

Arquitectura del Computador - LCC

Examen recuperatorio Parcial 2 - 29 de noviembre de 2024

Nombre y apellido: \_\_\_\_\_

Legajo: \_\_\_\_\_

1. (60 puntos) Dado el siguiente código en lenguaje C:

```
#include <stdio.h>
int a[4]={0xaaaaaaaa, 0x11223344, 16, 4};

int determinante(int a, int b, int c, int d){
    int r = a*d - b*c;
    printf("Determinante: %d * %d - %d * %d = %d\n", a, d, b, c, r);
    return r;
}

int main(){
    int x=a[0], y=a[1], z=a[2], w=a[3];
    printf("Vamos a calcular el determinante de la matriz \n %d \t %d \n %d \t %d \n",x,y,z,w);
    int r=determinante(x,y,z,w);
    return 0;
}
```

- (a) Completar el siguiente código equivalente en Assembler x86-64:

```

.data
a: .long Oxaaaaaaaa, 0x11223344, 16, 4
str: .asciz "Vamos a calcular el determinante de la matriz.\n %d \t %d\n %d \t %d\n"
str1: .asciz "Determinante: %d * %d - %d * %d = %d\n"

.text
.global main
main:
    pushq    %rbp
    movq    %rsp, %rbp
    subq    $32, %rsp
    movl    .d., %eax
    movl    %eax, -4(%rbp)      # a[0]
    movl    .d+4, %eax
    movl    %eax, -8(%rbp)      # a[1]
    movl    .d+8, %eax
    movl    %eax, -12(%rbp)     # a[2]
    movl    .d+12, %eax
    movl    %eax, -16(%rbp)     # a[3]
    movl    -4(%rbp), %esi
    movl    -8(%rbp), %edx
    movl    -12(%rbp), %ecx
    movl    -16(%rbp), %eax
    leaq    str, %rdi
    xorl    %eax, %eax
    call    printf
    movl    -4(%rbp), %edi
    movl    -8(%rbp), %esi

```

# rsp = 0x7fffff f e b 4 0 ✓

# rsp = 0x7fffff f f f f f f f f e b 2 0 ✓

# rsp = 0x7fffff e b 2 0 ✓

↳ (después de ejecutar p)

```

movl -12(%rbp), %edx
movl -16(%rbp), %ecx
call determinante
pushq %rdx
movl %eax, %edx
leave
ret

determinante:
pushq %rbp
movq %rsp, %rbp
subq $32, %rsp
movl %rdi, -4(%rbp)
movl %esi, -8(%rbp)
movl %edx, -12(%rbp)
movl %ecx, -16(%rbp)
movl -4(%rbp), %eax
imull -16(%rbp), %eax
movl %eax, %edx
movl -8(%rbp), %eax
imull -12(%rbp), %eax
subl %eax, %edx
movl %edx, -20(%rbp)
movl -4(%rbp), %esi
movl -16(%rbp), %edx
movl -8(%rbp), %ecx
movl -12(%rbp), %r3d
movl -20(%rbp), %r9d
leaq str1,%rdi
xori %eax, %eax
call printf
movl -20(%rbp), %eax
leave
ret

```

# rsp = 0x7fffffff2648 ✓

↳ leave vuelve al %rsp  
al %rbp y por ende al %rbp  
orig. dejando al %rsp  
donde estaba al llegar a  
main

# rsp = 0x7fffffff2610 ✓

# rsp = 0x7fffffff26f0 ✓

0x000000002222222228 ✓ (el resultado es truncado)

# rx = ..... , rdx = 0x000000000000000010 (c)

# rx = ..... , rdx = 0x000000002222222228 ✓

0x00000001233140 ✓

# a  
# d  
# b  
# c  
# r

↳ resultado producto  
anterior

(res.  
truncado )

# rsp = 0x7fffffff26f0 ✓

# rsp = 0x7fffffff2618 ✓

- (b) Mostrar cómo se guarda el arreglo de enteros en memoria, asumiendo que la etiqueta a corresponde a la dirección 0x404028.
- (c) Elaborar un diagrama de la pila que detalle el contenido de la memoria junto con las direcciones correspondientes, asumiendo un valor inicial de rsp=0x7fffffff2648.
- (d) Indicar el valor del registro **rsp** tras la ejecución de las instrucciones correspondientes al comentario.
- (e) Especificar el contenido de los registros **rax** y **rdx** después de la ejecución de las instrucciones **imull**.
- (f) Justificar el uso de la instrucción **subq \$32, %rsp** en la función **determinante**. ¿Por qué no se usó **subq \$20, %rsp**?
- (g) Analizar y justificar si el resultado obtenido será correcto.

2. (40 puntos) Escribir un programa en ensamblador x86-64 que calcule e imprima en pantalla el promedio de las notas ingresadas como argumentos por línea de comandos. Las notas deben ser números enteros y el promedio será un valor de tipo **double**. Finalmente, el programa debe retornar el valor “0”.

Si se ingresan menos de dos argumentos, el programa debe imprimir en pantalla el mensaje “Ingrese al menos una nota” y retornar el valor “1”.

Ayuda: La función **atoi** (ASCII to Integer) en C permite convertir una cadena de caracteres (de tipo **const char\***) que representa un número en formato de texto a un valor de tipo **int**. Su prototipo es:

```
int atoi(const char *str);
```

Ejercicio	1	2	Total
Puntos	60	40	100
Puntos obtenidos			