# R-222 Arquitectura del Computador - LCC

**Examen final condición regular**

Mesa 22 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 -0xfe, 0xff
f:      .quad 0x1122334455667788, 0xdeadbeef

.text
.global main
main:
        pushq   %rbp                # rbp = ....., rsp = .....
        movq    %rsp, %rbp          # rbp = ....., rsp = .....
        pushq   f                   # rbp = ....., rsp = .....
        pushq   f+8                 # rbp = ....., rsp = .....
        movl    -8(%rbp), %eax      # rax = .....
        movl    -12(%rbp), %ebx     # rbx = .....
        movb    a, %cl              # cl = .....
        cmpb    $0, %cl             # cl = .....
        jg      etiqueta
        xorl    %eax, %eax          # rax = .....
etiqueta:
        movb    a, %dil             # dil = .....
        movb    a+1, %sil           # sil = .....
        call    producto            # ax = ....., rsp = .....
        popq    %rbx                # rbx = ....., rsp = .....
        popq    %rax                # rax = ....., rsp = .....
        movq    %rbp, %rsp          # rsp = .....
        popq    %rbp                # rbp = ....., rsp = .....
        ret

producto:
        pushq   %rbp                # rbp = ....., rsp = .....
        movq    %rsp, %rbp          # rbp = .....
        movb    %dil, %al           # al = .....
        imulb   %sil                # ax = .....
        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`.

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

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

(d) Indicar el valor de retorno luego de ejecutarse el código.

(e) Indicar el valor de las banderas `CF` y `OF` luego de ejecutarse la instrucción `cmpb $0, %cl`

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

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

int main(int argc, char *argv[]) {
    if (argc < 2){
        printf("La cantidad de argumentos es insuficiente\n");
    } else {
        printf("Cantidad de argumentos ingresados: %d\n", argc-1);
        for (int i=1; i<argc; i++){
            printf("Argumento %d: %s\n", i, argv[i]);
        }
    }
    return 0;
}
```

Escribir un código equivalente en Assembler x86-64.

3. (25 puntos) Escribir una función denominada `calculo` en lenguaje ensamblador ARM que realice lo siguiente:

- La función `calculo` recibe cinco argumentos enteros y retorna un entero.
- En caso de que todos los argumentos sean positivos, realiza la suma de los mismos, imprime el resultado de la suma y retorna dicho valor.
- En caso de que alguno de los argumentos sea negativo o nulo, la función `calculo` debe llamar a otra función denominada `foo` que imprime un mensaje de error por pantalla y retornar -1.
- La función `foo` no tiene argumentos de entrada, debe imprimir el mensaje por pantalla "Todos los argumentos deben ser positivos" y retornar -1 a la función `calculo`.

4. (25 puntos) Se tiene un sistema de memoria virtual con paginación de dos niveles que puede direccionar un total de $2^{36}$ bytes. El primer nivel tiene 12 bits en el primer nivel y el segundo nivel 11 bits. Tiene 4 GB de memoria principal con páginas virtuales y físicas de 8 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. 0x7007847
2. 0x2ffdf8a
3. 0x7ffffe002

**TLB**

| V | Nro. Pág. Virtual | Nro. Pág. Física |
|---|---|---|
| 0 | | |
| 1 | 0x3ffff8 | 0x600 |
| 0 | | |
| 1 | 0x3fffff | 0x5ff |
| 0 | | |

**Tabla de páginas de primer nivel**

| | V | Dirección |
|---|---|---|
| 0xfff | 0 | |
| | 0 | |
| | 0 | |
| | : | : |
| | 1 | 0x2000 |
| | 0 | |
| | 1 | 0x1000 |
| | 0 | |
| | 0 | |
| | 1 | 0x1800 |
| | | |
| 0x0 | 0 | |

**Tablas de páginas de segundo nivel**

| | V | Nro. Página física |
|---|---|---|
| 0x27ff | 0 | |
| | 0 | |
| | 1 | 0x2ff |
| | : | : |
| | 1 | 0x2f0 |
| | 0 | |
| | 0 | |
| 0x2000 | 1 | 0x2fa |

| | V | Nro. Página física |
|---|---|---|
| 0x1fff | 0 | |
| | 1 | 0x1df |
| | 0 | |
| | : | : |
| | 0 | |
| | 1 | 0x1d0 |
| | 0 | |
| 0x1800 | 0 | |

| | V | Nro. Página física |
|---|---|---|
| 0x17ff | 0 | |
| | 1 | 0x5f0 |
| | 0 | |
| | : | : |
| | 0 | |
| | 0 | |
| | 1 | 0x600 |
| 0x1000 | 0 | |

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