



Calificación
1
2

SISTEMAS OPERATIVOS

Examen Final

10 de Junio de 2010

- PRIMERA PARTE -

Nombre	Titulación
SOLUCIONES	

Dispone de una hora y media para realizar el examen

1 (8 puntos) Test. En cada uno de los siguientes apartados, señale cuál opción es correcta. En caso de que existan varias opciones ciertas, *se considerará como correcta la más completa o precisa*. Las preguntas no contestadas no puntúan; las contestadas erróneamente puntúan negativamente restando un tercio de su valor. Señale sus respuestas rodeando con un círculo la opción correcta. Si se equivoca, tache la respuesta incorrecta y rodee la opción que considere correcta. Escriba con tinta. Las preguntas respondidas con lápiz o con varios círculos no tachados se considerarán no contestadas.

DEBE ADJUNTARSE JUSTIFICACIÓN DE TODAS LAS RESPUESTAS DEL TEST. Solo se considerarán correctas aquellas respuestas marcadas con la opción correcta y con la justificación correcta.

- Se tienen 3 procesos: P1, P2 y P3, con tiempos de ejecución: 85, 45 y 118 ms, respectivamente. Si actúa el planificador a largo plazo según el algoritmo SJF (*Shorttest Job First*) se obtiene que:
 - Los procesos se encuentran en la lista de preparados en el orden de llegada: P1, P2 y P3
 - Los procesos se encuentran en la lista de preparados en el orden: P2, P1 y P3**
 - Los procesos se ejecutan en el orden de llegada: P2, P1 y P3
 - Los procesos se ejecutan según la prioridad que posean los procesos
- Sean dos procesos: P1 con tiempo de ejecución de 20ms y P2 con 15ms. El planificador a corto plazo actúa según un *Round Robin* con *quanto* de 10ms y tiempo de cambio de contexto de 5ms. El tiempo de retorno de P1 es
 - 30ms
 - 40ms**
 - 45ms
 - 50ms
- El tiempo de retorno, para procesos que no quedan bloqueados en ningún momento, corresponde matemáticamente
 - Al porcentaje del tiempo medio de utilización
 - Al tiempo de espera menos el tiempo de ejecución
 - Al tiempo de ejecución más el tiempo de espera**
 - A una medida del número de procesos completados por unidad de tiempo
- La independencia de dispositivo implica:
 - lograr que el tiempo de lectura/escritura sea el mismo independientemente del dispositivo utilizado
 - El sistema operativo está liberado de realizar la gestión de la E/S
 - Utilizar todos los dispositivos de E/S con una visión uniforme**
 - Ninguna de las anteriores

5. Entre las distintas formas de conectarse los procesadores para formar un sistema multiprocesador se encuentra:
- El Sistema maestro/esclavo
 - El bus compartido**
 - El DMA
 - Ninguna de las anteriores
6. ¿Cuál de las siguientes afirmaciones es correcta?
- El sistema operativo ejecuta una serie de servicios que ofrece a los programas a través del intérprete de órdenes
 - Las órdenes del Shell son llamadas al sistema
 - El intérprete de órdenes se ejecuta en modo núcleo, puesto que se comunica directamente con el Hardware
 - Ninguna de las afirmaciones anteriores es correcta**
7. ¿Cuál de las siguientes afirmaciones acerca de los sistemas distribuidos es incorrecta?
- Son sistemas débilmente acoplados
 - Comparten una señal de reloj**
 - Pueden estar formados por sistemas heterogéneos
 - Usan una red de comunicación
8. De las siguientes afirmaciones acerca del núcleo de los sistemas operativos, señala la que es incorrecta
- Reside permanentemente en memoria
 - Se ejecuta en modo supervisor
 - Contiene el código de todos los programas del sistema**
 - Todas las opciones son correctas
9. Se tiene la siguiente carga de procesos: el proceso 0 llega en el instante 0 y tiene una duración de 6 unidades de tiempo (u.t.), el proceso 1 llega en el instante 1 y tiene una duración de 3 u.t., el proceso 2 llega en el instante 3 y tiene una duración de 2 u.t. Y el proceso 3 llega en el instante 5 y tiene una duración de 4 u.t.. Todos los procesos tienen la misma prioridad. ¿Cuál de los siguientes algoritmos de planificación produce el menor tiempo medio de espera?
- SJF
 - SRTF**
 - Round robin con cuanto de tiempo de 2 u.t.
 - Round robin con cuanto de tiempo de 3 u.t.
10. Con cuál de los siguientes algoritmos de planificación no se produce inanición:
- SJF
 - SRTF
 - Round robin**
 - Ninguno de los anteriores produce inanición

