String

Dr. Pradeep Kumar Mallick
Associate Professor-II
School of Computer Engineering, KIIT DU

String and Pointer

As we all know, the string is a character array which is terminated by the null '\0' character.

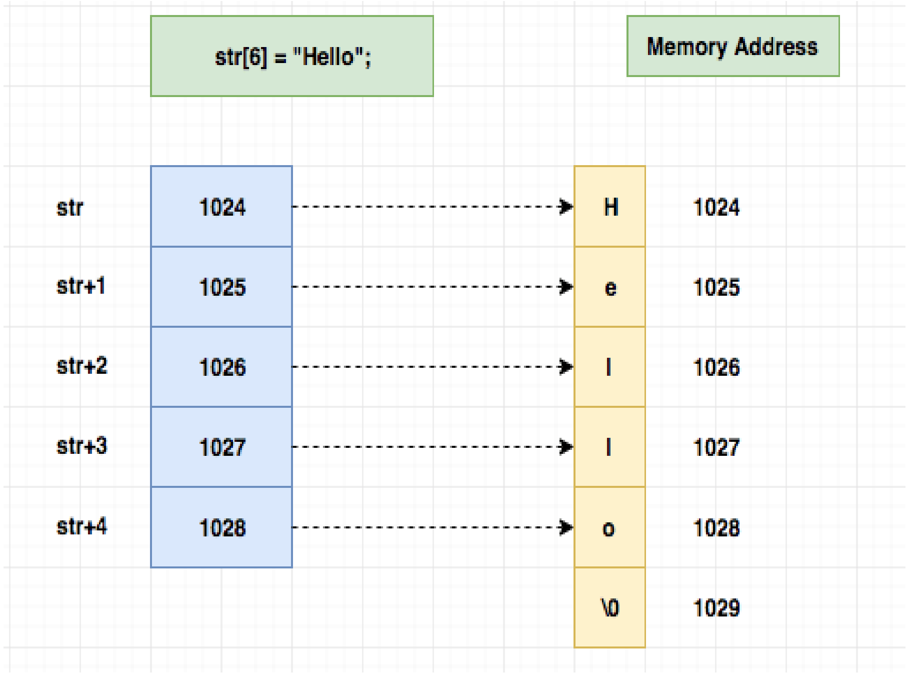
```
char str[6]="Hello";
```

Where,

- str is the string name and it is a pointer to the first character in the string. i.e. &str[0].
- Similarly, str+1 holds the address of the second character of the string. i.e. &str[1]
- To store "Hello", we need to allocate 6 bytes of memory.
- 5 byte for "Hello"
- 1 byte for null '\0' character.

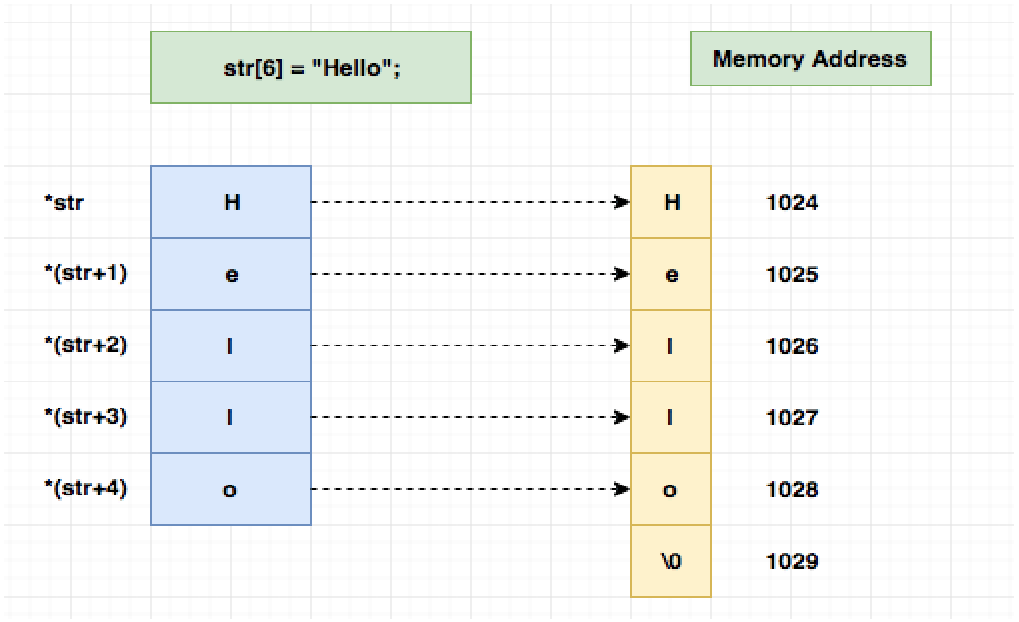
Character Array and Pointer

```
#include<stdio.h>
int main()
 {
char str[6] = "Hello";
 int i; //printing each char address
 for(i = 0; str[i]; i++)
       printf("&str[%d] = %p\n",i,str+i);
return 0;
 }
```



