

Esercitazione WEEK 22 D1

Assembly x86

Ettore Farris

Descrizione sintetica e svolgimento

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**

Sposta il valore 32 nell'accumulatore (ovvero registro EAX);

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

Sposta il valore 56 nel data register (Ovvero EDX);

- **0x00001155 <+28>: add EAX,EDX**

Somma 56 + 32 (Ovvero somma i valori contenuti nel data register e nell'accumulatore) e salva il risultato 88 nel registro EAX;

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

Sposta il valore 88 presente nell'accumulatore nel Base Pointer (Ovvero EBP);

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

Confronta 10 con 88, cioè il valore presente nel Base Pointer;

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

Effettua un salto condizionale alla locazione specificata se la destinazione è maggiore o uguale della sorgente nell'istruzione «cmp». (Sintassi dell'istruzione cmp = «cmp destinazione, sorgente»). In questo caso EBP 88 (destinazione) è maggiore di 10 (sorgente). Quindi l'esecuzione salta all'indirizzo di memoria 0x1176;

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

Setta l'accumulatore a zero;

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

Chiama la funzione printf.