|  |
| --- |
| Firma vigilante: |

**Examen ordinario 1C**

|  |
| --- |
| **Nombre y apellidos: JAVIER RODRÍGUEZ RODRÍGUEZ** |
| **NP: 130864** |
| **Firma:** |

**Sede:**………………………………………………………………………………………………**Modelo de Examen:** B

**Parte teórica**

1. Responda razonadamente a las siguientes preguntas:
2. **Indique cuatro diferencias entre procesos e hilos. (1 pto)**

- Contexto: Un proceso es una entidad independiente en el sistema operativo que tiene su propio espacio de direcciones y recursos, mientras que un hilo es una entidad dentro de un proceso compartiendo los mismos recursos.

- Creación: La creación de un proceso es un proceso costoso en términos de tiempo y recursos, mientras que la creación de un hilo es más ligera y rápida.

- Planificación: Los procesos son programados y ejecutados de forma independiente por el sistema operativo, mientras que los hilos son programados y ejecutados dentro de un proceso compartiendo los recursos del CPU.

- Comunicación: La comunicación entre procesos requiere mecanismos explícitos como señales, colas de mensajes, memoria compartida, etc., mientras que los hilos dentro de un proceso pueden compartir recursos y comunicarse fácilmente entre sí.

1. **Indique a qué se refiere la propiedad de viveza (0,5 pto)**

La propiedad de viveza ("liveliness" en inglés) en programación se refiere a la capacidad de un programa o componente de responder de manera rápida y eficiente a eventos o solicitudes. Se espera que un programa con una alta propiedad de viveza tenga un rendimiento óptimo y sea capaz de manejar grandes cantidades de trabajo en un tiempo limitado, sin causar retrasos o bloqueos. Esta propiedad es importante en muchas aplicaciones, especialmente en aquellas que requieren una respuesta en tiempo real, como los juegos, la animación y los sistemas de control de tiempo real.

1. **Indique a qué se refiere la propiedad de seguridad (0,5 pto)**

La propiedad de seguridad en programación se refiere a la capacidad de un programa o sistema de proteger los datos y recursos contra accesos no autorizados, manipulaciones y otros tipos de ataques. La seguridad se logra mediante la implementación de medidas técnicas, como autenticación, encriptación, control de acceso y gestión de permisos. La propiedad de seguridad es crucial en aplicaciones que manejan información confidencial, financiera o de salud, y en sistemas críticos para la infraestructura de una organización o de la sociedad en general.

1. **¿A qué se llama Sección Crítica? Ponga un ejemplo. (1 pto)**

La sección crítica en un programa es una porción de código que accede a recursos compartidos y requiere protección contra la interferencia o acceso concurrente por parte de otras partes del programa.

Un ejemplo de sección crítica sería una sección de código que incrementa una variable compartida entre múltiples hilos. Sin protección adecuada, varios hilos podrían intentar incrementar la variable al mismo tiempo, lo que podría causar resultados incorrectos. Para solucionar este problema, se puede usar un mecanismo de sincronización, como un bloqueo o un semáforo, para garantizar que sólo un hilo acceda a la sección crítica en cualquier momento.

1. **¿Cuándo se dice que un proceso sufre inanición (starvation)? (0,5 ptos)**

Se dice que un proceso sufre inanición (Starvation) cuando es bloqueado indefinidamente y no recibe la oportunidad de ejecutarse, lo que resulta en una falta de recursos y una falta de progreso. La inanición puede ocurrir cuando un proceso es bloqueado esperando un recurso que nunca se libera o cuando un proceso es retenido por otro proceso de forma permanente.

En el sistema operativo, el manejador de procesos utiliza un algoritmo de planificación para determinar qué procesos deben ejecutarse en cualquier momento. Si un proceso está bloqueado y no recibe la oportunidad de ejecutarse, puede sufrir inanición. Esto puede ser un problema en sistemas que requieren tiempo real o en los que un proceso debe garantizar un tiempo mínimo de ejecución para asegurar la correcta terminación.

1. **¿Cuándo se dice que un sistema se encuentra en interbloqueo (deadlock)? (0,5 ptos)**

Se dice que un sistema se encuentra en interbloqueo (deadlock) cuando dos o más procesos están bloqueados mutuamente, esperando el acceso a recursos que están siendo utilizados por el otro proceso. Como resultado, ninguno de los procesos puede continuar y ninguno de ellos puede liberar los recursos que el otro proceso necesita.

Un ejemplo de interbloqueo sería si dos procesos A y B solicitan el uso de dos recursos R1 y R2 en diferente orden. Si el proceso A solicita R1 primero y luego R2, y el proceso B solicita R2 primero y luego R1, ambos procesos quedarán bloqueados, esperando el acceso al recurso que el otro proceso está usando.

El interbloqueo puede ser un problema serio en sistemas operativos, ya que puede resultar en una falla total del sistema y una imposibilidad de recuperarse sin un reinicio manual. Por lo tanto, es importante diseñar y monitorear los sistemas para evitar interbloqueos y asegurar la continuidad del servicio.

**Parte Práctica**

**Ejercicio 1 (2ptos)**

Se desea crear un programa que simule un juego de dados. Para ello se utilizan N dados que se lanzan simultáneamente. Se desea crear un proceso para cada uno de los dados. Escriba un proceso que simule el lanzamiento de los dados y que devuelva un array con los resultados. La cabecera del método debe ser la siguiente:

public int[] tiradaDados(int numDados){

// Crea numDados procesos que lo que hacen es simular tirar un dado.

// Recoge los valores de todos los dados

// devuelve el array con los valores obtenidos.

}

**Ejercicio 2 (2ptos):**

Se desea escribir un programa que simule el proceso de matriculación en la universidad. Para ello se tiene el siguiente conjunto de actores:

![Forma

Descripción generada automáticamente con confianza baja]() Los alumnos llegan a la universidad con una matrícula formada. Cuando se van a matricular la universidad les asigna un número personal.

![Forma

Descripción generada automáticamente con confianza baja]() En la universidad existen varios puestos de matrícula. Los alumnos acceden a ellos utilizando una cola única. Cuando un puesto queda libre, pasa el siguiente alumno. Si cuando llega un alumno todos los puestos están ocupados, se queda esperando.

![Forma

Descripción generada automáticamente con confianza baja]() El sistema de la universidad, cuando se matricula un alumno asigna números de inscripción únicos de forma correlativa.

![Forma

Descripción generada automáticamente con confianza baja]() En el ejercicio, fíjese solamente en el mecanismo de asignación de los números de inscripción.

Escriba un programa en Java que realice la simulación del sistema solicitado utilizando semáforos como mecanismo de sincronización.

**Ejercicio 3 (2ptos):**

Resuelva el problema anterior utilizando monitores como mecanismo de sincronización.