PREAMBOLO DI UNA FUNZIONE

sabato 3 dicembre 2022 22:13

IL PREAMBOLO DI UNA FUNZIONE VIENE IMPLEMENTATO IN questo MODO IN ASSEMBLY.

```
void function() {
    int x = 128;
    ...
    return;
}
```

```
function:
    pushq   %rbp
    movq    %rsp, %rbp
    subq    $4, %rsp
    movl    $128, -4(%rbp)
    ...
    addq    $8, %rsp
    leave
    ret
```

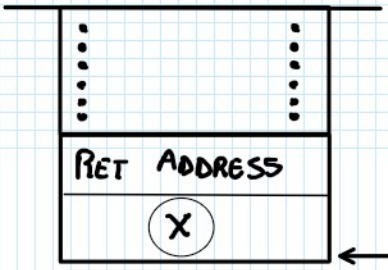
PREAMBOLO;

↓

Push del reg. RbP, seguito da una copia dello STACK POINTER (RSP), IN RbP;

STA SUCCEDENDO QUESTO:

Io entro nella mia funzione, sullo stack ci ho scritto varie cose, quando eseguo la call della mia funzione function, scriverò sulla cima dello stack l'indirizzo di ritorno. Dopodiché io li ho una variabile X, per quello che abbiamo detto prima io scriverò x li sopra. E abbiamo il corrispettivo RSP.



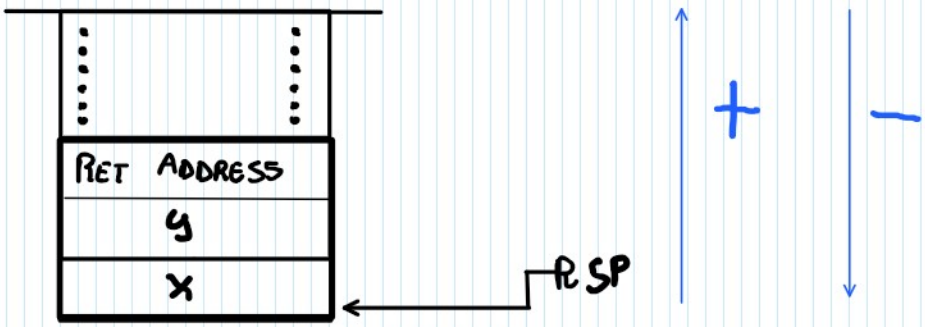
VOGLIO ACCEDERE A X!

X È INT, voglio copiare una longword IN UN REG.

VAI IN MEMORIA, all'INDIRIZZO CONTENUTO nel REGISTRO RSP, E PRENDITI 4 BYTE: `movl (%RSP), %EAX`

↳ Qual'è l'INDIRIZZO di questa variabile?

IMMAGINIAMO CHE IL MIO PROGRAMMA NON HA SOLO X COME VARIABILE, MA ANCHE Y!



SE VOGLIO ACCEDERE AD X COME INDIRIZZO NON CAMBIA NULLA: `movl (%RSP), %EAX`

SE VOGLIO ACCEDERE AD Y A CHE INDIRIZZO DEVO ACCEDERE? X+(longword)

SE VOGLIO ACCEDERE AD X COME INDIRIZZO NON CAMBIA NULLA : `movl (%rsp), %eax`

SE VOGLIO ACCEDERE AD Y A CHE INDIRIZZO DEVO ACCEDERE? X + (longword) = 4 BYTE

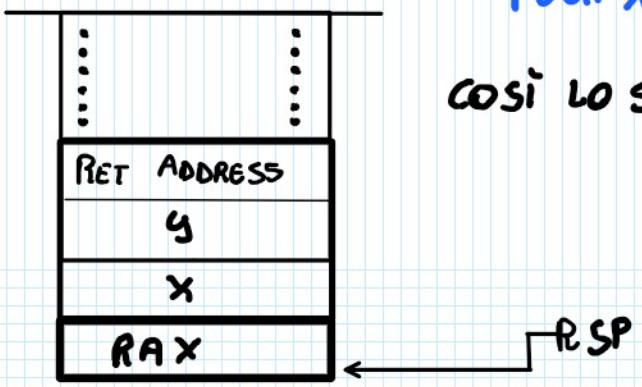
`MOVl 4(%RSP), %EAX`

↓

BASE + SPIAZZAMENTO

PERÒ MAGARI NEL MIO PROGRAMMA ESEGUO LA PUSH di un REGISTRO.

