



Ponteiros

Profa. Elloá B. Guedes

www.elloaguedes.com

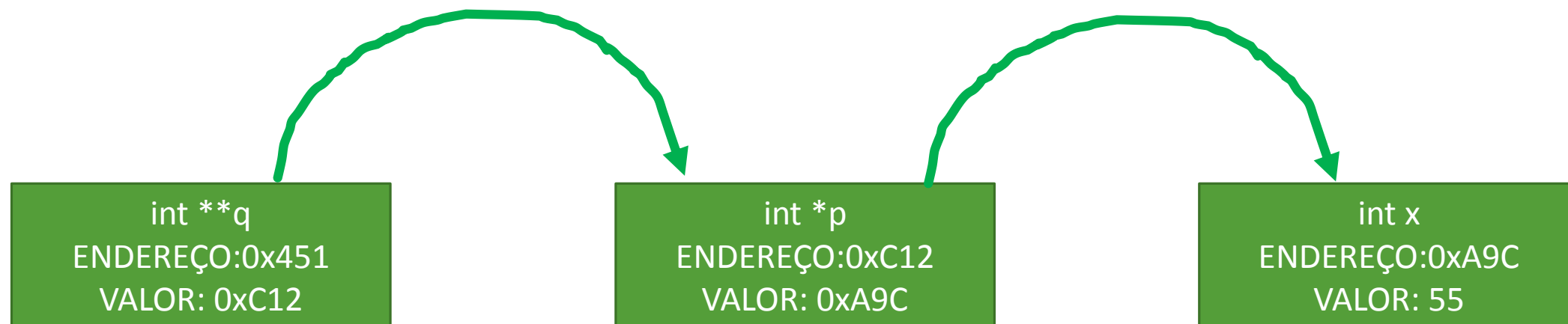
VOU ESTUDAR LP2! VOU ESTUDAR LP2!
VOU ESTUDAR LP2! VOU ESTUDAR LP2!
VOU ESTUDAR LP2! VOU ESTUDAR LP2!
VOU ESTUDAR LP2! VOU ESTUDAR LP2!
VOU ESTUDAR LP2! VOU ESTUDAR LP2!
VOU ESTUDAR LP2! VOU ESTUDAR LP2!
VOU ESTUDAR LP2! VOU ESTUDAR LP2!
VOU ESTUDAR LP2! VOU ESTUDAR LP2!
VOU ESTUDAR LP2! VOU ESTUDAR LP2!
VOU ESTUDAR LP2! VOU ESTUDAR LP2!
VOU ESTUDAR LP2! VOU ESTUDAR LP2!

Profa Elloá



Ponteiros para Ponteiros

- Um ponteiro para ponteiro guarda o endereço de um ponteiro
- Um ponteiro para ponteiro guarda um endereço que aponta para outro endereço



Ponteiros para Ponteiros

- Declaração

tipo **nome_do_ponteiro;

- Exemplos:

```
int **p; //declara ponteiro para um ponteiro de um int.
```

```
char **tmp; //declara ponteiro para um ponteiro de um char.
```

```
float **pont; //declara ponteiro para um ponteiro de um float.
```

Exercício

- É possível alterar o conteúdo de uma variável por meio de um ponteiro para um ponteiro?



