



Assembly: fondamenti



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>



Esercizio

Il valore **0x00001141 <+8>** è un indirizzo di memoria associato a un'istruzione nel codice assembly. In particolare, indica la posizione di memoria in cui inizia l'istruzione corrispondente nel programma. L' **<+8>** indica l'offset rispetto all'inizio della funzione o del blocco di codice corrente.

Quindi, **0x00001141** rappresenta l'indirizzo di memoria dell'ottava istruzione nel codice assembly che stai analizzando. Questo indirizzo può variare a seconda del carico del programma in memoria e dell'indirizzamento specifico del tuo sistema.



Esercizio

1. **mov EAX,0x20**: Carica il valore esadecimale 0x20 (32 in decimale) nel registro EAX.
2. **mov EDX,0x38**: Carica il valore esadecimale 0x38 (56 in decimale) nel registro EDX.
3. **add EAX,EDX**: Aggiunge il contenuto di EDX al registro EAX.
4. **mov EBP, EAX**: Muove il valore risultante in EAX nel registro EBP.
5. **cmp EBP,0xa**: Compara il contenuto di EBP con il valore esadecimale 0xa (10 in decimale).
6. **jge 0x1176 <main+61>**: Salta a 0x1176 (offset relativo rispetto all'inizio della funzione main) se il confronto precedente è maggiore o uguale (Greater or Equal).
7. **mov eax,0x0**: Muove il valore esadecimale 0x0 (0 in decimale) nel registro EAX.
8. **call 0x1030 <printf@plt>**: Chiama la funzione printf dalla libreria con l'indirizzo 0x1030.

Quindi, il codice sembra fare un calcolo, confrontare il risultato con 10 e, se il risultato è inferiore a 10, chiamare la funzione printf con il valore 0 come argomento.