CONVENZIONI DI CHIAMATA

domenica 4 dicembre 2022 12:53

Ogni architettura definisce delle convenzioni di chiamata perché bisogna mettersi d'accordo sull'interoperabilità del codice;

SE VOGLIO SCRIVERE UNA FUNZIONE IN ASSEMBLY, DEVO SAPERE DOVE MI ANDRÀ A METTERE I PARAMETRI IL CHIAMANTE!

I PARAMETRI VENGONO PASSATI TRAMITE STACK, REGISTRI, MIX delle DUE!

- TRAMITE LO STACK POSSO passare un numero qualsiasi di parametri;
- IL NUMERO di REGISTRI È LIMITATO

Quindi uso i registri fin dove posso e poi uso lo STACK.

- I primi sei parametri (interi e indirizzi) di una subroutine vengono passati tramite registri:
  - RDI, RSI, RDX, RCX, R8, R9
- Se una subroutine accetta più di sei parametri, si utilizza lo stack per quelli aggiuntivi
- Si utilizzano due registri per il valore di ritorno:
  - RAX e RDX

- I registri sono divisi in *callee save* e *caller save*
  - callee-save: RBP, RBX, R12–R15 → salvati dal chiamato
  - caller-save: tutti gli altri → salvati dal chiamante

Se il chiamato vuole utilizzare deve prima SCRIVERE sullo STACK e poi RIMETTERE appunto lo STACK PRIMA di RESTITUIRE il CONTROLLO al chiamante.

R8 è caller-salve!

- Le *calling conventions* definiscono, *per ogni architettura e sistema*, come è opportuno passare i parametri
- Le convenzioni principali permettono di passare i parametri tramite:
  - lo stack
  - i registri
  - un misto delle due tecniche
- Generalmente il valore di ritorno viene passato in un registro perché la finestra di stack viene distrutta al termine della subroutine
  - Se la subroutine chiamante vuole conservare il valore nel registro, deve memorizzarlo nello stack prima di eseguire la chiamata

QUANDO CHIAMO UNA FUNZIONE F(), VIVO NEL RECORD DI ATTIVAZIONE DI F SULLO STACK, QUINDI NON POSSO SCRIVERE IL VALORE DI RITORNO DI F SULLO STACK, AL TERMINE DI F, IL SUO RECORD DI ATTIVAZIONE VIENE DISTRUTTO!

SE F SCRIVESSE IL valore di RITORNO sul suo RECORD di attivazione, quando la funzione termina quel valore sarebbe logicamente non più valido, quindi devo utilizzare dei

SE F SCRIVESSE IL valore di RITORNO sul suo RECORD di attivazione, quando la funzione termina quel valore sarebbe logicamente non più valido, quindi devo UTILIZZARE dei REGISTRI.

I PARAMETRI AGGIUNTIVI VENGONO INSERITI SULLO STACK IN ORDINE INVERSO!

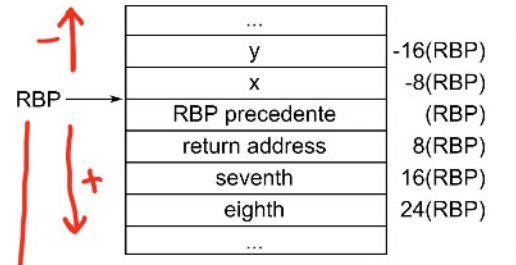
```
void f(int first, int second, int third, int fourth, int fifth,
       int sixth, int seventh, int eighth,) {
       int a, b, i, j;
       ...
}
```

da FIRST a SIXTH sono valori scritti nei REGISTRI, SEVEN ed EIGHT sullo STACK, poi eseguo la chiamata.

↓

SCRIVERÒ sullo STACK l'INDIRIZZO di RITORNO e POI POSTO IL PROLOGO della funzione

8 PARAMETRI



l'offset sarà calcolato a partire da RBP;