

Programação de Computadores II

Cap. 5 – Vetores

Livro: Waldemar Celes, Renato Cerqueira, José Lucas Rangel. Introdução a Estruturas de Dados, Editora Campus (2004)

Slides adaptados dos originais dos profs.: Marco Antonio Casanova e Marcelo Gattass (PUC-Rio)

Referências

Waldemar Celes, Renato Cerqueira, José Lucas Rangel,
Introdução a Estruturas de Dados, Editora Campus
(2004)

Capítulo 5 – Vetores e alocação dinâmica

Tópicos

- Vetores

Vetores

- Exemplo:
 - cálculo da média de um conjunto de 10 números reais

$$m = \frac{\sum x}{N}$$

- implementação
 - valores são lidos e somados
 - cálculo da média é feito a partir da soma

```
/* Cálculo da média de n números reais */
#include <stdio.h>

int main ( void ) {
    float nota, med = 0.0f;
    int i,n;

    printf("Entre com o número de notas:");
    scanf("%d",&n);

    /* leitura do conjunto de valores e cálculo do somatório */
    for (i = 0; i < n; i++) {
        scanf("%f", &nota);
        med = med + nota;
    }

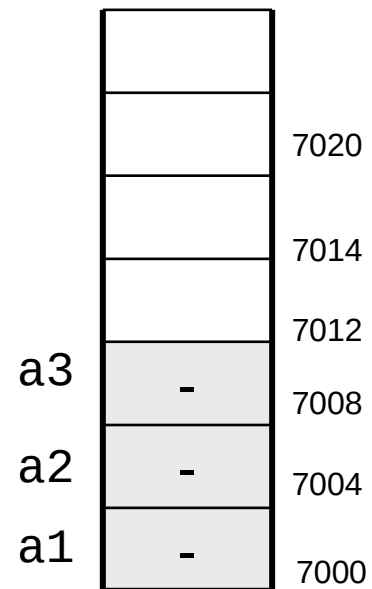
    /* cálculo da média */
    med = med / n;

    /* exibição do resultado */
    printf ("Media = %f \n", med);
    return 0;
}
```

Alocação de várias variáveis

```
int a1, a2, a3;
```

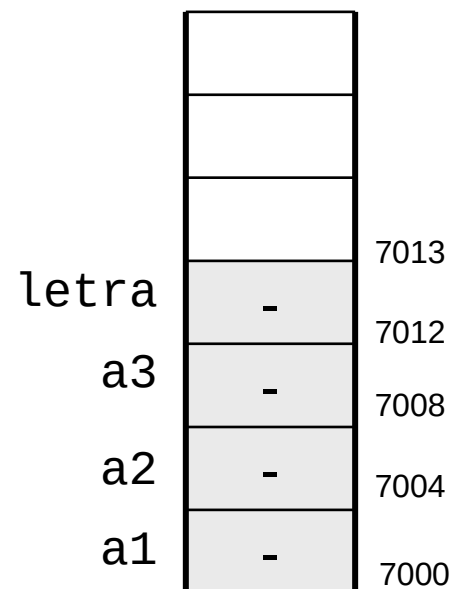
Pilha de memória



Suporte para conjuntos homogêneos

```
int a1,a2,a3;  
char letra;
```

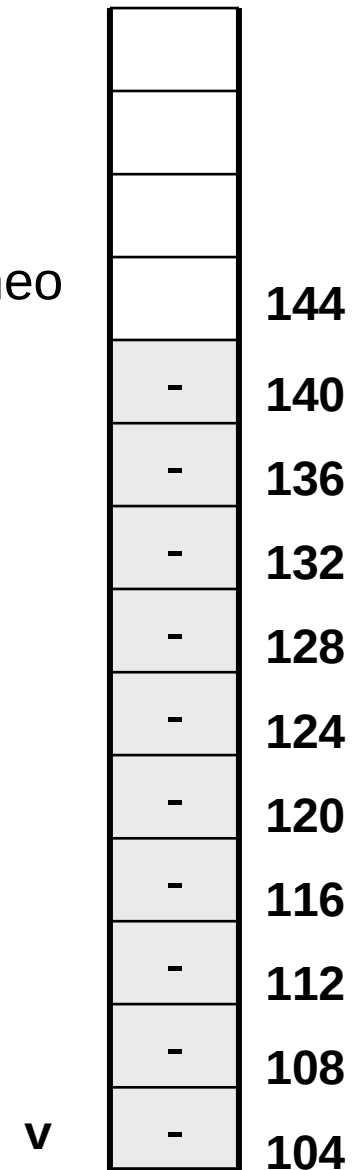
Pilha de memória



Vetores

- Vetor:
 - estrutura de dados definindo um conjunto homogêneo
 - Exemplo:
 - v = vetor de inteiros com 10 elementos

```
int v[10];
```



