



Apellido y Nombres	Legajo	Calificación

- 1) Cambio de Nivel de Privilegio de una tarea. Responda los siguientes ítems indicando el modo de trabajo que considera para sus respuestas (IA-32 o IA-32e):
 - a) Mencione dos mecanismos por los cuales una tarea pueda acceder cuando lo necesite, a un servicio residente en un segmento no conforming de mayor Nivel de Privilegio.
 - b) ¿Que valor de privilegio debe tener los mecanismos mencionados en (a)?.
 - c) Para los escenarios descriptos en (a), se pide una descripción detallada del tratamiento de pila. ¿Que valores contienen los registros SS:ESP una vez accedido el código privilegiado?. Indicar como se obtienen estos valores.
- 2) Protección de páginas:
 - a) Explicar como es el mapeo del CPL en los niveles de privilegio de las páginas.
 - b) Un código cuyo CPL mapea como supervisor, intenta escribir una página que tiene permiso Read Only. ¿Cual es el resultado?
 - c) Explicar que ocurre en el caso detallado en (b) si el acceso se realiza con CR0.WP=1. En función de su respuesta indique en que casos resulta útil el empleo de este bit.
- 3) En un esquema de segmentación Flat, escriba el código que permita implementar un esquema de paginación que permita clonar tareas de modo que al ser creada la tarea nueva comparta las mismas páginas que contienen el código y los datos de la original. Ni bien la original o alguna de las tareas clon intenten modificar un dato, el kernel le proveerá su propia copia a la tarea que ejecutó la modificación, de modo que sea esta copia “personal” la que utilizará a partir de ese momento.
- 4) Sea la función *void sobel (char *orig , char *dest, int ancho, int alto, profundidad)* una función que aplica el operador Sobel a una imagen monocromo para efectuar un gradiente clásico para detección de bordes. Se pide que implemente dicha función mediante threads, a razón de uno por cada fila. El algoritmo genérico consiste en la convolución de la matriz de 3x3 compuesta por el pixel en cuestión y su N8, con cada operador de Sobel.

$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}; Gy = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

El valor absoluto del gradiente es para calcular cada pixel resultante es:

$$|G| = \sqrt{Gx^2 + Gy^2}$$

Para simplificar el cálculo puede usar la siguiente aproximación (saturando el resultado):

$$|G| = |Gx| + |Gy|$$