



[← VOLVER A MIS CURSOS](#)

Mis cursos > [2024-1 ARQUITECTURA DE COMPUTADORAS \(1IEE14-0621\)](#) > Laboratorios  
> [LAbs 4 - Prueba de entrada \(0622\)](#)

## 2024-1 ARQUITECTURA DE COMPUTADORAS (1IEE14-0621)

Comenzado el	viernes, 19 de abril de 2024, 19:10
Estado	Finalizado
Finalizado en	viernes, 19 de abril de 2024, 19:28
Tiempo empleado	18 minutos 2 segundos
Calificación	4,00 de 5,00 (80%)

### Pregunta 1

Correcta  
Se puntúa 1,00 sobre 1,00

¿Cuál de las siguientes instrucciones convierte un número con representación en punto flotante de alta precisión a una de baja precisión?

- ☐ a. cvtss2sd
- ☒ b. cvtsd2ss
- ☐ c. cvtsi2sd
- ☐ d. cvtsd2si
- ☐ e. Ninguna de las anteriores



Respuesta correcta  
La respuesta correcta es:  
cvtsd2ss

## Pregunta 2

Correcta

Se puntúa 1,00 sobre 1,00

En la siguiente función:

```
void my_function(char a, short b, float c, double *d, double e)
```

¿A qué registro del CPU irá la variable d?

- ☐ a. xmm1
- ☐ b. xmm2
- ☐ c. xmm3
- ☐ d. rcx
- ☒ e. rdx



Respuesta correcta

Las respuestas correctas son:

xmm2,

rdx

## Pregunta 3

Incorrecta

Se puntúa 0,00 sobre 1,00

¿Cuál es la función de la bandera -c?

```
gcc -Os -c mul_esc_vec.c -o mul_esc_vec.o
```

- ☐ a. Permite usar el depurador gdb
- ☐ b. Permite generar el object file para desensamblar la función
- ☐ c. Optimiza el código para obtener un código ASM más compacto
- ☐ d. b y c
- ☒ e. Ninguna de las anteriores



Respuesta incorrecta.

La respuesta correcta es:

Permite generar el object file para desensamblar la función

Pregunta 4

Finalizado

Se puntúa 2,00 sobre 2,00

Se le brinda el código desensamblado de funcionX:

¿Qué realiza esta función?

```
funcionX.o:      file format elf64-x86-64

Disassembly of section .text:

0000000000000000 <funcionX>:
  0:  f3 0f 1e fa      endbr64
  4:  31 c0            xor    eax,eax
  6:  0f 57 c0        xorps  xmm0,xmm0
  9:  39 c2            cmp    edx,eax
  b:  7e 13            jle    20 <funcionX+0x20>
  d:  f3 0f 10 0c 87  movss  xmm1,DWORD PTR [rdi+rax*4]
 12: f3 0f 59 0c 86  mulss  xmm1,DWORD PTR [rsi+rax*4]
 17: 48 ff c0        inc    rax
 1a: f3 0f 58 c1      addss  xmm0,xmm1
 1e: eb e9            jmp    9 <funcionX+0x9>
 20: f3 0f 11 01      movss  DWORD PTR [rcx],xmm0
 24: c3              ret
```

[Ver Condiciones](#) [Reportar abuso](#)

Se limpia el registro `eax` y `xmm0` (se utilizan diferentes instrucciones ya que una es para flotantes y otra para enteros), se compara el valor de `edx` con `eax` y si el valor de `edx` <= `eax`, salta a la dirección de memoria indicada para después mover el valor del registro `xmm0` al puntero del primer elemento del registro `rcx`, si no se cumple la condición se mueve el valor del puntero del registro `rdi` al registro `xmm1`, para luego multiplicar el valor de `xmm1` con el puntero del registro `rsi`, se aumenta el valor de `rax` en 1 para ir cambiando la posición de elemento del puntero, se suma el valor de `xmm0` con el de `xmm1` y luego se hace un salto sin condición para evaluar si `edx` <= `eax`.

En conclusión lo que realiza esta función es multiplicar elemento por elemento de dos arreglos (ubicados en los registros `rsi` y `rdi`) en cada iteración e ir sumando el resultado, al final se tendrá la suma total en el primer elemento del arreglo del registro `rcx`.

Comentario:

◀ Lab3 - Parte Práctica (0622)

Ir a...

Lab4 - Parte Práctica (0622) ▶