



Apellido y Nombres	Legajo	Profesor	Año de Cursada	Cantidad de Hojas

### **Normas Generales**

Numere las hojas entregadas.

Complete en la primera hoja la cantidad total de hojas entregadas.

Cada ejercicio debe realizarse en hojas separadas. Debe identificarse cada hoja con: Nombre, Apellido y Legajo.

***Por favor entregar esta hoja y las restantes del tema junto al examen.***

1. En el sistema operativo LINUX
  - (a) Enumere los estados posibles de un proceso en LINUX.
  - (b) Cuál es la diferencia entre el estado TASK\_INTERRUPTIBLE y TASK\_UNINTERRUPTIBLE
  - (c) Con qué función se lleva a un proceso al estado TASK\_INTERRUPTIBLE
  - (d) Cómo identifica los procesos y los threads
  - (e) Con qué función se registra un dispositivo de tipo carácter y cual son sus argumentos.
2. Explique Scheduler Preemptivo, Scheduler Cooperativo y sus diferencias.
3. Mencione todas las formas/mecanismos disponibles en un procesador IA-32 para que **el código de una tarea** eleve su nivel de privilegio
4. Modelo de ejecución SIMD
  - (a) Explique como trabaja el modelo SIMD.
  - (b) Explique como trabaja Aritmética saturada vs. Aritmética de desborde. Cuál de estas incorpora el modelo SIMD.
  - (c) El siguiente cuadro muestra una suma empaquetada que corresponde a (marque la respuesta correcta):
    - i. Suma de bytes saturada no signada
    - ii. Suma de bytes saturada signada
    - iii. Suma de bytes por desborde
    - iv. Suma de words saturada signada

mm0	4D	23	9F	C0	11	4A	29	0B
mm1	32	F0	1A	0D	3F	AF	B0	36
Total	7F	FF	B9	CD	50	F9	D9	41



Apellido y Nombres	Legajo	Profesor	Año de Cursada	Cantidad de Hojas

5.Explique las ventajas y desventajas de que un procesador que implemente un pipeline de ejecución. ¿Cómo se puede optimizar el rendimiento del mismo?. ¿Cual es el efecto de un "caché miss" en un pipeline de ejecución?

6.Escriba el assembler que calcule el histograma a partir de una imagen en escala de grises de ocho bits. Se dispone de tres etiquetas con los siguientes significados:

(a)Cant\_pixels: dd ; indica la cantidad de elementos que componen la imagen

(b)Imagen: dd ; Posición de memoria al inicio de los valores de la imagen (bytes).

(c)Histograma: resd 256 ; Vector con el resultado del histograma

Nota: Un histograma es una representación gráfica de la frecuencia de ocurrencia de una variable. En las ordenadas se representan las frecuencias, y en las abscisas los diferentes valores de las variable, en este caso de 0 a 255.

7.Assembler embebido en C. Escriba el assembler que reciba parámetros de un código escrito en C según según la declaración que se muestra a continuación:

```
extern int suma_asm(unsigned int a, const char * name);
```

8.Continúe la siguiente porción de código de lenguaje C para que comience el proceso comience a recibir paquetes UDP en el primer puerto libre comprendido entre un límite inferior UDP\_Bajo y otro superior UDP\_Alto. En caso de no obtener ninguno la función retornará -1.

```
#define UDP_Bajo 4000
```

```
#define UDP_Alto 8000
```

```
int sockfd;
```

```
int UDP_Open() {
```

```
    int puerto
```

```
    struct sockaddr_in my_addr; /* direccion IP y numero de puerto local */
```

```
    struct sockaddr_in their_addr; /* direccion IP y numero de puerto del cliente */
```

```
    int addr_len;
```

```
    /* Continuar */
```

```
    return (puerto);
```

```
}
```



Apellido y Nombres	Legajo	Profesor	Año de Cursada	Cantidad de Hojas

Material de soporte:

```
my_addr.sin_family = AF_INET; /* host byte order */
```

```
my_addr.sin_port = htons(int puerto); /* network byte order */
```

```
my_addr.sin_addr.s_addr = INADDR_ANY; /* se asigna automaticamente la direccion IP local */
```

```
bzero(&(my_addr.sin_zero), 8); /* rellena con ceros el resto de la estructura */
```

```
int bind(int sockfd, struct sockaddr *my_addr, socklen_t addrlen);
```

```
int socket(int dominio, int tipo, int protocolo);
```

Los tipos definidos en la actualidad son: SOCK\_STREAM, SOCK\_DGRAM

9. A partir de CS:EIP = 0008:00009069 y dada la siguiente GDT:

Global Descriptor Table (base=0x00009002, limit=23):

GDT[0x00]=??? descriptor hi=0x00000000, lo=0x00000000

GDT[0x01]=Code segment, laddr=00000000, limit=00010 \* 4Kbytes, Execute-Only, 32-bit

GDT[0x02]=Data segment, laddr=00000000, limit=ffff \* 4Kbytes, Read/Write, Accessed

Se pide escribir la tabla de directorio y la o las tablas de páginas para el código solamente, asumiendo identity mapping.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Address of page directory <sup>1</sup>																Ignored				P C D	P W T	Ignored			CR3							
Bits 31:22 of address of 2MB page frame								Reserved (must be 0)				Bits 39:32 of address <sup>2</sup>				P A T	Ignored		G	1	D	A	P C D	P W T	U / S	R / W	1	PDE: 4MB page				
Address of page table																Ignored				0	I g n	A	P C D	P W T	U / S	R / W	1	PDE: page table				
Ignored																												0	PDE: not present			
Address of 4KB page frame																Ignored		G	P A T	D	A	P C D	P W T	U / S	R / W	1	PTE: 4KB page					
Ignored																												0	PTE: not present			



Apellido y Nombres	Legajo	Profesor	Año de Cursada	Cantidad de Hojas

10. Dado un sistema tiene RAM dinámica de 4 GB, y un cache L1 de 64Kbytes organizado en líneas de 32 bytes, con 8 líneas por set. Se pide:

(a) Determinar cantidad de Set's

(b) Determinar cantidad de bit's en el TAG

(c) Si el procesador accedió previamente a la dirección física 0x45671234, qué ocurre si el procesador luego requiere acceder a la dirección:

i. 0x45671242

ii. y luego a la 0x45671220.