SISTEMAS OPERATIVOS II – Examen de teoría
Convocatoria ordinaria de Junio – 6-7-2011

Apellidos: _____ **Nombre:**_____ **D.N.I.:**_____

Titulación: INGENIERÍA INFORMÁTICA **Grupo Teoría:**___ A (José Antonio Gómez) ___

1. [1,25] Responda a las siguientes cuestiones sobre sistemas de archivos:
   (a) ¿Cuál es el tamaño que ocupan todas las entradas de una tabla FAT32 que son necesarias para referenciar los cluster de datos, cuyo tamaño es de 16 KB, de una partición de 20 GB ocupados por la propia FAT32 y dichos cluster de datos?
   (b) Tenemos un archivo que ocupa 5 MBytes en un Sistema de Archivos con un tamaño de bloque de 2 KBytes y direcciones de 16 bits. Explique cómo se almacenaría y cuánto espacio para metadatos de localización de bloques necesitaría el sistema, para dicho archivo, suponiendo que se usa un método de asignación de espacio en disco mediante indexación a dos niveles.

2. **[1,25]** Supongamos que en un Sistema de Archivos *ext2* tenemos un archivo regular, *Datos*, que ocupa 100 MB. El tamaño de bloque lógico es de 2KB y cada bloque se direcciona usando 32 bits.
   (a) Si se realiza un enlace relativo (simbólico) y un enlace duro ("hard") a dicho archivo, ¿cuánto espacio de almacenamiento gasta el sistema para mantener la información necesaria de estos dos nuevos archivos?
   (b) ¿Cuáles y cuántos bloques índices se liberan si se borran los últimos 2MB del archivo *Datos*?

3. **[0,75]** Indique si un proceso perteneciente a la clase de tiempo compartido (SCHED_NORMAL) tipo batch de Linux puede sufrir inanición respecto de procesos interactivos, de la misma clase.

4. **[1]** Sea el siguiente fragmento de código correspondiente a *programa1.c*:

```
#include <mi_unica_biblioteca>

int var=3;

int thread(void *p) {
  var++;
  printf("El valor de var es %d\n", var);
}

main() {
  void **pila_hijo;
  int i;
  pila_hijo = (void **) malloc(16384);
  i = clone(thread, pila_hijo, CLONE_VM|CLONE_FILES|CLONE_FS|
      CLONE_THREAD|CLONE_SIGHAND, NULL);
  printf("En main() var vale: %d\n", var);
}
```

   Dibujar las estructuras de datos, junto a sus contenidos principales, que describen la memoria de usuario del programa principal y la hebra que crea en el instante en que la llamada `clone()` retorna, incluida la estructura `address_space`.

5. **[0,75]** Utilizando el código del programa1.c (del ejercicio anterior), dibujar los descriptores de proceso de las dos hebras y sus principales contenidos.

6. **[1,25]** Supongamos que una página de un proceso ha sido sacada de memoria principal al área de intercambio (*swapped out*). Si el proceso hace referencia de nuevo a esta página, ¿cómo sabe el kernel que esta página esta fuera de memoria? ¿cómo sabe dónde esta localizada?

7. **[1,25]** Los sistemas tipo Unix/Linux utilizan la misma interfaz de operaciones para acceder a archivos y dispositivos (`open`, `read`, `write`, etc.). Explicar a nivel de estructuras de datos cómo es posible que una operación, por ejemplo `write()`, acceda al dispositivo.