

Manuel Di Gangi

S10_L3

Assembly x86

27 marzo 2024

INDICE

Traccia.....	2
Sviluppo esercizio.....	3

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).

1. 0x00001141 <+8>: mov EAX,0x20
2. 0x00001148 <+15>: mov EDX,0x38
3. 0x00001155 <+28>: add EAX,EDX
4. 0x00001157 <+30>: mov EBP,EAX
5. 0x0000115a <+33>: cmp EBP,0xa
6. 0x0000115e <+37>: jge 0x1176 <main+61>
7. 0x0000116a <+49>: mov eax,0x0
8. 0x0000116f <+54>: call 0x1030 <printf@plt>

Sviluppo esercizio

ISTRUZIONE	DESCRIZIONE OPERAZIONE
0x00001141 <+8>: mov EAX,0x20	Sposta il valore 32 (conversione decimale di 20) nel registro EAX
0x00001148 <+15>: mov EDX,0x38	Sposta il valore decimale 56 nel registro EDX
0x00001155 <+28>: add EAX,EDX	Somma il registro EAX ed il registro EDX salvando il risultato nel registro EAX. 32 + 56 = 88 - EAX conterrà il valore 88
0x00001157 <+30>: mov EBP,EAX	Sposta il contenuto del registro EAX nel registro EBP
0x0000115a <+33>: cmp EBP,0xa	Compara il contenuto del registro EBP con il valore decimale 10
0x0000115e <+37>: jge 0x1176 <main+61>	Salta alla locazione 0x1176 <main+61> se la destinazione è >= della sorgente nell'istruzione "cmp". Nel nostro caso 88 > 10 quindi effettuo il salto.
0x0000116a <+49>: mov EAX,0x0	Sposta il valore decimale 0 nel registro EAX
0x0000116f <+54>: call 0x1030 <printf@plt>	Chiama la funzione printf allocata all'indirizzo 0x1030