## Manipulating C Strings

<table>
<thead>
<tr>
<th>String Operation</th>
<th>String in Stack</th>
<th>String in Rodata Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Allocating memory for a string</strong></td>
<td><code>{ char ac[5]; ... }</code></td>
<td><code>{ ... &quot;hi&quot;... ... }</code></td>
</tr>
</tbody>
</table>
| **Initializing a string** | ```
char acA[3] = {'h', 'i', '\0'};
char acB[] = {'h', 'i', '\0'};
char acD[10] = {'h', 'i', '\0'}/ *warning*/
char acE[3] = "hi";
char acF[] = "hi";
char acG[2] = "hi"; /* no warning!!! */
char acH[10] = "hi";
... ``` | ```
... "hi"...
... ``` |
| **Computing the length of a string** | ```
char ac[20] = "hello, world";
/* Evaluates to 12 */
... strlen(ac) ...
/* Evaluates to 20 */
... sizeof(ac) ...
``` | ```
char *pc = "hello, world";
/* Evaluates to 12 */
... strlen(pc) ...
/* Evaluates to 8 */
... sizeof(pc) ...
``` |
| **Changing the characters of a string** | ```
char ac[10] = "hi";
/* Compile-time error. */
ac = "bye";
/* The long way. */
ac[0] = 'b';
ac[1] = 'y';
ac[2] = 'e';
ac[3] = '\0';
/* The shortcut. */
strcpy(ac, "bye"); /* Dangerous. */
``` | (Runtime error to attempt to change the characters of a string that resides in the rodata section) |
| **Concatenating characters onto a string** | ```
char ac[10] = "hi";
/* Compile-time error. */
ac += "bye";
/* The long way. */
ac[2] = 'b';
ac[3] = 'y';
ac[4] = 'e';
ac[5] = '\0';
/* The shortcut. */
strcat(ac, "bye"); /* Dangerous. */
``` | (Runtime error to attempt to change the characters of a string that resides in the rodata section) |
### Comparing one string with another

```c
{  char acA[] = "hi";
    char acB[] = "bye";

    /* Legal, but compares addresses!!! */
    if (acA < acB) ...

    /* Compares strings */
    if (strcmp(acA, acB) < 0) ...
}
```

(Same as string in stack)

### Reading a string

```c
{  char ac[10];

    /* Reads a word as a string. */
    iConvCount = scanf("%s", ac);
    /* Dangerous. */

    /* Reads a line as a string, removing the \n character. */
    iRet = gets(ac);
    /* Dangerous. */

    /* Reads a line as a string, retaining the \n character. */
    iRet = fgets(ac, 10, stdin);
}
```

(Runtime error to attempt to change the characters of a string that resides in the rodata section)

### Writing a string

```c
{  char ac[] = "hi";

    /* Writes a string. */
    iCharCount = printf("%s", acStr);

    /* Writes a string, appending a \n character. */
    iSuccessful = puts(ac);

    /* Writes a string. */
    iSuccessful = fputs(ac, stdout);
}
```

(Same as string in stack)

### Converting a string to another type

```c
{  char ac[] = "123";
    int i;
    long l;
    double d;

    iConvCount = sscanf(ac, "%d", &i);
    i = atoi(ac);
    l = atol(ac);
    d = atof(ac);
}
```

(Same as string in stack)

### Converting another type to a string

```c
{  char ac[10];
    int i = 123;

    iCharCount = sprintf(ac, "%d", i);
    /* Dangerous. */
}
```

(Runtime error to attempt to change the characters of a string that resides in the rodata section)

---

Copyright © 2016 by Robert M. Dondero, Jr.