Strings

Algoritmos e Estruturas de Dados I

Prof. Cristiano Rodrigues

Prof. Lucas Malacarne Astore

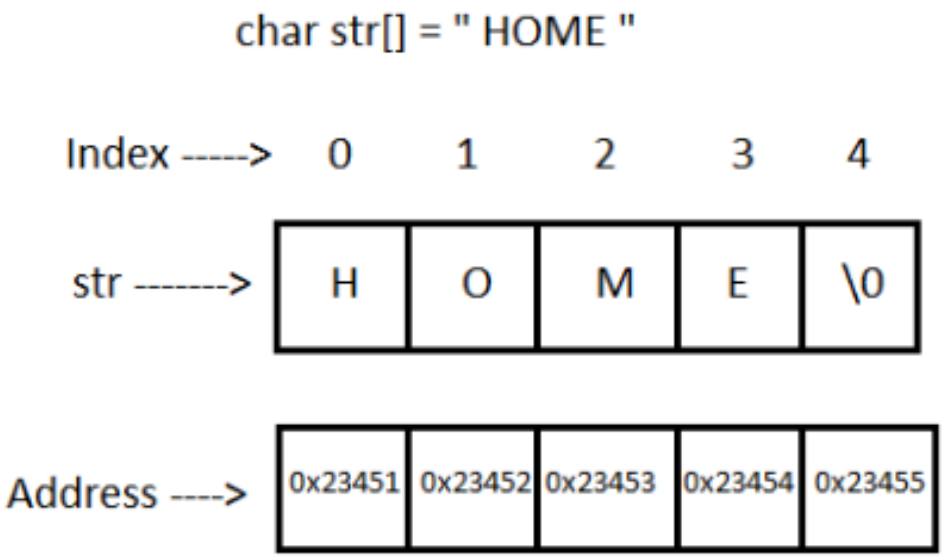
Vetores de Caracteres

String é uma sequência de caracteres tratada como um único item de dados e terminada por um caractere nulo '\0'.

Uma **string** é, na verdade, um vetor unidimensional de caracteres em linguagem C.

Vetores de Caracteres

A string "home" contém 5 caracteres incluindo o caractere '\0' que é automaticamente adicionado no final da string.



Declaração e inicialização de variáveis

```
// valid
char name[13] = "StudyTonight";
char name[10] = {'c','o','d','e','\0'};

// Illegal
char ch[3] = "hello";
char str[4];
str = "hello";
```

Declaração e inicialização de variáveis

Especificador de formato `%s` para ler uma entrada de string do terminal

A função `scanf()` termina sua entrada no primeiro espaço em branco que encontra

Solução: ditar conjunto de código de conversão, similar, mas não idual a "`%[..]`" que pode ser usado para ler uma linha contendo uma variedade de caracteres, incluindo espaços em branco

String

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

int main(){
     char str[20];
     printf("Digite uma string: ");
     scanf("%[^\n]", str);
     printf("%s \n", str);
     return 0;
}
```

String

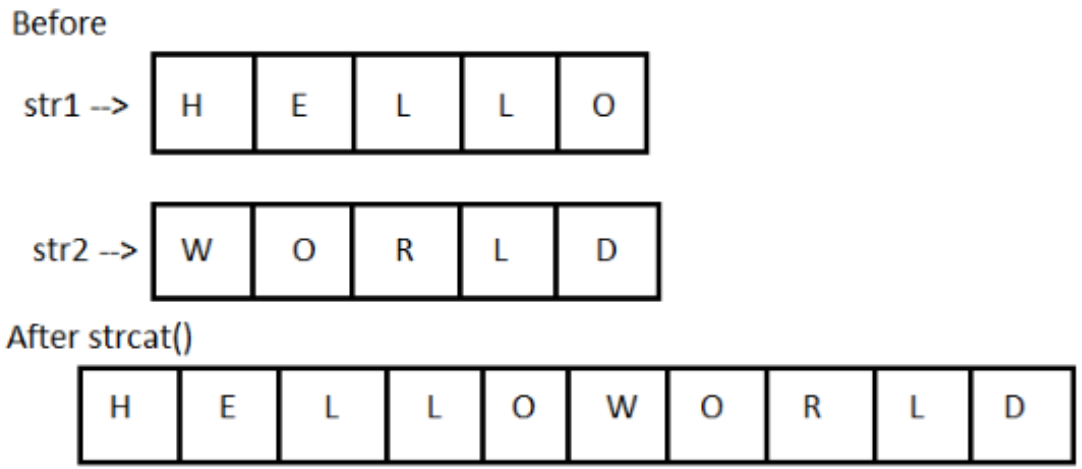
```c
#include <stdio.h>
#include <string.h>

