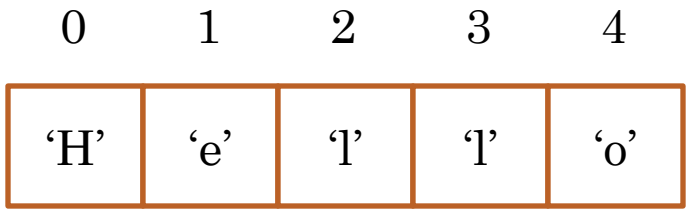
Символьные строки

Лекция №4

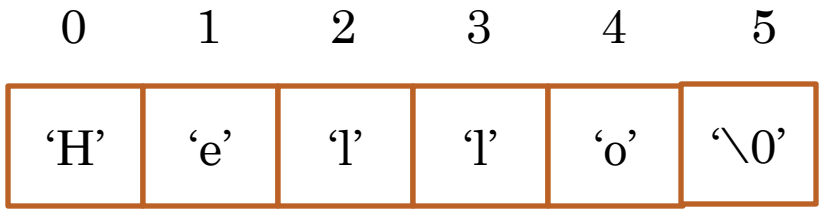
Массив символов

- Пронумерованная последовательность элементов одного и того же типа
  - Элементы одного и того же типа
  - Элементы расположены последовательно
  - Конечное количество элементов
  - Каждый элемент имеет свой номер
  - Элементы нумеруются с 0 до n-1, если всего элементов – n
  - Символьный массив имеет тип char



Исходные предпосылки

- В Си нет стандартного строкового типа
- Строка организуется как символьный массив с признаком конца
- В Си есть стандартная библиотека функций для работы со строками string.lib с заголовочным файлом string.h
- В Си управляющий символ ‘\0’ является признаком конца строки (нуль-символ)



Смысл нуль-символа

- символ – признак окончания строки
- увеличивает объем памяти на 1 байт
- позволяет обрабатывать строку неизвестной длины

- Потеря нуль символа может привести
  - к зацикливанию
  - к аварийному завершению

Формат описания символьного массива

- char имя[кол_во_эл]={знач1,знач2,знач3};
  - Количество элементов – целая константа
  - Номера элементов – целые
  - Границы контролирует программист!!!
- //Примеры описания символьных массивов
- char A[5]={‘2’,’1’,’A’,’!’,’v’};//это строка?
- char B[4]={‘9’, ‘F’, ‘?’,’\0’};//а это строка?
- char C[4];

- Символьные константы выделяются ‘

Способы кодирования символов

- ASCII
  - Каждый символ 1 байт = 8 бит
    - Английский алфавит, цифры и знаки – 128 кодовых комбинаций
    - Национальный алфавит (рус) и псевдографика – 128 кодовых комбинаций
- Unicode
  - Каждый символ – 2 байта = 16 бит
  - Диапазон разбит на поддиапазоны. Каждый поддиапазон – свой язык
    - Лат – совпадает с ASCII
    - Рус – 0400-04FFh

Способы задания строки

- С помощью символьных констант
  - char c1[ ]={‘A’,’B’,’C’,’\0’};
- С помощью строковой константы
  - char c2[ ]=“ABCDE”;//нуль в конце доб. автоматич.
  - В последних версиях VS от этого вар. отказались
- Заполнение в процессе работы
  - char c3[6];
  - for (i=0;i<5,i++)
    - c3[i]=‘+’;
  - c[i]=‘\0’;
- Заполнение с клавиатуры
  - printf(“Input a string”);
  - scanf(“%s”,c3);

Свойства строк

- Каждый символ занимает 1 байт (ASCII)
- Символы кодируются с учетом закономерностей (буквы, цифры)
- В конце строки – нуль символ
  - Возможность заранее не знать длину
  - Потеря нуль символа приводит к зацикливанию или аварийному выходу из программы

Закономерности кодирования символов

- Каждому графическому образу символа поставлено в соответствие целое число
  - ‘0’ – 48, ‘1’ – 49, ‘9’ – ?
  - ‘A’ – 65, ‘B’ – 66, ‘Z’ – ?
  - ‘a’ – 97, ‘b’ – 98, ‘z’ – ?
- Коды букв расположены в алфавитном порядке
- Коды цифр расположены по возрастанию
- Возможность преобразования числового формата в символьный и наоборот
- Возможность преобразования больших букв в маленькие и наоборот

Преобразования символов

- char c1,c2;
- int k1,k2;

- c1=‘5’;
- k1=c1-’0’;//?

- c1=‘h’;
- c2=c1-’a’+’A’;

Подсчет количества цифр

```
int a=123,k=0,b,i;
char c[10];
for (b=a; b!=0; b=b/10)
   k++;

//и перевод числа в символьную форму
b=a;
for (i=k-1; i>=0; i--)
   {c[i]=b%10+'0';
     b=b/10;}
c[k]=0;
```

Операции над строками

- Вычисление длины
- Копирование
- Сравнение
- Преобразования символов
- Поиск
- Добавление символов
- Удаление символов

Вычисление длины строки

- char s1[]=“AAbb1”;
- int n;
- for (n=0;s1[n]!=‘\0’;n++);

Подсчет количества пробелов

```
char s[]="Spring is coming";
int i, cnt=0;
for(i=0;s[i]!='\0';i++)
   if(s[i]==' ')
         cnt++;
```

Вставка пробела после точки

```
char s[ ] =“Abc.def”;
int i,k,n;
for (n=0;s[n]!=0;n++);

for (i=0;s[i]!=0;i++)
   if(s[i]==‘.’)
         {for(k=n-1;k>i+1;k--)
               s[k]=s[k-1];
         s[i+1]=‘ ‘;}
```

Библиотека для работы со строками

- string.lib
- Имеет заголовочный файл string.h

- Функции
  - strlen – длина строки
  - strcpy – копирование строк
  - strcmp – сравнение строк

Примеры вызовов библиотечных функций

```
#include <stdio.h>
#include <string.h>
void main(void)
{char s1[100],s2[100],s3[100];
int n,k,m;
scanf(“%s”,s1);
scanf(“%s”,s2);
n=strlen(s1);
if (strcmp(s1,s2)==0)
   printf(“strings are the same”);
strcpy (s3,s2);
```

Пример: поиск подстроки с синхрословом

```
char s1[]="abc123fdsg123khg"
char s2[]="123";
int i, j, k,f=-1,m1,m2;
m1=strlen(s1); m2=strlen(s2);
for(i=0;s1[i+m2]!=0;i++)
   {for(j=0;s2[j]!=0;j++)
         if(s1[i]==s2[j])
               for(k=0;k<m2;k++)
                     if(s1[i+k]!=s2[j+k])
                           break;
   if(k==m2)
         {f=i;
         break;
         }
   }
```

Выводы

- Строка обязательно должна иметь признак конца
- Для работы со строкой необязательно заранее знать количество символов
- Коды символов подчиняются закономерностям
- Операции над строками похожи на операции с массивами
- Имеется библиотека стандартных функций для работы со строками