

Programmazione in ASSEMBLY

-mov EAX,0x20

;L'istruzione mov è utilizzata nei linguaggi assembly per spostare dati da una posizione di memoria a un registro o viceversa(come un taglia ed incolla su un normale sistema operativo windows ma senza la rimozione del file iniziale).In questo caso,sta' assegnando il valore esadecimale 0x20 al registro EAX.

-mov EDX,0x38

;L'istruzione mov è utilizzata nei linguaggi assembly per spostare dati da una posizione di memoria a un registro o viceversa(come un taglia ed incolla su un normale sistema operativo windows ma senza la rimozione del file iniziale).In questo caso,sta' assegnando il valore esadecimale 0x38 al registro EDX.

-add EAX,EDX

;L'istruzione add è utilizzata nei linguaggi assembly per sommare due valori.In questo caso sta' sommando il valore di EDX ad EAX,notando che la somma verra eseguita in questo modo $EDX+EAX=$, il prodotto e' salvato/sovrascritto nella destinazione.

-mov EBP,EAX

;L'istruzione mov è utilizzata nei linguaggi assembly per spostare dati da una posizione di memoria a un registro o viceversa(come un taglia ed incolla su un normale sistema operativo windows ma senza la rimozione del file iniziale).In questo caso,sta' assegnando il valore del registro EAX a quello del registro EBP.

-cmp EBP,0xa

;L'istruzione cmp e' utilizzata nei linguaggi assembly per comparare destinazione e sorgente(si possono comarare due flag se specificato come per il piu' pratico esempio: cmp ZF,CF che sono 2 EFLAGS).

-jge 0x1176 <main+61>

;L'istruzione jge (salto condizionale)e' utilizzata nei linguaggi assembly per saltare alla allocazione specificata se la destinazione e' maggiore o uguale della sorgente nell'istruzione <<cmp>>

-mov eax,0x0

;L'istruzione mov è utilizzata nei linguaggi assembly per spostare dati da una posizione di memoria a un registro o viceversa(come un taglia ed incolla su un normale sistema operativo windows ma senza la rimozione del file iniziale).In questo caso,sta' assegnando il valore esadecimale 0x0 al registro eax.

-call 0x1030 <printf@plt>

L'istruzione call in assembly ed usata per chiamate a funzioni in questo caso la funzione printf la quale ci mostrera i dati legati ad essa sullo schermo.

Sviluppo

;valore di EAX=32
mov EAX,0x20

;valore di EDX=56
mov EDX,0x38

;somma $EAX+EDX=56+32=88$
add EAX,EDX

;valore di EBP=88

```
mov EBP,EAX
```

```
;comparare destinazione/sorgente dove EBP=88 0xa=10  
cmp EBP,0xa
```

```
;utilizzata per saltare alla allocazione specificata se la destinazione e' maggiore o uguale della sorgente nell'istruzione  
<<cmp>>,in questo caso saltera' al main (offset relativo al main 0x1176) dato 88>10  
jge 0x1176 <main+61>
```

```
;valore di eax=0  
mov eax,0x0
```

```
;call per chiamate a funzioni in questo caso la funzione printf la quale ci mostrera i dati legati ad essa sullo schermo.  
call 0x1030 <printf@plt>
```

IL programma si occupa di dividere per 10 la somma di due interi (x:a+b) in questo caso 10:88=0 printf ci riposta il risultato in ogni caso riportando 0 in caso di fallimento dopo il jge o riportando il risultato di una divisione possibile.