int main(){
	char str[20];
	printf("Digite uma string: ");
	fgets(str, sizeof(str), stdin);
	printf("%s \n", str);
	return 0;
}
```

Exemplos de funções da biblioteca <string.h>

strcat - concatenar

`strcat` adcionar uma string a outra.

```
strcat("hello", "world");
```


Before
str1 -->
H
E
L
L
O
str2 -->
W
O
R
L
D
After strcat()
H
E
L
L
O
W
O
R
L
D

strlen e strcmp

`strlen` retorna o tamanho da string

`strcmp` compara duas strings caractere por caractere. Retorna 0 se são iguais, <0 se a primeira for menor que a segunda e >0 se a primeira for maior que a segunda.

String 1	String 2	Comparação	Retorno
"abc"	"abc"	iguais até o final	0
"abc"	"abd"	'c' (99) < 'd' (100)	-1
"apple"	"banana"	'a' (97) < 'b' (98)	-1
"Zebra"	"apple"	'Z' (90) < 'a' (97)	-7 (*)

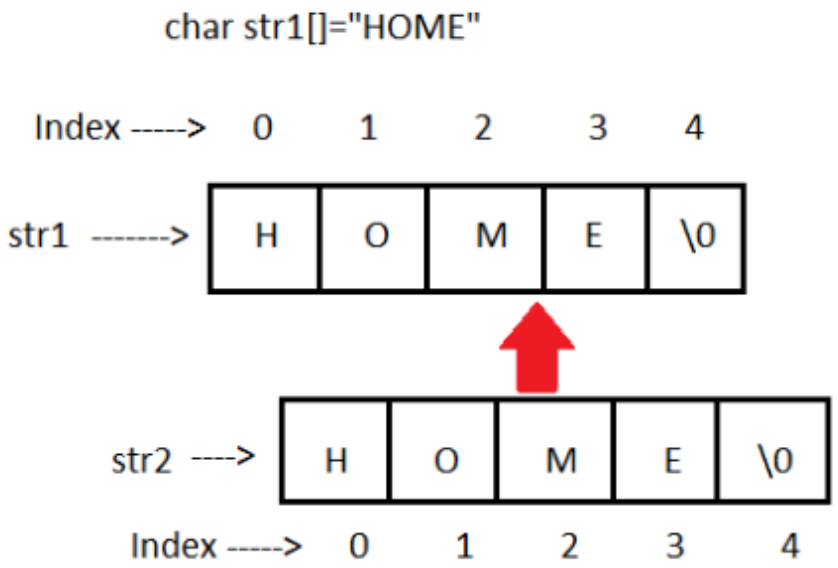
- A comparação é **interrompida no primeiro caractere diferente.**

`strlen` retorna o tamanho da string

`strcmp` compara duas strings caractere por caractere. Retorna 0 se são iguais, <0 se a primeira for menor que a segunda e >0 se a primeira for maior que a segunda.

```
int j = strlen("studytonight");
int i=strcmp("study ", "tonight");
printf("%d %d",j,i);
```

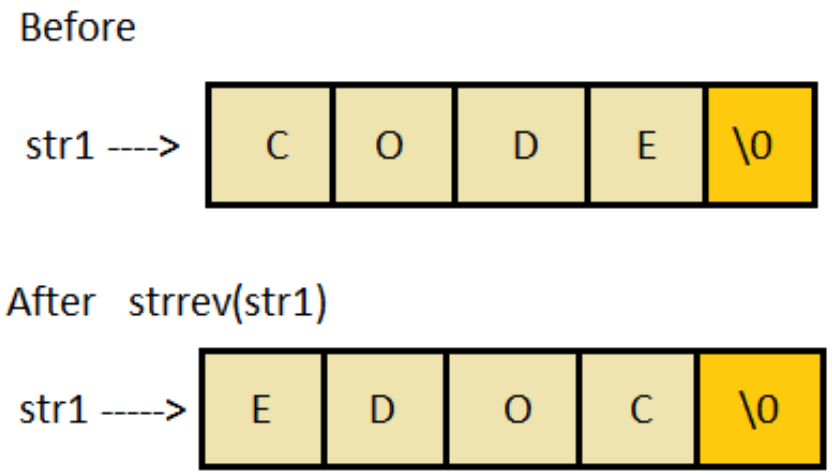
`strcpy` usada para copiar o conteúdo de uma string para outra.



strcpy

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

int main()
{
    char s1[50], s2[50];

    strcpy(s1, "StudyTonight");
    strcpy(s2, s1);

    printf("%s\n", s2);

    return(0);
}
```

