Курс «Основы программирования»

Федорук Елена Владимировна ст. преподаватель каф РК-6 МГТУ им.Н.Э.Баумана

Лекция №16

Функции обработки строк

Компиляторы языка Си обычно дополняются набором библиотечных функций для обработки строк. В системах ОС UNIX эти функции обычно содержатся в стандартной библиотеке Си **/lib/libc.a**

файл заголовков **<string.h>** — файл, в котором объявляются функции обработки строк, а так же типов, поименованных констант и макросов.

Программы, использующие эти функции, должны включать директиву препроцессора include:

#include <sting.h>

Один или более параметров этих функций имеют тип указателя на символ (**char ***). Это означает, что в качестве значения параметра ожидается адрес символа, например, начальный адрес символьного массива.

Некоторые функции обработки строк

char *strcat (char *s1, char *s2)

Функция присоединяет строку **s2** к строке **s1** и возвращает значение **s1**.

char *strncat (char *s1, char *s2, int n)

Функция присоединяет не более n символов строки **s2** к строке **s1** и возвращает значение **s1**.

int strcmp (const char *s1, const char *s2)

Функция сравнения двух строк, возвращает значение меньшее, равное или большее 0, если **s1** лексикографически меньше, равна или больше, чем**s2** (сравниваются коды символов).

int strncmp (const char *s1, const char *s2, int n)

Функция сравнения не более n символов двух строк, возвращает значение меньшее, равное или большее 0, если **s1** лексикографически меньше, равна или больше, чем **s2**.

char *strcpy (char *s1, const char *s2)

Функция копирования **s2** в **s1**, останавливается после копирования нулевого символа, возвращает **s1**.

char *strncpy (char *s1, const char *s2, int n)

Функция копирования не более n символов **s2** в **s1**, останавливается после копирования нулевого символа, возвращает **s1**.

`int strlen (const char *s)`

Функция определения длины строки, возвращает количество символов вплоть до нулевого символа, но не включая его.

`char *strchr (char *s, char c)`

Функция возвращает указатель на первое вхождение символа **`c`** в строке **`s`** или, если такового не оказалось, **`NULL`**.

`char *strrchr (char *s, char c)`

Функция возвращает указатель на последнее вхождение символа **`c`** в строке **`s`** или, если такового не оказалось, **`NULL`**.

Пример 1

```
#include <stdio.h>
#include <string.h>
#define LINELEN 256
#define MINLEN 6
#define MAXLEN 12
char input[LINELEN+1], passwd [MAXLEN+1];
int change_passwd(void)   /*функция вводит новый пароль, содержащий не менее 6 символов*/
{
printf("Enter new password: ");
gets(input);
if (strlen(input)<MINLEN)
  {
  printf("Password too short.\n");
  exit(1);
  }
}
int verify_passwd(void) /*функция проверяет введеный пароль*/
{
printf("Enter password: ");
gets(input);
getpassword (password); /*копирует текущий пароль по заданному указателем адресу*/
if (strlcmp(input, password) != 0)
  {
  printf("Sorry.\n";
  exit(2);
  }
}
```

Математические функции

Объявления математических функций находятся в заголовочном файле **<math.h>**, который необходимо подключать с помощью директивы препроцессора **#include <math.h>**

При использовании математических функций могут возникнуть ошибки области и ошибки диапазона.

Ошибка области возникает, если аргумент выходит за область значений, для которой определена функция.

Ошибка диапазона возникает, когда результат не может быть представлен в виде **double**.

Таблица 1

Объявления функции	Назначение функции
double sin(double x)	синус x
double cos(double x)	косинус x
double tan(double x)	тангенс x
double sin(double x)	арксинус x в диапазоне $[-\pi/2,\pi/2]$, $x\in[-1,1]$
double acos(double x)	арккосинус x в диапазоне $[0,\pi]$, $x\in[-1,1]$
double atan(double x)	арктангенс x в диапазоне $[-\pi/2,\pi/2]$
double atan2(double y, double x)	арктангенс y/x в диапазоне $[-\pi/2,\pi/2]$
double sinh(double x)	гиперболический синус x
double cosh(double x)	гиперболический косинус x
double tanh(double x)	гиперболический тангенс x
double exp(double x)	экспоненциальная функция e^x
double log(double x)	натуральный логарифм ln(x), x>0
double log10(double x)	десятичный логарифм $\log_{10}(x)$, x>0
double pow(double y, double x)	x^y. Ошибка области, если x=0 и $y\leq 0$ или x<0 и y-не целое
double sqrt(double x)	$\sqrt{x}$, $x\geq 0$
double ceil(double x)	наименьшее целое в виде **double**, которое не меньше x

