



27/03/2024

Assembly x86

Prepared by:
Manuel Buonanno

Organized by:



Indice

1) Traccia.....	3
2) Svolgimento.....	4
2.1) svolgimento.....	5

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>1
```

Svolgimento

- **0x00001141 <+8>: mov EAX,0x20**
Esadecimale -> decimale = 20 -> 32
Sposta il valore decimale 32 nel registro EAX.
- **0x00001148 <+15>: mov EDX,0x38**
Esadecimale -> decimale = 38 -> 56
Sposta il valore decimale 56 nel registro EDX.
- **0x00001155 <+28>: add EAX,EDX**
Addiziona il valore del registro EDX (52) al valore del registro EAX (32) e salva il risultato (88) nel registro EAX.
- **0x00001157 <+30>: mov EBP, EAX**
Sposta il valore del registro di EAX(88) nel registro EBP.
- **0x0000115a <+33>: cmp EBP,0xa**
Esadecimale -> decimale = a ->10
Controlla l'uguaglianza tra il valore decimale 10 con il valore contenuto in EBP (88).

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

Effettua un salto condizionale se la destinazione di «cmp» è maggiore o uguale del valore di controllo.

Considerato che $88 > 10$, il salto viene effettuato.

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

Esadecimale -> decimale = 0 -> 0

Sovrascrive il valore di EAX con il valore 0, ovvero sposta 0 in EAX.

- **0x0000116f <+54>: call 0x1030 <printf@plt>1**

Chiamata di funzione ad una funzione a noi nota (printf).

