2º Grado Informática
Estructura de Computadores
9 Septiembre 2013

Nombre:	
DNI:	Grupo:

Examen de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.
Cada respuesta vale 4/20 si es correcta, 0 si está en blanco o claramente tachada, -4/60 si es errónea.
Anotar las respuestas (a, b, c o 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. Considerar los siguientes dos bloques de código almacenados en dos ficheros distintos:

```
/* main.c */
int i = 0;
int main() {
    func();
    return 0;
}
```

```
/* func.c */
int i = 1;
void func() {
  printf("%d", i);
}
```

¿Qué sucederá cuando se compile, enlace y ejecute este código?

a. Error al compilar o enlazar, no se obtiene ejecutable
b. Escribe "0"
c. Escribe "1"
d. A veces escribe "0" y a veces "1"

2. ¿Qué hace gcc -S?

a. Compilar .c → .s (fuente C a fuente ASM)
b. Compilar .s → .o (fuente ASM a objeto)
c. Compilar optimizando tamaño (size), no tiempo
d. Compilar borrando del ejecutable la tabla de símbolos (strip)

3. ¿Qué modificador (switch) de gcc hace falta para compilar .c → .o (de fuente C a código objeto)?

a. Eso no se puede hacer con gcc
b. gcc -c
c. gcc -o
d. gcc -O

4. ¿Qué modificador (switch) de gcc hace falta para compilar una aplicación de 32bits (fuente C) en un sistema de 64bits en el que se ha instalado también el compilador de 32bits?

a. -m32
b. -32
c. -m elf_i386
d. No se puede conseguir ese efecto con modificadores de gcc, porque el gcc por defecto (/usr/bin/gcc) es el de 64bits, se debe usar el gcc de la instalación alternativa (de 32bits)

5. ¿Qué modificador (switch) de as hace falta para ensamblar una aplicación de 32bits en un sistema de 64bits en el que se ha instalado también el compilador de 32bits?

a. -m32
b. -32
c. -m elf_i386
d. Ninguna de las anteriores respuestas es correcta

6. Como parte del proceso de compilación de una aplicación en lenguaje C, enlazar .o → .exe (de objeto proveniente de fuente C a ejecutable) usando sólo as y ld, sin gcc...

a. Se puede, repartiendo entre as y ld los modificadores (switches) que corresponda
b. Se puede, repartiendo modificadores entre as y ld, y añadiendo al comando ld el runtime de C
c. Basta usar ld, con los modificadores de gcc que corresponda, y añadiéndole el runtime de C
d. Ninguna de las anteriores respuestas es correcta

---

7. La práctica "popcount" debía calcular la suma de bits de los elementos de un array. Un estudiante entrega la siguiente versión de popcount4:

```
int popcount4(unsigned* array, int len)
{
  int i, j, res = 0;
  for(i = 0; i < len; ++i) {
    unsigned x = array[i];
    int n = 0;
    do {
      n += x & 0x01010101L;
      x >>= 1;
    } while(x);
    for(j = 16; j == 1; j /= 2){
      n ^= (n >>= j);
    }
    res += n & 0xff;
  }
  return res;
}
```

Esta función popcount4:

a. Produce el resultado correcto
b. Fallaría con array={0,1,2,3}
c. Fallaría con array={1,2,4,8}
d. No es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

---

8. La práctica "paridad" debía calcular la suma de paridades impar (XOR de todos los bits) de los elementos de un array. Un estudiante entrega la siguiente versión de parity6:

```
int parity6(unsigned * array, int len)
{
  int  i, result = 0;
  unsigned x;
  for (i=0; i<len; i++){
    x = array[i];
    asm( "mov %[x], %%edx  \n\t"
         "shr $16,  %%edx  \n\t"
         "shr $8,   %%edx  \n\t"
         "xor %%edx,%%edx  \n\t"
         "setp %%dl        \n\t"
         "movzx %%dl, %[x] \n\t"
         : [x] "+r" (x)
         :
         : "edx"
         );
    result += x;
  }
  return result;
}
```

Esta función parity6:

a. Produce el resultado correcto
b. Fallaría con array={0,1,2,3}
c. Fallaría con array={1,2,4,8}
d. No es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

