

VARIABILI LOCALI

Quando è possibile il compilatore usa i registri per conservare i valori delle variabili locali.

Ci sono alcuni casi in cui questo non è possibile:

- 1 Non ci sono registri disponibili
- 2 È utilizzato l'operatore "address of" (&)
- 3 La variabile locale è un array o struct

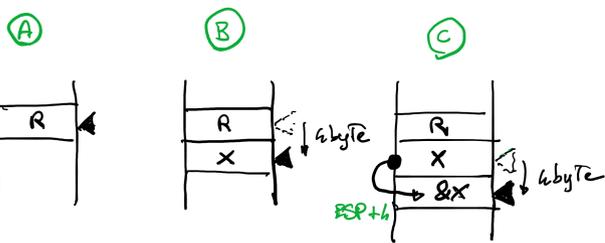
Esempio:

```
(A) int f() {
```

```
(B) int x;
```

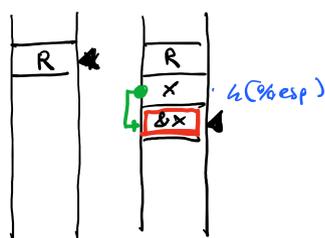
```
(C) f(&x);  
return x;
```

}



```
f: .globl f
```

```
# PROLOGO  
subl $8, %esp  
addl $4, %esp  
movl %esp, %eax  
addl $h, %eax  
movl %eax, (%esp)  
call f  
movl 4(%esp), %eax  
# EPILOGO  
addl $8, %esp  
ret
```



```
movl %esp, %eax } leal 4(%esp), %eax  
addl $h, %eax
```

LEA = Load Effective Address

LEA calcola un indirizzo di memoria senza accedere alla memoria

```
LEA O(B, i, S), D → MOV B, D  
D = D + 0  
D = D + I * S
```

↓
Indirizzo B + I * S + O

```
leal 4(%esp), %eax
```

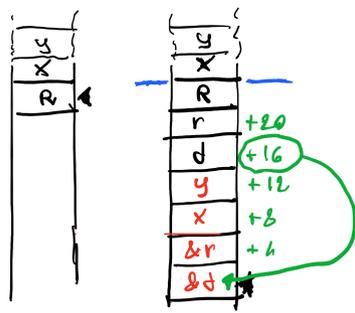
Calcola ESP+h e la scrive in EAX

```
movl 4(%esp), %eax
```

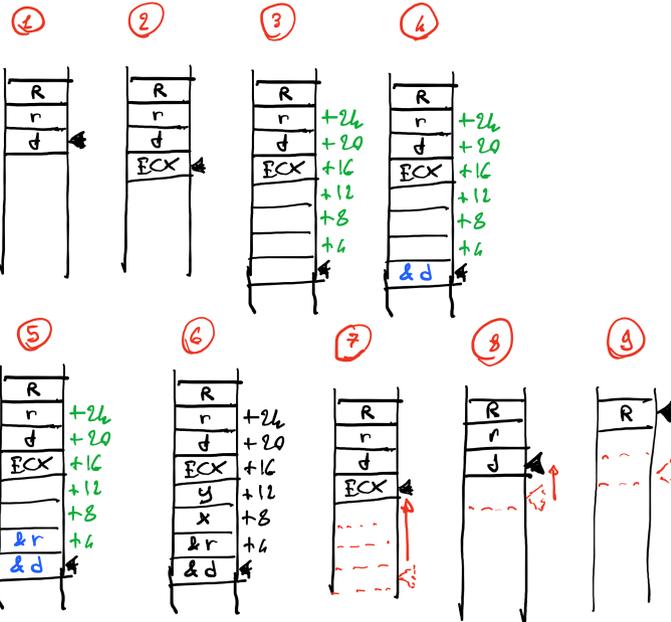
Calcola ESP+h, legge da memoria il valore presente a quell'indirizzo e lo scrive in EAX

NOTA > Se usiamo LEA per calcoli aritmetici lo status register (EFLAGS) NON viene alterato

```
subl $h, %esp
```



```
.globl f  
f:  
# PROLOGO  
movl 4(%esp), %eax # x  
movl 8(%esp), %ecx # y  
subl $8, %esp  
pushl %ecx  
subl $16, %esp  
leal 20(%esp), %edx # &d  
movl %edx, (%esp)  
leal 24(%esp), %edx # &r  
movl %edx, 4(%esp)  
movl %eax, 8(%esp)  
movl %ecx, 12(%esp)  
call get_div_rem  
addl $16, %esp  
popl %ecx  
movl (%esp), %eax  
# EPILOGO  
addl $8, %esp  
ret
```



ESPRESSIONI BOOLEANE

```
int f(int x, int y) {  
return x < y;  
}
```

```
int a = 1;  
if (x < y) goto E;  
a = 0;  
E:  
return a;
```

istruzione SET cc D
↑ ↑
condition code registro n
code bit

```
.globl f  
f:  
movl 4(%esp), %eax # x  
movl 8(%esp), %ecx # y  
cmpl %edx, %ecx  
setl %al  
movzbl %al, %eax  
ret
```

REGOLA DEL CORTOCIRCUITO

```
int is_space(char *s) {
```

```
return s != null && *s == ' ';
```

}

viene valutata se e solo se l'espressione soddisfa l'operazione