EPICODE-CS0124
S10/L3 - Pratica

Flaviano Sedici

Indice

Riferimenti e versioni

Responsabile del documento: Flaviano Sedici

Versionamento

Versione	Descrizione	Riferimento	Data
1.0	Redazione documento	Responsabile	27/03/2024

1. 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>

1.1. Definizioni

Assembly

Assembly è un linguaggio di programmazione a basso livello che rappresenta le istruzioni macchina utilizzando mnemonici leggibili dall'uomo. È specifico dell'architettura del processore e fornisce un controllo preciso sulle operazioni di un computer, ma richiede una conoscenza approfondita dell'architettura del processore.

1.2. Analisi del codice

1.2.1. 0x00001141 <+8>: mov EAX,0x20

Il codice muove il valore 0x20 (ovvero 32 in decimale) nel registro EAX.

1.2.2. 0x00001148 <+15>: mov EDX,0x38

Il codice muove il valore 0x38 (ovvero 56 in decimale) nel registro EDX.

1.2.3. 0x00001155 <+28>: add EAX,EDX

Il codice aggiunge il valore EDX (56) al valore di EAX (32), trasformando il valore di EAX in 88.

1.2.4. 0x00001157 <+30>: mov EBP, EAX

Il codice muove il valore di EAX (88) in EBP.

1.2.5. 0x0000115a <+33>: cmp EBP,0xA

Il codice compara il valore di EBP (88) con 0xa (ovvero 10 in decimale). Pone il valore ZF su 0 e CF su 0.

1.2.6. 0x0000115e <+37>: jge 0x1176 <main+61>

Il comando jump greater or equal (jge) verifica che il compare precedente restituisce ZF (Zero Flag) = 0 e CF (Carry Flag) = 0 (destinazione maggiore sorgente) o ZF = 1 e CF = 0 (destinazione uguale alla sorgente) e salta alla locazione di memoria indicata dal **0x1176 <main+61>**.

1.2.7. 0x0000116a <+49>: mov EAX,0x0

Il codice muove il valore 0x0 (ovvero 0 in decimale) nel registro EAX.

1.2.8. 0x0000116f <+54>: call 0x1030 <printf@plt>

Il codice call chiama una subroutine o una funzione presente alla locazione di memoria **0x1030 <printf@plt>**.