R-222 Arquitectura del Computador - LCC

Examen final condición regular

16 de febrero de 2023

Nombre y apellido: _____

Legajo: _____

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

1. (30 puntos) Dado el siguiente código en lenguaje C:

```
#include <stdio.h>
int a = 0xf0000000;

int foo1(){
    int b=1;
    a=a+(b<<28);
    return a;
}

int foo2(){
    int b=1;
    a=a+(b<<28);
    return a;
}

int main(){
    printf("0x%x 0x%x\n ", foo1(), foo2());
    return 0;
}
```

   (a) Indicar lo que se imprime en pantalla al ser ejecutado.

   (b) Indicar el valor de las banderas `CF` y `OF` luego de realizarse cada una de las sumas.

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

   (d) Mostrar en un diagrama la evolución de la pila, indicando los valores almacenados y los valores que va tomando el registro de pila (`rsp`).

2. (20 puntos) Dado el siguiente código en Lenguaje C:

```
#include <stdio.h>
struct alumno {
    char nombre[10];
    char apellido[10];
    int nota_1, nota_2, nota_3, nota_4;
    float promedio;
};
```

```
struct alumno a1 = {"Juan", "Perez", 9, 10, 7, 5, 7.75};

int main()
{
    printf ("Nombre: %s\n Apellido: %s\n Promedio: %f\n", a1.nombre, a1.apellido, a1.promedio);
    return 0;
}
```

(a) Dibujar el esquema de memoria, indicando los valores almacenados y las direcciones de memoria. Asumir que la estructura `a1` está alojada a partir de la dirección `0x404040`.

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

3. (30 puntos) Dado el siguiente código en C:

```
#include<stdio.h>
#include <stdlib.h>

int a[]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int suma(int n);

int main(int argc, char *argv[]){
    if(argc<2){
        printf("Ingrese un entero del 1 al 10\n");
        return 9;
    }
    else
    suma(atoi(argv[1]));
    return 0;
}

int suma(int n){
    int s=0;
    for (int i=0; i<n; i++){
        s=s+a[i];
    }
    printf("La suma de los %d enteros es %d\n", n, s);
    return 0;
}
```

(a) Indicar lo que se imprime si se compila y se ejecuta de la siguiente manera:

```
$ gcc -o main main.c
$ ./main 5
```

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

4. (20 puntos) Dado el siguiente código en Assembler x86-64:

```
                            .global main
                            main:
0x0000555555555125 <+0>:        movq    $0x100,%rdi
0x000055555555512c <+7>:        movq    $0x1122334455667788, %rsi
0x0000555555555136 <+17>:       pushq   %rdi                            # <--- %rdi=?
0x0000555555555137 <+18>:       pushq   %rsi                            # <--- %rsi=?
0x0000555555555138 <+19>:       call   foo
0x000055555555513d <+24>:       popq    %rdi
0x000055555555513e <+25>:       popq    %rsi
0x000055555555513f <+26>:       ret
```

```
                          foo:
0x0000555555555140 <+0>:     pushq   %rbp
0x0000555555555141 <+1>:     movq    %rsp,%rbp
0x0000555555555144 <+4>:     movq    0x8(%rbp), %rbx              # <--- %rbx=?
0x0000555555555148 <+8>:     movl    0x14(%rbp), %eax             # <--- %eax=?
0x000055555555514b <+11>:    movq    0x18(%rbp), %rdi             # <--- %rdi=?
0x000055555555514f <+15>:    movq    %rbp,%rsp
0x0000555555555152 <+18>:    popq    %rbp
0x0000555555555153 <+19>:    ret
```

(a) Escribir el contenido de la pila luego de ejecutarse el prólogo de la función `foo`, suponiendo como estado inicial `rbp=0x7fffffffe088` y `rsp=0x7fffffff0000`.

(b) Indicar el valor de los registros `rax`, `rdi` y `rsi` en hexadecimal luego de ejecutarse las líneas correspondientes a sus comentarios.

(c) Indicar el valor de retorno del programa si luego de ser ejecutado se ejecuta el comando `echo $?`.

5. (20 puntos) Dado el siguiente código en C:

```
#include <stdio.h>
int suma(int a, int b, int c, int d, int e);
int d=4, f=0;

int main(){
    int a=1, b=2, c=3, e=5;
    printf("La suma es %d\n", suma(a, b, c, d, e));
    return 0;
}

int suma(int a, int b, int c, int d, int e){
    f=a+b+c+d+e;
    if (f>20)
        return 999;
    else
        return f;
}
```

Escribir un código equivalente en Assembler ARM.

6. (20 puntos) Dados los siguientes números:

- $A = (256)_{10}$
- $B =$ `0x43C00001` (representación del número B en formato norma IEEE 754 simple precisión)
- $C = (1024)_{10}$

(a) Realizar el procedimiento para realizar la operación $(A + B + C) \times 8$ en punto flotante simple precisión IEEE 754, mostrando el resultado en formato hexadecimal.

(b) Indicar si el resultado es correcto. En caso negativo, indicar el error cometido.

(c) Indicar si el resultado de la operación $A \times 8 + B \times 8 + C \times 8$ es el mismo que el obtenido en el punto (a). En caso negativo, indicar en cuál de los cálculos se produce mayor error.

(d) Convertir el resultado obtenido en el punto (a) a doble precisión .

7. (30 puntos) Dado el siguiente código en lenguaje C que realiza el producto de dos enteros:

```
#include <stdio.h>
int main(){
    int a=4095, b=80;
    int c=a*b;
    return 0;
}
```

(a) Completar el siguiente código equivalente en Assembler ARM:

```
.global main:
    push    {r7}
    sub     sp, sp, #20
    add     r7, sp, #0
    ......  r3, #4095
    ......  r3, [r7, #12]
    ......  r3, #80
    ......  r3, [r7, #8]
    ......  r3, [r7, #12]
    ......  r2, [r7, #8]
    ......  r3, r2, r3
    ......  r3, [r7, #4]
    ......  r3, #0
    ......  r0, r3
    ......  r7, r7, #20
    ......  sp, r7
    ......  r7, [sp], #4
    bx      lr
```

(b) Dibujar el estado de la pila.

8. (30 puntos) Un computador posee un sistema de memoria virtual implementada mediante paginación que utiliza páginas de 4 KB. El computador proporciona un espacio de memoria virtual de 2^{32} bytes y tiene 2^{23} bytes de memoria física. Si la tabla de páginas correspondiente a un proceso en ejecución es la siguiente:

Bit de presencia	Bit de modificado	Número de página virtual	Número de marco de página
1	0	0	1
0	0	1	7
1	1	2	9
1	0	3	14
1	0	4	8
1	1	5	3
0	0	6	25
0	1	7	16
0	0	8	23
1	0	9	78

(a) Indique el formato de la dirección virtual, indicando los campos y el número de bits de los mismos.

(b) ¿Cuánta memoria principal ocupa este proceso?

(c) ¿Cuál es el tamaño de la memoria principal?

(d) Indique la dirección física correspondiente a la dirección virtual `0x0000308A`.

(e) Indique la dirección física correspondiente a la dirección virtual `0x0000508A`.

(f) ¿Qué cantidad de memoria puede ser necesario transferir desde/hacia el disco en el caso de un fallo de página?

Ejercicio	1	2	3	4	5	6	7	8	Total
Puntos	30	20	30	20	20	20	30	30	200
Puntos obtenidos									