UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela Universitaria de Informática
Facultad de Informática

Calificación
1
2
3

SISTEMAS OPERATIVOS,10 de diciembre de 2007
Examen Convocatoria Extraordinaria Especial

Nombre	Titulación

Dispone de tres horas para completar el examen

---

1 (5 puntos) Test. Marque las opciones correctas de cada apartado. En caso de que existan varias opciones ciertas, se considerará como correcta la más completa o precisa. Las preguntas no contestadas no puntúan; las mal contestadas puntúan negativamente restando un tercio de su valor. Marque la opción correcta rodeándola con un círculo. Si se equivoca, tache la respuesta equivocada y rodee la opción correcta. Escriba con tinta. Las preguntas respondidas con lápiz o con varios círculos no tachados se considerarán no contestadas.

1. Suponga un sistema de archivos tipo FAT en el que cada cluster posee 2048 bytes y cada entrada de directorio posee un tamaño de 32 bytes, si el directorio raíz posee 10 archivos y el directorio "/Mis_Aplicaciones" posee 23 archivos. Si la FAT ya está cargada totalmente en memoria ¿Cuántos accesos a disco se requerirían en el peor de los casos para abrir un archivo ubicado en "/Mis_Aplicaciones"?
    a. Un acceso
    b. Dos accesos
    c. Veintitrés accesos
    d. Treinta y tres accesos

2. Cuando realizamos una operación ***V*** sobre un semáforo:
    a. Se despertará siempre al primer proceso que realizó una operación P.
    b. En caso de existir algún proceso bloqueado, se despertará a uno de ellos según la política definida en la implementación del semáforo.
    c. Ningún otro proceso puede estar en mitad de la ejecución de otra operación V sobre dicho semáforo, aunque si es posible que exista otro proceso en mitad de la ejecución de una operación P sobre el mismo semáforo.
    d. Todas las anteriores son ciertas.

3. Supongamos que para resolver cierto problema de concurrencia empleamos un monitor y variables condición. En este caso:
    a. Una operación wait sobre una de las variables condición empleadas en dicho monitor provocará que el proceso que la realice se bloquee incondicionalmente.
    b. Una operación signal sobre una de las variables condición empleadas en dicho monitor provocará el desbloqueo de otro proceso que haya realizado una operación wait sobre cualquiera de las variables condición del monitor.
    c. Una operación signal nunca provoca el bloqueo del proceso que la realiza.
    d. Todas las anteriores son falsas.

4. ¿Cuál es la duración del cuanto de tiempo (*timeslice*) más apropiado para un sistema operativo que utiliza Round-Robin para garantizar interactividad? Como dato adicional, se sabe que en el peor caso habrá 20 procesos en la cola de preparados.
    a. un segundo
    b. una décima de segundo
    c. un milisegundo
    d. un microsegundo

5. Supongamos una política de planificación que actúa de la siguiente forma: cada vez que la CPU queda libre, se escoge a un proceso al azar de la cola de preparados. ¿Qué podemos afirmar sobre esta política?
    a. el tiempo medio de respuesta de esta política es similar al de un Round-Robin
    b. esta política tiene cierto riesgo de inanición
    c. esta política perjudica a los procesos con ráfagas de CPU muy largas
    d. esta política es expulsiva

6. Suponga un sistema con memoria virtual en el que el área de intercambio se soporta mediante un archivo de intercambio. Si el sistema permite establecer la política de asignación de espacio en disco de forma individual cada vez que un archivo se crea ¿qué política utilizaría para el archivo de intercambio?
    a. Asignación contigua
    b. Asignación encadenada
    c. Asignación indexada
    d. Asignación tipo FAT

7. Considere un sistema de archivos tipo UNIX nativo en el que el tamaño de bloque de sistema de archivo es de 1024 bytes; además dicho tamaño coincide con el tamaño de bloque físico. Para un archivo con registros de 256 bytes y que posee 2500 registros, acceder al registro 47 requeriría en el peor de los casos de:
    a. Un acceso
    b. Dos accesos
    c. Veintitrés accesos
    d. Cuarenta y seis accesos

8. El modo dual de operación:
    a. Permite ejecutar dos procesos concurrentemente
    b. Proporciona un mecanismo de protección de software
    c. Permite al usuario ejecutar una instrucción privilegiada mediante una llamada al sistema
    d. Ninguna de las anteriores

9. La tabla de páginas indica que la página 2 tiene asociado el marco número 5. ¿Cuál es la dirección física para la dirección virtual (5,215) dada en el formato (nº Pág., desplazamiento en la Pág):
    a. 5+215
    b. 5*1024+215
    c. Con los datos del enunciado no es posible obtenerla
    d. Si el tamaño del marco es T, la dirección física es 5 x T + 215

10. En un sistema operativo multitarea, con 16Kb de espacio lógico de procesos, con páginas de 512bytes y 64 marcos de página, sin memoria virtual. La dirección lógica está formada por:
    a. 6 bits para indicar la página y 9 bits para el desplazamiento.
    b. 9 bits para indicar la página y 6 para el desplazamiento.
    c. Dado que tenemos 16Kb de espacio lógico de procesos y 64 marcos, no tiene sentido indicar que se trata de un sistema sin memoria virtual.
    d. 5 bits para indicar la página y 9 bits para el desplazamiento.

