

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

- a. fich.c -> fich.o -> fich.s -> fich
- b. fich -> fich.s -> fich.o -> fich.c
- c. fich.c -> fich.s -> fich -> fich.o
- d. fich.c -> fich.s -> fich.o -> fich

2. ¿Qué hace gcc -O?

- a. Compilar con optimización suave
- b. Compilar .c->.o (fuente C a objeto)
- c. Compilar .s->.o (fuente ASM a objeto)
- d. Ambas (b) y (c), según la extensión de los ficheros que se usen como argumentos

3. ¿Qué modificador (switch) de gcc hace falta para compilar .c->.s (de fuente C a fuente ASM)?

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

4. ¿Qué modificador (switch) de ld hace falta para enlazar 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. No hace falta modificador, ld lo deduce del tipo de objeto a enlazar

5. Compilar .c->exe (de fuente C a ejecutable) usando sólo as y ld, sin gcc...

- a. No se puede
- b. Se puede, repartiendo entre as y ld los modificadores (switches) que corresponda
- c. Se puede, repartiendo modificadores entre as y ld, y añadiendo al comando ld el runtime de C
- d. Basta usar ld, con los modificadores de gcc que corresponda, y añadiéndole el runtime de C

6. ¿Cuál de las siguientes afirmaciones es falsa 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 inverso a como aparecen en la llamada de C, es decir, empezando por el último y acabando por el primero
- 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 dirección de memoria donde está almacenada la variable

7. El primer parámetro de printf:

- a. es un entero
- b. es un número en coma flotante
- c. es un puntero
- d. puede ser de cualquier tipo

8. Para averiguar la paridad de un número se puede usar la operación:

- a. NOT
- b. AND
- c. OR
- d. XOR

9. Utilizando la sentencia asm, las denominadas restricciones que se indican al final de dicha sentencia, involucran a:

- a. Solamente las entradas
- b. Solamente las salidas
- c. Solamente los sobrescritos
- d. Ninguna de las anteriores es cierta

10. Suponga la siguiente sentencia asm en un programa:

```
asm(" add (%[a],[i],4),%r"
```

```
:[r] "+r" (result)
```

```
:[i] "r" (i),
```

```
[a] "r" (array) );
```

¿Cuál de las siguientes afirmaciones es correcta?

- a. r es una posición de memoria de entrada/salida
- b. a es una posición de memoria de entrada
- c. i es un registro de entrada
- d. la salida de la función se fuerza a que esté en la variable result

11. En una bomba como las estudiadas en la práctica 4, del tipo...

```
0x0804873f <main+207>: call 0x8048504 <scanf>
```

```
0x08048744 <main+212>: mov 0x24(%esp),%edx
```

```
0x08048748 <main+216>: mov 0x804a044,%eax
```

```
0x0804874d <main+221>: cmp %eax,%edx
```

```
0x0804874f <main+223>: je 0x8048756 <main+230>
```

```
0x08048751 <main+225>: call 0x8048604 <boom>
```

```
0x08048756 <main+230>: ...
```

la contraseña es...

- a. el entero 0x804a044
- b. el entero almacenado a partir de la posición de memoria 0x804a044
- c. el string almacenado a partir de la posición de memoria 0x24(%esp)
- d. ninguna de las anteriores

12. En una bomba como las estudiadas en la práctica 4, del tipo...

```
0x08048705 <main+149>: call 0x80484c4
```

```
<gettimeofday>
```

```
...
```

```
0x08048718 <main+168>: cmp $0x5,%eax
```

```
0x0804871b <main+171>: jle 0x8048722 <main+178>
```

```
0x0804871d <main+173>: call 0x8048604 <boom>
```

```
0x08048722 <main+178>: ...
```

ejecutada paso a paso con el depurador ddd, interesaría...

