4, 6

R-222 Arquitectura del Computador - LCC

Examen Parcial - 4 de noviembre de 2022

Nombre y apellido: [redacted]

Legajo: _____

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

2

1. (10 puntos) Indicar por qué las siguientes instrucciones son inválidas o fallarán al ser ejecutadas (suponer que `a` y `b` están declaradas):
   A. `movq %rax, $a`
   B. `movq a, b`
   C. `movq %rax, 0x4000`
   D. `movq %rax, $0x4000`
   E. `movq 0x404028, %eax`

20

2. (30 puntos) Dado el siguiente código en lenguaje C que recibe un argumento `int x` por teclado e imprime por pantalla el valor escalado `double f = x*b+k`:

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

int imprime(double f);
double scale(int x, float b);
float b = 2.3;

int main(int argc, char* argv[]){
    int x = atoi(argv[1]);
    return scale(x, b);
}

double scale(int x, float b){
    int k = 8;
    double f = x*b+k;
    imprime(f);
    return f;
}

int imprime(double f){
    printf("El valor escalado es %f\n",f);
    return 0;
}
```

   (a) Dibujar un esquema mostrando el estado de la pila luego de entrar a la función `imprime`.
   (b) Escribir un código equivalente en Assembler x86-64.
   Aclaración: La función `atoi` convierte el argumento de entrada tipo cadena en un número entero (tipo `int`):
   `int atoi(const char *str)`

3. (30 puntos) Dado el siguiente código en Assembler x86-64:

```
.data
str:    .long 1, 2, 3, 4
g:      .quad 0x1122334455667788
msg:    .asciz "Hola mundo"

.text
.global main
main:
    movl    str, %eax           # rax=.....
    movl    str+4, %eax         # rax=.....
    movb    str+16, %al         # rax=.....
    movq    $3, %rbx
    movb    str(,%rbx,4), %al   # rax=.....
    leaq    msg, %rax           # rax=.....
    movb    g(, %rbx,1), %al    # rax=.....
    movl    $0, %eax            # rax=.....
    ret
```

(a) Dibujar el esquema de memoria a partir de la etiqueta `str` indicando el contenido de memoria para cada byte. Asumir que la dirección correspondiente a la etiqueta `str` es 0x404028.

(b) Indicar en el esquema de memoria la dirección de memoria correspondiente a cada etiqueta.

(c) Indicar el valor del registro `rax` en cada uno de los comentarios luego de ejecutarse la línea correspondiente.

4. (20 puntos) Dado el siguiente código en lenguaje C:

```
#include<stdio.h>
int fun(){
    static int count = 5;
    count++;
    return count;
}

int main(){
    printf("%d\n", fun());
    printf("%d\n", fun());
    return 0;
}
```

(a) Indicar lo que imprime en pantalla al ser ejecutado y por qué.

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

5. (10 puntos) Dada la siguiente función en lenguaje C:

```
int func(int a, int b, short c, long d, int *e, float f, double g, int h, int k){
    return a+b+h+*e+k;
}
```

Completar el siguiente código equivalente en Assembler x86-64 de manera que resulten códigos equivalentes:

```
func:
    addl    %esi, %edi
    addl    %edx, %edi
    addl    40(%rsp), %edi
    addl    48(%rsp), %edi
    movl    %edi, %edx
    ret
```

Ejercicio	1	2	3	4	5	Total
Puntos	10	30	30	20	10	100
Puntos obtenidos						