Programação em *assembly*

Variáveis escalares e controlo de fluxo

IA32

IA32 – Variáveis escalares

Contêm apenas um valor de um tipo de dados primitivo: caracter, inteiro, vírgula flutuante, etc.

Podem ser alocadas na memória ou em registos.

A utilização eficiente dos registos depende do compilador e do nível de optimização utilizado.

As variáveis globais são alocadas na memória.

O seu endereço só é determinado pelo *linker*.

Em *assembly* aparecem referenciadas pelo seu nome.

IA32 – Variáveis escalares

```
int i, j;
int main ()
{
   j=10;
   i = j * 4; }
```

```
main:
  pushl %ebp
  movl %esp, %ebp
  movl $10, j          ; j=10
  movl j, %eax         ; %eax=j
  sall $2, %eax        ; %eax = j*4
  movl %eax, i         ; i=%eax
  leave
  ret
```

```
leave ⇔ movl %ebp,%esp;popl %ebp
```

IA32 – Variáveis Escalares

```
int i, j, k;
int main ()
{  j = 10;
   k = 20;
   i = 100 + k/j; }
```

```
main:
  pushl %ebp
  movl %esp, %ebp
  movl $10, j          ; j=10
  movl $20, k          ; k=20
  movl k, %eax         ; %eax=k
  idivl j, %eax        ; %eax = %eax/j
  addl $100, %eax      ; %eax = %eax + 100
  movl %eax, i         ; i=%eax
  leave
  ret
```

IA32 – if .. then .. else

```
int i, j, k;
main ()
{
   i = 5;
   j = 10;
   k = j * i;
   if (k<10)
      return (0);
   else
      return (1);
   }
```

Optimização:
- `k` em `%eax`

```
main:
  pushl %ebp
  movl %esp, %ebp
  movl $5, i          ; i = 5
  movl $10, j         ; j = 10
  movl j, %eax
  imull i, %eax       ; %eax = j*i
  cmpl $10, %eax      ; %eax - 10
  jge else
  movl $0, %eax       ; return (0)
  jmp fim_if
else:
  movl $1, %eax       ; return (1)
fim_if:
  leave
  ret
```

IA32 – for (.. ; .. ; ..)

```
int i, ret;
int main ()
{
  ret = 1;
  for (i=10;i>1;i--)
   ret = ret * i;
  return (ret);
}
```

Optimização:
- i em `%eax`

```
main:
  pushl %ebp
  movl %esp, %ebp
  movl $1, ret      ; ret = 1
  movl $10, %eax    ; i = 10
  jmp teste         ; necessário?
ciclo:
  imull %eax, ret   ; ret *= %eax
  decl %eax         ; i--
teste:
  cmpl $1, %eax     ; i > 1 ??
  jg ciclo
  movl ret, %eax    ; return (ret)
  leave
  ret
```

Sumário

Tema	Hennessy [COD]	Bryant [CS:APP]
Variáveis escalares e controlo de fluxo		Sec 3.6