Найти

# Программирование на C и C++

Онлайн справочник программиста на C и C++



## Случайные статьи

installuserdriver
setgraphmode
_makepath
Конструктор копирования
Оператор ?
getdisk
fgetc, fgetchar
heapwalk, farheapwalk, _rtl_heapwalk
Указатели на объекты
fabs, fabsl
fopen
_dos_getftime
Комментарии
getpid
strupr, _fstrupr
sprintf
Ввод/вывод в массивы
Битовые операторы
strtol, strtoul
Типы данных
sound
perror
Использование множества сканирования
strerror
Лесенка if-else-if
_rotl, _rotr
fnmerge, fnsplit
_dos_findfirst, dos_findnext
chmod
sopen

# printf



## int printf(const char *format, arg-list)

**Прототип:**
stdio.h
**Описание:**

Функция printf() записывает в stdout аргументы из списка arg-list под управлением строки, на которую указывает аргумент format.

Строка, на которую указывает format, состоит из объектов двух различных назначений. Во- первых, это символы, которые сами должны быть выведены на экран. Во-вторых, это спецификаторы формата, определяющие вид, в котором будут выведены аргументы из списка arg-list. Спецификаторы формата состоят из символа процент, за которым следует код формата. Команды форматирования приведены в таблице. Количество аргументов должно точно соответствовать количеству спецификаторов формата, причем следовать они должны в одинаковом порядке. Например, следующий вызов функции printf()

```
printf ("Hi %c %d %s", 'c', 10, "there!");
```

приведет к выводу «Hi c 10 there!».

Таблица: Команды форматирования для printf()

| Код | Формат |
|---|---|
| %c | Символ типа char |
| %d | Десятичное число целого типа со знаком |
| %i | Десятичное число целого типа со знаком |
| %e | Научная нотация (e нижнего регистра) |
| %E | Научная нотация (E верхнего регистра) |
| %f | Десятичное число с плавающей точкой |
| %g | Использует код %e или %f – тот из них, который короче (при использовании %g используется e нижнего регистра) |
| %G | Использует код %E или %f – тот из них, который короче (при использовании %G используется E верхнего регистра) |
| %o | Восьмеричное целое число без знака |
| %s | Строка символов |
| %u | Десятичное число целого типа без знака |
| %x | Шестнадцатиричное целое число без знака (буквы нижнего регистра) |
| %X | Шестнадцатиричное целое число без знака (буквы верхнего регистра) |
| %p | Выводит на экран значение указателя |

| Код | Формат |
|---|---|
| %n | Ассоциированный аргумент — это указатель на переменную целого типа, в которую помещено количество символов, записанных на данный момент |
| %% | Выводит символ % |

Если количество аргументов меньше, чем количество команд форматирования, то вывод неопределен. Если же количество аргументов больше, чем команд форматирования, то лишние аргументы отбрасываются.

Функция printf() возвращает количество действительно выведенных символов. Возврат отрицательной величины означает ошибку.

Команды форматирования могут содержать модификаторы, означающие ширину поля, точность и флаг выравнивания влево. Переменная целого типа, помещенная между символом процент и командой форматирования, работает как спецификатор минимальной ширины поля, заполняя поле вывода пробелами или нулями так, чтобы обеспечить указанную минимальную ширину. Если строка или число больше, чем этот минимум, они будут полностью выведены. По умолчанию заполнение производится пробелами. При выводе числовых переменных, если надо использовать заполнение нулями, помещается нуль перед спецификатором минимальной ширины поля. Например, %05d будет дополнять числа, состоящие из менее чем 5 цифр, нулями до пяти цифр.

Результат использования модификатора точности зависит от типа модифицируемой команды форматирования. Чтобы использовать модификатор точности, надо поместить десятичную точку и точность вслед за ней после количества выводимых десятичных знаков. Например, %10.4f означает вывод числа шириной минимум 10 символов с четырьмя знаками после точки. Однако при использовании совместно со спецификаторами g или G модификатор точности задает максимальное количество отображаемых значащих цифр.

Когда модификатор точности применяется к целым числам, он указывает минимальное количество отображаемых цифр. (При необходимости отображаются предшествующие нули.)

Когда модификатор точности применяется к строкам, число после десятичной точки указывает максимальную длину поля. Например, %5.7s выводит строку длиной не менее пяти и не более семи символов. Если строка длиннее, чем максимальная ширина поля, то последние символы будут урезаны.

По умолчанию вывод производится с выравниванием вправо. Это значит, что если ширина поля больше, чем выводимые данные, то данные будут размещены на правом краю поля. Можно задать режим выравнивания влево, вставив знак минус сразу после знака процент. Например, %-10.2f прижмет влево в десятизнаковом поле число с плавающей точкой с двумя знаками после запятой.

Имеется два спецификатора формата, позволяющих printf() отображать целые числа типа short и long. Эти спецификаторы могут применяться совместно со спецификаторами типа d, i, o, u, x и X. Спецификатор l «говорит» printf() о том, что далее следуют данные типа long. Например, %ld означает, что нужно вывести long int. Спецификатор h указывает printf(), что нужно отобразить short int. Следовательно, %hu указывает, что данные имеют тип short unsigned int.

Хоть это и не оговорено в стандарте (да это и не нужно), модификатор l может также предшествовать спецификаторам типа с плавающей точкой e, E, f, g и G, указывая, что далее следует переменная double. L используется для указания long double.

Спецификатор n помещает количество выведенных до сих пор символов в переменную целого типа, на которую указывает аргумент, соответствующий спецификатору. Например, следующий фрагмент кода выводит число 15 после строки «this is a test».

```
int i;
printf ("this is a test %n", &i);
printf("%d", i);
```

Символ # имеет особое значение, когда используется с некоторыми спецификаторами формата функции printf(). Если поставить # перед спецификаторами a, g, G, f, e или E, то десятичная точка будет ставиться даже в отсутствие цифр после запятой. Если поставить # перед спецификатором формата x, то шестнадцатиричное число будет выведено с префиксом 0x. Использование # со спецификатором o приводит к выводу префикса 0. С другими спецификаторами формата символ # использоваться не может.

Спецификаторы минимальной ширины поля и точности могут быть представлены для printf() не только в виде констант, но и с помощью аргументов. Это делается с помощью символа *, который служит указателем места. При считывании форматной строки функция printf() проводит соответствие между символами * и аргументами в том порядке, в каком они встречаются.

**Пример:**

Данная программа отображает на экране то, что указано в комментариях:

```
#include <stdio.h>
int main(void)
{
/* выводит выровненное по левому краю "this is a test" в 20-символьном поле
*/
printf("%-20s", "this is a test");
/* выводит вещественное значение с 3 цифрами после запятой в 10-символьном поле.
В результате работы увидим "      12.235"
*/
printf("%10.3f", 12.234657);
return 0;
}
```



## Функции ввода/вывода

| | | |
|---|---|---|
| vprintf, vfprintf, vsprintf | access | chmod |
| chsize | clearerr | close, _rtl_close |
| creat, _rtl_creat, creatnew, creattemp | dup, dup2 | eof |
| fclose, fcloseall | fdopen | feof |
| ferror | fflush | fgetc, fgetchar |
| fgetpos | fgets | filelength |
| fileno | flushall | fopen |
| fprintf | fputc | fputchar |
| fputs | fread | freopen |

1 2 3 следующая › последняя »