

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

Organización del Computador 2

Recuperatorio del segundo parcial — 11/07/2019

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

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. (30 puntos)

- (20p) a. Considerando la siguiente tabla de traducciones de direcciones por segmentación y paginación. De ser posible, dar un conjunto de descriptores de segmento, directorio de paginas y tablas de paginas que cumplan con todas las traducciones **simultáneamente**. Detallar los campos de todas las estructuras involucradas. Además indicar si la traducción es *identity mapping* y en el caso de que alguna traducción no sea posible indicar por qué.

Lógica	Lineal	Física	Características
0x0013:0x00002FF0	0x00924FF0	0x00F00FF0	Datos Nivel 3 Acción: Lectura de 1KB, CPL:1
0x0111:0x00123123	0x00123123	0x00A00123	Código Nivel 1 Acción: Escritura de 4 Bytes, CPL:0
0x0122:0x000FF202	0x09FFF202	0x00211202	Código Nivel 0 Acción: Ejecución de 3 Bytes, CPL:2
0x0120:0x00000FFF	0x09F00FFF	0x00AAAFFF	Código Nivel 0 Acción: Lectura de 2 Bytes, CPL:0

- (10p) b. ¿Es posible activar paginación si la base del segmento actual es distinta de cero? Justificar. Además, de ser posible, proponer un conjunto de traducciones que respeten lo pedido, caso contrario explicar por qué no es posible.

Ej. 2. (50 puntos)

Un sistema en dos niveles de protección con segmentación y paginación activa, ejecuta exactamente n tareas de forma concurrente. A cada tarea se le asocia otra, contabilizando un total de n pares de contextos de ejecución que deben ser administrados por el sistema.

Se proveen dos servicios:

- **cambiarTarea**: La tarea que llame a este servicio es intercambiada con la otra tarea en el par de tareas. La tarea que estaba ejecutando debe dejar de correr inmediatamente. El servicio no toma ningún parámetro.
- **dameRet**: Este servicio obtiene valores de la otra tarea en el par, justo antes que dejará de ser ejecutada. En **EAX** retorna el valor contenido en **[EBP+4]**.

Considerar que cualquiera de las tareas puede cometer una excepción. En ese caso la tarea debe ser restaurada. Para esto se cuenta con la función **restaurarMemoriaDeLaTarea(cr3* memorymap)** que

toma un puntero a un valor de 4 bytes que representa una estructura de CR3 y restaura toda la memoria de la tarea.

- (10p) a. Definir un posible mapa de memoria para las tareas. Indicar el rango de direcciones de las páginas utilizadas para el kernel y tareas. Indicar qué información específica es almacenada en cada rango designado (*Código/Datos, Page Directory, Page Table* y *Pila Nivel Cero*). Definir las entradas de la IDT para los servicios pedidos, para las rutinas de excepciones y para la interrupción de reloj. Además indicar el contenido de la TSS de una tarea al iniciar el sistema, explicando el contenido de cada uno de sus campos o los valores que deben tomar.
- (20p) b. Implementar en ASM y/o C la rutina de atención de interrupción del reloj y una rutina de las excepciones. Para ambas se debe escribir todo el código necesario para su implementación y declarar las estructuras de datos que hagan falta.
- (20p) c. Implementar en ASM y/o C la rutina de atención de interrupciones de los servicios del sistema.

Nota: Suponer que se cuenta con las definiciones de las estructuras del sistema para operar desde C.

Ej. 3. (20 puntos)

Construir las siguientes funciones en C o en ASM:

- (10p) a. `uint32_t countUserInts(idt_descriptor* idtDesc)` toma un puntero a un descriptor de `idt` y cuenta la cantidad de rutinas de atención de interrupción que pueden ser llamadas desde nivel 2.
- (10p) b. `uint32_t countLevelZeroTasks(gdt_descriptor* gdtDesc)` toma un puntero a un descriptor de `gdt` y cuenta la cantidad de entradas de `tss` cuyo selector de segmento de código es de nivel 0.

```
typedef struct str_gdt_descriptor {
    uint16_t  gdt_limit;
    uint32_t  gdt_base;
} __attribute__((__packed__)) gdt_descriptor;

typedef struct str_gdt_entry {
    uint16_t  limit_0_15;
    uint16_t  base_0_15;
    uint8_t   base_23_16;
    uint8_t   type:4;
    uint8_t   s:1;
    uint8_t   dpl:2;
    uint8_t   p:1;
    uint8_t   limit_16_19:4;
    uint8_t   avl:1;
    uint8_t   l:1;
    uint8_t   db:1;
    uint8_t   g:1;
    uint8_t   base_31_24;
} __attribute__((__packed__)) gdt_entry;
```

```
typedef struct str_idt_descriptor {
    uint16_t  idt_limit;
    uint32_t  idt_base;
} __attribute__((__packed__)) idt_descriptor;

typedef struct str_idt_entry {
    uint16_t  offset_0_15;
    uint16_t  segsel;
    uint8_t   zero;
    uint8_t   GateType:4;
    uint8_t   s:1;
    uint8_t   dpl:2;
    uint8_t   p:1;
    uint16_t  offset_16_31;
} __attribute__((__packed__)) idt_entry;
```

Suponer que se cuenta además con la estructura de `tss`.