Character Array and Pointer

Str[i] == *(str+i)
str[i] will give the character stored at the string index i.
str[i] is a shortend version of *(str+i).
*(str+i) - value stored at the memory address str+i. (base address + index)



Character Array and Pointer

```
#include<stdio.h>
int main()
 {
          char str[6] = "Hello"; int i;     //printing each
char value
          for(i = 0; str[i]; i++)
                    printf("str[%d] = %c\n",i,*(str+i));
//str[i] == *(str+i)
          return 0;
 }
```

Accessing string using pointer

```
#include<stdio.h>
 int main()
 {
          char str[6] = "Hello";
           char *ptr; int i; //string name itself a base address of the
          string ptr = str; //ptr references str
           for(i = 0; ptr[i] != '\0'; i++)
                    printf("&str[%d] = %p\n",i,ptr+i);
           return 0;
 }
```

Library Functions

C supports a large set of library functions for manipulating the strings. These are the most commonly used string handling functions defined within the header file <string.h>

- strlen()
- strrev()
- strcpy()
- strcmp()
- strcat()
- Strlwr()
- Strupr()
- Strstr()

strlen() :

- This function is used to find the length of a string. It is defined within the header file <string.h>.This function accepts a string as its argument and returns an integer, which is the length of the string.

Q. Write a program which will accept a string from the user and will print the length of the string using library function.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
        int l;
        char str[30];
        clrscr();
        printf("\n Eenter a string : ");
        gets(str);
        l=strlen(str);
        printf("\n The length of the sting is %d ",l); }
```

Length of the string without using library function.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    int l=0,i;
    char str[30];
    clrscr();
    printf("\n Enter a string : ");
    gets(str);
    for(i=0;str[i]!='\0'; i++)
    {
            l++ ;
    }
    printf("\n The length of the sting is %d ",l);
```

strcpy() :

This function is used to copy one string into the second one. It is written in the following format.

Syntax :

```
strcpy (destination string ,source string);
```

Example :

```
strcpy(str2,str1);
```

With Library Function

Write a program to accept a string from the user, copy it to a second string and display the contains of both the string using library function .

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
        char str1[30],str2[30];
        clrscr();
        printf("\n Enter a string ");
        gets(str1);
        strcpy(str2,str1);
        printf("\n Original string is %s",str1);
        printf("\n Copied string is %s",str2);
}
```

Without Library Function

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
        int i;
        char a[80] ,b[80] ;
        printf(“Enter a string :”);
        gerts(a) ;
        for(i=0 ;a[i]!=’\0; i++)
        {
                b[i]=a[i];
        }
        b[i]=’\0’ ;
        printf(“%s \n %s” a, b);
}
```

strcmp() :

This function is used to check whether two strings are equal or not. It is written in the following format.
Syntax :

strcmp(string 1,string2);

This function returns 0 if the contains of the two strings are equal. Else it returns non zero. The header file is <string.h>.

Example :

strcmp("bls","bls");
returns 0 as the two strings are equal.

strcmp("abc","bbc");
returns -1 ,(a-b=97-98= =1).

strcmp("bbc","abc");
returns 1 (b-a=98-97=1).

With Library Function

Write a program to accept two strings and from the user and check it whether two strings are equal or not using strcmp().

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
        char str1[30], str2[30];
        int r;
        clrscr();
        printf("\n Enter 1st  strings");
        gets(str1);
        printf("\n Enter 2nd   strings");
        gets(str2);
        r=strcmp(str1,str2);
}
```

```
if(r==0)
printf("\n The two strings are
equal");
else
printf("\n The two strings are
not equal");
```

Without Library Function

```
void main()
{
char str1[30];
char str2[30];
int i=0;
clrscr();
printf("\n Enter 1st  strings:");
gets(str1);
printf("\n Enter 2nd   strings:");
gets(str2);
while(str1[i]!='\0' && str2[i]!='\0' && str1[i]==str2[i])
{
        i++;
}
if(str1[i] = = str2[i])
        printf("Equal");
else
        printf("Not equal ");

}
```

strrev() :With Library Function

Determine the reverse of a string and the result will be stored on the same variable. All characters are reverse except the Null character.

Syntax :

```
            strrev(string);
```

Write a program to input a string then reverse it using strrev().

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
            char a[50];
            printf("\nEnter a string:");
            gets(a);
            strrev(a);
            puts(a);
}
```

Without Library Function

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
            char a[50] ,b[50];
            int i,k=0;
            printf("\nEnter a string:");
            gets(a);
            for(i=strlen(a)-1; i>=0;i--)
            {
                        b[k++]=a[i];

