## Princeton University

**COS 217: Introduction to Programming Systems**

**Manipulating C Strings**

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