Vivekananda Institute of Professional Studies

योगः कर्मसु कौशलम्
IN PURSUIT OF PERFECTION

String Processing

Programming using C language
Dr. Nivedita Palia

String

In C , a string is a null-terminated character array.

String declaration: char str[size];

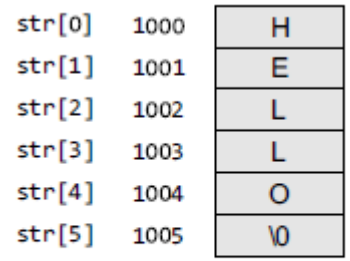

Figure 4.2 Memory representation of a character array

String Initialization
char str[]="Hello";
char str[5]="hello";
char str[]={'h', 'e' , 'l' , 'l' , 'o' , '\0'};
To store a string of length n , we need n+1 location(1 extra for null)

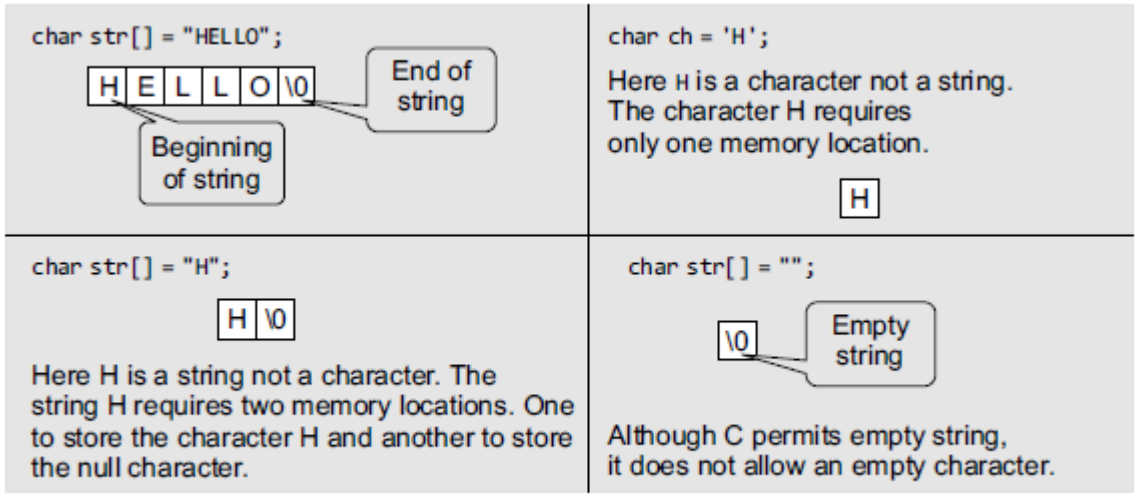

Figure 4.1 Difference between character storage and string storage

Reading String

Using scanf()

```
#include<stdio.h>
void main()
{char s1[10];
printf("enter a string \n");
scanf("%s",s1); // New
printf("\ns1=%s",s1); // New
}
```

using gets()

```
#include<stdio.h>
void main()
{char s1[10];
printf("enter a string");
gets(s1);
puts(s1);
}
```

Using getchar()

```
#include<stdio.h>
void main()
{char s1[10;
int i;
printf("create a character array using getchar");
for(i=0;i<10;i++)//string entered using getchar
s1[i]=getchar();
printf("%s",s1);
}
```

```
create a character array using getcharn e w y o r k
n e w y o

--------------------------------
Process exited after 10.94 seconds with return value 11
Press any key to continue . . .
```

Inserting a string in main string

```
INSERT("XYZXYZ", 3, "AAA") = "XYZAAAXYZ"
```

Text: main string
Pos: index where string to be inserted
Str: string to be inserted.

```
Step 1: [INITIALIZE] SET I=0, J=0 and K=0
Step 2: Repeat Steps 3 to 4 while TEXT[I] != NULL
Step 3: IF I = pos
          Repeat while Str[K] != NULL
              new_str[J] = Str[K]
              SET J=J+1
              SET K = K+1
          [END OF INNER LOOP]
        ELSE
          new_str[J] = TEXT[I]
          set J = J+1
        [END OF IF]
Step 4: set I = I+1
        [END OF OUTER LOOP]
Step 5: SET new_str[J] = NULL
Step 6: EXIT
```

Figure 4.9 Algorithm to insert a string in a given text at the specified position

Deletion from string

DELETE("ABCDXXXABCD", 4, 3) = "ABCDABCD"

M: the position from which deletion has to be done
N: the number of characters to be deleted

```
Step 1: [INITIALIZE] SET I=0 and J=0
Step 2: Repeat Steps 3 to 6 while TEXT[I] != NULL
Step 3:  IF I=M
               Repeat while N>0
                     SET I = I+1
                     SET N = N - 1
               [END OF INNER LOOP]
         [END OF IF]
Step 4: SET NEW_STR[J] = TEXT[I]
Step 5: SET J = J + 1
Step 6: SET I = I + 1
        [END OF OUTER LOOP]
Step 7: SET NEW_STR[J] = NULL
Step 8: EXIT
```

Figure 4.11 Algorithm to delete a substring from a text

Replacing a pattern with another pattern in string

```
Step 1: [INITIALIZE] SET POS = INDEX(TEXT, P1)
Step 2: SET TEXT = DELETE(TEXT, POS, LENGTH(P1))
Step 3: INSERT(TEXT, POS, P2)
Step 4: EXIT
```

Figure 4.12 Algorithm to replace a pattern P_1 with another pattern P_2 in the text

REPLACE(text, $pattern_1$, $pattern_2$).