}
b[k]='\0';
printf("\n reverse =%s", b)
}
```

strcat() :With Library Functio

1. This function is used to concatenate two strings and the result will be stored in the first string.
2. In other words we can say that it append a copy of source to end of the destination.
3. The length of the string is strlen(destination_string)+strlen(source_string) . It requires two string as its argument, separated by comma (,) and adds the content of second string (called as source string) with the content of the first string (called as destination string)

Syntax :

```
strcat(string1,string2);
```

With Library Function

Write a program to join two string using library function strcat().

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
		char str1[30];
		char str2[30];
		printf(“Enter the first string :”);
gets(str1);
printf(“Enter the 2nd  string :”);
gets(str2);
strcat(str1,str2);
printf(“%s”,str1);

}
```

Without Library Function

Write a program to join two string without using library function strcat().

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
        char str1[30];
        char str2[30];
        int i, x=0;le,,le1ef,k=0 ;
        printf("\n Enter the first string :");
gets(str1);
printf(" \n Enter the 2nd  string :");
gets(str2);
le=strlen(str1);
le1=strlen(str2);
```

```
for(i=le ;i<=le+le ;i++ )
{
        str1[i]=str2[k++] ;
}
printf("\n\n Final
string=%s",str1);

}
```

strlwr():With Library Function

It is used to convert a string to lower case.

syntax is:

strlwr(str);

Write a program input a string then convert it into lower case using strlwr().

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
		char str1[30];
		printf("\nEnter a string :");
gets(str1);
strlwr(str1);
printf("%s" , str1);

}
```

Without Library Function

Write a program input a string then convert it into lower case without using strlwr().

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
            char str1[30];
int i;
            printf("\nEnter a string :");
gets(str1);
for(i=0;str1[i]!='\0';i++)
{
            if(str1[i]>=65 && str1[i]<=90)
            {
                        str1[i]=str1[i]+32 ;
            }
}
printf("%s" , str1);

}
```

strupr()

This is used to convert string to uppercase. The syntax is given below.

strupr(string);

Write a program to converts string into uppercase . Use of strupr()

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
        char string[]={fakir mohan university};
        printf("String in upper case=%s",strupr(string));
}
```

strstr() :

strstr() :
This function finds second string in the first string .It returns the pointer location from where the second string starts in the first string . In case the first occurrence in the string is not observed , the function returns a NULL character .

Syntax:

```
        strstr(stging1, string2);
```

With Library Function

Write a program using strstr() function

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
          char name1[30] ,name2[30], *chp ;
          clrscr();
          puts("Enter name1 :");
          gets(name1);
                    puts("Enter name2 :");
          gets(name2);
          chp=strstr(name1,name2);
          if(chp)
          {
                    printf("%s 'String is present in Given string." , name2);
          }
else
{
printf("%s 'String is not
present in Given string." ,
name2);

}
}
```

Pattern Mathing

1. Pattern Matching algorithms are used to find patterns within a bigger set of data or text.
2. These algorithms work by comparing a pattern with a larger data set or text and determining whether or not the pattern is present.

Algorithms:

1. Brute Force Pattern Matching Algorithm
2. Naive Pattern Matching Algorithm
3. Knuth-Morris-Pratt Algorithm
4. The Boyer-Moore Algorithm, etc,..

Brute Force Pattern Matching Algorithm

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
    int i,j,k,n,m,flag=0;
    char t[40],p[30];
    clrscr();
    printf("Enter text: ");
    gets(t);
    printf("\nEnter pattern: ");
    gets(p);
    n=strlen(t);
    m=strlen(p);
   for(i=0;i<=n-m;i++)
    {
          j=0;
    while(j<m && p[j]==t[j+i])
{
    j++;
    if(j==m)
     {
          flag=1;
           k=i+1;
     }
     else
     flag=0;
  }// end while
} // end for
if(flag==1)
printf("nPattern found at
position: %dn ",k);
else
printf("nPattern not found in
text n");
getch();
}
```

Pattern Mathing

1. Pattern Matching algorithms are used to find patterns within a bigger set of data or text.
2. These algorithms work by comparing a pattern with a larger data set or text and determining whether or not the pattern is present.

Algorithms:

1. Brute Force Pattern Matching Algorithm
2. Naive Pattern Matching Algorithm
3. Knuth-Morris-Pratt Algorithm
4. The Boyer-Moore Algorithm, etc,..

Pattern Mathing

1. Pattern Matching algorithms are used to find patterns within a bigger set of data or text.
2. These algorithms work by comparing a pattern with a larger data set or text and determining whether or not the pattern is present.

Algorithms:

1. Brute Force Pattern Matching Algorithm
2. Naive Pattern Matching Algorithm
3. Knuth-Morris-Pratt Algorithm
4. The Boyer-Moore Algorithm, etc,..