Nombre _____

11. ¿En qué nivel de planificación de procesos se encuentra el componente llamado despachador o *dispatcher*?
    a. en el planificador de corto plazo
    b. en el planificador de medio plazo
    c. en el planificador de largo plazo
    d. dependiendo del sistema operativo, puede estar en el de medio plazo o en el de largo plazo
12. Supongamos que en un momento dado llegan a un sistema N procesos, cada uno de ellos con una sola ráfaga de CPU y todos ellos con exactamente la misma duración. Cuando se planifica esta carga de trabajo, dependiendo de la política, podemos afirmar lo siguiente sobre el tiempo medio de espera:
    a. el tiempo medio de espera con SJF y RR será el mismo
    b. el tiempo medio de espera con FCFS y SJF será el mismo
    c. el tiempo medio de espera con FCFS, SJF y RR será el mismo
    d. como las duraciones son iguales, el tiempo medio de espera será el mismo, cualquiera que sea la política de planificación empleada.
13. Un sistema multiprogramado:
    a. Permite gestionar varias tareas interactivas
    b. Evita el posible monopolio de la CPU por un proceso del sistema
    c. Aumenta el rendimiento del sistema ya que los procesos se ejecutan en paralelo
    d. Ninguna de las anteriores
14. La idea básica de mecanismo de E/S por interrupciones consiste en:
    a. Disponer de un mecanismo acceso directo a memoria
    b. Eliminar el bucle de espera
    c. Buscar mecanismos de interactividad
    d. Ninguna de las anteriores
15. Los búferes de E/S son:
    a. Un dispositivo de gestión de las entradas y salidas de datos al computador
    b. El espacio de memoria principal que s reserva para el almacenamiento intermedio de datos procedentes o con destino a los periféricos
    c. E/S controlada por programa
    d. Ninguna de las anteriores
16. Un manejador de dispositivos es:
    a. Hardware que controla un dispositivo
    b. Software que controla un dispositivo
    c. Un programa de usuario
    d. Ninguna de las anteriores

---

2 (2,5 puntos) Un computador utiliza gestión de memoria virtual *segmentada*. La memoria disponible para los procesos es de 64 bytes y el máximo tamaño de segmento es 32 bytes. En este computador vamos a ejecutar un proceso cuya imagen de memoria tiene 4 segmentos. En un determinado instante, la tabla de segmentos indica la siguiente información:

Nº de Segmento	Dirección Base	Tamaño (bytes)
1	$56 = 38_{16}$	8
3	$16 = 10_{16}$	$32 = 20_{16}$

El segmento 0 tiene un tamaño de 16 bytes y el segmento 2 un tamaño de 24 bytes.

Se realiza la siguiente secuencia de referencias virtuales a memoria: 30_{16}, $0C_{16}$, $6E_{16}$, $4F_{16}$ y 27_{16}.

Teniendo en cuenta que no hay más procesos en el sistema, se pide:

a) ¿Podemos asegurar un formato de dirección virtual y de dirección física? Justificar la respuesta indicando una propuesta para cada caso. (0.5p)
b) Indicar (con la mayor claridad posible) el contenido de las tablas que se utilizan para realizar la traducción dinámica de direcciones mediante traducción directa, indicando además cuándo hay que actualizar cada una de ellas, y el contenido de las mismas antes de realizar los accesos a memoria. (0.25p)
c) Calcular las direcciones reales correspondientes. Indicar el contenido de las tablas y estructuras de datos utilizadas en el apartado mientras se ejecuta el proceso. (1p)
d) Calcular la tasa de aciertos de la secuencia de referencias citada. (0.25p)

---

3 (2,5 puntos) Escoja y resuelva uno de los dos problemas de concurrencia que se describen a continuación.

OPCION A. Implemente un monitor que resuelva el primer problema de los lectores y escritores (prioridad para los lectores, posible inanición de los escritores). La solución deberá utilizar **UNA SOLA** variable condición.

Suponga una semántica Mesa. Si lo prefiere, puede utilizar la operación Broadcast().

El monitor deberá tener esta especificación:

```
monitor BaseDatos
{
  public abreLectura ();      // cuando un lector quiere empezar a leer
  public cierraLectura ();    // cuando un lector termina de leer
  public abreEscritura ();    // cuando un escritor quiere empezar a escribir
  public cierraEscritura ();  // cuando un escritor termina de escribir
}
```

NOTA: Sólo hay que implementar las operaciones del monitor, no hay que implementar los procesos lectores y escritores.

OPCION B. Vamos a simular un conjunto de N procesos que juegan al famoso juego adolescente-alcohólico del «marcianito». En este juego, un grupo de personas se reúnen en círculo; cada una tiene un número entre 1 y N. El número 1 agita sus dos manos al lado de su cabeza al grito de «marcianito número 1 llamando a marcianito número XXX», a lo que la persona con el número XXX debe contestar con un mensaje similar «marcianito número XXX llamando a marcianito número YYY», y así hasta que alguien se despiste.

Se trata de hacer una simulación del sistema usando semáforos, en la que cada proceso *i* tendrá la siguiente estructura (*i* será un número entre 1 y N):

```
while (true) {
  marcianito ( i, random(1..N) );
}
```

La rutina *marcianito(i,j)* debe funcionar así: el proceso *i* espera a que alguien le ceda el turno; una vez que le han cedido el turno, imprime por pantalla «marcianito número i llamando a marcianito número j». Y a continuación, cede el turno al proceso *j*, que hasta ese momento habrá estado bloqueado dentro de la rutina *marcianito(j,otro_proceso)*.

Además, el marcianito número 1 es el que arranca el sistema. Es decir, que en su primera iteración, el proceso 1 no debe esperar a que nadie le ceda el turno.

NOTA: lo que hay que implementar es únicamente la rutina *marcianito(i.j)*.