

Ejercicio 5. Sistemas Operativos, Curso 2009-2010

3-Mayo-2010, I.T. Informática de Gestión, URJC

Problema 1

Tenemos un sistema de cómputo que gestiona su memoria con memoria virtual implementada como paginación bajo demanda, de dos niveles, y usa política FIFO de reemplazo con asignación local. El sistema dispone de 16MB de memoria RAM. Las direcciones virtuales que llegan a la MMU tienen 16 bits, de los cuales los 4 primeros indexan en la tabla de primer nivel. Las páginas lógicas ocupan 1KB. En un determinado instante van a ejecutarse dos procesos independientes A y B. El sistema operativo ha asignado 2 marcos al proceso A y 3 al proceso B. Secuencialmente el proceso A leerá su página 1, escribirá en sus páginas 2, 5 y 6, y finalmente leerá la páginas 2 y 3. El proceso B presenta un patrón idéntico. Se pide responder razonadamente a las siguientes cuestiones:

- a) ¿Cuántos bits ocupa una dirección que sale de la MMU? ¿Qué tamaño tiene la memoria virtual? ¿Cuántas entradas tienen las tablas de segundo nivel?
- b) Describe el estado de las tablas de páginas de los procesos y su evolución temporal, indicando los fallos de página y los accesos a disco realizados, así como los bits de presente/ausente, referencia y modificación. Si el reemplazo fuera con el algoritmo de segunda oportunidad, ¿sería diferente la evolución? Si procede, indicar en qué.
- c) ¿Influye la planificación de CPU en la evolución de las tablas de páginas? ¿Cambiaría algo si en vez de ser dos procesos fueran dos hebras de kernel del mismo proceso?

Problema 2

Sea un sistema operativo donde los procesos son planificados siguiendo un algoritmo Round Robin con rodaja de 20 ms. Por otro lado los procesos pueden utilizar una librería de threads de usuario donde la planificación de los threads es bajo petición. Se pide dibujar el cronograma de ejecución del sistema (en el que aparezcan tanto procesos como threads) bajo la siguiente carga de trabajo:

- Proceso 1 thread de usuario 1: 10ms de CPU, 10ms de E/S, 20ms de CPU.
- Proceso 1 thread de usuario 2: 10ms de CPU, 10ms de E/S, 10ms de CPU.
- Proceso 1 thread de usuario 3: 20ms de CPU, 10ms de E/S, 20ms de CPU.
- Proceso 2: 10ms de CPU, 10ms de E/S, 10ms de CPU.
- Proceso 3: 15ms de CPU, 10ms de E/S, 5ms de CPU.

Se supondrá que la E/S de un thread de usuario no bloquea a los demás threads de su mismo proceso.

Problema 3

Sea un sistema operativo tipo UNIX con un sistema de ficheros de tipo nodo-i con 10 enlaces directos, 1 enlace de indirección simple, 1 enlace de indirección doble y 1 enlace de indirección triple. Los bloques son de 4 k y las direcciones de bloques de disco ocupan 4 octetos. Supóngase que se realiza en un momento dado la ejecución del siguiente código y sólo éste:

```
...
linea 1:  fd1 = open("/usr/pepe/practicasssoo/practica1.c");
linea 2:  fd2 = open("/usr/pepe/practicasssoo/practica2.c");
linea 3:  lseek(fd1, 1024*1024*1024);
linea 4:  a = read(fd1, buffer1, 10);
...
```

Responded razonadamente cuántos accesos a disco se realizan en cada línea de código.

Suponed que la cache de bloques del sistema de ficheros tiene espacio libre suficiente y que durante la ejecución de este código no se produce ningún fallo de cache.