

Cuestionario guía - Clases Teóricas 3 y 4

- 1- ¿Por qué las propiedades de vida dependen de la política de scheduling? ¿Cómo aplicaría el concepto de fairness al acceso a una base de datos compartida por n procesos concurrentes?
- 2- Dado el siguiente programa concurrente, indique cuál es la respuesta correcta (justifique claramente)


```
int a = 1, b = 0;
co <await (b = 1) a = 0 > // while (a = 1) { b = 1; b = 0; } oc
```

 - a) Siempre termina
 - b) Nunca termina
 - c) Puede terminar o no
- 3- ¿Qué propiedades que deben garantizarse en la administración de una sección crítica en procesos concurrentes? ¿Cuáles de ellas son propiedades de seguridad y cuáles de vida? En el caso de las propiedades de seguridad, ¿cuál es en cada caso el estado “malo” que se debe evitar?
- 4- Resuelva el problema de acceso a sección crítica para N procesos usando un proceso coordinador. En este caso, cuando un proceso $SC[i]$ quiere entrar a su sección crítica le avisa al coordinador, y espera a que éste le otorgue permiso. Al terminar de ejecutar su sección crítica, el proceso $SC[i]$ le avisa al coordinador. Desarrolle una solución **de grano fino** usando únicamente variables compartidas (ni semáforos ni monitores).
- 5- ¿Qué mejoras introducen los algoritmos Tie-breaker, Ticket o Bakery en relación a las soluciones de tipo spin-locks?
- 6- Modifique el algoritmo Ticket para el caso en que no se dispone de una instrucción Fetch and Add
- 7- Analice las soluciones para las barreras de sincronización desde el punto de vista de la complejidad de la programación y de la performance.
- 8- (OPCIONAL). Implemente una butterfly barrier para 8 procesos usando variables compartidas.
- 9-
 - a) Explique la semántica de un semáforo.
 - b) Indique los posibles valores finales de x en el siguiente programa (**justifique claramente su respuesta**):


```
int x = 4; sem s1 = 1, s2 = 0;
co P(s1); x = x * x; V(s1);
  // P(s2); P(s1); x = x * 3; V(s1);
  // P(s1); x = x - 2; V(s2); V(s1);
oc
```
- 10- Desarrolle utilizando semáforos una solución centralizada al problema de los filósofos, con un administrador único de los tenedores, y posiciones libres para los filósofos (es decir, cada filósofo puede comer en cualquier posición siempre que tenga los dos tenedores correspondientes).
- 11- Describa la técnica de *Passing the Baton*. ¿Cuál es su utilidad en la resolución de problemas mediante semáforos?
- 12- Modifique las soluciones de Lectores-Escritores con semáforos de modo de no permitir más de 10 lectores simultáneos en la BD y además que no se admita el ingreso a más lectores cuando hay escritores esperando.
- 13- (OPCIONAL) Broadcast atómico. Suponga que un proceso productor y n procesos consumidores comparten un buffer unitario. El productor deposita mensajes en el buffer y los consumidores los retiran. Cada mensaje depositado por el productor tiene que ser retirado por los n consumidores antes de que el productor pueda depositar otro mensaje en el buffer.
 - a) Desarrolle una solución usando semáforos
 - b) Suponga que el buffer tiene b slots. El productor puede depositar mensajes sólo en slots vacíos y cada mensaje tiene que ser recibido por los n consumidores antes de que el slot pueda ser reusado. Además, cada consumidor debe recibir los mensajes en el orden en que fueron depositados (note que los distintos consumidores pueden recibir los mensajes en distintos momentos siempre que los reciban en orden). Extienda la respuesta dada en (a) para resolver este problema más general.
- 14- (OPCIONAL) Implemente una butterfly barrier de n procesos usando semáforos (siendo n potencia de 2)