Universidad de Huelva

Sistemas Operativos

2º Ingeniería Técnica Informática

Convocatoria de Diciembre, curso 2005/2006

Examen de Teoría, 20 de Diciembre

NOTAS

- Apague el teléfono móvil.
- Debe razonar todas las respuestas dadas.

CUESTIÓN 1: BATLINUX (2 puntos)

Se desea modificar el algoritmo de planificación de Linux para implantarlo en un sistema donde no se disponga de procesos de tiempo compartido. En este sistema tan sólo se ejecutarán procesos de tiempo real y procesos BATCH. Para ello, cambiamos la implementación del estándar POSIX que hace LINUX. Los tres tipos de planificación que tendremos serán:

- Una FIFO de Tiempo Real, que nosotros denotaremos por SCHED_FIFO.
- Una Round Robin de Tiempo Real, que denotaremos por SCHED_RR, con quantum de 2 unidades de tiempo.
- Una SRT para los procesos BATCH, que denotaremos por SCHED_OTHER.

Las normas que rigen el comportamiento de dichas políticas son las mismas que usa LINUX.

Las prioridades de tiempo real tendrán el rango [1-32], mientras que la prioridad de la política SCHED_OTHER será 0. (Un número mayor indica mayor prioridad).

A dicho sistema llega la siguiente secuencia de procesos, y sabemos que tanto en el instante 0, como en el instante final, el proceso NULO es el que se ejecuta:

Proceso	Política	Prioridad	ti	Secuencia de ejecución
A	SCHED_OTHER	0	0	8
B	SCHED_RR	5	0	4 + 1 E/S + 1
C	SCHED_RR	5	1	2 + 2 E/S + 2
D	SCHED_OTHER	0	1	2 + 3 E/S + 2
E	SCHED_FIFO	20	3	2 + 2 E/S + 3

1. Obtenga el diagrama de ejecución de procesos.
2. Calcule los tiempos de servicio, espera e índice de servicio, junto con sus tiempos medios.
3. ¿Cuántas veces ha actuado el planificador?. ¿Y el despachador?. ¿Cuántos cambios de contexto se han producido?
4. ¿Con esta política de planificación, es posible la inanición?. Justifique la respuesta, y en caso de que sea afirmativa, proponga una solución.
5. Si en cada cambio de contexto se consume 0.1 unidades de tiempo, ¿Cuál sería el rendimiento del sistema?. Considerar el rendimiento como:

$$\text{Rendimiento} = \frac{\sum tiempo\ CPU\ \ proceso}{tiempo\ de\ ordenador}$$

NOTAS:

- Si en determinado momento, ocurren varios sucesos al mismo tiempo se considerará que en primer lugar ocurre la llegada de un proceso nuevo, en segundo lugar el final de una E/S, y como último el final de un quantum.
- El proceso nulo se está ejecutando antes de que llegue cualquier proceso y se seguirá ejecutando cuando terminen todos los procesos.
- Todas las operaciones de E/S se realizan sobre el mismo dispositivo reutilizable en serie.

CUESTIÓN 2: EL CRONOMETRO (2.5 puntos)

Un modelo de administración de memoria gestionado mediante paginación con memoria virtual tiene 32 tramas de 128 bytes. Los procesos, sin embargo, pueden llegar a tener hasta 64 páginas.

1. Dibuja el esquema de la MMU de este modelo, sabiendo que usa el algoritmo de reloj como algoritmo de reemplazo.

Si al sistema llegan dos procesos, primero A y luego B, el sistema le asigna a cada uno inicialmente 2 tramas, cargando las dos primeras páginas de cada proceso en las 2 primeras tramas libres (las tramas 0, 2, 3, 5, 6 y 8 están ya ocupadas). Se usa un Working Set con S=4 y cada proceso ejecuta las siguientes direcciones:

Proceso	***A***	***A***	***A***	***A***	***A***	***B***	***B***	***B***	***B***	***B***	***B***	***A***	***A***	***A***	***A***
@Log	002	131	004	384	005	130	258	128	000	133	259	257	385	129	256

1. ¿Cuántos no fallos, fallos y reemplazos produce la secuencia de direcciones de cada proceso?
2. ¿Cómo quedan las tablas de páginas de ambos procesos al final de la secuencia?

