

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>

```

0x00001141<+8>: mov EAX,0x20:

0x20= 32 in decimale, l'istruzione indica di copiare il valore 32 nel registro EAX. Indirizzo memoria: 0x00001149.

0x00001148<+15>: mov EDX,0x38:

0x38 = 56 in decimale, l'istruzione indica di copiare il valore di 56 nel registro EDX. Indirizzo memoria: 0x00001157.

0x00001155<+28>: add EAX,EDX: *32+56=88*

L'istruzione indica di sommare i valori contenuti nel registro EDX con quello di EAX, memorizzandolo in EAX. Indirizzo di memoria: 0x00001171.

0x00001157<+30>: mov EBP,EAX:

L'istruzione indica di copiare i valore contenuti in EAX nel registro di EBP. Indirizzo di memoria:0x00001175.

0x0000115a<+33>: cmp EBP,0xa: *88-10*

0xa = 10 in decimale, confronta il contenuto del registro EBP con il valore 10, in questo caso Zero flag e Carry flag saranno 0. Indirizzo di memoria: 0x0000117d.

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

0x1176: indirizzo di destinazione, <main+61> indica che 0x1176 si trova a un offset di +61byte dal punto di inizio della funzione chiamata 'main'. L'istruzione indica di eseguire un salto se la destinazione è maggiore o uguale alla sorgente. Indirizzo di memoria: 0x00001179.

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

0x0 = 0 in decimale, l'istruzione indica di copiare il valore 0 nel registro eax. Indirizzo di memoria: 0x00001195.

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

0x1030 è l'indirizzo di memoria dove si trova la funzione da chiamare, call serve appunto per chiamare la funzione, <printf.... specifica che la funzione è printf ..@plt> meccanismo utilizzato per chiamare funzioni in librerie condivise. Indirizzo memoria: 0x0000119d.