COSA DICE ABI IA32 SULL'USO DEI REGISTRI

I registri sono divisi in due gruppi

CALLER SAVED
REGISTRI A, C, D

- La funzione chiamante (caller) deve salvare il valore di questi registri prima di chiamare altra funzione
- La funzione chiamata (callee) può modificare liberamente il valore dei registri ("sporcare il reg.")

CALLEE SAVED
REGISTRI B, DI, SI, BP

- La funzione chiamata (callee) deve salvare il valore di questi registri e ripristinarlo prima di restituire il controllo alla funzione chiamante

COME SALVARE IL VALORE DI UN REGISTRO

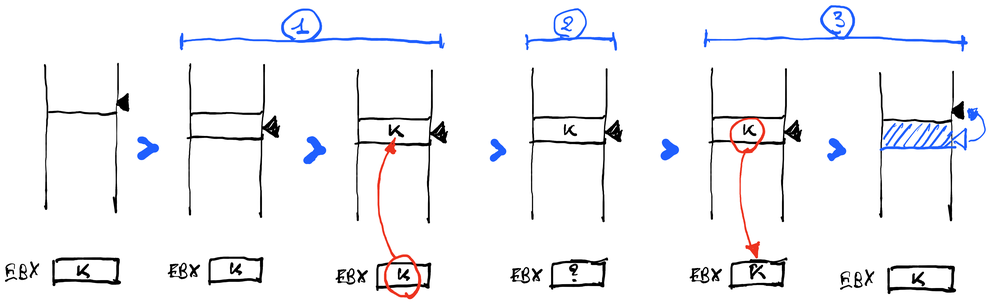
1. Salviamo il valore di %reg nella stack
2. Eseguiamo le istruzioni che "sporcano" %reg
3. Recuperiamo dalla stack il valore di %reg

METODO ESPLICITO

```
1  subl  $4, %esp
   movl  %ebx, (%esp)
2  ......   # %ebx viene sporcato
3  movl  (%esp), %ebx
   addl  $4, %esp
```

METODO PUSH + POP

```
1  pushl %ebx
2  .....
3  popl  %ebx
```



Es.:

```
int f() {
    return g() + h();
}
```

>

```
int f() {
    int a = g();
    int b = a;
    a = h();
    a = a + b;
    return a;
}
```

>

```
.globl f
f:
1  pushl %ebx
   call g
2  movl %eax, %ebx
   call h
   addl %ebx, %eax
3  popl %ebx
   ret
```

QUALI REGISTRI POSSO USARE?

- Ho una funzione che **NON** chiama altre funzioni
  - > Uso liberamente A, C, D
  - > Se non bastano uso B, DI, SI, BP ma devo preservare il valore (prologo + epilogo)
- Ho una funzione che chiama altre funzioni
  - > Uso B, DI, SI, BP preservandone il valore (prologo + epilogo)
  - > Uso liberamente A, C, D per valori temporanei **SOLO** in blocchi di istruzioni che **NON** chiamano altre funz
  - > Uso A, C, D ma preservo il valore (push + pop) prima di **QUALSIASI** chiamata a funzione