MINIAULA DE ALGORITMOS
STRINGS

Prof. Ivanilton Polato
Departamento Acadêmico de Computação (DACOM–CM)
ipolato@utfpr.edu.br

UTFPR

Strings: o que são?

- São um tipo específico de vetor de caracteres
  - *Conhecidos como cadeias de caracteres*
- São declarados com o tipo CHAR
  - *char str[50];*
- Utiliza de um caractere a mais na quantidade para o controle (‘\0’)
  - *Esse caractere é inserido automaticamente durante a leitura e utilizado por diversas funções para localizar o fim do preenchimento de uma cadeia.*

Um conjunto de funções predefinidas está disponível na biblioteca **`string.h`**

Strings: declaração

- Utilizamos a mesma forma dos vetores:

```
char str[11];
```

*No caso acima temos 11 posições na cadeia str, mas devemos ocupar no máximo 10, pois reservamos uma casa para o **caractere de controle**!*

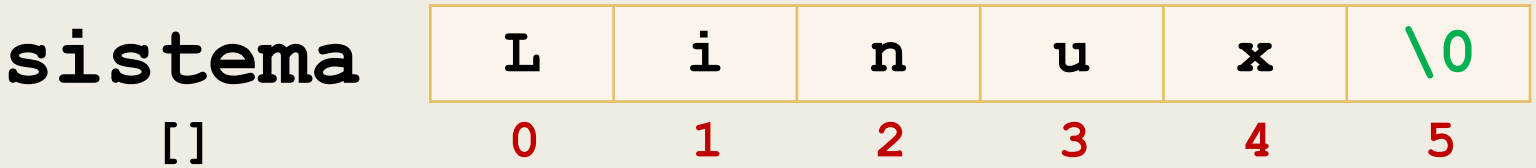
str	A	l	g	o	r	i	t	m	o	s	\0
[]	0	1	2	3	4	5	6	7	8	9	10

str	P	r	o	g	r	a	m	a	\0		
[]	0	1	2	3	4	5	6	7	8	9	10

UTFPR

Strings: inicialização (declaração)

- As cadeias podem ser declaradas automaticamente de acordo com o tamanho de uma palavra:
  - ***char sistema[] = {'L', 'i', 'n', 'u', 'x'};***
  - ***char sistema[]= "Linux";***
- As duas declarações acima têm o mesmo efeito, mas a segunda é mais utilizada por ser mais fácil.
- Note que nesse caso, não existe a necessidade de informar o tamanho da cadeia.
  - *O próprio compilador vai se encarregar de determinar o tamanho necessário para a cadeia.*



Strings: inicialização (leitura)

- Podemos inicializar uma cadeia com o scanf:

```
char str[50];
scanf("%s", str);
```

  - *Atenção: nesse uso da função scanf, o comando só vai ler até o primeiro espaço da frase e portanto múltiplas palavras não serão armazenadas em str!*

```
scanf("%[^\n]", str);
```

  - *Para resolver o problema, utilizamos uma expressão regular, em que serão lidos caracteres até que seja pressionada a tecla ENTER (\n).*

UTFPR

Strings: inicialização (leitura)

- Quando utilizando o scanf com cadeias, não devemos utilizar o caractere "&" e nem o par de caracteres "[ ]", apenas o nome da string!

```
scanf("%[^\n]", str);
```

  - *Diferentemente dos vetores numéricos, aqui não vamos preencher as posições uma a uma manualmente*

- A função scanf vai se encarregar de colocar cada caractere no seu índice respectivo e colocar o marcador \0 ao final do preenchimento!

UTFPR

Strings: atribuição

- Por se tratar de um vetor, a atribuição deve ser feita posição a posição.
  - *Mas como estamos trabalhando com palavras, existe uma função que faz a cópia da palavra completa de uma vez só para uma cadeia, a STRCPY.*

```
char strA[10];
strcpy(strA, "Teste123");
```

- Nesse exemplo a literal Teste123 será copiada e armazenada na cadeia strA, sobrescrevendo seu conteúdo anterior, se houver.
  - *O marcador de final (\0) será adicionado automaticamente.*

UTFPR

Strings: cópia

- A função STRCPY também pode ser utilizada para copiar valores de uma cadeia para outra.

```
char strA[10];
char strB[] = "Teste123";
strcpy(strA, strB);
```

- Nesse exemplo o conteúdo da string strB será copiado para a strA, sobrescrevendo qualquer caractere nas posições existentes.
  - *Só serão sobrescritos os caracteres necessários para copiar a palavra completa e o marcador final \0.*

UTFPR

Strings: concatenação

- Para juntar duas strings existe o comando **STRCAT**:

```
strcat(str1, str2);
```

- A função STRCAT concatena (junta) a str2 na str1. Lembre-se: é preciso ter espaço suficiente na str1!

```
strncat(str1, str2, n);
```

- A função **STRNCAT** concatena **apenas os n primeiros** caracteres da string str2 na str1.
- Ambas as funções estão presentes na biblioteca **`string.h`**!

UTFPR

Strings: comparação

- A função que compara cadeias retorna um número inteiro. É a função **STRCMP**.

```
int resultado = strcmp(str1, str2);
```

- A variável **resultado** vai conter:
  - ***Zero*** *se as duas cadeias forem iguais;*
  - *Um número negativo se a str1 for alfabeticamente menor que str2;*
  - *Um número positivo se a str1 for alfabeticamente maior que str2;*

UTFPR

Strings: comparação

- Também podem ser comparados apenas os primeiros caracteres de duas cadeias usando a função **STRNCMP:**

```
int resultado = strncmp(str1, str2, n);
```

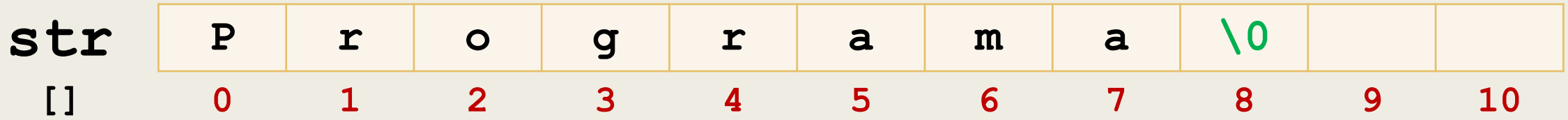
- Assim como na função anterior, quando os **n** primeiros caracteres forem iguais, o retorno será 0.
- Caso contrário um número positivo ou negativo será retornado.

Strings: tamanho

- Para descobrir quantos caracteres estão ocupados em uma cadeia, usamos a função STRLEN.

```
char str[11];
strcpy(str, "Programa");
int tamanho = strlen(str);
```

- Qual o valor armazenado em tamanho?
  - *A resposta é 8 (apenas os caracteres ocupados)!*



Strings: funções adicionais

- **STRREV** – inverte todos os caracteres da cadeia
  - *strrev(str);*
- **TOLOWER** – converte um caractere da cadeia para minúsculo. Utilize em um laço de repetição para converter a cadeia completa!

```
for(i=0; i<strlen(str); i++){
    str[i] = tolower(str[i]);
}
```

- **TOUPPER** – converte um caractere da cadeia para maiúsculo. Utilize em um laço de repetição para converter a cadeia completa!

```
for(i=0; i<strlen(str); i++){
    str[i] = toupper(str[i]);
}
```

UTFPR