R-222 Arquitectura del Computador - LCC

Examen final condición regular

Mesa 18 de abril 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:      .int -128, 127, 50, -25
f:      .float 7.25, 3.125
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 = .....
    movl    a+8, %eax           # rax = .....
    movl    %eax, -12(%rbp)     # rbp = .....
    movl    a+12, %eax          # rax = .....
    movl    %eax, -16(%rbp)     # rbp = .....
    movq    f, %rax             # rax = .....
    movq    %rax, -20(%rbp)     # rbp = .....
    movq    f+4, %rax           # rax = .....
    movq    %rax, -24(%rbp)     # rbp = .....
    COMPLETAR1
    call    calculo
    COMPLETAR2
    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 = .....
    COMPLETAR3
```

```
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 globales. Suponer que la etiqueta `a` corresponde a la dirección de memoria `0x404028`. Mostrar los valores en hexadecimal.

(b) Realizar la suma de los valores tipo `float` del arreglo `f`.

(c) Indicar el valor de las banderas `CF` y `OF` luego de ejecutarse la instrucción que realiza la suma `-128+127`. Explicar el resultado mostrando el resultado en la recta real, mostrando el rango de valores representables.

(d) Completar el código en los lugares donde dice COMPLETAR de manera tal que se imprima el resultado de sumar los datos tipo `int` a partir de `a`.

(e) Indicar el valor de los registros señalados en los comentarios luego de ejecutarse las instrucciones correspondientes. Suponer como valores iniciales `rbp=0x1` y `rsp=0x7fffebb8`.

(f) Realizar el diagrama de pila.

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

```c
#include <stdio.h>

int a[5] = {1, 2, 3, 4, 5};
int b[5] = {6, 7, 8, 9, 10};

void suma(int a[], int b[], int r[], int l) {
    for (int i=0; i<l; i++){
        r[i] = a[i] + b[i];
    }
}

int main() {
    int l = 5, r[5];
    suma(a, b, r, l);

    printf("Resultado de las sumas por componentes:\n");
    for (int i = 0; i < l; i++) {
        printf("%d ", r[i]);
    }
    printf("\n");
    return 0;
}
```

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

(b) Dibujar el estado de la pila antes de ejecutarse el epílogo de la función `suma`.

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

```c
#include <stdio.h>

double a[] = {4095.0, 80.0, 50.0, 1020.0, 540.0};

double promedio(double a[], int n){
    printf("Calculando el promedio...\n");
    double suma=0;
    for(int i=0; i<n; i++)
        suma+=a[i];
    return suma/n;
}
```

```
int main(){
    int n=5;
    printf("El resultado es: %f\n", promedio(a, n));
    return 0;
}
```

Escribir un código código equivalente en Assembler ARM.

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 el segundo nivel 10 bits. Tiene 8 GB de memoria principal con páginas virtuales y físicas de 1 KB.
   (a) ¿Cuántos bits tienen las direcciones virtuales?
   (b) Indicar cómo están compuestas la direcciones virtuales.
   (c) ¿Cuántos bits tienen las direcciones físicas?
   (d) Indicar cómo están compuestas las direcciones físicas.
   (e) ¿Cuál es el número máximo de páginas virtuales en el sistema?
   (f) ¿Cuál es el número máximo de páginas físicas en el sistema?
   (g) 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. `0xffbffb18`
      2. `0x7fb8f`
      3. `0x2ffb9f`

TLB

V	D	Nro. Pág. Virtual	Nro. Pág. Física
0	0		
1	0	0x3ffff0	0x600
0	0		
1	0	0x3ffff1	0x5fe
1	0	0xffff2	0x5ff

Tabla de páginas de primer nivel

	V	Dirección
0xfff	0	
0xffe	0	
0xffd	0	
0xffc	0	
0xffb	1	0x1400
:	:	
0x5	1	0x1000
0x4	0	
0x3	0	
0x2	1	0x1800
0x1	0	
0x0	0	

Tablas de páginas de segundo nivel

	V	D	Nro. Página física
0x1bff	0	0	
0x1bfe	0	0	
0x1bfd	1	0	0x7ffff
:	:		:
0x1803	1	0	0x7fffe
0x1802	0	0	
0x1801	0	0	
0x1800	1	0	0x7fff0

	V	D	Nro. Página física
0x17ff	0	0	
0x17fe	1	0	0x1df
0x17fd	0	0	0x1de
:	:		:
0x1403	0	0	
0x1402	1	0	0x1d0
0x1401	0	0	
0x1400	0	0	

	V	D	Nro. Página física
0x13ff	0	0	
0x13fe	1	1	0x5f0
0x13fd	0	0	
:	:		:
0x1003	0	0	
0x1002	0	0	
0x1001	1	0	0x600
0x1000	0	0	

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