

Algoritmos e Programação II

Ponteiros: cadeias de caracteres

Prof. Viviane Bonadia
Aula baseada nos tópicos de aula e slides
do Prof. Tiago A. Almeida

Cadeias de caracteres

- ***String*** = cadeia de caracteres
- Toda *string* é armazenada na memória como uma sequência contígua (vetor) de caracteres. O final da *string* é marcado pelo caractere nulo '`\0`'
- **Exemplo:**
 - A *string* "abc" é armazenada em um vetor de quatro caracteres:

a	b	c	\0
---	---	---	----
 - Se a *string* for vazia:

\0

Cadeias de caracteres

- ***String*** = cadeia de caracteres
- Toda *string* é armazenada na memória como uma sequência contígua (vetor) de caracteres. O final da *string* é marcado pelo caractere nulo '`\0`'
- **Exemplo:**
 - A *string* "abc" é armazenada em um vetor de quatro caracteres:

a	b	c	\0
---	---	---	----
 - Se a *string* for vazia:

\0

Cadeias de caracteres: ponteiros

- Uma vez que *strings* são armazenadas como vetores, o compilador as processa como um ponteiro do tipo **char ***
 - Tanto a função **printf** quanto **scanf** esperam um valor do tipo **char *** como primeiro argumento
 - `printf("abc");`
 - Quando `printf` é executado, o endereço de "abc" é passado como parâmetro (um ponteiro que aponta para o endereço de memória onde a letra a está armazenada)
- Sempre nos referimos a uma *string* pelo endereço (ponteiro) do primeiro caractere.

Cadeias de caracteres: ponteiros

- **Possibilidades de declaração:**

```
char date[] = "Junho 14";    (Vetor)      [1]  
char *date  = "Junho 14";    (Ponteiro)   [2]
```

- Em [1], os caracteres podem ser alterados como um elemento de um vetor. Em [2], date aponta para uma string literal que não pode ser alterada!

Cadeias de caracteres: ponteiros

- **Possibilidades de declaração:**

```
char date[] = "Junho 14";    (Vetor)      [1]  
char *date  = "Junho 14";    (Ponteiro)   [2]
```

- Em [1], os caracteres podem ser alterados como um elemento de um vetor. Em [2], date aponta para uma string literal que não pode ser alterada!
- Em [1] date é o nome do vetor. Em [2], date é uma variável que pode apontar para outras *strings* durante a execução do programa.

Cadeias de caracteres: ponteiros

- Para manipular uma *string* usando um ponteiro
 - É necessário declarar um ponteiro e apontá-lo para a *string*.

```
char date[] = "Junho 14", p = date;
```

```
*p = 'j';    /* date = "junho 14" */
```

- **p** aponta para o primeiro caractere de `date`.
- Podemos usar aritmética de ponteiros para processar *strings*.

Cadeias de caracteres: aritmética de ponteiros

- Função para contar a quantidade de espaços em branco

```
int count_spaces(const char *s){  
    int count = 0;  
  
    for (; *s != '\0'; s++)  
        if (*s == ' ')  
            count++;  
  
    return count;  
}
```


Cadeias de caracteres: aritmética de ponteiros

- Função para contar a quantidade de espaços em branco

```
int count_spaces(const char *s){  
    int count = 0;
```

const especifica que um valor de variável é constante e informa o compilador para impedir que o programador a modifique.

Se dentro desta função tentarmos alterar o conteúdo apontado por **s**, teremos um erro de compilação!

assignment of read-only location '*s'

Cadeias de caracteres: aritmética de ponteiros

- Função para calcular o tamanho da *string*

```
size_t my_strlen(const char *s){  
    size_t n = 0;  
  
    for (; *s != '\0'; s++)  
        n++;  
  
    return n;  
}
```

Cadeias de caracteres: aritmética de ponteiros

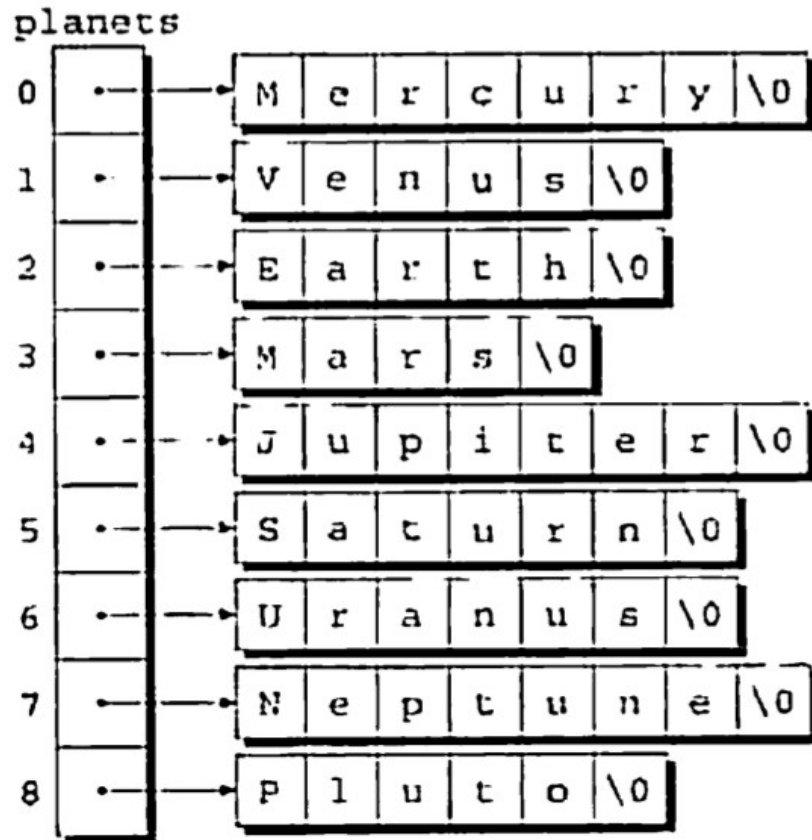
- Função para concatenar duas *strings*

```
char * my_strcat(char *s1, char *s2){  
    char *p = s1;  
  
    while (*p != '\0')  
        p++;  
    while (*s2 != '\0') {  
        *p = *s2;  
        p++;  
        s2++;  
    }  
    *p = '\0';  
  
    return s1;  
}
```

Cadeias de caracteres: vetor de ponteiros

- **Matriz de strings:** vetor de ponteiros para *strings*

```
char *planets[] = {"Mercury", "Venus", "Earth",  
                  "Mars", "Jupiter", "Saturn",  
                  "Uranus", "Neptune", "Pluto"};
```



Cadeias de caracteres: vetor de ponteiros

- **Matriz de strings:** vetor de ponteiros para *strings*
 - Impressão de todos os planetas que iniciam com 'M'

```
for(i=0; i<9; i++)  
    if (planets[i][0] == 'M')  
        printf("%s inicia com M\n", planets[i]);
```