. | Identifica conceptos y responde con ellos a los contextos presentados, pero no logra explicar adecuadamente el impacto que ellos tienen en las soluciones de ingeniería. | Identifica algunos aspectos en los que los sistemas operativos y las tecnologías relacionadas impactan en las soluciones de ingeniería, pero no es del todo claro el argumento presentado. | Comprende las funciones del sistema operativo sobre la gestión de proceso, y los relaciona con el impacto que tiene en la generación de sistemas informáticos. | Comprende las funciones del sistema operativo sobre la gestión de procesos, y explica y sustenta de forma coherente y clara el impacto en los sistemas informáticos. |

**Rúbrica de Evaluación General:**

|  | **Indicadores de Desempeño** | | | | |
| --- | --- | --- | --- | --- | --- |
| **Criterio de evaluación** | **Nivel Inferior**  **0≥58%** | **Nivel Bajo**  **59%≥70%** | **Nivel Medio**  **71%≥82%** | **Nivel Alto**  **83%≥92%** | **Nivel Superior**  **93%≥100%** |
| Redactado por el docente. |  |  |  |  |  |

**Observaciones Generales:**

* El desarrollo del segundo previo se realiza en la fecha, hora y sitio estipulado en la Resolución del Plan de Estudios.
* La entrega del parcial se realiza en el tiempo estipulado a través de la UVirtual.
* El entregable corresponde a la guía y rúbrica y al código fuente de los dos ejercicios.

**CONTENIDO DE LA EVALUACIÓN**

Parcial en grupos de 2 personas.

El parcial se centrará en la programación concurrente y los mecanismos para la sincronización, de forma que se protejan las secciones críticas del código fuente. Cada grupo, tendrá uno de los problemas descritos más abajo.

Tenga en cuenta las siguientes recomendaciones:

* Cada grupo debe trabajar en un problema diferente.
* Deberán programar una solución que represente el problema descrito usando dos lenguajes: un lenguaje compilado y uno interpretado.
* La solución debe implementarse utilizando hilos. Si algún grupo desea trabajar con procesos, puede hacerlo.
* Pueden implementar semáforos o monitores para realizar la sincronización.
* Cada grupo debe presentar la descripción del problema correspondiente, y explicar el algoritmo realizado, el código implementado, haciendo énfasis en las secciones críticas.
* Deben realizar una comparación entre las dos implementaciones realizadas. Para ello, prueben ambos códigos en una misma máquina y realicen pruebas de rendimiento. Para el informe, documenten similitudes y diferencias entre los dos lenguajes.
* El día del parcial, cada grupo sustentará lo realizado.

**Problemas de concurrencia**

|  | **Integrantes** | **Problema** | **Lenguajes asignados** |
| --- | --- | --- | --- |
| 1 | 191974 - 192007 | Productores y consumidores | Rust - Python |
| 2 | 192040 - 192052 | Escritores y lectores | Go - Javascript |
| 3 | 191942 - 191996 | Filósofos cenando | Rust - Javascript |
| 4 | 192103 - 192109 | Barbero dormilón | Go - Python |
| 5 | 191991 - 192100 | Fumadores de cigarrillo | Kotlin - PHP |
| 6 | 192019 - 191940 | Misioneros y caníbales | Java - Ruby |
| 7 | 191926 - 191984 | Control de acceso a una base de datos | C# - Perl |
| 8 | 191995 - 192035 | Simulador de procesos en un sistema operativo | Java - Javascript |
| 9 | 192033 - 192114 | Simulador de procesos en un sistema operativo | Kotlin - Perl |
| 10 | 191999 - 191958 | Sistema de reserva de entradas | C# - PHP |
| 11 | 192055 - 192063 | Simulador de producción y distribución | Java - Ruby |

**Problema de productores y consumidores**

El programa describe dos tipos de procesos, productor y consumidor, ambos comparten un buffer de tamaño finito. La tarea del productor es generar un producto, almacenarlo y comenzar nuevamente; mientras que el consumidor toma (simultáneamente) productos uno a uno. El problema consiste en que el productor no añada más productos que la capacidad del buffer y que el consumidor no intente tomar un producto si el buffer está vacío. Deben trabajarlos bajo la situación que hay varios procesos productores y varios procesos consumidores.

**Problema de los escritores y lectores**

Existen dos tipos de procesos, lectores y escritores, los cuales requieren del acceso a una base de datos. Se deben considerar las siguientes reglas de acceso a la base de datos:

* Solamente puede haber un escritor al mismo tiempo, si un escritor o un lector quiere utilizar la base de datos y hay un escritor utilizándola, entonces debe esperar.
* Puede haber varios lectores utilizando la base de datos al mismo tiempo, pero si un escritor quiere usar la base de datos mientras los lectores la usan, deberá esperar.
* El último lector en usar la base de datos debe habilitarla para que pueda ingresar el siguiente escritor que estaba esperando la base de datos o en todo caso al siguiente proceso que requiera usarla.

**Problema de los filósofos cenando**

Cinco filósofos se sientan alrededor de una mesa y pasan su vida cenando y pensando. Cada filósofo tiene un plato de fideos y un tenedor a la izquierda de su plato. Para comer los fideos son necesarios dos tenedores y cada filósofo sólo puede tomar los que están a su izquierda y derecha. Si cualquier filósofo toma un tenedor y el otro está ocupado, se quedará esperando, con el tenedor en la mano, hasta que pueda tomar el otro tenedor, para luego empezar a comer.

Si dos filósofos adyacentes intentan tomar el mismo tenedor a una vez, se produce una condición de carrera: ambos compiten por tomar el mismo tenedor, y uno de ellos se queda sin comer.

