



Apellido y Nombres	Legajo	Calificación

### Ejercicio 1

Se trabaja en el desarrollo de un kernel para un embedded IA-32. Su tarea es construir el módulo de paginación. Para ello dispone de los siguientes datos:

- El sistema trabaja con un esquema de segmentación FLAT idéntico al de Linux, con dos segmentos de datos y dos de código (uno por cada nivel de privilegio 00 y 11), de 4 Gbytes de tamaño.
- El sistema mapeará los 4 Gbytes de espacio lineal en la memoria física instalada, cuya cantidad se averigua mediante una función ya existente denominada **GetMemSize**, que retorna en el EAX la cantidad de memoria en Mbytes (1 Mbyte =  $2^{20}$  bytes).
- El mapa de memoria física es:
  - 00000000 - 000007FF IDT
  - 00000800 - 000107FF GDT
  - 00010800 - 0007FFFF Código del Kernel
  - 00080000 - 0087FFFF Datos Kernel
  - 00880000 - Resultado de GetMemSize Datos y código de aplicaciones a demanda

Se pide:

1. Escribir el código para generar las páginas correspondientes al kernel, considerando las siguientes particularidades:
  1. Tamaño de página 4Kbytes.
  2. El buffer de video en modo texto debe corresponder a la dirección lineal 0x0087F000.
2. Escribir una función basada en la anterior para crear la DPT de una nueva tarea con su primer tabla de Páginas (PT), y otra para agregar páginas a tareas existentes. Debe estar perfectamente claro en que área de memoria se armarán dichas estructuras de paginación

### Ejercicio 2

Escriba un proceso que mantenga activos y vivos permanentemente 10 instancias child que esperen por el named pipe: /tmp/pipe[pid] (en donde **pid** es el process ID del child), un comando para ejecutar.

Una vez recibido el comando el child reemplaza su código por el del comando recibido y una vez ejecutado el mismo finaliza.

Siempre debe tenerse 10 instancias child listas para ejecutar.

El padre recibe los comandos por el port UDP 12345.

No deben quedar instancias defunct (EXIT\_ZOMBIE)