R-222 Arquitectura del Computador - LCC

Examen final condición regular

Mesa 29 de febrero de 2024

Nombre y apellido: _____

Legajo: _____

IMPORTANTE: Justificar las respuestas e indicar de manera clara y detallada los procedimientos realizados para llegar a las soluciones.

1. (25 puntos) Dado el siguiente código en Assembler x86-64:

```
.data
a:      .byte -99, 57
f:      .float 7.5, 3.5
str:    .asciz "%f %d\n"

.text
.global main
main:
        pushq   %rbp                # rbp = ....., rsp = .....
        movq    %rsp, %rbp          # rbp = ....., rsp = .....
        subq    $32, % rsp          # rsp = .....
        movl    a, %eax             # rax = .....
        movl    %eax, -4(%rbp)      # rbp = .....
        movl    a+4, %eax           # rax = .....
        movl    %eax, -8(%rbp)      # rbp = .....
        movq    f, %rax             # rax = .....
        movq    %rax, -16(%rbp)     # rbp = .....
        movq    f+4, %rax           # rax = .....
        movq    %rax, -24(%rbp)     # rbp = .....
        COMPLETAR
        call    calculo             # ax = ....., rsp = .....
        COMPLETAR
        call    printf
        popq    %rbp                # rbp = ....., rsp = .....
        movq    %rbp, %rsp          # rbp = ....., rsp = .....
        addq    $32, % rsp          # rsp = .....
        xorl    %eax, %eax          # rax = .....
        ret

calculo:
        pushq   %rbp                # rbp = ....., rsp = .....
        movq    %rsp, %rbp          # rbp = .....
        COMPLETAR
        xorl    %eax, %eax          # rax = .....
        movq    %rbp, %rsp          # rsp = .....
        popq    %rbp                # rbp = ....., rsp = .....
        ret
```

(a) Realizar un esquema para mostrar la ubicación en memoria de las variables. Suponer que la etiqueta `a` corresponde a la dirección de memoria `0x404028`. Mostrar los valores en hexadecimal.

(b) Realizar un esquema mostrando el estado de la pila antes de ejecutarse el epílogo de la función `calculo`. Suponer como valores iniciales `rsp=0x7fffffffebc8` y `rbp=0x1` al comenzar la ejecución del programa.

(c) Indicar el valor de los registros señalados en los comentarios luego de ejecutarse las instrucciones correspondientes.

(d) Completar el código en los lugares donde dice COMPLETAR de manera tal que se imprima el resultado de sumar los dos datos tipo `float` almacenados a partir de la etiqueta `f` y los datos tipo `char` a partir de `a`.

(e) Indicar el valor de las banderas `CF` y `OF` luego de ejecutarse la instrucción que realiza la suma $-99 + 57$

2. (25 puntos) Dado el siguiente programa en lenguaje C:

```c
#include <stdio.h>

float vector1[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
float vector2[5] = {6.6, 7.7, 8.8, 9.9, 10.0};
float resultado[5];


void suma_vectores(float vector1[], float vector2[], float resultado[], int longitud) {
    for (int i = 0; i < longitud; i++){
        resultado[i] = vector1[i] + vector2[i];
    }
}

int main() {

    suma_vectores(vector1, vector2, resultado, 5);

    printf("Resultado de la suma:\n");
    for (int i = 0; i < 5; i++) {
        printf("%.2f ", resultado[i]);
    }
    printf("\n");
    return 0;
}
```

(a) Escribir un código equivalente en Assembler x86-64.

(b) Realizar el procedimiento para realizar la primera de las sumas (es decir, $1.1 + 6.6$) en formato IEEE 754.

(c) Indicar el error cometido al realizar la suma debido al formato de representación (se puede dejar expresado).

3. (25 puntos) Dado el siguiente código en lenguaje C que llama a la función `promedio`, la cual realiza el promedio de tres valores tipo `float` e imprime el resultado:

```c
#include <stdio.h>

float promedio(float a, float b, float c){
    return (a+b+c)/3;
}

int main(){
    float a = 4095, b = 80, c;
    c=a*b;
    printf("El resultado es: %f\n", promedio(a,b,c));
    return 0;
}
```

(a) Escribir un código código equivalente en Assembler ARM.

(b) Dibujar el estado de la pila antes de ejecutarse el prólogo de la función `promedio`.

4. (25 puntos) Se tiene un sistema de memoria virtual con paginación de dos niveles que puede direccionar un total de 2^{32} bytes. El primer nivel tiene 12 bits en el primer nivel y 10 bits en el segundo nivel. Tiene 1 GB de memoria principal con páginas virtuales y físicas de 1 KB.
   (a) Indicar cómo están compuestas las direcciones virtuales y cuántos bits tienen.
   (b) Indicar cómo están compuestas las direcciones físicas y cuántos bits tienen.
   (c) ¿Cuál es el número máximo de páginas virtuales y de páginas físicas en el sistema?
   (d) Indicar el tamaño de la TLB, de la tabla de páginas de primer nivel y de las tablas de páginas de segundo nivel (redondear a un número entero de bytes).
   (e) Atendiendo al contenido de la TLB y de las tablas de páginas de un proceso mostradas a continuación, indicar la dirección de memoria accedida para las siguientes direcciones virtuales:
      1. 0xffb00b18
      2. 0xffffc78f
      3. 0xffbff79f

TLB

V	D	Nro. Pág. Virtual	Nro. Pág. Física
0	0		
1	0	0x3ffff0	0x600
0	0		
1	0	0x3ffff1	0x5ff
0	0		

Tabla de páginas de primer nivel

	V	Dirección
0xfff	0	
	0	
	0	
	0	
	1	0x1400
	:	
	1	0x1000
	0	
	0	
	1	0x1800
0x0	0	

Tablas de páginas de segundo nivel

	V	D	Nro. Página física
0x1bff	0	0	
	0	0	
	1	0	0x7ffff
	:		:
	1	0	0x7fffe
	0	0	
	0	0	
0x1800	1	0	0x7fff0

	V	D	Nro. Página física
0x17ff	0	0	
	1	0	0x1df
	0	0	0x1de
	:		:
	0	0	
	1	0	0x1d0
	0	0	
0x1400	0	0	

	V	D	Nro. Página física
0x13ff	0	0	
	1	1	0x5f0
	0	0	
	:		:
	0	0	
	0	0	
	1	0	0x600
0x1000	0	0	

Ejercicio	1	2	3	4	Total
Puntos	25	25	25	25	100
Puntos obtenidos					