("AAABBBCCC", "BBB", "X") = AAAXCCC
("AAABBBCCC", "X", "YYY")= AAABBBCC

Pattern matching/ substring

```
Step 1: [INITIALIZE] SET I=0 and MAX = Length(TEXT)-Length(STR)+1
Step 2: Repeat Steps 3 to 6 while I < MAX
Step 3:   Repeat Step 4 for K = 0 To Length(STR)
Step 4:          IF STR[K] != TEXT[I + K], then Goto step 6
          [END OF INNER LOOP]
Step 5:   SET INDEX = I. Goto Step 8
Step 6:   SET I = I+1
        [END OF OUTER LOOP]
Step 7: SET INDEX = -1
Step 8: EXIT
```

Figure 4.10 Algorithm to find the index of the first occurrence of a string within a given text

Text: Welcome
Str: come
Max:7-4+1=4

Inbuilt string function (string.h)

function	Action	Syntax
strcat()	Concatenates two strings	strcat(str1,str2)
strcmp()	Compares two strings Return 0 if equal else numeric difference of between the first non matching	strcmp(str1,str2)
strcpy()	Copies one string to another	strcpy(str1,str2)
strlen()	Find length of string	N=strlen(str1)
strncpy()	Copies leftmost n characters of one string to another	strncpy(str1,str2,n)
strncmp()	Compares leftmost n characters of two strings	strncmp(str1,str2,n)
strncat()	Concatenates leftmost n characters of string str2	strncat(str1,str2,n)
strstr()	Used to locate substring in astring	strstr(str1,str2)

strcat()

```
#include<stdio.h>
#include<string.h>
void main()
{char s1[20],s2[10];
gets(s1);
gets(s2);
puts(strcat(s1,s2));
}
```

```
new
york
newyork
```

strcmp()

```
#include<stdio.h>
#include<string.h>
void main()
{char s1[20],s2[10],s3[10];
int a,b;
gets(s1);
gets(s2);
gets(s3);
a=strcmp(s1,s2);
printf("a=%d\n",a);
b=strcmp(s1,s3);
printf("b=%d",b);
}
```

```
new
new
york
a=0
b=-1
```

String Concatenate

```
#include <stdio.h>
int main() {
        char str1[20],str2[20],str3[40];
        int i,j;
        printf("enter the string1");
        gets(str1);
        printf("enter the string2");
        gets(str2);
        printf("string 1= %s string 2%s\n",str1,str2);
        for(i=0;str1[i]!='\0';i++)
        str3[i]=str1[i];
        str3[i]=' ';
        i=i+1;
        for(j=0;str2[j]!='\0';j++)
        str3[i+j]=str2[j];
        printf("string3%s\n",str3);
        return 0;
}
```

```
enter the string1abc
enter the string2efg
string 1= abc string 2efg
string3abc efg
```

String Compare

```
#include <stdio.h>
int main() {
        char str1[20],str2[20];
        int i=0;
        printf("enter the string1");
        gets(str1);
        printf("enter the string2");
        gets(str2);
        printf("string 1= %s string 2%s\n",str1,str2);
        while((str1[i]==str2[i])&&(str1[i]!='\0')&&(str2[i]!='\0'))
        i=i+1;
        if(str1[i]=='\0'&&str2[i]=='\0')
        printf("strings are equal");
        else
        printf("strings are not equal");
        return 0;
}
```

```
enter the string1abc
enter the string2efg
string 1= abc string 2efg
strings are not equal
```

String Copy and string length

```
#include <stdio.h>
int main() {
        char str1[20],str2[20];
        int i;
        printf("enter the string1");
        gets(str1);
        printf("string 1 %s\n",str1);
        for(i=0;str1[i]!='\0';i++)
        str2[i]=str1[i];
        str2[i]='\0';
        printf("string2%s\nthe number of characters in a string=%d",str2,i);
}
```

```
enter the string1abcd
string 1 abcd
string2abcd
the number of characters in a string=4
--------------------------------
```

Pointers and Strings

```
#include <stdio.h>
int main() {
        char *name, *p;
        name= "New";
        p=name;
        puts(name);
        while(*p!='\0')
        {
        printf("%c is stored at address %u\n",*p, p);
        p++;
        }
        return 0;
}
```

```
New
N is stored at address 4210688
e is stored at address 4210689
w is stored at address 4210690
```

Array of pointers

Char name[3][25]; declare 75 bytes

```
#include <stdio.h>
int main() {
        char *name[3]={"New","Newyork","New Zealand"};
        puts(name[0]);
        puts(name[1]);
        puts(name[2]);
        return 0;
}
```

```
New
Newyork
New Zealand

--------------------------------
```

char *name[3] uses 24 bytes

END