

S10 L3

Davide Andreozzi

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).

Assembly	Descrizione
0x00001141 <+8>: mov EAX,0x20	0x00001141 <+8> indica l'indirizzo di memoria dell'istruzione. L'istruzione mov EAX,0x20 copia il valore 0x20 (32) nel registro EAX.
0x00001148 <+15>: mov EDX,0x38	L'istruzione va a copiare il valore 0x38 (56) nel registro EDX
0x00001155 <+28>: add EAX,EDX	Aggiunge/Somma il valore contenuto in EDX con il contenuto di EAX e lo «salva» in EAX
0x00001157 <+30>: mov EBP, EAX	Copia il valore contenuto in EAX nel registro EBP
0x0000115a <+33>: cmp EBP,0xa	Confronta il valore nel registro EBP con 0xa (10) sottrae il secondo operando dal primo operando e imposta i flag di stato della CPU in base al risultato. Non salva il risultato ma ne modifica solo il flag (ZF,CF, ecc.)
0x0000115e <+37>: jge 0x1176 <main+61>	Causa un salto (salto condizionale) all'indirizzo 0x1176 < main+61 > se il risultato dell'ultima operazione di confronto (CMP) ha determinato che il primo operando era maggiore o uguale al secondo.
0x0000116a <+49>: mov eax,0x0	Copia il valore 0x0 (0) nel registro EAX
0x0000116f <+54>: call 0x1030 <printf@plt>	Inizia una chiamata alla funzione printf@plt all'indirizzo 0x1030