



Apellido y Nombres	Legajo	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. ¿En qué estructura del kernel se almacena?

(b) ¿Cuál es el resultado de `kill -9 7149` si el proceso 7149 está en estado `TASK_UNINTERRUPTIBLE`?

(c) ¿Con qué función el kernel “despierta” a un proceso en estado `TASK_UNINTERRUPTIBLE`?

(d) ¿Cuáles son los dos recursos que usa el kernel para “dormir” un proceso?.

(e) Mencione como se crea un thread, cual es la función que se utiliza, que argumento/s recibe, cuales son las funciones de sincronización, y que recaudos se toman al compilar / linkear el programa que va a paralelizarse.

2. Explique cómo funciona el mecanismo conocido como Copy-On-Write y que recursos del procesador utiliza el kernel de Linux en su versión para IA-32.

3. Una tarea en un procesador IA-32 llama a dos call gates. Una la lleva a un segmento de código con `DPL=00` y `C=0`, y la otra a un segmento de código con `DPL=00` y `C=1`. ¿Existe diferencia en el tratamiento de la pila entre ambos casos?. Justifique

4. Modelo de ejecución SIMD

(a) Explique cómo funcionan las instrucciones de comparación empaquetadas, cual es su resultado y explique alguna aplicación en procesamiento digital de señales o imágenes.

(b) Tratamiento de datos en punto flotante: Diferencia entre instrucciones que manejan escalares y vectores de números en punto flotante. Formato que se utiliza. Mencione ejemplos

(c) Lectura de datos alineados. Bajo qué condiciones ocurre la excepción `#AD`.



Apellido y Nombres	Legajo	Cantidad de Hojas

(d)¿Qué significa una denormalización de un número en punto flotante?. ¿Cuál es su efecto dentro del formato del número? ¿Cuál es su consecuencia?

5.Paralelismo a nivel de instrucción en procesadores: Motor de ejecución fuera de orden. Ventajas que introduce. Implementación Three Core Engine vigente en IA-32 desde microarquitectura P6: Funcionamiento, concepto de microoperación, pool de instrucciones, puertos de ejecución. Unidad de retiro.

6.Escriba el assembler que binarice una imagen en escala de grises de ocho bits. El procedimiento recibe:

- (a)en al el valor de ocho bits correspondiente al umbral
- (b)en ecx la cantidad de bytes que componen la imagen
- (c)en esi el puntero a la tira de bytes (o pixeles) de la imagen

Nota: Binarizar una imagen es reemplazarla por una del mismo tamaño pero cuyos pixeles toman dos valores posibles: negro absoluto o blanco absoluto (0 o 255 respectivamente), de acuerdo al valor del pixel original respecto de uno definido como umbral. Los valores por debajo del umbral saturan a 0 y los que lo superan saturan a 255.

Pista: Alguna de las instrucciones que explica en el ítem 4.a. pueden resultarle de utilidad seguramente

7.Assembler embebido en C. Escriba el assembler que reciba parámetros de un código escrito en C según la declaración que se muestra a continuación, y que genere tres variables locales de tipo doble Word:

```
extern int binarize_asm(unsigned char threshold, const char *pixelmap, int img_size);
```

8.Complete el fragmento de código a continuación, para que el proceso comience a recibir paquetes TCP por el primer puerto libre en el rango de TCP_Bajo a TCP_Alto. En caso de no haber ningún puerto habilitado para su uso, la función deberá retornar -1 como código de error.

```
#define TCP_Bajo 5000
#define TCP_Alto 5600
int sockfd;
int TCP_Open() {
    int puerto
```



Apellido y Nombres	Legajo	Cantidad de Hojas

```
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);
}
```

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. Dada la siguiente GDT:

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

GDT[0x01]=Code segment, laddr=00000000, limit=00006 * 4Kbytes, Execute-Only, 32-bit

GDT[0x02]=Data segment, laddr=00000000, limit=0000C * 4Kbytes, Read/Write, Accessed

Se pide escribir la tabla de directorio y la o las tablas de páginas para que el código mapee a partir de 0x00100000, y los datos a partir de 0x00800000.



Apellido y Nombres	Legajo	Cantidad de Hojas

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 ¹																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 ²		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										

10. Dado un procesador capaz de direccionar 4Gbytes de RAM dinámica, y que dispone en su core de un cache L1 asociativo de 4 vías de 64Kbytes de capacidad organizado en líneas de 16 bytes, las cuales se agrupan en sets de 8 líneas, se pide:

- Determinar cantidad de Set's en que se divide el sistema
- Determinar el tamaño en bits del TAG de cada set
- Si el procesador requiere el byte de la dirección física 0x00197665
 - El rango de direcciones físicas que se "cachean"
 - ¿Qué valor se escribe en el tag correspondiente?
 - ¿A qué set pertenece el tag?
 - Si a continuación el procesador requiere el byte que está en la dirección física 0x0019766D ¿Se produce un hit o un miss?. Justifique
- Para la siguientes secuencias de accesos consecutivos suponiendo que ninguno estaba cacheado, se desea saber si alguno genera desalojo del caché. En tal caso indique cual reemplaza a cual en cada caso
 - 0x0019B662, 0x001D7669, 0x0019B66E, 0x00167664, 0x001E7660
 - 0x0039B56E, 0x003D7567, 0x0039B561, 0x0036756F, 0x003E7660