#### UTN FRD

### Ingeniería en Sistemas de Información

Dpto. Ingeniería en Sistemas de Información Area: Computación

## Revisión – Unidad III – Sincronización entre Procesos

1. ¿Podría aplicarse la exclusión mutua sobre una sección crítica?

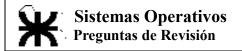
2. Unir con flechas:

Inanición mutual exclusion

Condición de Carrera livelock Exclusión Mutua deadlock Interbloqueo race condition Circulo Vicioso startvation

- 3. ¿Puedo coordinar n procesos teniendo en cuenta las velocidades de ejecución de cada proceso?
- 4. ¿Qué relación Ud. Encuentra entre la dificultad de encontrar errores y la condición de carrera?
- 5. ¿Qué tipo de problemas puede provocar la implementación de la exclusión mutua?
- 6. ¿Qué sucede si un proceso queda indefinidamente dentro de una sección crítica?
- 7. Responda V o F: La espera activa es una buena solución de software, puesto que no consume tiempo de CPU.
- 8. Responda V o F: El algoritmo de Dekker es mejor que el algoritmo de Peterson, porque tiene menor carga de procesamiento.
- 9. ¿La deshabilitación de interrupciones es una técnica de hardware que da soporte a la exclusión mutua en un SMP?
- 10. Responda V o F: Deshabilitar interrupciones en un SMP implica evitar el entrelazado de procesos.
- 11. Responda V o F: Deshabilitar interrupciones en un computador con una sola CPU implica evitar el entrelazado de procesos.
- 12. ¿Las soluciones de hardware son mejores que las soluciones de software, en cuanto al problema de la exclusión mutua?
- 13. ¿Las instrucciones de CPU atómicas surgen como una mejora a la deshabilitación de interrupciones?
- 14. Responda V o F: Tanto las soluciones de software como las instrucciones atómicas de hardware evitan la espera activa.
- 15. Responda V o F: Las instrucciones de CPU atómicas permiten implementar algoritmos de exclusión mutua de forma más eficiente que las soluciones de software.
- 16. ¿Por qué razón Ud. Cree que el SO es el que puede dar el mejor soporte a la exclusión mutua?
- 17. Responda V o F: Según Dijkstra, la exclusión mutua no puede resolverse usando semáforos.
- 18. Responda V o F: semSignal(s) es una instrucción atómica, no interrumpible, que envía una señal al semáforo s.
- 19. Responda V o F: semWait(s) incrementa s.
- 20. Responda V o F: semSignal(s) decrementa s.
- 21. Responda V o F: Una de las ventajas de los semáforos es que su valor se puede cambiar sin necesidad de usar semWait() o semSignal().
- 22. Responda V o F: Un semáforo binario es más rápido y potente que un semáforo no binario.
- 23. ¿Qué es un proceso productor?
- 24. ¿Qué es un proceso consumidor?
- 25. ¿Por qué razón se desarrollaron los monitores?
- 26. ¿Qué relación hay entre los monitores y la OOP?
- 27. ¿Qué es una variable de condición?
- 28. En cuanto a cola de mensajes, ¿Qué tipo de primitivas receive() existen?

