



---

[\[ Главная \]](#) [\[ Гостевая \]](#)  
[Содержание](#) | [Глава 13](#)

## Функция vprintf, vfprintf, vsprintf, vsnprintf

```
#include <stdarg.h>;
#include <stdio.h>
int vprintf(char *format, va_list arg_ptr);
int vfprintf(FILE *stream, const char *format, va_list arg_ptr);
int vsprintf(char *buf, const char *format, va_list arg_ptr);
int vsnprintf(char * restrict buf, size_t num, const char * restrict format, va_list arg_ptr);
```

Действия функций vprintf(), vfprintf(), vsprintf() и vsnprintf() эквивалентны действиям функций printf(), fprintf(), sprintf() и snprintf() соответственно, но список аргументов заменяется указателем на список аргументов. Этот указатель должен иметь тип va\_list, который определен в заголовке <stdarg.h>.

В версии C99 к параметрам *buf* и *format* применен квалификатор restrict. Функция vsnprintf() добавлена в версии C99.

### Пример

Данный фрагмент программы иллюстрирует, как нужно вызывать функцию vprintf(). Вызов функции va\_start() приводит к созданию указателя на список аргументов переменной длины, причем этот указатель указывает на начало списка аргументов. Этот указатель должен быть использован при вызове функции vprintf(). Вызов функции va\_end() очищает указатель на список аргументов переменной длины.

```
#include <stdio.h>
#include <stdarg.h>

void print_message(char *format, ...);

int main(void)
{
    print_message("Не удастся открыть файл %s.", "test");

    return 0;
}

void print_message(char *format, ...)
{
    va_list ptr; /* извлечение аргумента ptr */

    /* инициализация ptr, он становится указателем на первый аргумент,
       следующий за строкой форматирования */
    va_start(ptr, format);

    /* вывод сообщения */
    vprintf(format, ptr);

    va_end(ptr);
}
```

### Зависимые функции

[vscanf\(\)](#) [vfscanf\(\)](#) [vsscanf\(\)](#) [va\\_arg\(\)](#) [va\\_start\(\)](#) [va\\_end\(\)](#)

---

[Содержание](#) | [Глава 13](#)  
[\[ Главная \]](#) [\[ Гостевая \]](#)