Si todos los filósofos toman el tenedor que está a su derecha al mismo tiempo, entonces todos se quedarán esperando eternamente, porque alguien debe liberar el tenedor que les falta. Nadie lo hará porque todos se encuentran en la misma situación (esperando que alguno deje sus tenedores). Entonces los filósofos se morirán de hambre. Este bloqueo mutuo se denomina interbloqueo o deadlock.

El problema consiste en encontrar un algoritmo que permita que los filósofos nunca se mueran de hambre.

**Problema del barbero dormilón**

El problema consiste en una barbería en la que trabaja un barbero que tiene un único sillón de barbero y varias sillas para esperar. Cuando no hay clientes, el barbero se sienta en una silla y se duerme. Cuando llega un nuevo cliente, éste o bien despierta al barbero o —si el barbero está afeitando a otro cliente— se sienta en una silla (o se va si todas las sillas están ocupadas por clientes esperando). El problema consiste en realizar la actividad del barbero sin que ocurran condiciones de carrera. La solución implica el uso de semáforos y objetos de exclusión mutua para proteger la sección crítica.

**Problema de los fumadores de cigarrillo**

El caso de los fumadores consiste en un agente y tres fumadores que, para fumar, necesitan los ingredientes que les faltan para hacer un cigarrillo y fumárselo. Cada fumador posee un ingrediente en cantidades ilimitadas, pero les faltan otros dos. El agente posee cantidades ilimitadas de todos los ingredientes que son papel, tabaco y cerillas, pero solo deja en una mesa dos de estos ingredientes a la vez. Cada fumador posee un ingrediente distinto de los tres necesarios y según los ingredientes que deje el agente uno de los fumadores podrá fumar con los dos ingredientes que el agente deja.

**Problema de los fumadores misioneros y caníbales**

Tres caníbales y tres misioneros se encuentran en la orilla izquierda de un río. Hay un bote que puede llevar a una o dos personas. Si los misioneros quedan en minoría en cualquiera de las orillas al efectuar los viajes, los caníbales se los comerán. La tarea es determinar una sucesión de cruces que logre llevar a todos al lado opuesto del rio.

**Problema de control de acceso a una base de datos**

Para el problema de control de acceso a una base de datos, varios hilos o procesos necesitan acceder a la base de datos simultáneamente para realizar consultas, inserciones, actualizaciones o eliminaciones de datos. Es fundamental garantizar que solo un hilo o proceso pueda modificar una parte específica de la base de datos en un momento dado para evitar condiciones de carrera y mantener la consistencia de los datos.

**Problema de la gestión de inventario en una cadena de suministro**

Diferentes almacenes y puntos de venta compiten por el acceso a los productos en una cadena de suministro. La cadena de suministro consta de varios almacenes de distribución y puntos de venta. Cada uno de estos lugares necesita gestionar su inventario de productos, que puede incluir artículos de diferentes categorías y cantidades. El desafío radica en garantizar que los productos estén disponibles cuando se necesiten, minimizando al mismo tiempo los costos de almacenamiento y transporte. Dado que múltiples hilos estarán accediendo y modificando el inventario al mismo tiempo, se requerirá sincronización para evitar condiciones de carrera y garantizar la consistencia de los datos.

* Los hilos que representan los puntos de venta pueden generar pedidos cuando necesiten reabastecer su inventario de productos.
* Los hilos que representan los almacenes pueden procesar estos pedidos, verificar la disponibilidad de productos en el inventario central y coordinar la entrega de productos reabastecedores a los puntos de venta.

**Simulador de procesos en un sistema operativo**

Implemente un simulador que modele la ejecución de procesos en un sistema operativo. Utiliza hilos para representar procesos y algún mecanismo para sincronizar el acceso a recursos compartidos, como la CPU, la memoria, datos. Los recursos pueden ser definidos previamente, y se crearán N cantidad de procesos que van a acceder a los recursos.

**Sistema de reserva de entradas**

Implemente un sistema de reserva de entradas para un cine o teatro. Los clientes pueden intentar reservar asientos simultáneamente, por lo que necesitarás utilizar semáforos para garantizar la consistencia de la reserva.

**Simulador de producción y distribución**

El problema a resolver consiste en diseñar un simulador de producción y distribución que modele el flujo de productos a través de una cadena de suministro. Esta cadena de suministro incluye múltiples etapas, como la producción de productos, su almacenamiento en almacenes y su distribución a los clientes.

El sistema debe simular varias entidades, como fábricas, almacenes y clientes, así como el flujo de productos entre ellas. Cada fábrica produce ciertos tipos de productos a una tasa determinada, los almacena temporalmente y luego los distribuye a los clientes. Los clientes realizan pedidos de ciertos productos en cantidades específicas.

El objetivo del proyecto es diseñar e implementar un simulador de producción y distribución que modele eficazmente el flujo de productos a través de una cadena de suministro, para garantizar un funcionamiento seguro y eficiente del sistema.

**SOLUCIÓN DE LA ACTIVIDAD**

**1. Pregunta 1**

Responda en esta sección

| **Observaciones** |  | **Nota** |
| --- | --- | --- |
|  |

**2. Pregunta 2**

Responda en esta sección

| **Observaciones** |  | **Nota** |
| --- | --- | --- |
|  |

**3. Pregunta 3**

Responda en esta sección

| **Observaciones** |  | **Nota** |
| --- | --- | --- |
|  |

**4. Pregunta 4**

Responda en esta sección

| **Observaciones** |  | **Nota** |
| --- | --- | --- |
|  |