



ESERCITAZIONE

S10 - L3



MICHELANGELO BORROMEO

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).

- 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>

ASSEMBLY

0x00001141 <+8>: mov EAX,0x20

Sposta il valore decimale **32** all'interno di EAX

0x00001148 <+15>: mov EDX,0x38

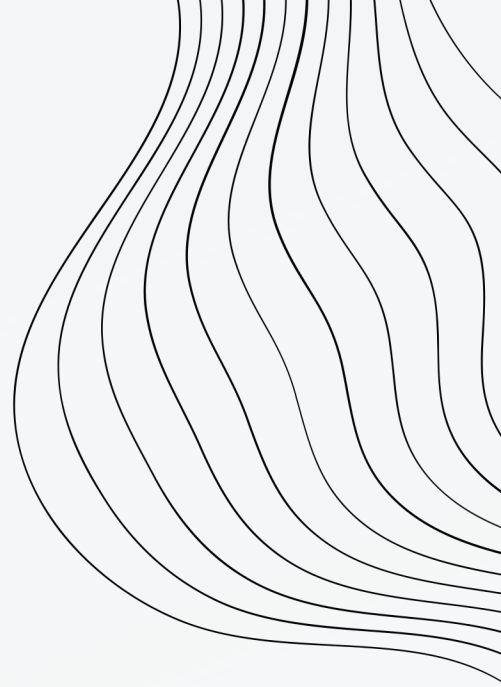
Sposta il valore decimale **56** all'interno di EDX

0x00001155 <+28>: add EAX,EDX

Somma il valore di EAX= **32** con il valore di EDX= **56** e ne aggiorna il contenuto **EAX= 88**

0x00001157 <+30>: mov EBP, EAX

Sposta il contenuto di EAX all'interno di EBP



ASSEMBLY

0x0000115a <+33>: cmp EBP,0xa

Compara se 0xa è uguale al valore di EBP

0x0000115e <+37>: jge 0x1176 <main+61>

Salta alla locazione specificata se la destinazione è maggiore o uguale della sorgente l'esecuzione del programma continuerà a partire dall'indirizzo 0x1176. Altrimenti, il flusso di esecuzione continuerà con l'istruzione successiva immediatamente dopo jge.

0x0000116A <+49>: MOV EAX,0x0

Sposta 0x0 in EAX quindi il valore sarà aggiornato a 0

call 0x1030 <printf@plt>

Chiama la funzione printf