

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное учреждение высшего образования  
«Чувашский государственный университет И.Н. Ульянова»  
Факультет информатики и вычислительной техники  
Кафедра вычислительной техники

Системное программирование  
Лабораторная работа 7  
«Позиционирование в файле»

**Выполнил:**

Студент группы ИВТ-41-20  
Галкин Д.С.

**Проверил:**

Яковлев С.В.

## Цель работы:

Создать пустое консольное приложение и добавить к нему служебные функции:

1. Напишите программу **tail** имя, которая будет выводить на стандартное устройство десять последних строк текстового файла. Программа должна демонстрировать позиционирование в файле на основании его размера, и 64-разрядную арифметику с использованием типа Microsoft C **LARGE\_INTEGER**
2. Алгоритм может быть следующим: от конца файла назад отсчитывается длина в предположении, что средняя длина строки меньше 256 байт. Затем ищутся последовательности символов CR, LF и запоминаются положение следующих байтов как позиции начала строк. Циклический буфер хранит 11 позиций

## Полный текст программы:

text\_utils.h

```
//  
// Created by Dmitry Galkin on 20.02.2024.  
//  
  
#ifndef TEXT_UTILS_H  
#define TEXT_UTILS_H  
#include <string>  
  
const int MAX_LINE_LENGTH = 256; // Предполагаемая максимальная длина строки  
const int NUM_LINES = 10; // Количество выводимых строк  
const std::string PATH_FILES_LAB7 = "../Lab7/resources/";  
  
void tail(const std::string& filename);  
void getLines(const std::string& filename, int startLine, int numLines);  
  
#endif //TEXT_UTILS_H
```

```

//
// Created by Dmitry Galkin on 20.02.2024.
//
#include "text_utils.h"

#include <algorithm>
#include <fstream>
#include <iostream>
#include <vector>

void tail(const std::string& filename) {
    std::string full_path = PATH_FILES_LAB7 + filename;
    std::ifstream file(full_path, std::ios::binary | std::ios::ate); // Открываем файл
    в конце
    if (!file) {
        std::cerr << "Не удастся открыть файл: " << filename << std::endl;
        return;
    }

    std::streamsize size = file.tellg(); // Получаем размер файла
    std::vector<std::string> lines; // Вектор для хранения последних строк
    std::string current_line;
    int line_count = 0;

    // Позиционирование с конца файла
    for (std::streamoff i = 1; i <= size && line_count < NUM_LINES + 1; ++i) {
        file.seekg(-i, std::ios::end); // Перемещаем указатель файла
        char c;
        file.read(&c, 1); // Читаем один символ

        // Проверяем символы конца строки (Windows - CR LF, Unix - LF)
        if (c == '\n' || c == '\r' || file.tellg() <= 1) {
            if (!current_line.empty()) {
                lines.push_back(current_line); // Сохраняем строку
                current_line.clear();
                line_count++;
            }
        } else {
            current_line = c + current_line; // Собираем строку в обратном порядке
        }
    }

    // Если файл начинается без новой строки, добавляем последнюю строку в список
    if (!current_line.empty() && lines.size() < NUM_LINES) {
        lines.push_back(current_line);
    }

    // Выводим сохраненные строки
    std::reverse(lines.begin(), lines.end()); // Переворачиваем строки в правильном
    порядке
    for (const auto& line : lines) {
        std::cout << line << std::endl;
    }
}

```

```
void getLines(const std::string& filename, int startLine, int numLines) {
    std::string full_path = PATH_FILES_LAB7 + filename;
    std::ifstream file(full_path);
    if (!file) {
        std::cerr << "Cannot open file: " << filename << std::endl;
        return;
    }

    std::string line;
    int currentLine = 1; // Номер текущей строки

    // Пропускаем строки до начальной    while (currentLine < startLine &&
std::getline(file, line)) {
        ++currentLine;
    }

    // Читаем и выводим нужное количество строк
    while (currentLine < startLine + numLines && std::getline(file, line)) {
        std::cout << line << std::endl;
        ++currentLine;
    }
}
```

main.cpp

```
#include "Lab4/write_utils.h"
#include <iostream>
#include <vector>
#include <sys/termios.h>

using namespace std;

void start_lab7(int argc, char* argv[]);

int main(int argc, char* argv[]) {

    start_lab7(argc, argv);
    return 0;
}

void start_lab7(int argc, char* argv[]) {
    if (argc < 2) {
        std::cerr << "Использование: " << argv[0] << " <filename>" << std::endl;
        return 1;
    }
    else if (argc < 4 && argc > 2) {
        std::cerr << "Usage: " << argv[0] << " <filename> <startLine> <numLines>" <<
std::endl;
        return 1;
    }

    if (argc == 2)
        tail(argv[1]);
    else if (argc == 4) {
        const std::string filename = argv[1];
        int startLine = std::atoi(argv[2]); // Преобразуем строку в число
        int numLines = std::atoi(argv[3]); // Преобразуем строку в число

        // Проверяем, что номера строк и количество строк являются положительными числами
        if (startLine < 1 || numLines < 1) {
            std::cerr << "Start line and number of lines must be greater than 0." <<
std::endl;
            return 1;
        }

        getLines(filename, startLine, numLines);
    }
    return 0;
}
```

## Пример работы:

```
1. George Washington would serve as president of the convention.  
2. Each state, large or small, would have one vote at the convention.  
3. They would not follow the instructions Congress had given them. They would not even try to improve the Articles of confederation. They thought the Articles had too many weaknesses. Instead, they  
4. They would keep their discussions private. They also decided that what they said at the convention would remain a secret for 30 years. There were 2 reasons for this:  
* They believed they needed to speak freely to create the best possible constitution. If people could listen to them and tell others what they said, they would not feel as free to discuss their ideas  
* They wanted the new constitution to be accepted by the people. They were afraid that the people might not accept it if they knew all the disagreements the Framers had. The agreement meant that each
```