Western University
Computer Science Department

Part Six: String Functions

The String Library in C

String functions are provided in an ANSI standard string library.

Access this through the include file:

```
#include <string.h>
```

Includes functions such as:

Computing length of string
Copying strings
Concatenating strings

This library is guaranteed to be there in any ANSI standard implementation of C.

The String Library in C

strlen returns the length of a NULL terminated character string:

```
int count;
char d[8] = “Magic”;
/* char: 1 byte */
```

Defined in string.h

```
count = strlen(d);
```

- Returns 5 (even though there are 6 values if you

Label	Address	Value
d[0]	400	M
d[1]	401	a
d[2]	402	g
d[3]	403	i
d[4]	404	c
d[5]	405	\0
d[6]	406	0
d[7]	407	0

The String Library in C

strcpy copies a character string into another string:

A copy of source is made at destination

source should be NULL terminated

destination should have enough room

(its length should be at least the size of source)

The return value also points at the destination.

The String Library in C

strcpy example

output:
```
C programming
```

```
#include <stdio.h>
#include <string.h>

int main() {
    char str1[20] = "
    char str2[20];

    // copying str1 to str2
    strcpy(str2, str1);

    puts(str2); // C programming

    return 0;
}
```

Note: When you use strcpy() the size of the destination string should be large enough to store the copied string.

Otherwise, it may result in **undefined behavior**

The String Library in C

strcat : included in <string.h>:

Appends a copy of str2 to the end of str1
A pointer equal to str1 is returned

Ensure that str1 has sufficient space for the concatenated string!

Array index out of range will be the most popular bug in your C programming career.

The String Library in C

strcat example

output:

```
This is programiz.com
programiz.com
```

```
#include <stdio.h>
#include <string.h>
int main() {
   char str1[100] = "This

   // concatenates str1 ar
   // the resultant string is stored in str1.
   strcat(str1, str2);

   puts(str1);
   puts(str2);

   return 0;
}
```

Note: When we use strcat() the size of the destination string should be large enough to store the resultant string.

If not, we will get the segmentation fault error.

The String Library in C

C strings can be compared for equality or inequality

If they are equal - they are ASCII identical

If they are unequal the comparison function will return an int that is interpreted as:

< 0 : str1 is less than str2
0 : str1 is equal to str2
> 0 : str1 is greater than str2

The String Library in C

Basic comparison functions:

```
int strcmp (str1, str2);
```

Does an ASCII comparison one char at a time until a difference is found between two chars in the same position.

Return value is as stated before

If both strings reach a '\0' at the same time, they are considered equal.

```
int strncmp (str1, str2, n);
```

Compares n chars of str1 and str2

Continues until n chars are compared or

The end of str1or str2 is encountered

Also have **strcasecmp()** and **strncasecmp()**

which do the same as above, but ignore case in letters.

The String Library in C

strcmp example

output:

```
strcmp(str1, str2) = 32
strcmp(str1, str3) = 0
```

```
#include <stdio.h>
#include <string.h>

int main()
{
    char str1[] = "abcd", str
    int result;

    // comparing strings str1
    result = strcmp(str1, str
    printf("strcmp(str1, str2

    // comparing strings str1
    result = strcmp(str1, str
    printf("strcmp(str1, str3) = %d\n", result);

    return 0;
}
```

The first unmatched character between string str1 and str2 is third character.

The ASCII value of 'c' is 99 and the ASCII value of 'C' is 67.
so, when strings str1 and str2 are compared, the return value is 32.

When strings str1 and str3 are compared, the result is 0 because both strings are identical.

The String Library in C

There are a number of searching functions:

```
char * strchr (char * str, int ch) ;
```

strchr search str until ch is found or NULL character is found instead.
If found, a (non-NULL) pointer to ch is returned.
Otherwise, NULL is returned instead.

You can determine its location (index) in the string by:
Subtracting the value returned from the address of the start of the string

More pointer arithmetic … more on this later!

The String Library in C

strchr example

output:

```
String after |.| is - |.tutorialspoint.com|
```

```
#include <stdio.h>
#include <string.h>

int main () {
   const char str[] = "http
   const char ch = '.';
   char *ret;

   ret = strchr(str, ch);

   printf("String after |%c| is - |%s|\n", ch, ret);

   return(0);
}
```

strchr() finds the first ‘.’ after www

strchr() returns a pointer

(pointer????)
- later …

The String Library in C

String functions are provided in an ANSI standard string library.

Access this through the include file:

```
#include <stdlib.h>
```

This header defines one variable type, one macro, and various functions for manipulating arrays of characters.

This library is guaranteed to be there in any ANSI standard implementation of C.

```
double atof(const char *str)
```

Converts the string pointed to, by the argument *str* to a floating-point number (type double).

```
int atoi(const char *str)
```

Converts the string pointed to, by the argument *str* to an integer (type int).

```
long int atol(const char *str)
```

Converts the string pointed to, by the argument *str* to a long integer (type long int).

Simple String Functions

END OF PART 4