Arquitectura del Computador - LCC

Examen final

Mesa 12 de septiembre 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. (20 puntos) Dado el siguiente código en Assembler x86-64:

```
.data
a:  .long   1, 2, 3, 4, 5, 6, 7, 8, 9
    .zero   4
b:  .ascii  "abcde"
msg: .string "Hola mundo"
LC0: .string "%p  %d %d\n"
LC1: .string "%p  %c %c\n"

.text
.global main
main:
    pushq   %rbp                # rsp = ....................., rbp = .....................
    movq    %rsp, %rbp          # rsp = ....................., rbp = .....................
    movl    a+16, %edx          # rdx = .....................
    movl    a, %eax             # rax = .....................
    movl    %edx, %ecx          # rcx = .....................
    movl    %eax, %edx          # rdx = .....................
    movl    $a, %esi            # rsi = .....................
    movl    $LC0, %edi          # rdi = .....................
    movl    $0, %eax            # rax = .....................
    call    printf              # rax = .....................
    movl    %edx, %ecx          # rcx = .....................
    movl    %eax, %edx          # rdx = .....................
    movl    $b, %esi            # rsi = .....................
    movl    $LC1, %edi          # rdi = .....................
    movl    $0, %eax            # rax = .....................
    call    printf              # rax = .....................
    movl    $msg, %edi          # rdi = .....................
    call    printf              # rax = .....................
    movl    $0, %eax            # rax = .....................
    popq    %rbp                # rsp = ....................., rbp = .....................
    ret
```

   (a) Realizar un esquema para mostrar la ubicación en memoria de las variables. Suponer que la dirección de memoria correspondiente a la etiqueta `a` es `0x404030`.

   (b) Completar los comentarios con el valor resultante luego de ejecutar cada una de las instrucciones. Asumir los siguientes valores iniciales: `rsp = 0x7fffffffeb88` y `rbp = 0x1`.

(c) Indicar lo que se imprime en pantalla al ejecutar cada una de las instrucciones `call printf`.

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

```
#include <stdio.h>

int main(int argc, char *argv[]) {
    if (argc < 5){
        printf("Número insuficiente de argumentos\n");
    }
    else {
        printf("Número total de argumentos: %d\n", argc);
        printf("Argumentos ingresados:\n");
        for (int i = 1; i < argc; i++) {
            printf("Argumento %d: %s\n", i, argv[i]);
        }
    }
    return 0;
}
```

(a) Indicar qué se imprime si se ejecuta de la siguiente manera:
`./a.out uno dos tres cuatro`

(b) Escribir un código equivalente en Assembler x86-64 que imprima los mismos resultados.

3. (20 puntos) Dado el siguiente código en Assembler ARM:

```
.data
numbers: .word 3, 6, 9, 12, 15

.text
.global main
main:
    LDR R0, =numbers
    MOV R1, #5
    MOV R2, #0
    MOV R3, #0

loop:
    LDR R4, [R0, R3, LSL #2]
    ADD R2, R2, R4
    ADD R3, R3, #1
    CMP R3, R1
    BNE loop
    BX LR
```

(a) Describe lo que hace este código ensamblador, línea por línea. ¿Cuál es el resultado final en el registro `R2` al finalizar la ejecución?

(b) Modifica el código para que calcule el valor máximo entre los números e imprima dicho valor en pantalla.

4. (15 puntos) Indique si es verdadero o falso, justificando su respuesta, si los siguientes valores inmediatos son válidos en la arquitectura ARM:

A. `0x00108000`

B. `0x00408000`

C. `0x00fff000`

D. `0x00fff00f`

E. `0x00000124`

En los casos positivos, indique cómo es la codificación del valor inmediato. En los casos negativos, indique una forma alternativa para poder cargarlo en un registro.

5. (20 puntos) Se tiene un sistema de memoria virtual con paginación de un nivel que puede direccionar un total de 2^{48} bytes y tiene 4 GB de memoria secundaria. Asumiendo que las páginas virtuales y físicas tienen un tamaño de 4 KB cada una:
   (a) ¿Cuántos bits tiene la dirección virtual?
   (b) ¿Cuántos bits tiene la dirección física?
   (c) ¿Cuál es el número máximo de páginas virtuales en el sistema?
   (d) ¿Cuál es el número máximo de páginas físicas en el sistema?
   (e) Suponga que, además del número de página física, cada entrada de la tabla de paginación también contiene información de estado en forma de un bit válido (V) y un bit sucio (D). ¿Cuál es el tamaño total de la tabla de paginación en bytes? Redondear a un número entero de bytes.
   (f) Atendiendo al contenido de la TLB y de la tabla de páginas de un proceso mostradas a continuación, indicar la dirección de memoria accedida para las siguientes direcciones virtuales:
      1. 0x4040fe
      2. 0x405a10
      3. 0x7ffffffec04
      4. 0x7ffffffdb05

TLB

V	D	Nro. pág. virtual	Nro. pág. física
1	0	0x406	0x7000
1	0	0x7fffffffd	0xa000
1	1	0x404	0x1000
1	1	0x405	0x2000
1	0	0x401	0xb000

Tabla de paginación

	V	D	Nro. pág. física
⋮	⋮	⋮	⋮
0x7fffffff	0	0	
0x7ffffffe	1	0	0x4000
0x7ffffffd	0	0	
0x7ffffffc	1	0	0x6000
0x7ffffffb	1	1	0x3000
0x7ffffffa	1	0	0x8000
⋮	⋮	⋮	⋮

Ejercicio	1	2	3	4	5	Total
Puntos	20	25	20	15	20	100
Puntos obtenidos						