

Assembly x86

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

- Si sposta il valore esadecimale 0x20 (32 in decimale) nel registro EAX.

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

- Si sposta il valore esadecimale 0x38 (56 in decimale) nel registro EDX.

0x00001155 <+28>: add EAX,EDX

- Si aggiunge il valore contenuto in EDX a quello in EAX.

0x00001157 <+30>: mov EBP,EAX

- Si sposta il valore risultante dall'aggiunta di EAX e EDX nel registro EBP.

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

- Si confronta il valore contenuto in EBP con il numero esadecimale 0xa (10 in decimale).

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

- Se il valore in EBP è maggiore o uguale a 10, salta all'indirizzo 0x1176.

0x0000116a <+49>: mov eax,0x0

- Si sposta il valore esadecimale 0x0 (0 in decimale) nel registro EAX.

0x0000116f <+54>: call 0x1030 <printf@plt>

- Viene chiamata la funzione printf dall'indirizzo 0x1030.

Valori esadecimali convertiti in decimale:

- 0x20 in decimale è 32.
- 0x38 in decimale è 56.
- 0xa in decimale è 10.

Le istruzioni in linguaggio assembly, con i valori esadecimali convertiti in decimali, diventano:

- mov EAX, 32
- mov EDX, 56
- add EAX, EDX (che porterà EAX a contenere 88)
- mov EBP, EAX (EBP conterrà anch'esso 88)
- cmp EBP, 10
- jge 0x1176 <main+61> (questa istruzione salterà se il valore in EBP è maggiore o uguale a 10)
- mov EAX, 0
- call 0x1030 <printf@plt> richiamo funzione

