



Apellido y Nombres	Legajo	Calificación

**Parte Práctica: 1 hora 30 minutos**

Se desea desarrollar un kernel IA32. Los descriptors de la GDT e IDT ya están correctamente declarados y cargados en el sistema, y no debe inicializar los mismos. Su código ya se ejecuta en Modo Protegido.

Se le requiere a ud. que desarrolle las siguientes partes.

1. Un scheduler que corra en CPL=0. y que procese una cola de ejecución a modo de lista simplemente enlazada donde cada elemento de la lista posee la siguiente estructura:

TSS\_Number    dw ?    ; descriptor de la TSS  
Max\_Ticks      db ?    ; Prioridad en ciclo de timer tick continuos  
Current\_Ticks   db ?    ; Cantidad de timer ticks que lleva ejecutándose la tarea  
                              ; desde que le fue asignada la CPU  
Next            dd ?    ; Puntero a la dirección lineal del siguiente elemento de la  
                              ; lista.

La lista está enlazada de forma circular, es decir, la última tarea apunta en su campo Next a la primera.

Las entradas **no estarán** en direcciones de memoria consecutivas.

El scheduler es "non-preemptive", y por lo tanto no soporta tareas real time.

Como desarrollador, Ud. puede optar por implementar el scheduler a través de una puerta de interrupción o de una puerta de tarea. **Debe** elegir y aclarar de forma explícita cuál es la solución preferida, no siendo necesario que justifique su elección.

Tenga en cuenta que la lista varía dinámicamente, por lo tanto debe recorrerla desde la entrada de la tarea actual. Si la tarea a la que se le deben ajustar los ticks ya ha concluido (tal como se describe en el punto 3), debe ignorar los datos recibidos.

2. Rutina de interrupción para IRQ4 (debe ser una puerta de tarea) que deberá leer por el puerto serie 0x2E8 2 valores consecutivos:

- Un byte con el nuevo valor de la cantidad máxima de ticks.
- Dos bytes que componen el **TSS\_Number** de la tarea a la cual aplicarle el valor leído en el byte anterior.

Con estos dos valores debe ajustar la estructura de la tarea con la nueva cantidad de ticks. En el caso de que reciba un nuevo valor máximo de ticks para la tarea actual que sea menor a los ya ejecutados, debe actuar como si la misma hubiese alcanzado el máximo valor en el presente ciclo.

3. Puerta de llamada (ya que las tareas trabajan en nivel 3) para que cualquiera de ellas finalice donde el kernel deberá remover dicha tarea de la lista enlazada, y dejar el sistema funcionando de forma correcta. La Call Gate no debe

(Condición de aprobación: Punto 1 correctamente resuelto y al menos uno de los puntos subsiguientes)



Apellido y Nombres	Legajo	Calificación

**Parte Teórica: Tiempo límite 45 minutos**

- 1) Protocolo TCP. Mecanismo de sliding window. Explique su funcionamiento.
- 2) Se desea implementar en un driver la system call *sleep ()*. Se pide que escriba la porción de código (o pseudo código) de las funciones del driver que se necesitan para este fin. Considere el estado en el que debe estar el proceso invocante de la función *sleep ()*.
- 3) Cuales son los métodos que conoce para que una tarea logre escalar el nivel de privilegio en IA32? cuales de ellos pueden ser escenario del “Caballo de Troya”?
- 4) ¿Como se puede lograr que las tareas de un S.O. Para acceder al Buffer de Video en modo texto, que físicamente se encuentra a partir de la dirección 0x000B8000, se logre cuando estas escriban a partir de la dirección lineal 0x00100000? Escriba las estructuras con los valores adecuados y los registros involucrados inicializados como corresponde para obtener este resultado.