NOTA:

- Se considera que llega primero el proceso A y después el B y ya sen encuentran cargados cuando se inicia la secuencia de referencias. El sistema asigna tramas cuando llegan los procesos y cuando se ejecuta el Working Set de un proceso.

CUESTIÓN 3: La cadena fragmentada (1 punto)

En un sistema con 1 Mb. de memoria, el S.O. ocupa 100 Kb. y se encuentra en la parte baja de la memoria (primeras posiciones). La memoria está gestionada mediante particiones variables, con una lista única para gestionar huecos libres y ocupados.

Si se produce la siguiente secuencia de llegada de procesos y sabemos que existe una política BEST FIT de asignación de huecos:

Proceso	Tamaño	t_i	t en memoria
P1	448 Kb.	0	9
P2	256 Kb.	3	12
P3	140 Kb.	4	10
P4	60 Kb.	5	3
P5	80 Kb.	6	11
P6	20 Kb.	7	5
P7	400 Kb.	8	10

¿Qué fragmentación interna y externa existiría y cómo estaría la lista encadenada en el:

1. instante 6 (tras la llegada del proceso P5)?
2. instante 8 (tras la llegada del proceso P7)?
3. instante 10?

CUESTIÓN 4: FORJAMARI (2.5 puntos)

La empresa Forjamari S.A. desea realizar una actualización de sus sistemas de almacenamiento. Para ello ha comprado dos unidades de un disco cuya geometría es 16x128x64 (superficies, sectores/pista, cilindro), los sectores son de 512 bytes y un bloque está formado por dos sectores. Durante este proceso de actualización la primera duda que surge a la hora de decidir qué método de asignación de espacio a ficheros se va a utilizar, y así se proponen dos alternativas:

- Administración de espacio libre y asignación a ficheros mediante FAT.
- Administración de espacio libre con mapa de bits y asignación a ficheros mediante i-nodos de estructura semejante a la del sistema operativo UNIX.

Para cada elemento del sistema de ficheros se reservan 11 bytes para el nombre y otros 5 para atributos. Si cada entrada de la FAT ocupa 16 bits y cada i-nodo 29 bytes. Para ambos tipos de sistemas de ficheros:

1. ¿Cuánto ocupa cada entrada de un directorio?
2. ¿Cuántos ficheros regulares como máximo se pueden almacenar?¿qué porcentaje del disco se usa para almacenar los datos de los ficheros? ¿y para guardar la metainformación (información que se necesita para poder acceder a los datos contenidos)?
3. ¿Cómo organizaría los discos para aumentar la disponibilidad de los datos? ¿Cuál sería el tanto por ciento de redundancia?

Una vez que tenemos instalados los sistema de ficheros en las dos unidades de disco tenemos que migrar los datos.

4. Suponga que en el disco sólo están libres los bloques 121, 541, 759 y 1001, y se quiere crear el fichero *passwd* de 4000 Bytes en el directorio raíz. **Describa los pasos a llevar a cabo, así como el estado de las estructuras de control que se modifiquen en el proceso de creación**. Haga todas las suposiciones que considere. ¿Coincide el tamaño del fichero con el espacio que ocupa el fichero en el disco?
5. Detalle los pasos para crear un enlace llamado *lasclaves* que apunte al fichero creado anteriormente, así como las posibles modificaciones en las estructuras de datos originales.

Pasemos ahora a tratar dos cuestiones de seguridad. Si el fichero *passwd* contiene claves encriptadas mediante algún algoritmo de condensado o resumen, y sabemos que tanto el sistema original como el nuevo tienen un par de claves (clave pública y clave privada) diferentes:

6. ¿Qué tendríamos que hacer para copiar los datos contenidos en el fichero *passwd* original en el que creamos en el apartado 4?
7. Detalle cómo implantaría un sistema de protección basado en listas de control de acceso.

NOTAS:

- No existe bloque de arranque
- El superbloque está formado únicamente por mapas de bits y la tabla de i-nodos.