

Simulación Primer parcial de Sistemas Operativos

xx de setiembre de 2019

Lea detenidamente las siguientes instrucciones. No cumplir los requerimientos puede implicar la pérdida del parcial.

Formato

- Indique su nombre completo y número de cédula en cada hoja (no se corregirán las hojas sin nombre). Numere todas las hojas e indique la cantidad total de hojas en la primera.
- Escriba las hojas de un solo lado y empiece cada problema en una hoja nueva.
- Si se entregan varias versiones de un problema solo se corregirá la primera de ellas.

Dudas

- Sólo se contestarán dudas de letra.
- No se aceptarán dudas en los últimos 15 minutos del parcial.

Material

- El parcial es **SIN** material (no puede utilizarse ningún apunte, libro ni calculadora). Sólo puede tenerse las hojas del parcial, lápiz, goma y lapicera en su banco. Todas sus demás pertenencias debe colocarlas en el piso debajo de su asiento.

Finalización

- El parcial dura **3 horas**.
- Al momento de finalizar el parcial no se podrá escribir absolutamente nada en las hojas, debiéndose parar e ir a la fila de entrega. Identificar cada una de las hojas con nombre, cédula y numeración forma parte de la duración del parcial.

Problema 1 (6 pts)

- Describa brevemente el proceso de invocar un llamado al sistema (system call).
- Mencione ventajas y desventajas entre los sistemas simétricos y asimétricos.
- ¿Cuál es la principal ventaja de la multiprogramación? ¿Que problemas nuevos le acarrea el sistema operativo?
- Describa los 2 comportamientos de la primitiva signal en monitores. Cuando se programa utilizando monitores, ¿cómo se puede lograr que el programa funcione correctamente independientemente del comportamiento del signal? ¿En qué se diferencia la operación `signal()` asociada a monitores de la operación `V()` asociada a semáforos?

Problema 2 (9 pts)

- Implemente con `cobegin` y `coend` la solución de la expresión 1 con el máximo nivel de concurrencia posible.

$$(2 \times a \times c^2 + 34/2 - 5 \times b) + (6/a - 4 \times 2) \quad (1)$$

- Implemente con semáforos la solución al problema del productor-consumidor con buffer finito de tamaño N pero considerando que el sistema tiene la capacidad de devolver un valor útil (al consumidor) cuando el buffer está vacío. Se dispone de los siguientes funciones/procedimientos:

`producir()` : Función, produce y devuelve un producto.

`consumir(p)` : Procedimiento, consume el producto p.

`insertar(p)` : Procedimiento, inserta el producto p en el buffer.

`obtener()` : Función, devuelve el primer producto del buffer. En caso que el buffer esté vacío igual devuelve un valor válido que debe ser consumido.

`vacío()` : Función, devuelve `true` si el buffer está vacío, `false` en caso contrario.

Problema 3 (25 pts)

Se desea modelar usando ADA una fábrica de automóviles. La misma cuenta con una línea de producción con 25 estaciones de trabajo. En cada estación pueden trabajar hasta 3 empleados simultáneamente.

Los empleados trabajan en turnos de 60 minutos, luego descansan 10 minutos y vuelven a trabajar. Pueden trabajar en cualquier estación pero deberán darle prioridad a las estaciones con menos empleados cuando terminan su descanso. Si alguna de las estaciones está mas de 5 minutos sin al menos una persona se debe activar una alarma.

Hay un supervisor que verifica el estado de la diferentes estaciones de trabajo. Debe tener acceso exclusivo sobre la estación para poder supervisar y tiene prioridad sobre los empleados. El supervisor cuenta para no hacer sonar la alarma.

Se desean modelar en ADA las estaciones de trabajo, el supervisor y los empleados. Se pueden usar tareas auxiliares.

Se dispone de las siguientes funciones:

1. `trabajar()` ejecutada por el empleado para trabajar 60 minutos.
2. `descanso()` ejecutada por el empleado para descansar 10 minutos.
3. `alarma(estación: integer)` ejecutada por la estación de trabajo cuando no hay empleados por más de 5 minutos.
4. `elegir_estacion():integer` ejecutada por el supervisor para elegir la estación a supervisar.
5. `supervisar()` ejecutada por el supervisor para hacer su trabajo.