

Nombre: Dai, Natalia

Grupo: 43

Nombre: Pérez Castillo, Pol

Hoja de respuesta al Estudio Previo

1. Explica en detalle qué hacen las siguientes instrucciones. Si están sintácticamente mal escritas, indica el motivo:

`movl $1, %eax`

Mueve un 1 longword al registro %eax.

`movl 1, %eax`

Mueve el longword de la posición de memoria 1 al registro %eax.

`movl $1, eax`

Está mal porque el registro debe tener %.

`movl 1, eax`

Está mal porque el registro debe tener %.

2. Dada la siguiente secuencia de instrucciones:

`cmpl %eax, %ebx`

`jge fin`

¿en qué condiciones se efectúa el salto?

Cuando el longword guardado en %ebx sea mayor o igual al longword guardado en %eax.

3. Dada la siguiente secuencia de instrucciones:

`cmpl %ebx, %eax`

`jge fin`

¿en qué condiciones se efectúa el salto?

Cuando el longword guardado en %eax sea mayor o igual al longword guardado en %ebx.

4. La traducción a código ensamblador del código C es:

```
        movl $0, i
for:    cmpl $100, i
        jge fi-for
        cmpl b, a
        jge fi-if
if:     movl b, %eax
        subl a, %eax
        idivl $2
        addl %eax, a
```

```
fi-if:  incl i
        jmp for
fi-for:
```

5. La traducción a código C del código ensamblador es:

```
do {  
    if (a != b) {  
        b -= a; a = b; a = 2 * a + b;  
    }  
    --i;  
} while (i > 0);
```

6. Explica cuál es la diferencia entre un *breakpoint* y un *watchpoint*.

Un breakpoint se para en ese punto de código que marques mientras que un watchpoint se para cuando el valor de algo es modificado

7. Explica cómo se puede hacer en *ddd* que un *breakpoint* sólo se pare cuando se ha ejecutado 15 veces.

Source → Breakpoints → Edit properties → Ignore Count = 15

8. Explica cómo se puede hacer en *ddd* para modificar el contenido de una dirección de memoria.

Selecciono una variable y se pulsa Set para modificar el contenido.