Vetores: inicialização

```
int v[10]={2,3,5,7,11,13,17,19,23,29};
```

Liberalidade da linguagem:

```
int v[]={2,3,5,7,11,13,17,19,23,29};
```

v		
		144
	29	140
	23	136
	19	132
	17	128
	13	124
	11	120
	7	116
	5	112
	3	108
	2	104

Vetores: acesso aos elementos

```
int v[10]={2,3,5,7,11,13,17,19,23,29};
```

~~v[10]=31;~~

não pode!
invade a memória da
próxima variável!

v[9]	29	140
v[8]	23	136
v[7]	19	132
v[6]	17	128
v[5]	13	124
v[4]	11	120
v[3]	7	116
v[2]	5	112
v[1]	3	108
v[0]	2	104

Acesso aos elementos: exemplo

```
int v[7]={1,3,5,7,9,11,13};  
int i;
```

ou:

```
int v[7];  
int i;  
  
for (i=0; i<7; i++) {  
    v[i]=2*i+1;  
}
```

		140
		136
i		132
v[6]	13	128
v[5]	11	124
v[4]	9	120
v[3]	7	116
v[2]	5	112
v[1]	3	108
v[0]	1	104

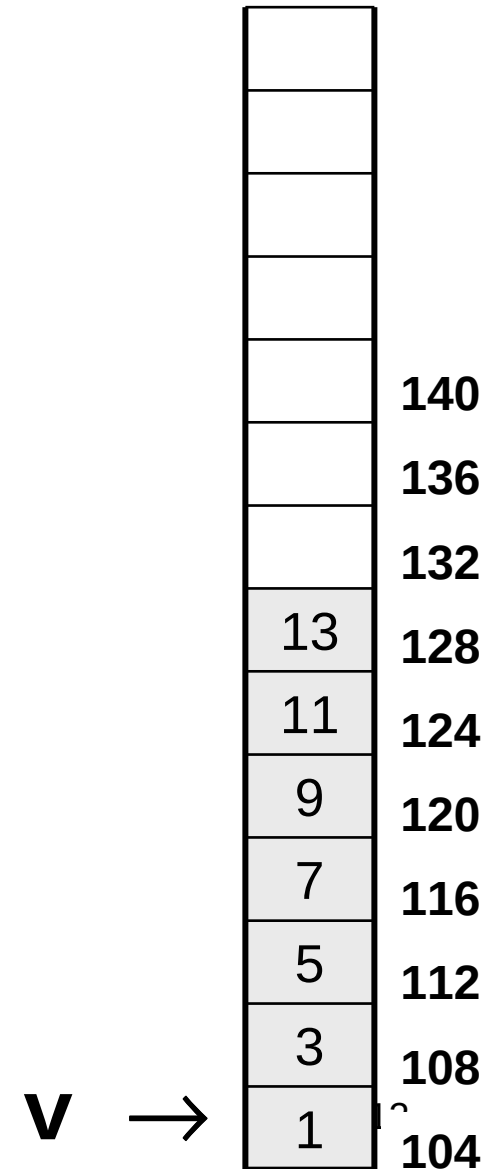
Vetores e Ponteiros

```
int v[7];
```

Quem é v?

v é um ponteiro de valor constante que aponta para o primeiro elemento do vetor de inteiros!

12/03/2014



Vetores e Aritmética de Ponteiros

```
int v[7];
```

$v+i$ aponta para $v[i]$

Ou seja: se $i=1$:

$*(v+i)=3; \Leftrightarrow v[i]=3;$

$\text{scanf}("%d", \&v[i]);$

$\Leftrightarrow$

$\text{scanf}("%d", v+i);$

$v+6 \rightarrow$		140
		136
		132
		128
$v+5 \rightarrow$	13	124
$v+4 \rightarrow$	11	120
$v+3 \rightarrow$	9	116
$v+2 \rightarrow$	7	112
$v+1 \rightarrow$	5	108
$v+0 \rightarrow$	3	104