- a. ejecutar hasta jle, ajustar %eax a 6, y continuar ejecutando paso a paso
- b. ejecutar hasta jle, ajustar %eax a 4, y continuar ejecutando paso a paso
- c. cambiar jle por jmp usando ddd o un editor hex, salvar el programa y reiniciar la depuración con el nuevo ejecutable
- d. Ninguna de las opciones anteriores es de interés (bien porque no se pueda hacer eso o porque no sirva para evitar la bomba)

13. En una bomba como las estudiadas en la práctica 4, del tipo...

```

0x080486e8 <main+120>: call 0x8048524 <strcmp>
0x080486ed <main+125>: test %eax,%eax
0x080486ef <main+127>: je 0x80486f6 <main+134>
0x080486f1 <main+129>: call 0x8048604 <boom>
0x080486f6 <main+134>: ...

```

la contraseña es...

- el valor que tenga %eax
- el string almacenado a partir de donde apunta %eax
- el entero almacenado a partir de donde apunta %eax
- ninguna de las anteriores

14. Respecto a las bombas estudiadas en la práctica 4, ¿en cuál de los siguientes tipos de bomba sería más difícil descubrir la contraseña? Se distingue entre strings definidos en el código fuente de la bomba, y strings solicitados al usuario mediante scanf(). Por "cifrar" podemos entender la cifra del César, por ejemplo. "Invertir" es darle la vuelta al string de manera que la primera letra se convierta en la última y viceversa.

- 1 string del usuario se cifra, y se compara con el string del fuente
- 2 strings del fuente se invierten, se concatenan, se cifra el resultado, y se compara con el string del usuario
- 2 strings del fuente se concatenan, se invierte el resultado, se cifra, y se compara con el string del usuario
- Las 2 (o 3) opciones más difíciles son de la misma dificultad, así que no se puede marcar ninguna como la más difícil

15. Respecto a las bombas estudiadas en la práctica 4, ¿en cuál de los siguientes tipos de bomba sería más difícil descubrir la(s) contraseña(s)? Se distingue entre enteros definidos en el código fuente de la bomba, y enteros solicitados al usuario mediante scanf(). Por "procesar" podemos entender

calcular el n-ésimo elemento de la serie de Fibonacci, por ejemplo.

- 1 entero del fuente se procesa, y se compara con el entero del usuario
- 2 enteros del usuario se suman, se procesa la suma, y se compara con el entero del fuente
- 2 enteros del usuario se procesan, se suman los resultados, y se compara con el entero del fuente
- Las 2 (o 3) opciones más difíciles son de la misma dificultad, así que no se puede marcar ninguna como la más difícil

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

- Todas las instrucciones deben realizar las cuatro etapas del cauce
- Las etapas ID y MEM duran un 80% del tiempo de ciclo de reloj
- Una operación de suma de enteros (operandos y resultado en el banco de registros) necesitará 3.8 ciclos en el procesador sin segmentar
- ld f2,a en el procesador ssrDLX requiere 4.6 ciclos

17. Dado un programa en un procesador DLX, se puede aplicar la técnica que recibe el nombre de desenrollado de bucle, la cual repercute en:

- realizar los cálculos de varias iteraciones en diferentes subrutinas
 - reducir el número de instrucciones de salto que se tienen que ejecutar
 - aumentar el número de instrucciones en el código para aumentar la probabilidad de que existan riesgos de control
 - reorganización de las instrucciones para reducir el efecto de las dependencias entre ellas
- ¿Cuál de las anteriores afirmaciones es cierta?

18. 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 256
 - b. La primera instrucción del programa se ubicará en la posición 0x100
 - c. Existe una etiqueta de salto denominada .text en la posición 256
 - d. Todas las anteriores afirmación son incorrectas
-

19. ¿Cómo se almacenaría como palabra de 32 bits el número -128 en un sistema que utilice el criterio del extremo menor ("little endian")?

- a. posición 0: FF pos.1:FF pos.2: FF pos.3: 00
 - b. 0:00 1:FF 2:FF 3:FF
 - c. 0:00 1:01 2: 00 3:80
 - d. Ninguna de las anteriores
-

20. En un procesador de la familia 80x86 las posiciones de memoria que representan una variable entera contiene los bytes: F0 FF FF FF. ¿Cuánto vale dicha variable?

- a. -16
 - b. 4043309055
 - c. 16
 - d. 4294967280
-