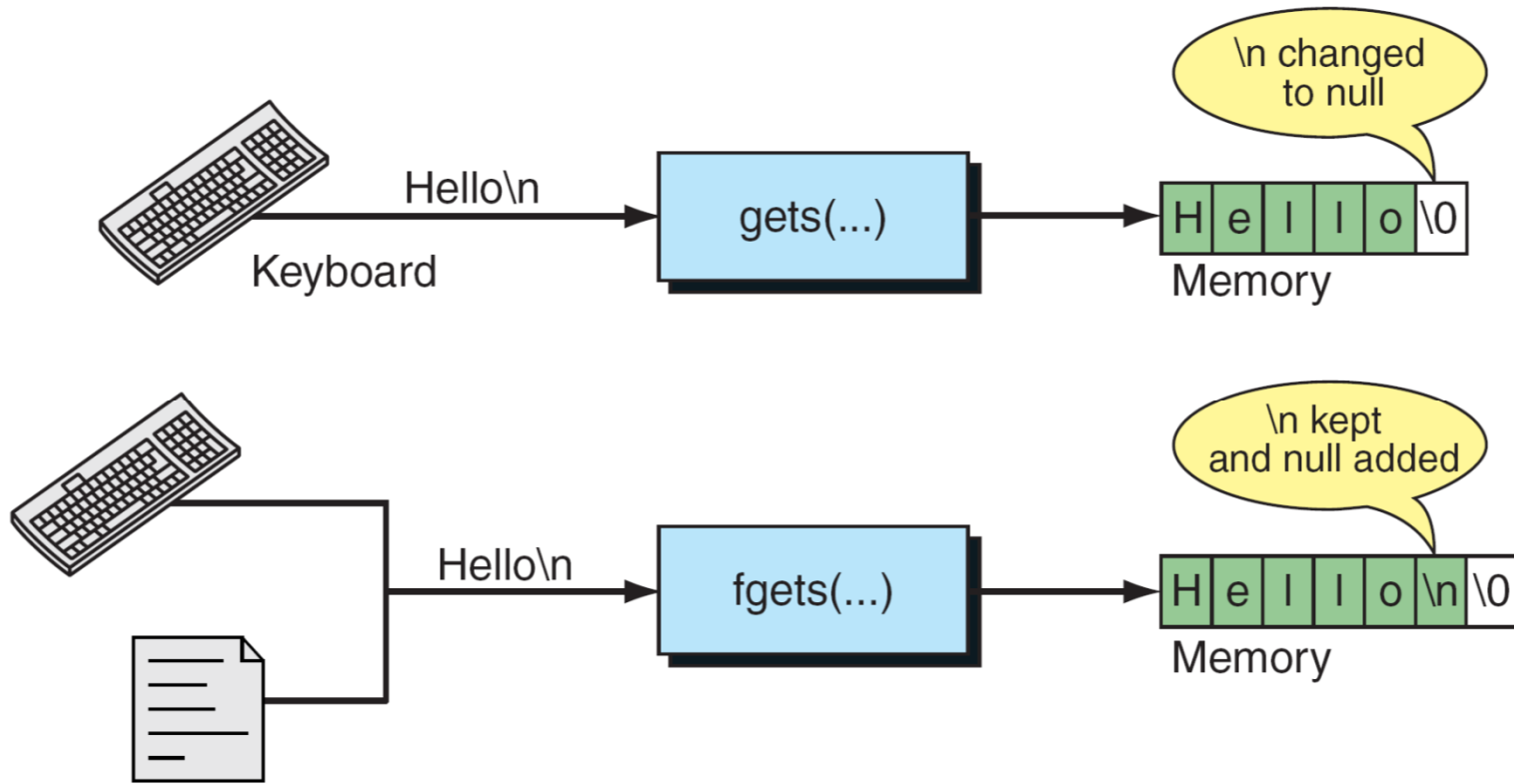


# Strings & Character Arrays

## Part - 2



# gets ( ) & fgets ( )



*from Figure 11-11 in Forouzan & Gilberg, p. 682*

# Determining String Length

- We can use the `strlen()` (from `string.h`) function to determine the length of a string.
- It counts the number of characters in a string, excluding the null character.
- To use `strlen()`:  
`strCnt = strlen(myString);`