Vetores

- Exemplo:
 - cálculo da média e da variância de um conjunto de 10 números reais

$$m = \frac{\sum x}{N}, \quad v = \frac{\sum (x - m)^2}{N}$$

- implementação
 - valores são lidos e armazenados em um vetor de 10 posições
 - cálculos da média e da variância efetuados sobre o conjunto de valores armazenado

```

/* Cálculo da média e da variância */
#include <stdio.h>

#define MAX 10

int main ( void ) {
    float v[MAX], med=0.0, var=0.0;
    int i,n;

    for (i = 0; i < MAX; i++)
        scanf("%f", &v[i]);

    for (i = 0; i < MAX; i++)
        med = med + v[i];
    med = med / MAX;

    for ( i = 0; i < MAX; i++ )
        var = var+(v[i]-med)*(v[i]-med);
    var = var / MAX;

    printf ( "Media = %f    Variancia = %f  \n", med, var );
    return 0;
}

```

Vetores

- Passagem de vetor para função:
 - consiste em passar o endereço da primeira posição do vetor
 - função deve ter parâmetro do tipo ponteiro para armazenar esse endereço
 - “passar um vetor para uma função” é equivalente a “passar o endereço inicial do vetor”
 - elementos do vetor não são copiados para a função
 - argumento copiado é apenas o endereço do primeiro elemento
 - Exemplo:
 - Na chamada à função que passa um vetor de `int`, a função deve ter um parâmetro do tipo `int *`


```
/* Cálculo da média e da variância de n reais (segunda versão) */  
  
#include <stdio.h>  
  
/* Função para cálculo da média */  
float media (int n, float *v)  
{  
    int i;  
    float s = 0.0;  
    for (i = 0; i < n; i++)  
        s += v[i];  
    return s/n;  
}
```



parâmetro do tipo
ponteiro para float

```
/* Função para cálculo da variância */  
float variancia (int n, float *v, float m)  
{  
    int i;  
    float s = 0.0;  
    for (i = 0; i < n; i++)  
        s += (v[i] - m) * (v[i] - m);  
    return s/n;  
}
```

```
#define MAX 10

int main ( void )
{
    float v[MAX];
    float med, var;
    int i;

    /* leitura dos valores */
    for ( i = 0; i < MAX; i++ ) {
        scanf("%f", &v[i]);
    }

    med = media(MAX,v);
    var = variancia(MAX,v,med);

    printf ( "Media = %f    Variancia = %f  \n", med, var);
    return 0;
}
```

Vetores

- Passagem de vetor para função (cont.):
 - função pode alterar os valores dos elementos do vetor pois recebe o endereço do primeiro elemento do vetor (e não os elementos propriamente ditos)
 - Exemplo:
 - função incrementando todos os elementos de uma unidade

```
/* Incrementa elementos de um vetor */
#include <stdio.h>
void incr_vetor ( int n, int *v )
{
    int i;
    for (i = 0; i < n; i++)
        v[i]++;
}

int main ( void )
{
    int a[ ] = {1, 3, 5};
    incr_vetor(3, a);
    printf("%d %d %d \n", a[0], a[1], a[2]);
    return 0;
}
```

saída do programa será 2 4 6

Exercício

- Escreva uma função que recebe dois vetores de tamanho n e um terceiro vetor de tamanho $2n$. Ao chamar a função os elementos dos vetores de tamanho n são inseridos de modo intercalado no vetor de tamanho $2n$.

```

/* Incrementa elementos de um vetor */
#include <stdio.h>
void une_intercalado ( int n, int *u, int *v, int *z )
{
    int i;
    for (i = 0; i < n; i++) {
        z[2*i] = u[i];
        z[2*i+1] = v[i];
    }
}

int main ( void )
{
    int a[ ] = {1, 3, 5};
    int b[ ] = {2, 4, 6};
    int c[6];
    int i;
    une_intercalado(3, a, b, c);
    for (i = 0; i < 6; i++)
        printf("%d\n", c[i]);
    return 0;
}

```

Revisão

`n++` incrementa `n` de uma unidade, depois de ser usado

`n--` decrementa `n` de uma unidade, depois de ser usado

`if ( expr ) { bloco de comandos } else { bloco de comandos }`

`for ( expr_inicial; expr_booleana; expr_de_incremento ) { bloco de comandos }`

“passar um vetor para uma função” = “passar o endereço inicial do vetor”

exemplo 1: `float variancia (int n, float* v, float m)`

exemplo 2: `void incr_vetor ( int n, int *v )`

alocação dinâmica de vetor: através da função `malloc`

exemplo 3: `int *v;`

`v = (int *) malloc(10*sizeof(int));`