#### UTN FRD



### Ingeniería en Sistemas de Información

Dpto. Ingeniería en Sistemas de Información Area: Computación

# En cuanto a los trabajos prácticos:

- 1. ¿Es posible aplicarle cierto time-out a una operación bloqueante read()?
- 2. ¿Las señales se lanzan como eventos sincrónicos o asincrónicos?
- 3. ¿Qué sucede si un programa entra en loop y comienza a lanzar en forma repetitiva una misma señal a otro proceso?
- 4. ¿Una señal de interrupción tiene mayor prioridad que una señal de usuario?
- 5. ¿Qué cuidados debe tener el programador que utiliza la función pause()?
- 6. ¿La llamada a la función sleep() implica una espera activa del proceso que la invocó?
- 7. En un contexto de multithreading, ¿Tiene sentido escribir sentencias luego de invocar a la función pthread exit()?
- 8. ¿Qué sucede cuando un thread invoca a la función pthread join()?
- 9. ¿Qué tipo de atributo debe tener un thread sobre el cual se ejecuta una operación de join?
- 10. ¿Qué diferencia hay entre un semáforo y una variable de condición?
- 11. ¿Para qué sirve el comando "time"? ¿Cómo se usa?
- 12. Supongamos que tengo la sección crítica C y hay dos funciones f1() y f2() que acceden a dicha sección crítica. La función f1() tarda unos 200 milisegundos, f2() tarda unos 500 milisegundos; entonces ejecuto f1() y luego f2(); al comienzo de f2() pongo una demora de unos 300 milisegundos. Ambas funciones se lanzan con pthread create(). Responda:
- a) Ud. es el Analista a cargo del desarrollo de la aplicación X y un programador vino con este planteo para un proceso que Ud. le encargó ¿Aprueba Ud. este diseño?
  - b) ¿Funciona el diseño planteado en el punto 12?
  - c) ¿Qué otro diseño podría aplicarse al proceso del punto 12?
  - 13. ¿Por qué razón Ud. cree que las prácticas proponen devolver valores distintos en la función main() para las distintas situaciones de error detectadas?
  - 14. ¿Qué es una señal nula? ¿Para qué sirve?
  - 15. ¿Qué significa que una función no es re-entrante?
  - 16. ¿Una variable de tipo volatile implica que su valor vuelve a su valor por defecto luego de cierto tiempo o bajo determinada condición?
  - 17. ¿Es posible cambiar la imagen del proceso actual por la de otro proceso?
  - 18. ¿Qué significa que un programa no se comporte acorde con un "Modelo Determinista"?
  - 19. ¿Qué es un coproceso?
  - 20. Tengo un programa interactivo (requiere que el operador tipee datos por teclado y obtenga salidas por pantalla), ¿Puedo -a través de otro proceso- reemplazar al operador (obviamente suponiendo un comportamiento determinístico)?
  - 21. Desde el punto del vista del SO, ¿Cómo se identifica cada recurso de tipo IPC y Cómo podemos consultarlo y/o administrarlo?
  - 22. De la misma forma que un SO tiene su API (system calls), cualquier aplicación que desarrollemos también podría proveer (hacia el exterior) su propia API, ¿En qué archivo se especificaría esta API?
  - 23. ¿Qué otra razón -además de la indicada en el TP VII Cola de Mensajes- se le ocurre a Ud. que un Analista puede tener para definir una API implementada en la librería X para ser reusada en las aplicaciones A,B,C,D?
  - 24. Responda Verdadero o Falso y justifique: "La memoria compartida es la forma más lenta de IPC".
  - 25. Enumere algunos ejemplos de funciones bloqueantes vinculadas con IPC.
  - 26. Responda Verdadero o Falso y justifique: "El programa cliente que se conecta a un area de memoria compartida debe especificar (en la conexión) el tamaño de la misma".
  - 27. Responda Verdadero o Falso y justifique: "Una de las principales ventajas de la memoria

#### UTN FRD

## Ingeniería en Sistemas de Información

Dpto. Ingeniería en Sistemas de Información Area: Computación

compartida es que nos libera de la preocupación de la compartición de datos, puesto que implementa automáticamente mecanismos de exclusión mutua".

28. ¿Qué es lo que hace esta sentencia C: for (i=0; i<10; i++, p++) \*p \*= 1.1; ?

# En cuanto a Concurrencia, Interbloqueo e Inanición:

- 1. Supongamos que dos procesos (A,B) que se ejecutan concurrentemente, requieren los recursos R1 y R2. Respnda Verdadero o Falso:
  - a. Siempre -sin excepción- se producirá el interbloqueo
  - **b.** Es factible que exista una o más trayectorias (en la ejecución concurrente de A y B) que eviten el interbloqueo
  - c. El interbloqueo se evita cuando los procesos entran en la "región fatal"
  - **d.** Evitar el interbloqueo implica encontrar "un camino seguro"
  - e. "Un camino seguro" es un camino que no pasa por una "región fatal"
- 2. Responda Verdadero o Falso: El interbloqueo siempre ocurre por competencia de recursos.
- 3. ¿Cuáles son las condiciones necesarias (aunque no suficientes) para la ocurrencia del interbloqueo?
- 4. ¿Para qué sirve un "Grafo de Asignación de Recursos"?
- 5. ¿Qué forma tiene un "Grafo de Asignación de Recursos" cuando estamos en presencia de una "espera circular"?
- 6. ¿Cuáles son las estrategias utilizadas para el tratamiento del problema del interbloqueo? ¿Qué tratan de lograr estas estrategias?
- 7. Teniendo en cuenta el grado de concurrencia entre los procesos que permiten las estrategias para el tratamiento del interbloqueo ¿Cuál es la estrategia más conservadora, la menos conservadora y la más moderada?
- 8. Supongamos que no puedo determinar -por anticipado- los recursos que requerirá cada proceso, en este caso, ¿Puedo utilizar la estrategia de "Predicción del Interbloqueo"?
- 9. ¿Cómo funciona el "Algoritmo del Banquero" y en qué tipo de estrategia se usa?
- 10. ¿Cuál es la estrategia que requiere de un algoritmo que se ejecute periódicamente para detectar la "condición de espera circular"?