`PUSH %RAX`



COSÌ LO STACK POINTER AUMENTA;

ORA SE VOGLIO ACCEDERE NUOVAMENTE A X?

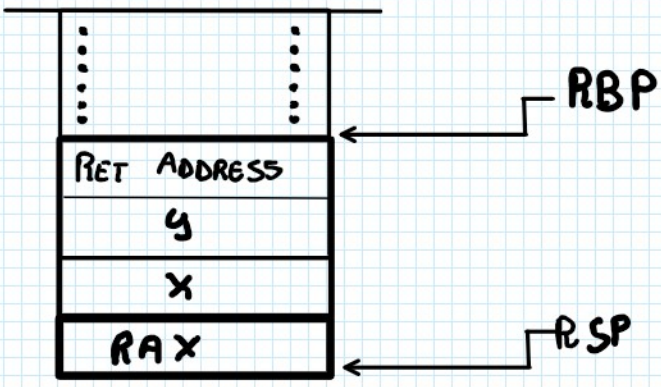
`movl 8(%RSP), %EAX`

LE ULTIME DUE ISTRUZIONI ACCEDONO ALLA STESSA VARIABILE MA UTILIZZANDO DUE INDIRIZZAMENTI DIVERSI!

QUESTA COSA È SCOMODA!

ÈMEGLIO TENERE TRACCIA DELLA BASE DELLO STACK!

ACCESSO A Y:



y = RBP - 4

x = RBP - 8

↓

questi offset sono costanti anche se il mio stack sale o scende.

LA BASE LA salviamo nel REGISTRO BASE POINTER (RBP)

LA BASE LA SALVIAMO NEL REGISTRO BASE POINTER (RBP)

OGNI VOLTA CHE ENTRO IN UNA FUNZIONE SALVO IL VALORE DI RBP PRECEDENTE SULLO STACK, E IMPOSTO IL NUOVO RBP ALLA CIMA DELLO STACK.

RBP=RSP e tutte le variabili le leggo da questo spiazzamento

```
pushq   %rbp
movq    %rsp, %rbp
subq    $4, %rsp
```

all'atto della chiamata della funzione!

PER CIASCUN Record di attivazione!

SOTTRAGGO 4 alla cima dello stack per fare spazio alla mia variabile x, di 4 byte!

A questo punto ci scrivo 128, che è il valore della mia variabile!

```
function:
    pushq   %rbp
    movq    %rsp, %rbp
    subq    $4, %rsp
    movl    $128, -4(%rbp)
    ...
    addq    $8, %rsp
    leave
    ret
```

a partire dalla BASE:

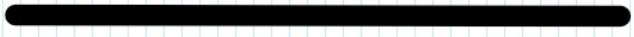
Rbp - 4

4 → DISTRUZIONE RDA all'ISTRUZIONE di RETURN;

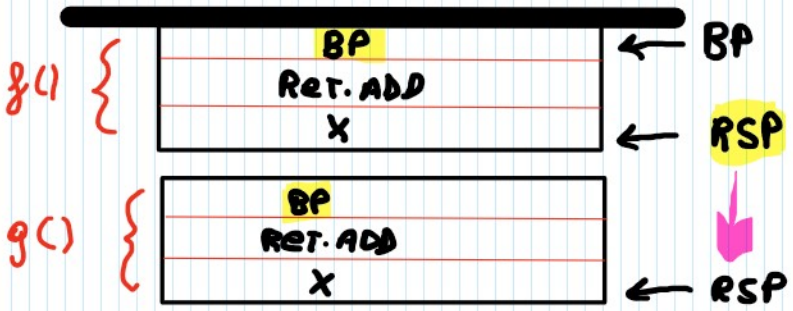
RDA di attivazione PRIMA di eseguire

RESTITUISCO IL CONTROLLO al chiamante, OSSIA togliere tutto quello che riguarda la funzione corrente dalla cima dello stack.
INVALIDO logicamente tutto lo spazio delle variabili globali, TOGLIERE RBP dalla cima dello STACK e RIMETTERLO nel REGISTRO, effettuare IL RITORNO;

ESEMPIO

1.  ← BP, RSP

2.



Prima l'indirizzo
dopo BP

E PARTE IL PREAMBOLO