

**SISTEMAS OPERATIVOS II – Examen de teoría –
Convocatoria extraordinaria de Septiembre – 6-9-2011**

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

Titulación: ING. INFORMÁTICA **Grupo Teoría:** _ A (José Antonio Gómez) _____

1. [1,25] Sea un Sistema de Archivos en el que cada bloque (cluster) ocupa 8KBytes y las direcciones son de 32 bits. En él existe un archivo con un tamaño de 3GBytes. Calcule el número de accesos a disco necesarios para acceder al byte número 255.529 de dicho archivo en las situaciones que a continuación se indican:
 - (a) El sistema de archivos usa una política de asignación **enlazada pura**.
 - (b) El sistema de archivos usa una política de asignación **FAT**.
 - (c) El sistema de archivos usa un esquema de asignación combinado tipo Unix/Linux.

2. [0,75] Explique detalladamente el mecanismo que utiliza el núcleo de UNIX/Linux para garantizar la consistencia de los objetos compartidos del núcleo que residen en memoria principal. Por ejemplo, un *i-nodo*.

3. [1] Considere el caso en el cual un proceso A realiza un fork y como consecuencia se crea un proceso B (proceso hijo) ¿Puede haber algún momento durante la ejecución de A y B en el que haya páginas de datos que estén compartidas por los dos procesos y páginas de datos que no lo estén? SI/NO: _____ Justifíquelo.

4. [1] Si el kernel (o cualquier otro proceso) envía una señal a un proceso ¿puede esto afectar al estado del proceso destinatario de la señal? Justifíquelo con un ejemplo concreto.

5. [1,5] Responda a las siguientes cuestiones sobre la gestión de memoria que lleva a cabo el núcleo de UNIX descrito en clase:
 - (a) Explique como implementa la idea del conjunto de trabajo. Describa la información que utiliza el núcleo y el algoritmo.
 - (b) Cuando se produce una falta de página, ¿cómo se detecta en esta implementación y qué posibles situaciones pueden darse a la hora de hacer que dicha página vuelva a formar parte del conjunto de trabajo?

6. [0,75] ¿Qué diferencias existen en Linux entre un inodo de un archivo regular y un inodo de un archivo de dispositivo?

7. [1,25] Describa en pseudocódigo los pasos que sigue el núcleo UNIX visto en clase y las estructuras de datos que utiliza para llevar a cabo la llamada al sistema `symlink`, la cual permite crear un enlace blando (soft link) cuyo nombre está especificado por `newpath`, al archivo especificado en `oldpath`. La sintaxis de la llamada es:

```
...  
int symlink(const char *oldpath, const char *newpath);  
...
```