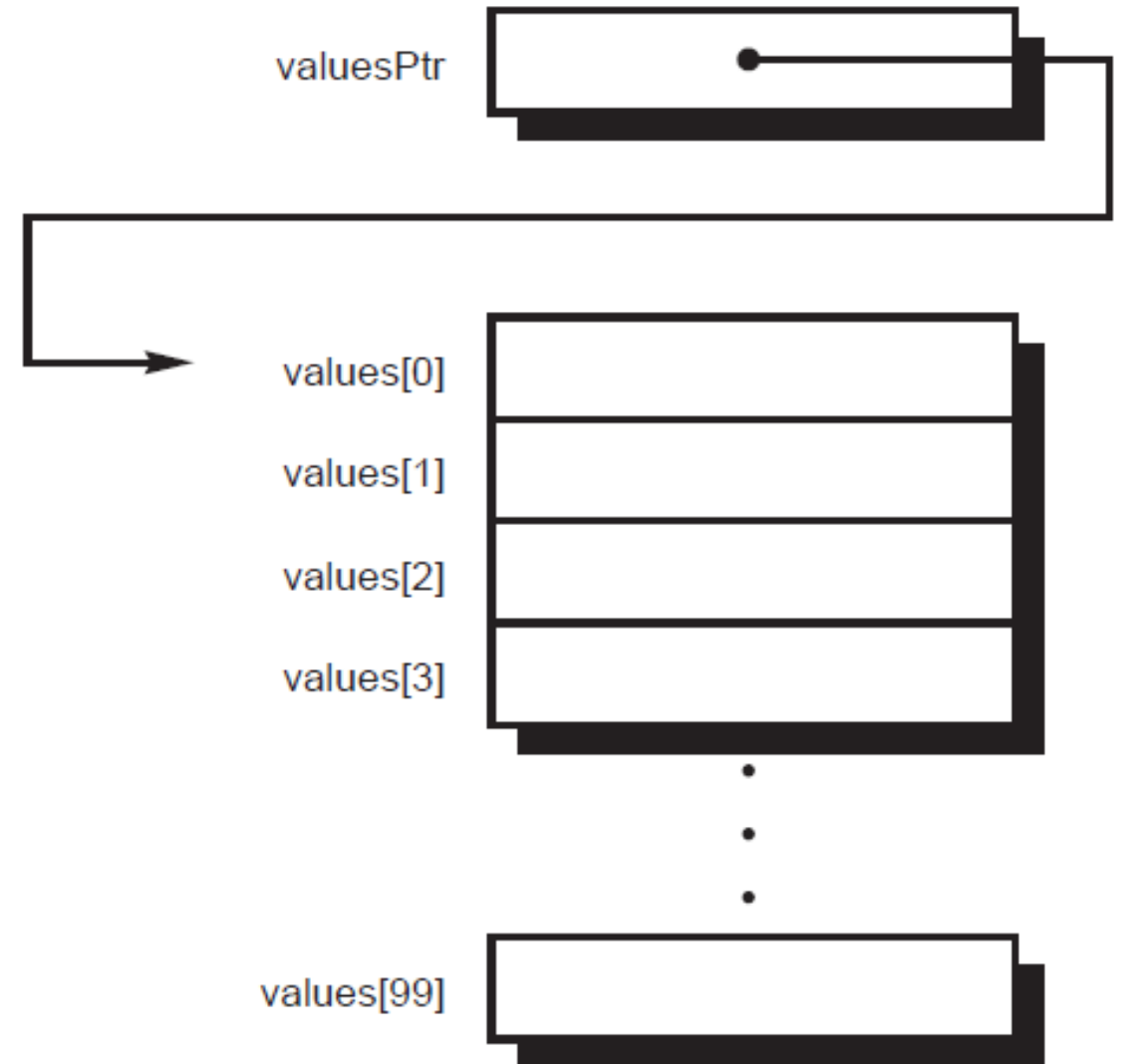
Ponteiros para Vetores

- Vetores são intimamente ligados ao conceito de ponteiros
- Os elementos de um vetor são sempre guardados sequencialmente, a uma distância fixa um do outro
- É possível facilmente passar de um elemento a outro, percorrendo sempre uma mesma distância para frente ou para trás na memória
- Em C, vetores são ponteiros
- Vantagem: eficiência!

