

CONSEGNA S10/L3

ASSEMBLY X86

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 0x0
000116f <+54>:  call 0x1030 <printf@plt>
```

Traduzione

0x00001141 <+8>: mov EAX,0x20 - Sposta il valore intero decimale 32 nel registro EAX

0x00001148 <+15>: mov EDX,0x38 - Sposta il valore intero decimale 56 nel registro EDX

0x00001155 <+28>: add EAX,EDX - Somma il registro EDX ad EAX, praticamente somma 56 a 32 ed aggiorna il registro EAX con la somma = 88

0x00001157 <+30>: mov EBP, EAX - Muove il contenuto del registro EAX, ovvero 88 nel registro EBP

0x0000115a <+33>: cmp EBP,0xa - Controlla (CMP= compare) l'uguaglianza tra il valore 0xa che in decimale è il numero 10 con il valore contenuto in EBP ovvero 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 - Sovrascrive il valore di EAX con il valore 0, ovvero sposta 0 in EAX

0x0000116f <+54>: call 0x1030 <printf@plt> - Chiamata di funzione ad una funzione a noi nota, è la funzione printf