UNIVERSIDAD DE GRANADA — ETSIIT Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

2º Grado Informática
Estructura de Computadores
20 de julio de 2017

ATC Departamento de Arquitectura y Tecnología de Computadores UNIVERSIDAD DE GRANADA

Nombre:	
DNI:	**Grupo:**

Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.
Cada respuesta vale 0.2p si es correcta, 0 si está en blanco o claramente tachada, -0.0$\widehat{6}$p si es errónea.
Anotar las respuestas (a, b, c ó d) en la siguiente tabla.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

1. ¿Cuál de las siguientes instrucciones máquina copia en EAX la dirección efectiva resultante de la operación EDX*4 + EBX?

a. leal 4(%edx, %edx), %eax
b. leal (%ebx, %edx, 4), %eax
c. movl 4(%edx, %edx), %eax
d. movl (%ebx, %edx, 4), %eax

2. ¿Cuál de las siguientes instrucciones máquina copia en EAX el entero almacenado en la posición de memoria cuya dirección efectiva es el resultado de la operación EDX*4 + EBX?

a. leal 4(%edx, %edx), %eax
b. leal (%ebx, %edx, 4), %eax
c. movl 4(%edx, %edx), %eax
d. movl (%ebx, %edx, 4), %eax

3. Los switches `-m elf_i386` y `-m elf_x86_64` para trabajar en 32 bits / 64 bits corresponden a la herramienta...

a. gcc
b. as
c. ld
d. nm

4. Si ECX vale 0, la instrucción adc $0,%ecx

a. Pone CF=1
b. Pone CF=0
c. Cambia CF
d. No cambia CF

5. Dada la siguiente definición de datos:

```
lista:   .int 0x10000000, 0x50000000,
              0x10000000, 0x20000000
longlista:    .int (.-lista)/4
resultado:    .quad 0x123456789ABCDEF
formato: .ascii "suma=%llu=%llx hex\n\0"
```

La instrucción para copiar la dirección de memoria donde comienza lista en el registro EBX es:

a. movl lista, %ebx
b. movl $lista, %ebx
c. movl (lista), %ebx
d. movl $lista, (%ebx)

6. Dada la siguiente definición de datos:

```
lista:   .int 0x10000000, 0x50000000,
              0x10000000, 0x20000000
longlista:    .int (.-lista)/4
resultado:    .quad 0x123456789ABCDEF
formato: .ascii "suma=%llu=%llx hex\n\0"
```

la instrucción `movl longlista, %ecx` copia el siguiente valor:

a. 4
b. 8
c. 16
d. 32

---

7. Dada la siguiente definición de datos:

```
lista:  .int 0x10000000, 0x50000000,
            0x10000000, 0x20000000
longlista:   .int (.-lista)/4
resultado:   .quad 0x123456789ABCDEF
formato: .ascii "suma=%llu=%llx hex\n\0"
```

y suponiendo que hemos llamado a una función *suma* que devuelve un número de 64 bits en la pareja EDX:EAX, las instrucciones que copian ese número en *resultado* son:

a. movl %eax, resultado
   movl %edx, resultado+4
b. movl (%eax), resultado
   movl (%edx), resultado+4
c. movl %eax, resultado+4
   movl %edx, resultado
d. movl (%eax), resultado+4
   movl (%edx), resultado

---

8. Dada la siguiente definición de datos:

```
lista:  .int 0x10000000, 0x50000000,
            0x10000000, 0x20000000
longlista:   .int (.-lista)/4
resultado:   .quad 0x123456789ABCDEF
formato: .ascii "suma=%llu=%llx hex\n\0"
```

la llamada correcta a *printf* será:

a. push resultado+4
   push resultado
   push $formato
   call printf
   add $12, %esp
b. push resultado+4
   push resultado
   push resultado+4
   push resultado
   push $formato
   call printf
   add $20, %esp
c. push resultado
   push resultado+4
   push $formato
   call printf
   add $12, %esp
d. push resultado
   push resultado+4
   push resultado
   push resultado+4
   push $formato
   call printf
   add $20, %esp

---

9. En la práctica "media" se pide sumar una lista de enteros ***con*** signo de 32 bits en una plataforma de 32 bits sin perder precisión, esto es, evitando overflow. ¿Cuál es el menor valor positivo que repetido en los ***dos*** primeros elementos de la lista causaría overflow con 32 bits al realizar la suma de ***esos dos*** primeros elementos de la lista?

a. 0x0400 0000
b. 0x0800 0000
c. 0x4000 0000
d. 0x8000 0000

---

10. ¿Cuál de las siguientes afirmaciones es cierta respecto al lenguaje C?