2 Contestar brevemente a las siguientes cuestiones:

1. Escribe un algoritmo de gestión de la sección crítica que cumpla la propiedad de exclusión mutua, pero no cumpla la propiedad de progreso.

Hay una infinidad de posibles respuestas a esta pregunta. Quizás el algoritmo más sencillo que cumple exclusión mutua pero no progreso sería este:

```
while (true) { /* no hacer nada */ }
```

... sección crítica ...

... sección no crítica ...

O sea, simplemente impedir entrar a todos los procesos. Nunca hay más de un proceso en sección crítica, sencillamente porque nunca hay procesos. Evidentemente, este algoritmo no cumple la propiedad de progreso.

2. Los semáforos binarios, ¿son más potentes o menos potentes que los semáforos generales?

Si entendemos por «potencia» la capacidad de resolver problemas de sincronización, unos y otros tienen la misma potencia. Esto se puede demostrar indirectamente, si conseguimos lo siguiente:

construir un semáforo general a partir de semáforos binarios (lo cual demuestra que los binarios son al menos tan potentes como los generales)

construir un semáforo binario a partir de semáforos generales (que demuestra que los generales son al menos más potentes que los binarios).

Lo primero es perfectamente viable (el algoritmo se puede consultar en cualquier texto sobre concurrencia). Lo segundo es trivial, ya que los semáforos binarios son versiones restringidas de los semáforos generales. Con ello se demuestra la equivalencia de ambas herramientas: todo lo que podemos hacer con un semáforo general lo podemos hacer con semáforos binarios.

3. Cuando un proceso ejecuta una operación V sobre un semáforo y hay varios procesos esperando por ese semáforo, ¿cuál o cuáles de los procesos en espera se desbloquea? ¿Uno al azar? ¿El más antiguo? ¿El de más prioridad? ¿Todos los procesos?

Se desbloqueará alguno de los procesos en espera. La especificación de los semáforos no prescribe cuál debe ser el criterio de elección del proceso desbloqueado. Lo único que debe cumplirse es que debe ser un solo proceso.

4. A continuación se presenta un algoritmo para gestionar el problema del búfer finito o búfer limitado, que utiliza semáforos como herramienta de sincronización. Identifique y corrija los errores que tiene este algoritmo.

<p>Variables globales mutex : Semáforo = 0; huecos : Semáforo = N; // N es el tamaño del búfer llenos : Semáforo = 0;</p>	
<p>Productor (código que se ejecuta cada vez que produce un elemento) P(mutex) P(huecos) ... inserta el elemento en el búfer ... V(llenos) V(mutex)</p>	<p>Consumidor (código que se ejecuta cada vez que se consume un elemento) P(mutex) P(llenos) ... extrae un elemento del búfer ... V(huecos) V(mutex)</p>

Un primer error es que el semáforo «mutex» está inicializado a cero. Este semáforo se está intentando usar para manipular el búfer de forma exclusiva: para ello debería haberse inicializado a uno. Tal y como está el código, ningún proceso podrá producir ni consumir.

Aun si corregimos el valor inicial de «mutex», la solución tiene riesgo de interbloqueo. Si un proceso consumidor intenta trabajar con el búfer vacío, se encuentra con que el semáforo «llenos» está a cero y se bloquea, lo cual es correcto. Pero como el consumidor previamente se ha apropiado de «mutex», impide a cualquier otro proceso avanzar, así que nadie podrá desbloquear a este consumidor: el sistema queda en un estado de bloqueo total.

Lo mismo ocurriría con un productor que se encontrara con el búfer lleno (semáforo «huecos» a cero). En ambos casos, lo que habría que hacer es trasladar las operaciones «P(mutex)» a continuación de las operaciones P con los semáforos «huecos» y «llenos».