# Copying Strings

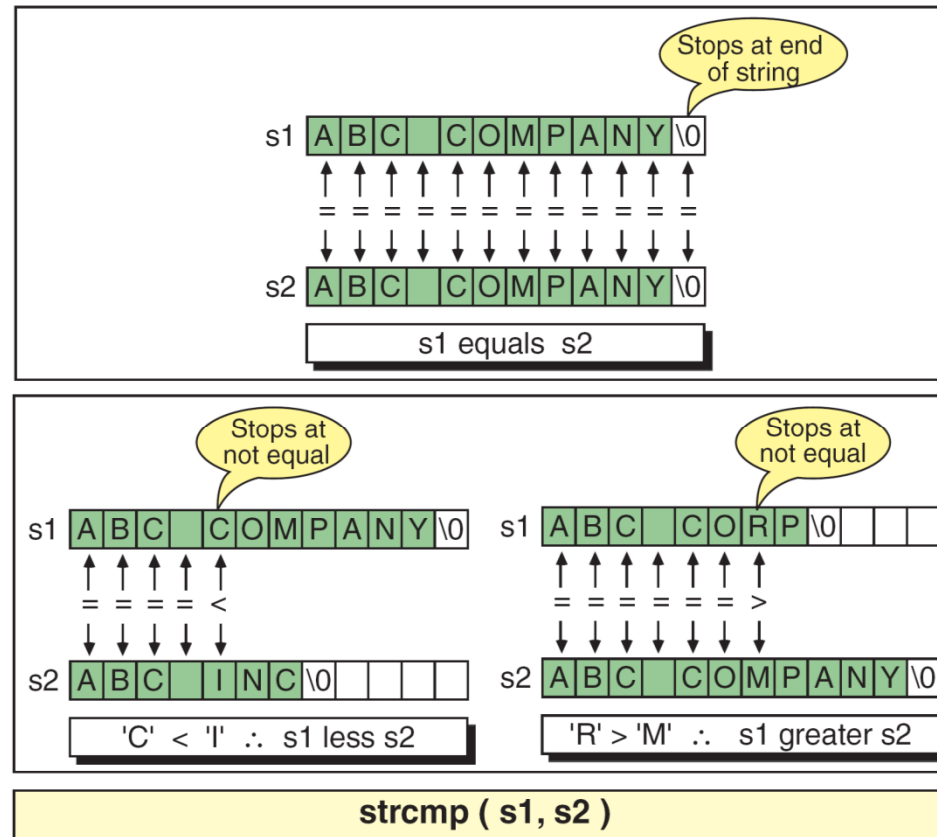
- Remember, we cannot simply assign one string to another due to the fact that strings are character arrays!
- However, we can use the `strcpy( )` function from `string.h` to copy one string to another:

```
strcpy(destinationStr, sourceStr);
```

# Comparing Strings

- We can use the `strcmp( )` function, from `string.h`, to compare two strings:  
`strcmp(str1, str2) ...`
- If the two strings are equal, `strcmp( )` returns `0`.
- If `str1` is greater than `str2`, `strcmp( )` returns a positive number.
- If `str1` is less than `str2`, `strcmp( )` returns a negative number.

# Comparing Strings



*from Figure 11-17 in Forouzan & Gilberg, p. 698*

# Combining Strings

- We can use the `strcat()` function to combine two strings into a new string.
- We need to be certain that the array to which we assign the resulting string is large enough to hold all the characters from the two contributing strings.
- To use `strcat()`:  
`strcat(str1, str2);`
- The above example adds the value of `str2` to the end of `str1`, replacing the delimiter of `str1`.

## Other Built-in Functions

- `memchr();`

- `memcmp();`

- `memcpy();`

- `memmove();`

- `memset();`

`strcat();`

`strncat();`

`strchr();`

`strcmp();`

`strncmp();`



## Other Built-in Functions

- `strcoll();`

- `strcpy();`

- `strncpy();`

- `strcspn();`

- `strerror();`

- `strxfrm();`

`strlen();`

`strpbrk();`

`strrchr();`

`strspn();`

`strstr();`

`strtok();`

# Questions?

