Estructura de Computadores
Grado en Ingeniería Informática
27 de Octubre de 2016

Nombre: **DNI:** **Grupo:**

Sobre 10, cada respuesta vale 2 si es correcta, 0 si está en blanco o claramente tachada, y -2/3 si es errónea.
Anotar las respuestas (**a**, **b**, **c** o **d**) en la siguiente tabla.

1	2	3	4	5
c	c	a	a	c

1. Para desplazar %eax a la derecha un número variable de posiciones, indicado en %ebx, se puede hacer…

a. sar %ebx, %eax — no existe

b. sar %bl, %eax — no existe

c. mov %ebx, %ecx
sar %cl, %eax — p.213 libro, Ej.3.8, p.214

d. No se puede, sar sólo admite un número fijo de posiciones (debe ser un valor inmediato)

2. Indicar cuál es la dirección de salto (en qué dirección empieza la subrutina <main>) para esta instrucción call
0804854e: e8 3d 06 00 00 call <main>
08048553: 50 pushl %eax

a. 0804854e + 3d060000 — no, offset little-endian

b. 0804854e + 0000063d = 08048b8b — no, %eip=08048553

c. 08048b90 — Tema2.3, tr.7

d. No se puede deducir de la información proporcionada, faltan datos

3. El motivo para usar una estructura de pila para almacenar las direcciones de retorno es…

a. que, obviamente, como no puede ser de otra manera, se debe retornar de la subrutina invocada antes que de la invocante, con lo cual se consume la última dirección de retorno anotada antes que la anterior — Tema2.3, tr.10

b. que así se pueden almacenar argumentos y variables locales, cosa que no se podría hacer sin usar una pila (no serviría una cola, ni un montículo (heap), ni por supuesto los registros de la CPU)

c. que así se pueden gestionar funciones recursivas y reentrantes, además de funciones anidadas “normales” (no reentrantes o recursivas) que sí podrían haberse implementado con otros mecanismos distintos a una pila

d. elección del fabricante, puesto que existen fabricantes que usan colas, otros usan montículos, otros usan registros de la CPU, y todos pueden gestionar funciones “normales”, anidadas, recursivas y reentrantes con la misma facilidad

4. Considerando que un marco de pila va desde una dirección de retorno a la siguiente, los argumentos de llamada a una función…

a. están en el marco de pila del invocante, y es responsabilidad del invocante ponerlos al llamar — Tema2.3, tr.24

b. están en el marco de pila del invocante, y es responsabilidad del invocado ponerlos al llamar

c. están en el marco de pila del invocado, y es responsabilidad del invocante ponerlos al llamar

d. están en el marco de pila del invocado, y es responsabilidad del invocado ponerlos al llamar

5. En la convención de llamada cdecl seguida por gcc Linux/IA-32…

a. EAX es un registro salva-invocante, por eso en cualquier función hay que salvarlo antes de usarlo — no para usarlo

b. EBX es un registro salva-invocante, por eso si se usa hay que salvarlo antes de llamar a una función — salva-invocado

c. ESI es un registro salva-invocado, por eso en cualquier función hay que salvarlo antes de usarlo — Tema2.3, tr.36

d. EDI es un registro salva-invocado, por eso si se usa hay que salvarlo antes de llamar a una función — no para llamar