

# Programação em *Assembly*

## Variáveis estruturadas

IA32

# Variáveis estruturadas

Variáveis estruturadas são compostas por conjuntos de variáveis escalares.

Exemplos:

- array unidimensional – vector de variáveis escalares
- estrutura – várias variáveis escalares e/ou vectoriais encapsuladas

O tamanho destas variáveis inviabiliza o seu armazenamento em registos.

## **Afectação em memória**

# Espaço de Endereçamento

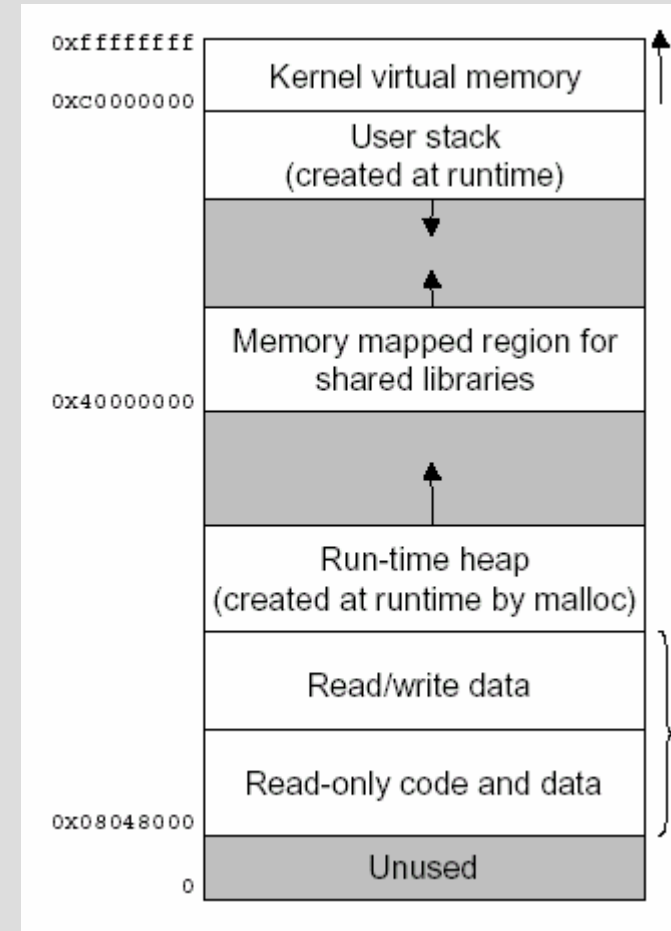
Um processo é a abstracção do sistema operativo para um programa em execução.

Cada processo tem um espaço de endereçamento virtual, que vê como se fosse físico e exclusivamente seu.

Este espaço de endereçamento é dividido em várias secções lógicas que incluem o código, dados estáticos, dados dinâmicos, *stack*, etc.

