Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по Системному программированию

Практическая работа «Разработка программного кода для работы с процессами и потоками в Windows»

Вариант 15

Выполнил: Рубцова Д.А

Группа: ПР-31

Преподаватель: Морозов В.В

2025 год

1. **Описание задачи**

Реализуйте программу, в которой несколько потоков читают

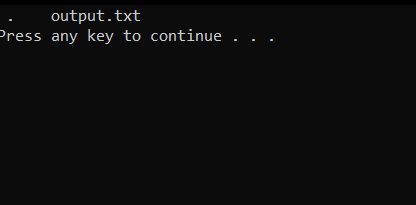
данные из одного большого текстового файла, ищут заданные

ключевые слова и подсчитывают, сколько раз они встречаются.

Результаты необходимо записать в отдельный файл,

обеспечивая потокобезопасность.

1. **Результаты действия программы:**



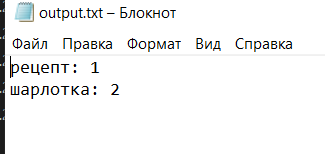


Рис. 1: Работа программы

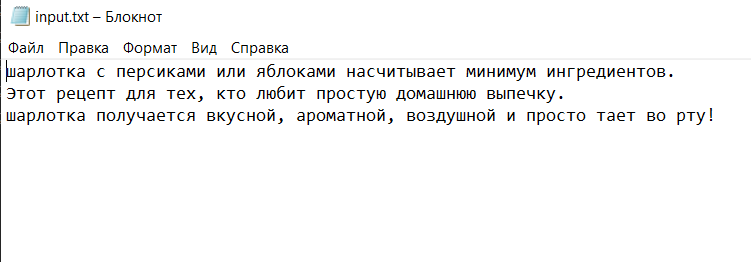


Рис. 3: Исходный файл

1. **Текст программы:**

#include <windows.h>

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <map>

#include <mutex>

#include <locale>

#include <codecvt>

#include <sstream>

#include <algorithm>

const std::wstring inputFile = L"input.txt";

const std::wstring outputFile = L"output.txt";

const std::vector<std::wstring> keywords = { L"шарлотка", L"рецепт" };

const int threadCount = 4;

std::map<std::wstring, int> wordCount;

std::mutex countMutex;

std::vector<std::wstring> lines;

// Приведение строки к нижнему регистру

std::wstring toLower(std::wstring str) {

std::transform(str.begin(), str.end(), str.begin(), towlower);

return str;

}

// Очистка слова от знаков препинания

std::wstring cleanWord(const std::wstring& word) {

std::wstring clean;

for (wchar\_t ch : word) {

if (iswalpha(ch)) {

clean += ch;

}

}

return clean;

}

// Функция поиска ключевых слов в строках

DWORD WINAPI SearchKeywords(LPVOID param) {

int threadNum = \*(int\*)param;

int linesPerThread = lines.size() / threadCount;

int start = threadNum \* linesPerThread;

int end = (threadNum == threadCount - 1) ? lines.size() : start + linesPerThread;

std::map<std::wstring, int> localCount;

for (int i = start; i < end; ++i) {

std::wstring line = toLower(lines[i]);

std::wstringstream ss(line);

std::wstring word;

while (ss >> word) {

word = cleanWord(word);

if (!word.empty()) {

for (const std::wstring& keyword : keywords) {

if (word == keyword) {

localCount[keyword]++;

}

}

}

}

}

std::lock\_guard<std::mutex> guard(countMutex);

for (const auto& pair : localCount) {

wordCount[pair.first] += pair.second;

}

return 0;

}

int main() {

SetConsoleOutputCP(CP\_UTF8);

std::wcout.imbue(std::locale(""));

std::wifstream file(inputFile);

file.imbue(std::locale(file.getloc(), new std::codecvt\_utf8<wchar\_t>));

if (!file) {

std::wcerr << L"Ошибка открытия файла!\n";

return 1;

}

std::wstring line;

while (std::getline(file, line)) {

lines.push\_back(line);

}

file.close();

HANDLE threads[threadCount];

int threadIds[threadCount];

for (int i = 0; i < threadCount; ++i) {

threadIds[i] = i;

threads[i] = CreateThread(nullptr, 0, SearchKeywords, &threadIds[i], 0, nullptr);

if (!threads[i]) {

std::wcerr << L"Ошибка создания потока " << i << L"!\n";

return 1;

}

}

WaitForMultipleObjects(threadCount, threads, TRUE, INFINITE);

for (int i = 0; i < threadCount; ++i) {

CloseHandle(threads[i]);

}

std::wofstream outFile(outputFile);

outFile.imbue(std::locale(outFile.getloc(), new std::codecvt\_utf8<wchar\_t>));

for (const auto& pair : wordCount) {

outFile << pair.first << L": " << pair.second << L"\n";

}

outFile.close();

std::wcout << L"Анализ завершён. Результаты записаны в output.txt\n";

system("pause");

return 0;

}

1. **GitHub**

Ссылка на проект:<https://github.com/diana767/pra2.git>

1. **Вывод**

Я получила навыки объектно-ориентированного программирования по разработке многозадачных приложений в OC Windows