

Linguaggio Assembly

Esercizio

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>
```

ISTRUZIONE	SIGNIFICATO
0x00001141 <+8>: mov EAX,0x20	Spostare il valore 32(0x20) in EAX
0x00001148 <+15>: mov EDX,0x38	Spostare il valore 56(0x38) in EDX
0x00001155 <+28>: add EAX,EDX	Somma il valore contenuto in EDX ad EAX e salva il risultato in EAX. Quindi 56 + 32 = 88
0x00001157 <+30>: mov EBP,EAX	Sposta il valore contenuto in EAX in EBP Quindi EBP = 88
0x0000115a <+33>: cmp EBP,0xa	Compara il valore contenuto di EBP (88) con 0xa (10)
0x0000115e <+37>: jge 0x1176 <main+61>	Esegue un salto condizionale ad una locazione specifica se la destinazione è maggiore o uguale della sorgente nell'istituzione "cmp". Visto che 88 > 10 il salto avviene
0x0000116a <+49>: mov EAX,0x0	Sposta il valore di 0 (0x0) in EAX
0x0000116f <+54>: call 0x1030 <printf@plt>	Chiama la funzione printf