CS10003:
Programming & Data Structures

Dept. of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Autumn 2020

Strings

Strings

- 1-d arrays of type char
- By convention, a string in C is terminated by the end-of-string sentinel '\0' (null character)
- char s[21] - can have variable length string delimited with \0
  - Max length of the string that can be stored is 20 as the size must include storage needed for the '\0'
- String constants : "hello", "abc"
- "abc" is a character array of size 4

String Constant

- A string constant is treated as a pointer
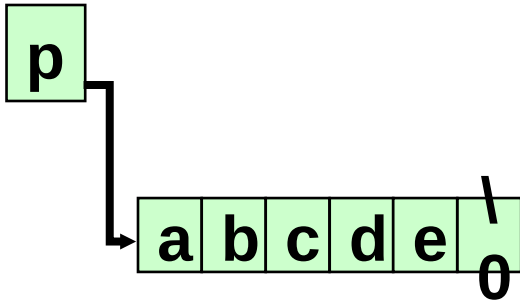- Its value is the base address of the string

  char *p = "abc";
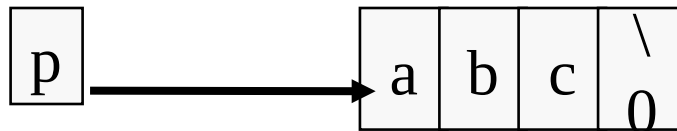
  printf ("%s %s\n",p,p+1); /* abc bc is printed */

Differences : array & pointers
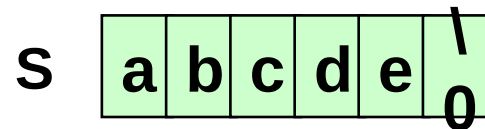
char *p = "abcde";

The compiler allocates space for p, puts the string constant "abcde" in memory somewhere else, initializes p with the base address of the string constant

char s[ ] = "abcde";

≡ char s[ ] = {'a','b','c','d','e'.'\0'};

The compiler allocates 6 bytes of memory for the array s which are initialized with the 6 characters

p

| a | b | c | d | e | \0 |

s | a | b | c | d | e | \0 |

5

Library Functions for String Handling

- You can write your own C code to do different operations on strings like finding the length of a string, copying one string to another, appending one string to the end of another etc.

- C library provides standard functions for these that you can call, so no need to write your own code

- To use them, you must do

  #include <string.h>

  At the beginning of your program (after #include <stdio.h>)

String functions we will see

- strlen : finds the length of a string
- strcat : concatenates one string at the end of another
- strcmp : compares two strings lexicographically
- strcpy : copies one string to another

strlen()

int strlen(const char *s)

■ Takes a null-terminated string (we routinely refer to the char pointer that points to a null-terminated char array as a string)

■ Returns the length of the string, not counting the null (\0) character

You cannot change contents of s in the function

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

strcat()

■ char *strcat (char *s1, const char *s2);

■ Takes 2 strings as arguments, concatenates them, and puts the result in s1. Returns s1. Programmer must ensure that s1 points to enough space to hold the result.

```
char *strcat(char *s1, const char *s2)
{
    char *p = s1;
    while (*p != '\0')   /* go to end */
    ++p;
    while(*s2 != '\0')
        *p++ = *s2++;  /* copy */
    *p = '\0';
    return s1;
}
```

9

Dissection of the strcat() function

char *p = s1;

p is being initialized, not *p. The pointer p is initialized to the pointer value s1. Thus p and s1 point to the same memory location

Dissection of the strcat() function

char *p = s1;

p is being initialized, not *p. The pointer p is initialized to the pointer value s1. Thus p and s1 point to the same memory location

while (*p != '\0') ++p;

As long as the value pointed to by p is not '\0', p is incremented, causing it to point at the next character in the string. When p points to \0, the control exits the while statement

Dissection of the strcat() function

char *p = s1;

p is being initialized, not *p. The pointer p is initialized to the pointer value s1. Thus p and s1 point to the same memory location

while (*p != '\0') ++p;

As long as the value pointed to by p is not '\0', p is incremented, causing it to point at the next character in the string. When p points to \0, the control exits the while statement

while(*s2 != '\0') *p++ = *s2++; /* copy */

At the beginning, p points to the null character at the end of string s1. The characters in s2 get copied one after another until end of s2

Dissection of the strcat() function

char *p = s1;

p is being initialized, not *p. The pointer p is initialized to the pointer value s1. Thus p and s1 point to the same memory location

while (*p != '\0') ++p;

As long as the value pointed to by p is not '\0', p is incremented, causing it to point at the next character in the string. When p points to \0, the control exits the while statement

while(*s2 != '\0') *p++ = *s2++; /* copy */

At the beginning, p points to the null character at the end of string s1. The characters in s2 get copied one after another until end of s2

*p = '\0';   put the '\0' at the end of the string

13

strcmp()

int strcmp (const char
  *s1, const char *s2);

Two strings are passed
  as arguments. An
  integer is returned
  that is less than,
  equal to, or greater
  than 0, depending on
  whether s1 is
  lexicographically less
  than, equal to, or
  greater than s2.

strcmp()

int strcmp (const char *s1, const char *s2);

Two strings are passed as arguments. An integer is returned that is less than, equal to, or greater than 0, depending on whether s1 is lexicographically less than, equal to, or greater than s2.

```
int strcmp(const char *s1, const char *s2)
{
    for (;*s1!='\0'&&*s2!='\0'; s1++,s2++)
    {
     if (*s1>*s2) return 1;
     if (*s2>*s1) return -1;
    }
    if (*s1 != '\0') return 1;
    if (*s2 != '\0') return -1;
    return 0;
}
```

strcpy()

char *strcpy (char *s1, const char *s2);

The characters in the string s2 are copied into s1 until '\0' is moved. Whatever exists in s1 is overwritten. It is assumed that s1 has enough space to hold the result. The pointer s1 is returned.

strcpy()

char *strcpy (char *s1, const char *s2);

The characters in the string s2 are copied into s1 until '\0' is moved. Whatever exists in s1 is overwritten. It is assumed that s1 has enough space to hold the result. The pointer s1 is returned.

```
char * strcpy (char *s1, const char *s2)
{
    char *p = s1;
    while (*p++ = *s2++) ;
    return s1;
}
```

Example: Using string functions

```
int main()
{
 char s1[ ] = "beautiful big sky country",
      s2[ ] = "how now brown cow";
 printf("%d\n",strlen (s1));
 printf("%d\n",strlen (s2+8));
 printf("%d\n", strcmp(s1,s2));
 printf("%s\n",s1+10);
 strcpy(s1+10,s2+8);
 strcat(s1,"s!");
 printf("%s\n", s1);
 return 0;
}
```

Output

```
25
9
-1
big sky country
beautiful brown cows!
```