Esercizio S10L3

Traccia:

Nella lezione teorica del mattino, abbiamo visto i fondamenti del linguaggio Assembly. Dato il codice in Assembly per la CPU x86 allegato qui di seguito, identificare lo scopo di ogni istruzione, inserendo una descrizione per ogni riga di codice. Ricordate che i numeri nel formato 0xYY sono numeri esadecimali. Per convertirli in numeri decimali utilizzate pure un convertitore online, oppure la calcolatrice del vostro computer (per programmatori).

```
0×00001141 <+8>:    mov   EAX,0×20
0×00001148 <+15>:   mov   EDX,0×38
0×00001155 <+28>:   add   EAX,EDX
0×00001157 <+30>:   mov   EBP, EAX
0×0000115a <+33>:   cmp   EBP,0xa
0×0000115e <+37>:   jge    0×1176 <main+61>
0×0000116a <+49>:   mov   eax,0×0
0×0000116f <+54>:   call   0×1030 <printf@plt>
```

Numeri in esadecimale

```
0x00001141 <+8>:    mov   EAX,0x20
0x00001148 <+15>:   mov   EDX,0x38
0x00001155 <+28>:   add   EAX,EDX
0x00001157 <+30>:   mov   EBP, EAX
0x0000115a <+33>:   cmp   EBP,0xa
0x0000115e <+37>:   jge    0x1176 <main+61>
0x0000116a <+49>:   mov   eax,0x0
0x0000116f <+54>:   call   0x1030 <printf@plt>
```

Numeri in decimale

```
0x00001141 <+8>:    mov   EAX,32
0x00001148 <+15>:   mov   EDX,56
0x00001155 <+28>:   add   EAX,EDX
0x00001157 <+30>:   mov   EBP, EAX
0x0000115a <+33>:   cmp   EBP,10
0x0000115e <+37>:   jge    4470 <main+61>
0x0000116a <+49>:   mov   eax,0
0x0000116f <+54>:   call   4144 <printf@plt>
```

1) 0x00001141 <+8>: mov EAX,32
2) 0x00001148 <+15>: mov EDX,56
3) 0x00001155 <+28>: add EAX,EDX 32+56=88
4) 0x00001157 <+30>: mov EBP, EAX
5) 0x0000115a <+33>: cmp EBP,10
6) 0x0000115e <+37>: jge 4470 <main+61>
7) 0x0000116a <+49>: mov eax,0
8) 0x0000116f <+54>: call 4144 <printf@plt>

1) Carica il valore decimale 32 nel registro EAX.
2) Carica il valore decimale 56 nel registro EDX.
3) Fa la somma tra il contenuto di EDX e il contenuto di EAX e memorizza il risultato in EAX.
4) Muove il contenuto di EAX nel registro EBP.
5) Compara il contenuto di EBP con il valore decimale 10.
6) Salta all'indirizzo 4470 se il risultato della comparazione è maggiore o uguale a 10.
7) Muove il valore decimale 0 nel registro EAX.
8) Chiama la funzione printf dall'indirizzo 4144 nella Procedure Linkage Table (PLT).

mov : consente di copiare una variabile o un dato da una locazione ad un'altra.

add: somma b al valore di a e salvare/aggiornare il valore di a con il nuovo valore dopo l'addizione.

cmp: è simile all'istruzione «sub» (=sottrazione), ma a differenza di «sub» non modifica gli operandi ma modifica i flag ZF (zero flag) e CF (carry flag).

jge: Salta alla locazione specificata se la destinazione è maggiore o uguale della sorgente nell'istruzione «cmp».

call: chiama una funzione .