

1. [1,25] Con respecto al apoyo hardware al SO:
 - a. Explica los modos de direccionamiento de memoria de la arquitectura IA32: *memory flat, segmented y real-address mode*.
 - b. Describe los pasos, hardware y software, que se llevan a cabo para la resolución de una llamada al sistema.
 2. [1,25] Con respecto a virtualización:
 - a. ¿Cuál es la diferencia entre los dos enfoques de virtualización explicados en clase: *hipervisor tipo1 e hipervisor tipo2*?
 - b. Describe en qué consiste la técnica de virtualización denominada *virtualización asistida por hardware*. Bases para la explicación en la arquitectura x86 y los *rings* del procesador.
 3. [2,5] Responda a las siguientes cuestiones sobre el concepto de hebra:
 - a. ¿Qué ventajas proporciona el modelo de hebras frente al modelo de proceso tradicional?
 - b. ¿Cuál es el inconveniente en la implementación de hebras de usuario a la hora de que se realice una llamada al sistema bloqueante por parte del programa?
 - c. Justifique el grado de paralelismo real alcanzado por una aplicación con varias hebras, teniendo en cuenta que los programas utilizan una biblioteca de hebras a nivel usuario y el núcleo no planifica hebras sino procesos.
 4. [1,25] ¿Cómo implementa Linux el concepto de hebra? Explíquelo utilizando el PCB de Linux (`struct task_struct`) y la llamada al sistema `clone()`.
 5. [1,25] En un SO con una política de planificación apropiativa, enumere las distintas partes del SO que deben comprobar la posibilidad de desplazar al proceso que actualmente se está ejecutando y proponga un pseudocódigo que describa cómo se realizaría dicha comprobación en cada parte.
- [1,25] Con respecto a la planificación de procesos responda las siguientes cuestiones:
- a. ¿Qué algoritmo de planificación provoca una mayor penalización a los procesos limitados por E/S frente a los procesos limitados por CPU? ¿Por qué?
 - b. Describa los factores a considerar a la hora de diseñar un algoritmo de planificación basado en colas múltiples con realimentación. En particular, justifique como asociaría los conceptos de quantum y prioridades a su diseño.
- [1,25] Con respecto al núcleo de Linux visto en clase:
- a. Describe los pasos que ejecuta el núcleo de Linux en la función `do_exit()`.
 - b. Describa como se comporta el planificador de Linux, `schedule()`, para los procesos planificados mediante la clase de planificación CFS (*Complete Fair Scheduling*).