Ponteiros para Vetores

```
valuesPtr = &values[0];
```

```
valuesPtr = values;
```



Ponteiros para Vetores

- Como percorrer um vetor usando ponteiros?
- Dica: usar aritmética de ponteiros!
- O final do vetor pode ser armazenado em um ponteiro
 - Ponteiro para o começo do vetor + tamanho do vetor
 - Iterações com for ou while

Exemplo: Iteração pelos Elementos de um Vetor

```
#include <stdio.h>
```

```
int main(){
```

```
    int vet[10] = {12, 1, 9, 48, 52, 37, 22, 16, -5, 4};
```

```
    int *inicio = vet;
```

```
    int *final = vet + 10;
```

```
    for (int *p = inicio; p < final; p++){
```

```
        printf("%d\n", *p);
```

```
    }
```

```
    return 0;
```

```
}
```

Exercício

- Faça um programa que some os elementos de um vetor!
- Restrições:
 - Use ponteiros
 - Percorra o vetor de trás pra frente
 - Armazene o resultado por meio de um ponteiro pra inteiro



Ponteiros para Strings

- Ponteiros para strings são altamente úteis!
- Diferenças na manipulação dos vetores de outros tipos (int, float, etc.)
 - Não é necessário criar um vetor que aponta para o último elemento do array
 - Basta olhar o conteúdo do ponteiro (pare de iterar quando for igual a '\0')
- Várias funções da biblioteca string.h consideram a manipulação com ponteiros

Imprimindo uma String com Ponteiros

```
#include <stdio.h>
```

```
int main(){
```

```
    char string[] = "C eh uma linguagem muito poderosa!";
```

```
    char *ptr = string;
```

```
    for (char *p = ptr; *p != '\0'; p++){
```

```
        putchar(*p);
```

```
    }
```

```
    return 0;
```

```
}
```

Função strstr() em <string.h>

- Procura uma substring dentro de uma string
 - Se há, retorna um ponteiro
 - Se não há, retorna NULL
- Protótipo:
 - `char* strstr(const char *string, const char *substring);`

Função strstr() em <string.h>

```
#include <stdio.h>
#include <string.h>
```

```
int main(void)
{
```

```
    char *cptr;
```

```
    char str[] = "Wikipedia, be bold";
```

```
    char substr[] = "edia, b";
```

```
    cptr = strstr(str, substr);
```

```
    printf("%c\n", *cptr);
```

```
    printf("%s\n", cptr);
```

```
    return 0;
```