`strrev` Pode ser usada para reverter uma dada string.



`strtok` Separa uma string em palavras

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

int main ()
{
  char str[] ="- This, a sample string.";
  char * pch;
  printf ("Splitting string \"%s\" into tokens:\n",str);
  pch = strtok (str," ,.-");
  while (pch != NULL)
  {
    printf ("%s\n",pch);
    pch = strtok (NULL, " ,.-");
  }
  return 0;
}
```

Função	Descrição curta	Protótipo	Observação importante
strcat	Concatena (junta) duas strings.	char *strcat(char *dest, const char *src);	Adiciona src no final de dest. dest deve ter espaço suficiente.
strlen	Retorna o tamanho da string (sem contar o \0).	size_t strlen(const char *str);	Só mede o tamanho, não altera a string.
strcmp	Compara duas strings.	int strcmp(const char *str1, const char *str2);	Retorna <0, 0 ou >0 dependendo da ordem alfabética.
strcpy	Copia uma string para outra.	char *strcpy(char *dest, const char *src);	dest deve ter espaço suficiente. Cuidado para não sobrescrever memória.
strrev	Inverte a string	char *strrev(char *str);	Algumas bibliotecas têm; pode precisar implementar manualmente.
strtok	Divide a string em "tokens" baseado em delimitadores.	char *strtok(char *str, const char *delim);	Modifica a string original; não é segura para multithreading.

EXEMPLOS

1) Um programa que leia uma string e a imprima

```c
#include <stdio.h>

int main() {
    char texto[100];

    printf("Digite uma string: ");
    scanf("%99[^\n]", texto);

    printf("String digitada: %s\n", texto);

    return 0;
}
```

EXEMPLOS

2) Um programa que mostra o comprimento de uma string (sem a função strlen)

```c
int main() {
    char texto[100];
    int comprimento = 0;

    printf("Digite uma string: ");
    scanf("%99[^\n]", texto);

    while (texto[comprimento] != '\0') {
        comprimento++;
    }

    printf("O comprimento da string é: %d\n", comprimento);

    return 0;
}
```

EXEMPLOS

3) O que o programa abaixo faz?

```c
#include <stdio.h>

int main() {
    char nome[100];

    printf("Digite um nome: ");
    scanf("%99[^\n]", nome);

    printf("%.4s\n", nome);

    return 0;
}
```

4) Um programa em C que confere se uma pessoa digitou a senha corretamente

Vamos na tabela ver qual das funções a gente pode usar pra auxiliar a resolver o problema..

4) Um programa em C que confere se uma pessoa digitou a senha corretamente

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

#define SENHA_CORRETA "minhasenha123"

int main() {
    char senha_digitada[50];

    printf("Digite a senha: ");
    scanf("%[^\n]", senha_digitada);

    if (strcmp(senha_digitada, SENHA_CORRETA) == 0) {
        printf("Acesso permitido!\n");
    } else {
        printf("Senha incorreta. Acesso negado!\n");
    }

    return 0;
}
```

EXEMPLOS

5) O que o programa abaixo faz?

```c
int main() {
    char str[100];
    int i, j;
    char temp;

    printf("Digite uma string: ");
    scanf("%[^\n]", str);
    int tamanho = strlen(str);

    for (i = 0, j = tamanho - 1; i < j; i++, j--) {
        temp = str[i];
        str[i] = str[j];
        str[j] = temp;
    }

    printf("%s\n", str);

    return 0;
}
```