

ГУАП

КАФЕДРА № 14

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

8

ассистент

должность, уч. степень, звание

Іль

подпись, дата

П. В. Шпигун  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 1

ОСНОВЫ РАБОТЫ С КОНСОЛЬНЫМ ПРИЛОЖЕНИЕМ И  
СИНТАКСИЧЕСКИЙ РАЗБОР ТЕКСТА

по курсу: АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

1446

А. С.

25.10.2025

подпись, дата

А. С. Пырву  
инициалы, фамилия

Санкт-Петербург 2025

**Цель работы:** изучение работы с консолью на Linux.

**Постановка задачи:** Вывести на экран содержимое файла с программой на C, подсвечивая другим цветом все статические (static )переменные.

### Формализация:

Константы:

```
#define CL_GREEN "\033[92m"
#define CL_RESET "\033[0m"
#define LINE_SIZE 256
```

Реализованные функции:

prev\_symbol\_allowed - Проверка допустимости символа перед ключевым словом.  
process\_line\_symbolwise - Обработка одной строки.  
main - Управляющая функция программы.

Использованные библиотеки:

```
#include <stdio.h>
#include <string.h>
#include <stdbool.h>
```

### Листинг программы:

```
#include <stdio.h>
#include <string.h>
#include <stdbool.h>

// коды цвета и сброса цвета
#define CL_GREEN "\033[92m"
#define CL_RESET "\033[0m"

// размер буфера строки
#define LINE_SIZE 256

// нужно, чтобы проверить, что символ перед словом static
// разрешенный (не было int loop_static, где оно в имени переменной)
bool prev_symbol_allowed(char c)
{
    return c == '\0' || c == '\t' || c == ' ';
}

// обработка одной строки посимвольно
void process_line_symbolwise(char* line)
{
    // флаги, означающие, находимся ли мы
    // сейчас в комментарии или строковом литерале
    bool in_com1 = false;
    bool in_str = false;
    // это делаем static, чтобы переменная сохраняла значение в разные вызовы
    // так как комментарий /* может быть открыт или закрыт на предыдущей строке
    static bool in_com2 = false;

    int i = 0;
    char prev = '\0';
    while (line[i] != '\0')
    {
        if (prev == '/' && line[i] == '/')
            in_com1 = true;

        if (prev == '*' && line[i] == '/')
            in_com2 = true;
    }
}
```

```

    in_com2 = true;

    if (prev == '*' && line[i] == '/')
        in_com2 = false;

    if (line[i] == '\"')
        in_str = !in_str;

    if (in_com1 || in_com2 || in_str)
    {
        printf("%c", line[i]);
        prev = line[i];
        i++;
        continue;
    }

    // strncmp сравнивает первые N символов у строк
    // "стоит ли на позиции i слово static с пробелом?"
    if (prev_symbol_allowed(prev) && strncmp(line + i, "static ", 7) == 0)
    {
        // начало цветной области
        printf(CL_GREEN);
        // идем до 2го пробела: static int abc; подсветим static int
        int k_spaces = 0;
        while (line[i] != '\0' && k_spaces < 2)
        {
            if (line[i] == ' ')
            {
                k_spaces += 1;
            }
            printf("%c", line[i]);
            i++;
        }
        // отключаем подсветку
        printf(CL_RESET);
    }

    printf("%c", line[i]);
    prev = line[i];
    i++;
}
}

int main()
{
    FILE* fp = fopen("static.c", "r");

    char line[LINE_SIZE];
    // [...] fgets кладет в конец строки символ \n!
    while (fgets(line, LINE_SIZE, fp) != NULL)
    {
        // вызов для каждой строки
        process_line_symbolwise(line);
    }
}

```

Листинг 1. Код программы

### Тестирование:

CON Консоль отладки Microsoft Visual Studio

```
#include <stdio.h>

static int file_scope_counter = 0;

static int test_after = 999; // here's comment
int loop_static = 555;

/*
static int abc = 777;
*/

char string[] = "static int v = 0; static int after_literal = 888;
// static int main_static = 0;
```