2º Grado Informática
Estructura de Computadores
7 Septiembre 2016

Nombre:	
DNI:	**Grupo:**

Test 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. El switch de gcc para que únicamente compile de lenguaje C a ensamblador, y no realice ningún paso adicional (ensamblar, enlazar, etc), es...

a. -c
b. -S
c. -o
d. -g

2. Los switches --32 y --64 para trabajar en 32bit/64bit corresponden a la herramienta...

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

3. El switch -l para indicar librerías ***NO*** funciona con la herramienta...

a. gcc
b. as
c. ld
d. no se puede marcar una y solo una de las anteriores

4. ¿Cuál de las siguientes no es una sección de un fichero ELF?

a. .text
b. .static
c. .data
d. .bss

5. ¿Cuál de los siguientes contenidos no está incluido en un fichero ELF ejecutable?

a. código máquina
b. variables globales
c. pila del usuario
d. tabla de símbolos

6. En la práctica "media" se programa la suma de una lista de 32 enteros de 4 B para producir un resultado de 8 B, primero sin signo y luego con signo. Si la lista se rellena con el valor que se indica a continuación, ¿en qué caso ambos programas producen el mismo resultado?

a. `0x1111 1111`
b. `0x9999 9999`
c. `0xAAAA AAAA`
d. `0xFFFF FFFF`

7. En la práctica "media" se programa la suma de una lista de 32 enteros de 4 B para producir un resultado de 8 B, primero sin signo y luego con signo. Si la lista se rellena con el valor 0x0400 0000, ¿en qué se diferencian los resultados de ambos programas?

a. no se diferencian
b. en uno ocupa 32 bits, en otro 64 bits
c. en uno se interpreta como negativo, en otro como positivo
d. en uno los 32 bits superiores son 0xFFFF FFFF, en el otro no

8. En la práctica "media" se suma una lista de 32 enteros de 4 B con signo para producir una media y un resto usando la instrucción IDIV. ¿Cuál de las siguientes afirmaciones es falsa?

a. IDIV produce el mismo cociente que el operador / en lenguaje C
b. IDIV produce el mismo resto que el operador % en lenguaje C
c. La media se redondea al entero más próximo
d. El resto siempre tiene el mismo signo que la suma

---

9. En la práctica "media" un estudiante usa el siguiente bucle para acumular la suma en EBP:EDI antes de calcular la media y el resto

```
bucle:
        mov (%ebx,%esi,4), %eax
        cltd
        add %eax, %edi
        adc %edx, %ebp
        jnc nocarry
        inc %edx
nocarry:
        inc %esi
        cmp  %esi,%ecx
        jne bucle
```

Estando bien programado todo lo demás, este código

a. produce siempre el resultado correcto
b. fallaría con lista: .int 0,1,2,3
c. fallaría con lista: .int -1,-2,-4,-8
d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

---

10. Alguno de los siguientes no es un nombre de registro en una máquina IA-32 en modo 32 bits

a. ebp
b. ax
c. dh
d. sil

---

11. Alguno de los siguientes no es un nombre de registro en una máquina x86-64 en modo 64 bits

a. r8d
b. r12w
c. sih
d. spl

---

12. Para comprobar si el entero almacenado en EAX es cero (y posiblemente saltar a continuación usando JZ/JNZ), gcc genera el código

a. cmp %eax, $0
b. test %eax
c. cmp %eax
d. test %eax, %eax

---

13. 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 parity3:

```
int parity3(unsigned* array,
                        int len){
  int  i,res=0,val;
  unsigned x;
  for (i=0; i<len; i++){
    x=array[i];
    val=0;
    do {
      val += x;
      x >>= 1;
    } while (x);
    val &= 0x1;
    res+=val;
    }
  return res;
}
```

Esta función parity3:

a. produce siempre el resultado correcto
b. fallaría con array={0,1,2,3}
c. fallaría con array={1,2,4,8}
d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

14. Un estudiante entrega la siguiente versión de parity4:

```
int parity4(unsigned* array,
                        int len){
  int val,i,res=0;
  unsigned x;
  for (i=0; i<len; i++){
    x=array[i];
    val=0;
    asm("\n"
"ini3:              \n\t"
    "xor  %[x],%[v] \n\t"
    "shr  %[x]      \n\t"
    "test %[x], %[x]\n\t"
    "jne   ini3     \n\t"
    :[v]"+r" (val)
    :[x] "r" (x)
    );
    val = val & 0x1;
    res+=val;
  }
  return res;
}
```

Esta función parity4:

a. produce siempre el resultado correcto
b. fallaría con array={0,1,2,3}
c. fallaría con array={1,2,4,8}
d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

---

15. La sentencia asm() del listado anterior tiene las siguientes restricciones

a. ninguna
b. arquitectura de 32 bits
c. dos entradas y una salida
d. un registro y dos sobrescritos (clobber)

---

16. Un estudiante entrega la siguiente versión de parity5:

```
int parity5(unsigned* array,
                        int len){
  int  i,j,res=0;
  unsigned x;
  for (i=0; i<len; i++){
    x=array[i];
    for (j=sizeof(unsigned)*4;
         j>0; j=j/2){
      x^=x>>j;
    }
    x = x & 0x1;
    res+=x;
  }
  return res;
}
```

Esta función parity5:

a. produce siempre el resultado correcto
b. fallaría con array={0,1,2,3}
c. fallaría con array={1,2,4,8}
d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

---

17. Un estudiante entrega la siguiente versión de parity6:

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

Esta función parity6:

a. produce siempre el resultado correcto
b. fallaría con array={0,1,2,3}
c. fallaría con array={1,2,4,8}

d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

---

18. La sentencia asm() del listado anterior tiene las siguientes restricciones

a. ninguna
b. arquitectura de 32 bits
c. dos entradas y una salida
d. un registro y dos sobrescritos (clobber)

---

19. En el programa "size" de la práctica de la cache, si el primer escalón pasa de tiempo = 1 para todos los tamaños de vector menores o iguales que 32 KB a tiempo = 3 para los tamaños 64 KB y 128 KB, podemos asegurar que:

a. la cache L1 es al menos tres veces más rápida que la cache L2.
b. la cache L1 es como mucho tres veces más rápida que la cache L2.
c. la cache L2 es al menos el doble de rápida que la memoria principal.
d. la cache L2 es como mucho el doble de rápida que la memoria principal.

---

20. El código del programa "size" de la práctica de la cache accede al vector saltando...

a. de byte en byte.
b. de 64 en 64 bytes.
c. de 1 KB en 1 KB.
d. de 64 KB en 64 KB.