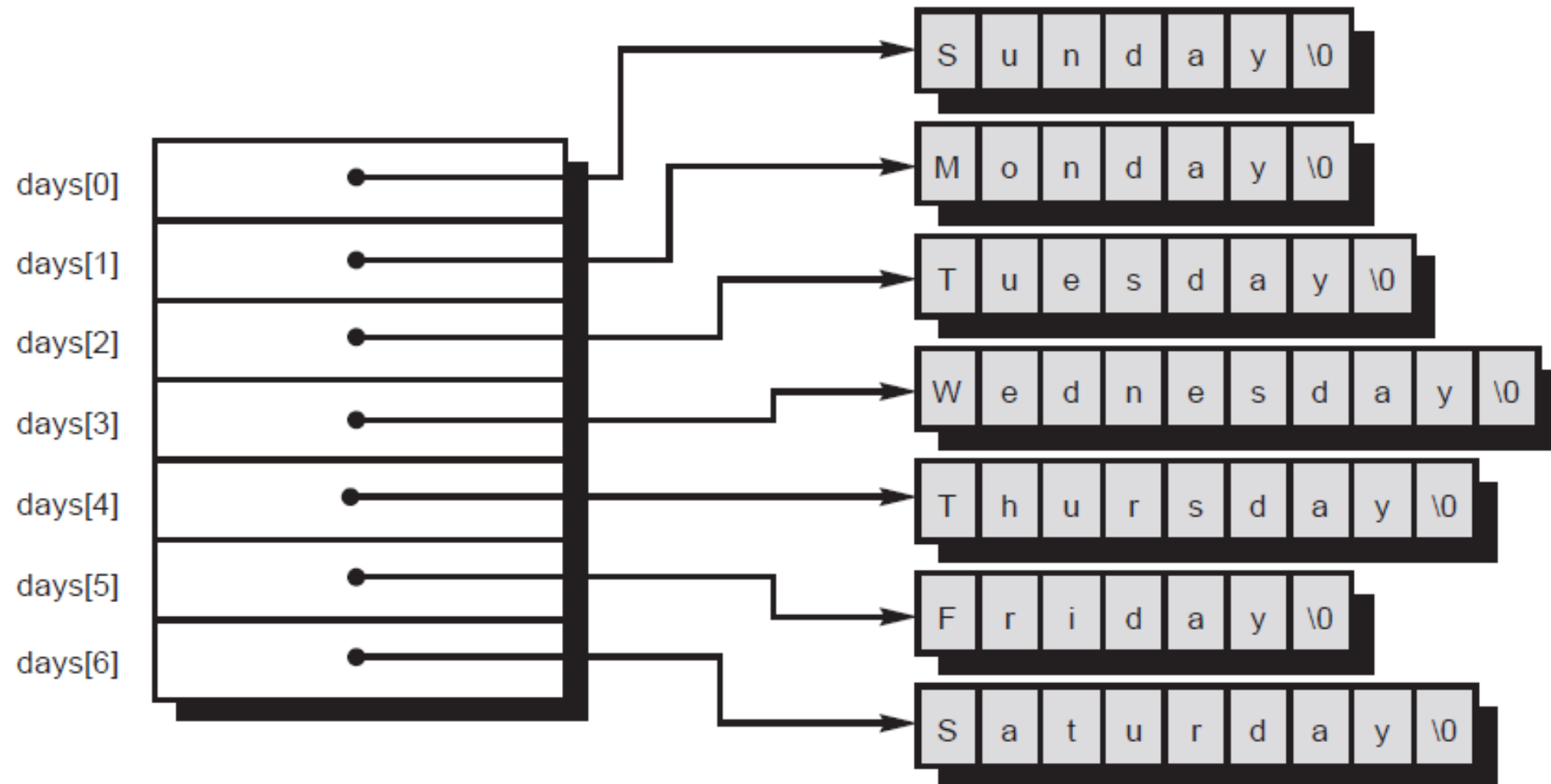
```
}
```

Strings e Ponteiros para Ponteiros

- Podemos ter um vetor de strings agora
- Cuidados:
 - Utilização de ponteiros para ponteiros
 - Opcional: Acrescentar NULL ao final do vetor para marcar o final

```
char *days[] =  
    { "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",  
      "Saturday" };
```

Ideia geral:



Strings e Ponteiros para Ponteiros

```
#include <stdio.h>

int main(){
    char *semana[] = {"Segunda", "Terca", "Quarta", "Quinta",
                      "Sexta", "Sabado", "Domingo", NULL};

    char **i = semana;

    while (*i) {
        puts(*i);
        i++;
    }

    return 0;
}
```

Exercício

- Escreva um programa em C que armazene um vetor de strings contendo os nomes dos dozes meses do ano.
- Por meio de funções, responda:
- Quantas vezes a letra 'a' aparece nos nomes dos meses?
- Qual o mês com maior nome?



Solução

```
#include <stdio.h>
#include <string.h>

int contaA(char *string){
    int cont = 0;
    int i = 0;

    while (*string != '\0'){
        if (*string == 'a')
            cont++;
        string++;
    }

    return cont;
}
```

Solução

```
int main(){
    char *meses[] = {"janeiro", "fevereiro", "marco", "abril", "maio", "junho",
                    "julho", "agosto", "setembro", "outubro", "novembro", "dezembro", NULL};

    char **nomeMeses = meses;
    int totalAs = 0;
    int maior = 0;
    char maiorMes[12];
```

Solução

```
while (*nomeMeses){
    totalAs += contaA(*nomeMeses);
    if (strlen(*nomeMeses) > maior) {
        maior = strlen(*nomeMeses);
        strcpy(maiorMes,*nomeMeses);
    }
    nomeMeses++;
}

printf("Total de As: %d\n",totalAs);
printf("O maior mes tem %d letras e eh %s",maior-1,maiorMes);

}
```