

Nº Orden	Apellido y nombre	L.U.	Cantidad de hojas

Organización del Computador 2

Segundo parcial — 14/11/17

1 (20)	2 (50)	3 (30)	
--------	--------	--------	--

Normas generales

- Numere las hojas entregadas. Complete en la primera hoja la cantidad total de hojas entregadas.
- Entregue esta hoja junto al examen, la misma **no** se incluye en la cantidad total de hojas entregadas.
- Está permitido tener los manuales y los apuntes con las listas de instrucciones en el examen. Está prohibido compartir manuales o apuntes entre alumnos durante el examen.
- Cada ejercicio debe realizarse en hojas separadas y numeradas. Debe identificarse cada hoja con nombre, apellido y LU.
- La devolución de los exámenes corregidos es personal. Los pedidos de revisión se realizarán por escrito, antes de retirar el examen corregido del aula.
- Los parciales tienen tres notas: I (Insuficiente): 0 a 59 pts, A- (Aprobado condicional): 60 a 64 pts y A (Aprobado): 65 a 100 pts. No se puede aprobar con A- ambos parciales. Los recuperatorios tienen dos notas: I: 0 a 64 pts y A: 65 a 100 pts.

Ej. 1. (20 puntos)

Responder detalladamente las siguientes preguntas, ejemplificar de ser posible.

- (4p) 1. ¿Cuántos bytes de tamaño, tiene un segmento de límite 0 y granularidad 0?
- (4p) 2. ¿Qué diferencia hay entre el bit *dirty* y el bit *accessed* en una entrada de tabla de páginas?
- (4p) 3. ¿Qué permisos efectivos tiene una pagina si su *Page Directory Entry* es de sólo lectura con nivel de usuario y su *Page Table Entry* es de lectura/escritura con nivel supervisor?
- (4p) 4. ¿Cuál es el nivel de privilegio necesario en un descriptor de interrupciones que atiende interrupciones de hardware? ¿Cómo se establece?
- (4p) 5. ¿Qué mecanismo se utiliza para saber si puedo acceder a un segmento de código?

Ej. 2. (30 puntos)

Se desea tener una función que dado una dirección de memoria correspondiente a la base del directorio de páginas, y una dirección física, devuelva un valor correspondiente a la cantidad de direcciones virtuales distintas por las que se puede acceder a dicha dirección.

La signature debe ser

```
unsigned int cantidad_direcciones(unsigned int base_dir, unsigned int fisica);
```

1. (15p) Escriba el código en C de la función, contando solo direcciones que se puedan acceder en nivel Supervisor.
2. (5p) Modifique la función anterior para considerar las direcciones en nivel de Usuario.
3. (10p) Si el valor devuelto por los llamados a las funciones anteriores con una determinada dirección física son 0, ¿se puede asumir que la página que corresponde es una página física libre? **Justifique.**

Nota: Considerar que los marcos de página son de 4kb.

Ej. 3. (50 puntos)

En un sistema con segmentación flat, se propone el esquema de paginación que muestra la figura a continuación. Cada tarea ocupa exactamente 20 KB de memoria física, que corresponden a 5 páginas.

El mapeo a memoria virtual de cada tarea corresponderá a mapear el código, datos y la pila de nivel 0 según corresponda a cada tarea.

Las tareas en el sistema son ejecutadas en orden, una por cada ciclo de reloj. Inicialmente las tareas comienzan con el EIP en la primer dirección de memoria de código y todos los registros de uso general en 0x00, excepto EAX, que contendrá el número de tarea que se esta ejecutando y EBX que contendrá el valor 0xFF la primera vez que se ejecute, o el código de error de la excepción correspondiente en el caso de que la tarea se haya reniciado.

Las tareas pueden cometer cualquier tipo de excepción, en ese caso deben ser reiniciadas y comenzar a ejecutar inmediatamente respetando las condiciones descriptas en el párrafo anterior. Puede suponer que la pila de nivel 3 apunta la tope de la página de datos de nivel 3.

- (10p) 1. Indicar los campos relevantes de todas las estructuras involucradas en el sistema para administrar segmentación, paginación, tareas, interrupciones, privilegios, registros de control y funciones del scheduler. Explicar como deben instanciarse las estructuras de datos y explicar su funcionamiento.
- (15p) 2. Programar en C la función `mapear_tarea`, que dado el puntero al directorio de paginas de una tarea se encarga de construir todo el mapa de paginación de la misma.
- (15p) 3. Programar en ASM/C la rutina de atención de interrupciones de alguna excepción del procesador.
- (10p) 4. Programar en ASM/C la rutina de atención de interrupciones del reloj.

