

Assembly x86

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

Sposta il sorgente, ovvero il valore esadecimale 0x20 (32 decimale) nella destinazione, ossia il registro EAX.

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

Carica il valore esadecimale 0x38 (56 decimale) nel registro EDX.

0x00001155 <+28>: add EAX,EDX

Somma i valori dei registri EDX a quelli di EAX, salvando il risultato in EAX. Più precisamente dunque svolge $32+56=88$ e salva 88 nel registro EAX.

0x00001157 <+30>: mov EBP,EAX

Sposta il contenuto del registro EAX, cioè 88, nel registro EBP.

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

Confronta il valore di EBP con il valore esadecimale 0xa (10 decimale). Controlla se 88 è maggiore, minore o uguale a 10.

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

Salta all'indirizzo 0x1176 perché il valore di EBP (88) è maggiore o uguale a 10.

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

Sposta il valore esadecimale 0x0 (0 decimale) nel registro EAX, sovrascrivendo il valore precedente.

0x0000116f <+54> call 0x1020 <printf@plt>

Chiama la funzione printf per stampare il valore di EAX, ovvero 0.

In breve, il codice carica due valori nei registri EAX ed EDX, li somma, salva il risultato in EBP, confronta il valore di EBP con 10 e stampa il valore di EAX se EBP è inferiore a 10.

Appunti:

i registri sono risorse preziose e veloci della CPU, utilizzati per contenere dati e indirizzi temporanei durante l'esecuzione di un programma. In particolare, i registri EAX, EDX e EBP sono tutti registri a 32 bit della CPU x86.

- EAX (Accumulator Register) è un registro generale utilizzato principalmente per le operazioni aritmetiche e logiche. In particolare, è spesso utilizzato per contenere il valore di ritorno di una funzione.
- EDX (Data Register) è un registro generale che viene spesso utilizzato in coppia con EAX per contenere i risultati delle operazioni a 64 bit.
- EBP (Base Pointer) è un registro utilizzato come punto di riferimento per l'accesso alle variabili locali e ai parametri di una funzione. Viene spesso utilizzato per l'implementazione della gestione delle pile.