double floor(double x)	наибольшее целое в виде **double**, которое не больше x
double fabs(double x)	абсолютное значение|x|
double ldexp(double x, int n)	$x.2^n$
double frexp(double x, int *exp)	разбивает x на два сомножителя, первый из которых — нормализованная дробь в интервале [1/2,1), которая возвращается, а второй — степень двойки, эта степень запоминается в ***exp**. Если x - нуль, то обе части результата равны нулю
double modf(double x, double *ip)	разбивает на целую и дробную части, обе имеют тот же знак, что и x. Целая часть запоминается в ***ip**. Дробная часть возвращается как результат
double fmod (double x, double y)	остаток от деления x на y в виде числа с плавающей точкой. Знак результата совпадает со знаком x. Если y равен нулю, результат зависит от реализации

Примечание 1

Углы в тригонометрических функциях задаются в радианах.

Примечание 2

При компиляции программы, использующей математические функции, необходимо использовать флаг **-lm** в команде **gcc** (**cc**).

Функции случайных чисел

Функции случайных чисел являются функциями библиотеки общего назначения.

Объявления этих функций и необходимых макроопределений находится в заголовочном файле **<stdlib.h>**, поэтому в программах, использующих эти функции, необходимо включать следующую директиву препроцессора: **#include <stdlib.h>**.

int rand(void)

Функция **rand()** возвращает псевдослучайное число в диапазоне от 0 до **RAND_MAX** (**RAND_MAX** не менее 32767)

int srand(unsigned int seed)

Функция **srand()** использует **seed** в качестве семени для новой последовательности псевдослучайных чисел. Изначально параметр **seed** равен 1.

Функцию **srand()** достаточно использовать один раз в начале программы для того, чтобы при каждом исполнении программы была получена новая последовательность случайных чисел.

Часто в качестве семени используют значение текущего времени.

Пример 1

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
void fun(void)
{int i, a;
srand(time(NULL));
for (i=0; i<10; i++)
  {a=rand()%15-10; /*получаем случайное число в интервале от -10 до 4*/
   printf ("%d|n", a);
  }
}
```

Функции преобразования ASCII-строки в числовые значения

Функции преобразования ASCII-строки в числовые значения являются функциями библиотеки общего назначения.

Эти функции принимают в качестве аргумента строку в коде ASCII и возвращают соответствующее число: целое, длинное целое или двойной точности с плавающей точкой. Если первый символ в строке не цифра, то возвращается 0.

Объявления этих функций находится в заголовочном файле **<stdlib.h>**. поэтому в программах, использующих эти функции, необходимо включать следующую директиву препроцессора:

#include <stdlib.h>

int atoi(const char *str) - функция переводит строку **str** в целое типа **int**

long atol(const char *str)- функция переводит строку **str** в целое типа **long**

double atof(const char *str)- функция переводит строку **str** в вещественное типа **double**

Пример 1

```
#include <stdio.h>
#include <string.h>
int main(0
{
int quantity;
double percentage;
char line[81];
gets(line);
quantity=atoi(line);
gets(line);
percentage=atof(line);
}
```

Функция exit()

Функция exit() являются функцией библиотеки общего назначения.

Объявление этой функций находится в заголовочном файле **<stdlib.h>**, поэтому в программах, использующих эту функцию, необходимо включать следующую директиву препроцессора:

#include <stdlib.h>.

void exit(int status)

Функция **exit()** вызывает нормальное завершение программы. При этом производится очищение буферов открытых файлов, открытые потоки закрываются, и управление возвращается в среду, из которой был произведен запуск программы. Значение **status**, передаваемое в среду, зависит от реализации, однако, при успешном завершении программы принято передавать нуль.