EPICODE

Esercizio S10 – L3

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>

SOLUZIONE

- 0x00001141 <+8>: mov EAX,0x20  
Copia il valore 0x20 (32 in decimale) nel registro EAX. I registri sono piccole locazioni di memoria all’interno del processore che contengono dati temporanei durante l’esecuzione del programma.

- 0x00001148 <+15>: mov EDX,0x38  
Copia il valore 0x38 (56 in decimale) nel registro EDX.

- 0x00001155 <+28>: add EAX,EDX  
Aggiunge il valore nel registro EDX a quello nel registro EAX, e salva il risultato in EAX. Quindi, dopo questa istruzione, EAX conterrà 0x58 (88 in decimale).

- 0x00001157 <+30>: mov EBP, EAX  
Copia il valore del registro EAX nel registro EBP. Quindi, dopo questa istruzione, EBP conterrà 0x58 (88 in decimale).

- 0x0000115a <+33>: cmp EBP,0xa  
Confronta il valore nel registro EBP (0x58, 88 in decimale) con 0xa (10 in decimale). Dato che la destinazione (88 in decimale) è maggiore della sorgente (10 in decimale) Zero flag (ZF) e Carry Flag (CF) vengono entrambi impostati a 0. Zero flag viene impostato a zero in quanto il risultato è diverso da 0, mentre Carry Flag viene impostato a 0 in quanto non c’è un “riporto”.

- 0x0000115e <+37>: jge 0x1176 <main+61>  
Salta all’indirizzo 0x1176 (che corrisponde all’etichetta main+61) solo se (si tratta di un salto condizionale) la destinazione è maggiore della sorgente. Questo significa che il flusso del programma salterà a quella istruzione dato che il valore di destinazione in EBP (88 in decimale) è maggiore o uguale al valore di sorgente 0xa (10 in decimale).

- 0x0000116a <+49>: mov eax,0x0  
Copia il valore 0x0 (0 in decimale) nel registro eax.

- 0x0000116f <+54>: call 0x1030 <printf@plt>  
Chiama la funzione printf, che si trova all’indirizzo 0x1030 (che corrisponde all’etichetta printf@plt). La funzione printf stampa una stringa sullo schermo in base ai parametri passati nei registri.  
  
Si specifica che il programma non esegue le ultime due istruzioni scritte, in quanto salta direttamente a 0x1176.