Variáveis globais – dados estáticos

Variáveis locais - *stack*



# Vectores unidimensionais

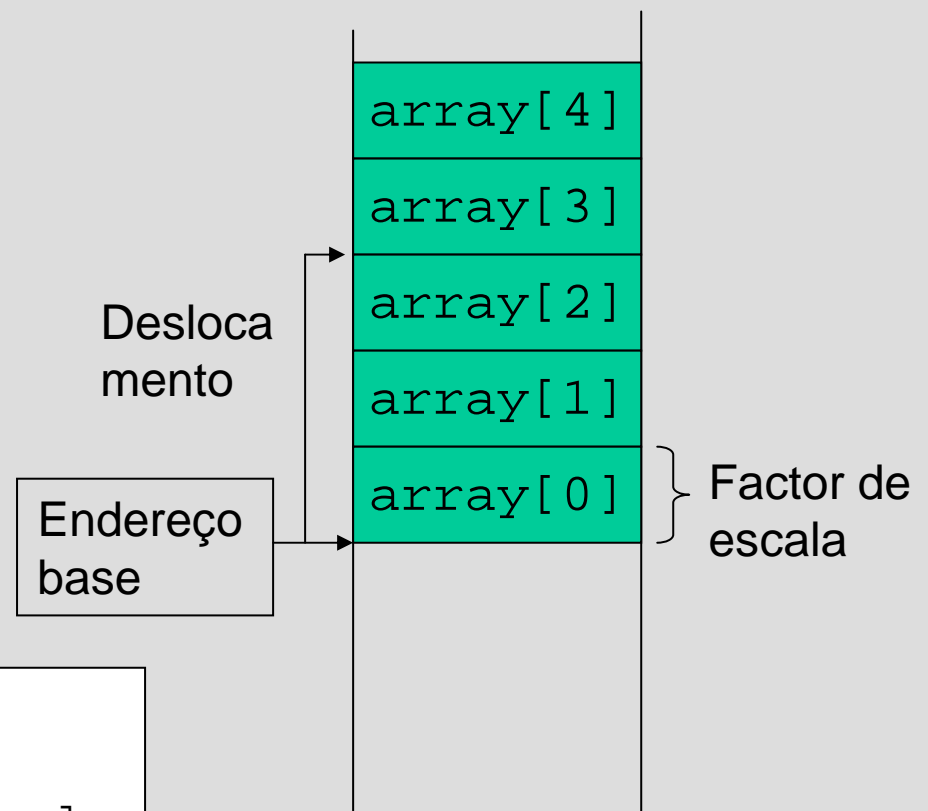
Endereço base – início do *array*

Factor de escala – número de bytes de cada elemento do array

Índice – número do elemento do array ao qual se pretende aceder

Deslocamento – valor a somar ao endereço base para aceder ao elemento do array = índice\*factor de escala

```
int array[5];
```



## Exemplo:

```
&array[3] = base+3*factor de escala
```

# IA32 – vectores unidimensionais

```
int array[100];

int main ()
{ int i;

  for (i=0;i<100;i++)
    array[i] = 10;
}
```

```
array - global
i - %eax
```

```
main:
  pushl %ebp
  movl %esp, %ebp
  movl $0, %eax ; i = 0
  movl $array, %edx ; %edx=&array
ciclo:
  movl %eax, (%edx, %eax, 4) ; array[i] = 10
  incl %eax ; i++
  cmpl $100, %eax ; i < 100 ?
  jl ciclo
  leave
  ret
```

# IA32 – vectores unidimensionais

```
char a[100], b[100];
int main ()
{ int i;

  for (i=0;i<100;i++)
    a[i] = b[99-i];
}
```

a, b - globais

i - %eax

**NOTA:** A utilização de %ebx e %esi viola a convenção de utilização de registos. Ver 7ª aula

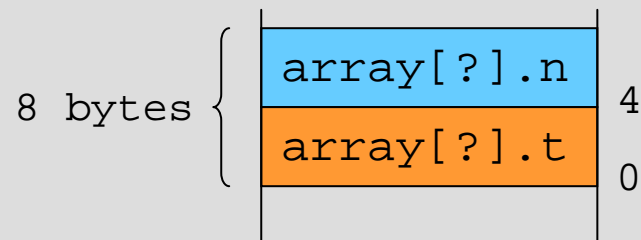
```
main:
  pushl %ebp
  movl %esp, %ebp
  movl $0, %eax      ; i = 0
  movl $a, %edx      ; %edx = &a
  movl $b, %ecx      ; %ecx = &b
ciclo:
  movl $99, %esi
  subl %eax, %esi    ; %esi = 99-i
                    ; %bl = b[99-i]
  movb (%ecx, %esi), %bl
                    ; a[i] = %bl
  movb %bl, (%edx, %eax)
  incl %eax          ; i++
  cmpl $100, %eax
  jl ciclo
  leave
  ret
```

# IA32 – vectores unidimensionais

```
struct {
    int t;
    int n; } array[100];

int main ()
{ int i;

  for (i=0;i<100;i++) {
    array[i].t = 10;
    array[i].n = -i;
  }
}
```



```
main:
    pushl %ebp
    movl %esp, %ebp
    movl $0, %eax
    movl $array, %edx
ciclo:
    movl $10, (%edx, %eax, 8)
    movl %eax, %ecx
    negl %ecx ; %ecx = -i
    movl %ecx, 4(%edx, %eax, 8)
    incl %eax
    cmpl $100, %eax
    jl ciclo
    leave
    ret
```

# IA32 – vectores unidimensionais

(uso de apontadores)

```
int a[100];

int main ()
{ int i;

  for (i=0;i<100;i++)
    a[i] = 10;
}
```

```
a - global
i - %eax
```

```
main:
  pushl %ebp
  movl %esp, %ebp
  movl $0, %eax      ; i=0
  movl $a, %edx     ; %edx=&a
ciclo:
  movl $10, (%edx) ; a[i]=10
  addl $4, %edx
  incl %eax          ; i++
  cmpl $100, %eax
  jl ciclo
  leave
  ret
```



# Sumário

<b>Tema</b>	<b>Hennessy [COD]</b>	<b>Bryant [CS:APP]</b>
Variáveis estruturadas	Sec 3.3 e 3.5 Sec 3.11	Sec 3.8 a 3.8.3 Sec. 3.9.1