---

9. Respecto a las bombas estudiadas en la práctica 3, ¿en cuál de los siguientes tipos de bomba sería más **fácil** descubrir la(s) contraseña(s)? Se distingue entre strings definidos en el código fuente de la bomba, y strings solicitados al usuario mediante scanf(). Por "cifrar" se entiende aplicar la cifra del César (sumar o restar una constante fija a los códigos ASCII).

a. 1 string del fuente se cifra, y se compara con el string del usuario
b. 2 strings del usuario se concatenan, se cifra el resultado y se compara con el string del fuente
c. 2 strings del usuario se cifran, se concatenan los resultados, y se compara con el string del fuente
d. Las opciones más fáciles son de la misma dificultad, así que no se puede marcar ninguna como la más fácil

---

10. Respecto a las bombas estudiadas en la práctica 3, ¿en cuál de los siguientes tipos de bomba sería más **difícil** descubrir la contraseña? Se distingue entre enteros definidos en el código fuente de la bomba, y enteros solicitados al usuario mediante scanf(). Por "procesar" se entiende calcular el n-ésimo elemento de la serie de Fibonacci.

a. 1 entero del usuario se procesa, y se compara con el entero del fuente
b. 2 enteros del fuente se suman, se procesa la suma, y se compara el resultado con el entero del usuario
c. 2 enteros del fuente se procesan, se suman los resultados, y se compara la suma con el entero del usuario
d. Las opciones más difíciles son de la misma dificultad, así que no se puede marcar ninguna como la más difícil

11. ¿Cuál de los siguientes es el orden correcto en el ciclo de compilación de un programa escrito en lenguaje C? (el fichero sin extensión es un ejecutable):

a. file.s → file.c → file → file.o
b. file.c → file.s → file.o → file
c. file.c → file.s → file → file.o
d. file.s → file.c → file.o → file

12. ¿Qué hace gcc -O?

a. Compilar sin optimización, igual que -O0
b. Compilar .c → .o
c. Compilar .s → .o
d. Compilar con optimización, igual que -O1

13. ¿Qué modificador (switch) de gcc hace falta para compilar .s → .o sin llamar al enlazador?

a. Eso no se puede hacer con gcc
b. gcc -c
c. gcc -s
d. gcc -S

14. ¿Qué modificador (switch) de gcc hace falta para compilar una aplicación de 32 bits en un sistema de 64 bits?

a. -m32
b. -m64
c. -32
d. -64

15. Compilar .s → ejecutable, usando sólo as y ld, sin gcc...

a. No se puede
b. Se puede, usando en as y ld los modificadores (switches) que corresponda
c. Basta usar as, con los modificadores que corresponda
d. Basta usar ld, con los modificadores que corresponda

16. ¿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

17. El primer parámetro de printf:

a. es un char
b. es un entero
c. es un puntero
d. puede ser de cualquier tipo, incluso no existir

18. Respecto al procesador que denominamos ssrDLX (procesador DLX sin segmentar), ¿cuál de las siguientes afirmaciones es falsa?

a. Todas las instrucciones deben realizar las cuatro etapas del cauce

b. Las etapas ID y WB duran un 80% del tiempo de ciclo de reloj
c. Una operación de suma de enteros (operandos y resultado en el banco de registros) necesitará 3.6 ciclos en el procesador sin segmentar
d. ld f2,a en el procesador ssrDLX requiere 4.6 ciclos

19. La directiva .text en el simulador DLX con la dirección 256, (.text 256), indica en un programa que:

a. la variable text tiene el valor 0x100
b. Existe una etiqueta de salto denominada .text en la posición 0x100
c. La primera instrucción del programa se ubicará en la posición 0x100
d. Todas las anteriores afirmaciones son incorrectas

20. En un procesador de la familia 80x86 una variable de 32 bits, entera con signo, almacenada a partir de la dirección n contiene: 0xFF en la dirección n, 0xFF en la dirección n+1, 0xFF en la dirección n+2 y 0xF0 en la dirección n+3. ¿Cuánto vale dicha variable?

a. -16
b. -251658241
c. 16
d. 4294967280