Estructura de Computadores
Grado en Ingeniería Informática
6 de Noviembre de 2015

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
d	b	d	d	d

1. En x86_64 se pueden referenciar los registros… p.202 libro (F.3.2), p.307-8 (**F.3.35**), Prác2 Ap.2 p.15, T2.4 tr.4, T2.1 tr.26,43

a. %rax, %eax, %ax, %ab — %ab no existe
b. %rsi, %esi, %si, %sib — %sib no existe
c. %r8q, %r8d, %r8w, %r8b — %r8q no existe, es %r8 a secas
d. %r12, %r12d, %r12w, %r12b — ok

2. En general, gcc Linux x86_64 evitará crear un marco de pila tanto en el invocante como en el invocado, a menos que por algún motivo no se pueda evitar. Alguno de los siguientes motivos no es inevitable:

a. Realizar en el invocante una llamada con más de 6 argumentos — ok, T2.4 tr.7, <=6 -> regs
b. Usar en el invocado una variable local declarada con la palabra reservada volatile — no, T2.4 tr.8, usa zona roja
c. Necesitar usar (conservar) un registro salva-invocado mientras se llama a otra subrutina — ok, T2.4 tr.10, %rsp baja
d. Necesitar usar (conservar) un registro salva-invocante mientras se llama a otra subrutina — ok, mismo motivo

3. Si declaramos `int val[5]={1,5,2,1,3};` entonces… — T2.4, tr.18

a. val[5] es de tipo int y vale 3 — no, val[4]==3
b. val+1 es de tipo int y vale 2 — no, int*, val+1==&val[1]
c. val+4 es de tipo int* y se cumple que *(val+4)==5 — no, *(val+4)==val[4]==3
d. &val[2] es de tipo int* y apunta 8 posiciones por encima de val — ok, T2.4 tr.18

4. Una función C llamada `get_el()` genera el siguiente código ensamblador. Se puede adivinar que…

```
movl  8(%ebp), %eax
leal  (%eax,%eax,4), %eax
addl  12(%ebp), %eax
movl  var(,%eax,4), %eax
```

a. var es un array multi-nivel (punteros a enteros) de cuatro filas
b. var es un array multi-nivel pero no se pueden adivinar las dimensiones
c. var es un array bidimensional de enteros, no se pueden adivinar dimensiones
d. var es un array bidimensional de enteros, con cinco columnas — ok, T2.4 tr.28

5. Al traducir la sentencia C `r->i = val;` gcc genera el código ASM `movl %edx, 12(%eax)`. Se deduce que…

a. r es un puntero que apunta a la posición de memoria 12
b. i es un entero que vale 12
c. val es un entero que vale 12
d. el desplazamiento de i en *r es 12 — ok, T2.4 tr.40