a. En lenguaje C, al llamar a una subrutina o función se introducen los parámetros en la pila y después se realiza una llamada a la subrutina
b. Los parámetros se introducen en la pila en el orden en el que aparecen en la llamada de C, es decir, empezando

por el primero y acabando por el último

c. Antes de volver de la rutina llamada, el programa en C se encarga de quitar de la pila los parámetros de llamada realizando varios pop

d. Pasar a una función un puntero a una variable se traduce en introducir en la pila el valor de la variable

---

11. ¿Cuál de los siguientes registros tiene que ser salvaguardado (si va a modificarse) dentro de una subrutina según la convención cdecl para IA32?

a. eax

b. ebx

c. ecx

d. edx

---

12. ¿En qué registro se pasa el primer argumento a una función en Linux gcc x86-64?

a. ecx

b. edx

c. esi

d. edi

---

13. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount3:

```
int popcount3(unsigned* array,
              int len){
    int i, res = 0;
    unsigned x;
    for( i=0; i<len; i++ ) {
      x = array[i];
      asm("ini3:       \n"
      "shr %[x]        \n"
      "adc $0, %[r]    \n"
      "add $0, %[x]    \n"
      "jne ini3        \n"
        : [r] "+r" (res)
        : [x] "r" (x) );
    }
    return res;
}
```

Esta función produce siempre el resultado correcto, a pesar de que una instrucción máquina en la sección **asm()** es distinta a la que se esperaba después de haber estudiado `pcount_r` en teoría. La instrucción distinta también se podría haber cambiado por…

a. sar %[x]

b. adc $0, %[x]

c. test %[x], %[x]

d. cmp %[x], %[r]

---

14. En la práctica de la bomba necesitamos estudiar el código máquina de la bomba del compañero. A veces dicho código no se visualiza directamente en el depurador ddd, y algunas de las técnicas que se pueden probar para conseguir visualizarlo son… (marcar la opción ***falsa***)

a. comprobar que está activado el panel *View → Machine Code Window*

b. escribir `info line main` en el panel de línea de comandos gdb

c. recompilar con información de depuración, por si se nos había olvidado, ya que sin –g el ejecutable no contiene información de depuración

d. asegurarse de que se ha escrito correctamente el nombre del ejecutable

---

15. En la práctica de la bomba, el primer ejercicio consistía en "saltarse" las "explosiones", para lo cual se puede utilizar…

a. objdump o gdb
b. gdb o ddd
c. ddd o hexedit
d. hexedit u objdump

16. En la práctica de la bomba, el segundo ejercicio consistía en crear un ejecutable sin "explosiones", para lo cual se puede utilizar…

a. objdump o gdb
b. gdb o ddd
c. ddd o hexedit
d. hexedit u objdump

17. En la práctica de la bomba, el tercer ejercicio consistía en usar un editor hexadecimal para crear un ejecutable sin "explosiones". Para saber qué contenidos del fichero hay que modificar, se puede utilizar… (marcar la opción falsa)

a. objdump
b. gdb
c. ddd
d. hexedit

18. Suponer una memoria cache con las siguientes propiedades: Tamaño: 512 bytes. Política de reemplazo: LRU. Estado inicial: vacía (todas las líneas inválidas). Suponer que para la siguiente secuencia de direcciones enviadas a la cache: 0, 2, 4, 8, 16, 32, la tasa de acierto es 0,33. ¿Cuál es el tamaño de bloque de la cache?

a. 4 bytes
b. 8 bytes
c. 16 bytes
d. Ninguno de los anteriores

19. Abajo se ofrece el listado de una función para multiplicar matrices C = A x B.

```
void mult_matr(float A[N][N],
float B[N][N],float C[N][N]){
/*Asume val.ini. C={0,0...}*/
int i,j,k;
for (i=0; i<N; i++)
 for (j=0; j<N; j++)
  for (k=0; k<N; k++)
   C[i][j]+= A[i][k]*B[k][j];
}
```

Suponer que:

• El computador tiene una cache de datos de 8 MB, 16-vías, líneas de 64 bytes.

• N es grande, una fila o columna no cabe completa en cache.

• El tamaño de los tipos de datos es como en IA32.

• El compilador optimiza el acceso a C[i][j] en un registro.

Aproximadamente, ¿qué tasa de fallos se podría esperar de esta función para valores grandes de N?

a. 1/2
b. 1/4
c. 1/8
d. 1/16

20. Sea un computador de 32 bits con una memoria cache L1 para datos de 32 KB y líneas de 64 bytes asociativa por conjuntos de 2 vías. Dado el siguiente fragmento de código:

```
int v[262144];
for (i=0; i<262144; i+=8)
   v[i] = 9;
```

¿Cuál será la tasa de fallos aproximada que se obtiene en la ejecución del bucle anterior?

a. 0 (ningún fallo)
b. 1/2 (mitad aciertos, mitad fallos)
c. 1/8 (un fallo por cada 8 accesos)
d. 1 (todo son fallos)