Examen Final de Sistemas Operativos (90 min.) 24/07/2010

Nota: Tenga en cuenta que las unidades se encuentran expresadas en milisegundos.

Apellido y Nombre:..... Profesor:.....

A			B		C				NOTA	FIRMA ALUMNO (Sólo en caso de revisión de examen)
1	2	3	1	2	1	2		3		

A) **Explícitamente defina como VERDADERA o FALSA cada una de las siguientes afirmaciones, FUNDAMENTANDO su respuesta en no más de 3 renglones.**

1. En el filesystem FAT, se podría simular un “hard link” similar al de UNIX si se creara una entrada de directorio apuntando al mismo número de cluster que otro archivo ya existente, sin que eso ocasione otros problemas con el manejo de atributos de los archivos.

2. En un sistema con muchos procesos de tipo I/O bound ejecutando concurrentemente, un esquema de RAID 4, 5 ó 6 sería más eficiente que uno de Raid 2 ó 3.

3. Una condición de carrera (Race Condition) ocurre cuando, independientemente del orden de ejecución de dos o más procesos, el recurso compartido queda siempre en un estado inconsistente.

B) Responda las siguientes preguntas en no más de 5 renglones.

1. Un sistema operativo que fue diseñado para una sola CPU se requiere adaptarlo para que corra en múltiples procesadores. Mencione un cambio o agregado que incluiría en el software o hardware relacionados con las primitivas de sincronización (Locks, Semaphores, etc) y dé una explicación del porque de su decisión.

2. En el año 1992 se llevó a cabo lo que hoy en día se conoce como “The Tanenbaum-Torvalds Debate”, debate en el cual el reconocido investigador y profesor universitario Andrew Tanenbaum decía que Linux era obsoleto debido a que contenía un kernel monolítico, a diferencia de la entonces novedosa propuesta de “microkernel” que se estaba gestando en esa época. Sin embargo, 18 años después, todavía siguen predominando los kernels monolíticos. Explique brevemente en que se diferencia un microkernel de un kernel monolítico y cite al menos tres ventajas y una desventaja del microkernel.

C) Resuelva en forma clara y detallada los siguientes ejercicios.

1. Esta mañana en Ciudad Gótica Batman tiene un problema que le está impidiendo combatir el crimen. Acaba de comprar una nueva BatiNotebook, pero por alguna razón no está funcionando correctamente con la actualización de la nueva versión del sistema operativo, el BatiSO 2.0. Luego de investigar en varios foros, Batman descubrió que en el BatiSO cada vez que el planificador de corto plazo interviene el SO consume 2 ms; mientras que cada vez que un proceso pasa del estado Blocked al estado Ready el SO consume 1 ms. También descubrió que el kernel no deshabilita las interrupciones cuando está ejecutando y que las operaciones sobre los dispositivos de E/S se manejan con DMA.
Sabiendo que el algoritmo de planificación a corto plazo es SPN (Shortest Process Next), Batman quiere ejecutar el siguiente conjunto de procesos:

Proceso	T.Llegada	CPU	I/O	CPU
A	0	2	3	1
B	2	4	4	2
C	4	5	1	1

Como justo aparece una Batiseñal y Batman debe atenderla, finalmente le pide a Alfred que confeccione un **Diagrama de Gantt** para poder analizar el problema.

2. Continuando la problemática anterior: Alfred hace lo que le pidió Batman, pero resulta que el BatiSO 2.0 tiene algún bug, porque la BatiNotebook se tilda y no vuelve a bootear correctamente. Alfred sabe que Batman tiene guardados en el disco rígido algunos capítulos de la famosa serie “Batineras” y que si no logra recuperarlos tendrá que buscarse un nuevo trabajo con otro superhéroe. Luego de analizar el disco, encuentra que es de 40 GiB con 20 platos de 2048 pistas cada uno y 1024 sectores por pista, formateado con FAT32 bajo clusters de 1 KiB. Mediante el software “BatiBackup”, Alfred logra extraer un snapshot parcial de la FAT que se muestra a continuación (todas las demás entradas contienen 0x00000000).

20	0xFFFFFFF7
21	0x00000017
22	0x00000015
23	0xFFFFFFFF
24	0x0000001E
25	0xFFFFFFFF
26	0x00000000
27	0x00000016
28	0x00000000
29	0x00000018
30	0xFFFFFFFF
31	0xFFFFFFF7

Sabiendo que 0x00000000, 0xFFFFFFF7 y 0xFFFFFFFF significan cluster libre, dañado y final respectivamente, Alfred se propone:

a) Averiguar la cantidad total de archivos que existen, el tamaño en disco de cada uno y graficar su disposición en disco de la siguiente forma: $^{x}y_{j/k}$, con x: número de cluster en el disco, y: archivo, j: número de cluster del archivo, k: total de clusters del archivo.

Para evitar que esto vuelva a pasar, Alfred le pide un consejo a Robin mientras le sirve el desayuno. Entonces, Robin sugiere considerar, para tener mayor recuperabilidad, cambiarle el filesystem por ext3 con bloques de 1 KiB, inodos con 72 bytes de información del archivo, con 12 punteros directos y 2 indirectos, y 256 punteros por bloque. Como información administrativa tendría un superbloque de 4 KiB, un bitmap de inodos de 128 KiB y una tabla de inodos. Como Alfred nunca entendió bien de filesystems, le pide a Robin que averigue:

b) El filesystem que ocuparía más espacio en datos administrativos considerando el actual de FAT32 y el EXT3 planteado como opción. **Justifique**.

c) Para cada filesystem, la cantidad de accesos a memoria y a disco para leer el byte número 3070 de un archivo, asumiendo que se encuentran en memoria todas las estructuras administrativas necesarias.

3. Continuando (una vez más) con la problemática anterior, Robin hace algunas pruebas pero se da cuenta que teniendo tan solo 128MiB de memoria principal nunca podrá sacar alguna conclusión válida de qué filesystem utilizar. Como se le hace tarde y tenía planeada una salida con la mujer maravilla, le pide a la BatiChica que verifique si esa cantidad de memoria es suficiente para hacer sus pruebas.
La BatiChica observa que la BatiNotebook tiene un solo procesador, memoria virtual con paginación bajo demanda, un espacio direccionable máximo de 4 GiB y utiliza páginas de 128 bytes. El tiempo de acceso a memoria es de 150 nanosegundos, la tasa de fallos de página es del 5% y el tiempo medio de servicio de fallo de página es de 6 milisegundos. La política de reemplazo es local, utiliza el LRU y permite un tamaño máximo de área de trabajo por proceso de tres frames. La búsqueda de frames libres se realiza mediante una búsqueda lineal empezando por el frame cero. El kernel del BatiSO 2.0 tiene un tamaño de 134 216 448 bytes y está ubicado de forma contigua a partir del primer frame de la memoria.
La BatiChica decide entonces hacer una prueba ejecutando el proceso “BatiDogancia memTest” que tiene un tamaño de 5300 byte y que generará las siguientes direcciones de memoria principal (direcciones lógicas en decimal): 898, 100, 150, 260, 127, 400, 110, 515, 180, 420, 120, 425, 256, 175, 270.

Ella sabe que un buen diagnóstico consistirá en indicar el estado final de la tabla de páginas y la cantidad de page faults producidos. Deberá incluir en su solución el bit de validez de cada frame, como así también el instante de último acceso.

Condición de Aprobación: Para aprobar este examen deberá tener como mínimo 3 (tres) preguntas teóricas (partes A y B) y 2 (dos) ejercicios correctamente resueltos (parte C).