APS105 Lecture 16
7.1, 7.2 without putchar/getchar, 7.3 strlen

Dan Zingaro

March 3, 2010

Feedback from Reading Quiz

- "When is [the NULL character] present and when is it not?"
  - It is present whenever we create a string
- "If a string is `char friend[]`, does the first character of the string start from `friend[1]` rather than `friend[0]`?"
  - No. A string is an array, so it starts at index 0
- `char a[3] = {'D', 'A', 'n', '\0'};`
  - Is this a string?

Creating Strings

- String: array of characters that ends with the special character '\0'
- Without the '\0', we would not be able to detect the end of the string
- So, to store a string of length n, we use an array of at least size n+1
- One way to create a string:

```
char name[4];
name[0] = 'd';
name[1] = 'a';
name[2] = 'n';
name[3] = '\0';
```

Creating Strings...

Instead of specifying each character, we can initialize an array with a string. C automatically adds the `'\0'`:

```
char name[] = "Dan";
```

We can also specify an array size:

```
char name[10] = "dan";
```

- C adds seven `'\0'` characters after the three characters in the string

String Constants

We can also create string constants, and refer to them through char pointers.

```
char *name2 = "joe";
```

- Again, C adds the `'\0'` terminator to the end of the string
- We can still obtain characters of the string (e.g. `name2[1]`)
- But we are not allowed to change the characters in the string

```
name2[2] = 't'; //No!
```

ConcepTest

Fill in the comment with the correct code.

```
char s[50];
char *t = "abc";
//make s the string "abc"
```

- A.
  ```
  s = "abc";
  ```
- B.
  ```
  s = t;
  ```
- C.
  ```
  s[0] = t[0];
  s[1] = t[1];
  s[2] = t[2];
  ```
- D.
  ```
  s[0] = t[0];
  s[1] = t[1];
  s[2] = t[2];
  s[3] = t[3];
  ```
- E. A or B

Writing Strings

- `printf` can be used to write a string, with the %s placeholder

```
char s[] = "hi";
printf ("The string is: %s\n", s);
```

Reading Strings

- `scanf` can be used to read strings, with a `%s` placeholder
- `scanf` skips leading whitespace, then begins storing the characters it reads
- `scanf` adds the `'\0'` terminator

```
char t[10];
scanf ("%s", t);
```

Reading Strings...

- We generally do not use this form of `scanf` to read strings, for two reasons
- Problem 1: `scanf` stops reading when it gets a whitespace character, so it will read only the first word into the string
  - Solution: use `gets`; it reads until a newline, but does not store the newline in the string
  - `gets` stores any leading whitespace in the string
  - `gets` adds the `'\0'` terminator

```
char t[10];
gets (t);
```

Reading Strings...

- Problem 2: nothing stops the user from entering more characters than can be stored in the string
- This will overwrite other data and cause undefined program behavior
  - Solution: use `fgets`; it takes a parameter indicating the maximum length of the string to store
  - The array should have at least this number of elements

```
#define T_LENGTH 10;
char t[T_LENGTH];
fgets (t, T_LENGTH, stdin);
```

- `fgets` stops reading when it hits a newline, or when it reads `T_LENGTH - 1` characters
- If `fgets` reads a newline, it stores it in the string
- `fgets` adds a `'\0'` terminator

ConcepTest

Assume that the user types `abc defg`, but that only `abc` is stored in the string `s`. Which of the following could have been used to read the string?

- A. `scanf ("%s", s);`
- B. `gets (s);`
- C. `fgets (s, 3, stdin);`

Length of a String

```
size_t strlen (const char *s)
```

- `strlen` is a function that returns the length of a string
- Length of a string: number of characters that are found prior to the first `'\0'` character

```
char greet[10] = "hi";
int i = strlen (greet) //stores 2
```

ConcepTest

What is the result of this code?

```
char s[] = "hi\0hey";
```

- A. The three characters `'h', 'i', '\0'` are stored in `s`; `strlen (s) == 2`
- B. The seven characters `'h', 'i', '\0', 'h', 'e', 'y', '\0'` are stored in `s`; `strlen (s) == 2`
- C. The seven characters `'h', 'i', '\0', 'h', 'e', 'y', '\0'` are stored in `s`; `strlen (s) == 6`
- D. Error!