## String Operations

<table>
<thead>
<tr>
<th>String Operation</th>
<th>String in Stack</th>
<th>String in Rodata Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocating memory for a string</td>
<td><code>{ char pcStr[5]; ... }</code></td>
<td><code>{ ... &quot;hi&quot;... ... }</code></td>
</tr>
<tr>
<td>Initializing a string</td>
<td><code>{ char pcStr1[3] = {'h', 'i', '\0'}; char pcStr2[] = {'h', 'i', '\0'}; char pcStr3[3] = &quot;hi&quot;; char pcStr4[] = &quot;hi&quot;; char pcStr5[2] = &quot;hi&quot;; /* truncation */ char pcStr6[20] = &quot;hi&quot;; ... }</code></td>
<td><code>{ ... &quot;hi&quot;... ... }</code></td>
</tr>
<tr>
<td>Computing the length of a string</td>
<td><code>{ char pcStr[20] = &quot;hi&quot;; ... sizeof(pcStr) ... /* Evaluates to 20 */ ... strlen(pcStr) ... /* Evaluates to 2 */ }</code></td>
<td><code>{ char *pcStr = &quot;hi&quot;; ... sizeof(&quot;hi&quot;) ... ... sizeof(pcStr) ... /* Evaluate to 4 */ ... strlen(&quot;hi&quot;) ... ... strlen(pcStr) ... /* Evaluate to 2 */ }</code></td>
</tr>
<tr>
<td>Changing the characters of a string</td>
<td><code>{ char pcStr[20] = &quot;hi&quot;; pcStr = &quot;bye&quot;; /* compiletime error */ pcStr[0] = 'b'; pcStr[1] = 'y'; pcStr[2] = 'e'; pcStr[3] = '\0'; strcpy(pcStr, &quot;bye&quot;); /* Danger of memory corruption. */ }</code></td>
<td><code>{Runtime error to attempt to change the characters of a string that resides in the rodata section}</code></td>
</tr>
</tbody>
</table>
### Comparing one string with another

```c
{ 
    char pcStr1[] = "hi";
    char pcStr2[] = "bye";

    if (pcStr1 < pcStr2) ...  
    /* Legal, but compares pointers!!! */
    if (strcmp(pcStr1, pcStr2) < 0) ... 
    /* Compares strings */
}
```

(Same as string in stack)

### Reading a string

```c
{ 
    char pcStr[20];

    iConvCount = scanf("%s", pcStr);
    /* Reads a word as a string. 
       Grave danger of memory corruption. */

    iRet = gets(pcStr);
    /* Reads a line as a string, 
       removing the \n character. 
       Grave danger of memory corruption. */

    iRet = fgets(pcStr, 20, stdin);
    /* Reads a line as a string, 
       retaining the \n character. */
}
```

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

### Writing a string

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

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

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

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

(Same as string in stack)

### Converting a string to another type

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

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

(Same as string in stack)

### Converting another type to a string

```c
{ 
    char pcStr[20];
    int i = 123;

    iCharCount = sprintf(pcStr, "%d", i);
    /* Danger of memory corruption. */
}
```

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

---

Copyright © 2004 by Robert M. Dondero, Jr.