

Nombre: Héctor Castaños

Grupo: 11

Nombre: Alicia Pallarol

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`

MOVE LONG WORD → pone el literal 1 en el registro %eax.

`movl 1, %eax`

MOVE LONG WORD → pone a lo que apunta la dirección de memoria 1 en el registro %eax.

`movl $1, eax`

Está sintácticamente mal ya que el nombre de un registro siempre va precedido del carácter '%',

`movl 1, eax`

Idem anterior.

2. Dada la siguiente secuencia de instrucciones:

`cmpl %eax, %ebx`

`jge fin`

¿en qué condiciones se efectúa el salto?

Salta si el contenido de `%eax < %ebx`.

3. Dada la siguiente secuencia de instrucciones:

`cmpl %ebx, %eax`

`jge fin`

¿en qué condiciones se efectúa el salto?

Salta si el contenido de `%ebx < %eax`.

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

```
    movl $0, i
for: cmpl b, a
    jge endif
    movl b, %eax
    subl a, %eax
    movl $2, %ebx
    idivl %ebx
    addl %eax, a
```

```
endif: incl i
    jmp for
```

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

```
while (i != 0) {  
    if (a != b) {  
        b = b - a;  
        a = 3 * b;  
        --i;  
    }  
}
```

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

El breakpoint detiene el programa en la línea establecida. Mientras que watchpoint detiene el programa según la condición establecida.

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

Una vez establecido el breakpoint → Source → breakpoints → properties. Una vez en properties ignore count = 15.

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

Clicar en la variable y luego en Set.