ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

ПРОГРАММА ”ПЕРВАЯ ВОЕННАЯ ЗАДАЧА” (ВАРИАНТ 23)

Пояснительная записка

Исполнитель
студент группы БПИ196
_____ / Татаринов Н.А. /
«29» ноября 2020 г.

Москва 2020

Пояснительная записка

Листов 15

Москва 2020

АННОТАЦИЯ

В данном программном документе приведена пояснительная записка к программе «Первая военная задача».

В разделе «Введение» указано наименование программы и краткая характеристика области её применения.

В разделе «Технические характеристики» содержатся следующие подразделы:

постановка задачи на разработку программы;

описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами;

описание и обоснование выбора метода организации входных и выходных данных.

В разделе «Код программы» исходный текст программы на языке Ассемблер с подробными комментариями необходимыми для понимания программы сторонним человеком.

Содержание

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы – «Первая военная задача».

1.2. Краткая характеристика области применения

Текст задания: «Темной-темной ночью прапорщики Иванов, Петров и Нечепорчук занимаются хищением военного имущества со склада родной военной части. Будучи умными людьми и отличниками боевой и строевой подготовки, прапорщики ввели разделение труда: Иванов выносит имущество со склада, Петров грузит его в грузовик, а Нечепорчук подсчитывает рыночную стоимость добычи. Требуется составить многопоточное приложение, моделирующее деятельность прапорщиков. При решении использовать парадигму «производитель-потребитель»».

Программа выполняется в рамках курса «Архитектура вычислительных систем» в соответствии с учебным планом подготовки бакалавров по направлению 09.03.04 «Программная инженерия» Национального исследовательского университета «Высшая школа экономики», факультет компьютерных наук, департамент программной инженерии.

2. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

2.1. Постановка задачи на разработку программы

Программа должна создать файл из случайного количества строк [10; 50], в каждой из которых должны быть наименование товара (случайная строка из латинских символов и цифр длины [10;15]) и его цена (целое число [1000; 100000]), разделённые одним пробелом. Далее, должны быть созданы 3 потока (с помощью технологии OpenMP), 1 из которых - поток-производитель, а 2 других - потоки-потребители. Поток-потребитель получает данные из файла, сохраняя их в 2 буфера, а потоки-потребители формируют 2 других соответствующих файла с наименованиями и ценами по отдельности (в конце файла с ценами должна быть суммарная стоимость всех товаров).

2.2. Описание алгоритма и функционирования программы

Генерация строк представлена методом static std::string generate_random_string (std::mt19937& engine), где engine - движок для генерации случайных чисел. Создаётся массив всех возможных символов строки, из которого по случайному индексу берутся элементы для случайной строки.

За создание файла отвечает метод static void create_input_file(). В нём генерируется случайным образом количество товаров на складе и стоимости каждого из них. Все наименования товаров (получаются из вышеописанного метода случайных строк) и их цены выводятся в цикле в файл, откуда потом их читает поток-производитель.

Потоки-производители и потоки-потребители представлены классами class Producer и template<class T> class Consumer соответственно. Буфер данных для хранения наименований товаров и их цен представлен классом template<class T> class Buffer.

Полями класса буфера данных являются максимальное число элементов static const size_t st_max_size = 5, одинаковое для всех экземпляров класса, и контейнер std::stack<T>* st стекового типа.

Методоми класса буфера данных являются добавление - void add(const T& newElement) - и удаление - T remove().

Полями класса производителя являюся буфер наименований товаров - Buffer<std::string>* units,- буфер цен товаров - Buffer<unsigned int>* prices,- и переменная для ввода данных из файла - std::ifstream fin (путь к нему передаётся через конструктор, как и буферы - по ссылке).

Методом класса производителя является метод считывания данных из файла и их обработки - void run().

Полями класса потребителя являются буфер данных (произвольного типа; в программе используются строковые - наименования - и целочисленные - цены) - Buffer<T>* objects, - переменная для вывода данных в файл - std::ofstream fout,- и целочисленная переменная суммарной стоимости товаров - unsigned long long overall_price (используется только в потребителе, хранящем в буфере цены товаров).

Методом класса потребителя является метод обработки данных и вывода их в файл - void run().

2.3. Описание и обоснование выбора метода организации входных и выходных данных

Генерация случайных строк и стоимостей товаров с сохранением их в файл является более удобной, чем ввод данных из консоли с каждым запуском программы. Проверять в таком формате также удобнее.

3. КОД ПРОГРАММЫ

```
#include "omp.h"
#include <iostream>
#include <fstream>
#include <thread>
#include <stack>
#include <condition_variable>
#include <random>

///Блокировщик потоков в моменты добавления в стек (удаления из стека) и
///вывода комментариев в консоль.
omp_lock_t lock;
///Отражает, считал ли поток-производитель все данные из входного файла.
bool file_is_empty = false;

///Представляет собой класс для обмена данными между
///производителями и потребителями.
template<class T>
class Buffer {
private:
    ///Максимальное количество элементов в буфере.
    static const size_t st_max_size = 5;
    ///Контейнер для хранения элементов буфера.
    std::stack<T>* st;

public:
    Buffer();
    ~Buffer();

    ///Добавляет новый элемент в буфер.
    void add(const T& newElement);

    ///Удалаяет верхний элемент из буфера.
    T remove();
};

///Представляет собой класс производителя.
class Producer{
private:
    ///Буфер наименований товаров.
    Buffer<std::string>* units;
    ///Буфер цен товаров.
    Buffer<unsigned int>* prices;
    ///Файл, откуда считываются товары.
    std::ifstream fin;

public:
    Producer(Buffer<std::string>* _units,
             Buffer<unsigned int>* _prices,
             const std::string& path);
    ~Producer();

    ///Метод работы с данными.
    void run();
};

///Представляет собой класс потребителя.
template<class T>
class Consumer{
```

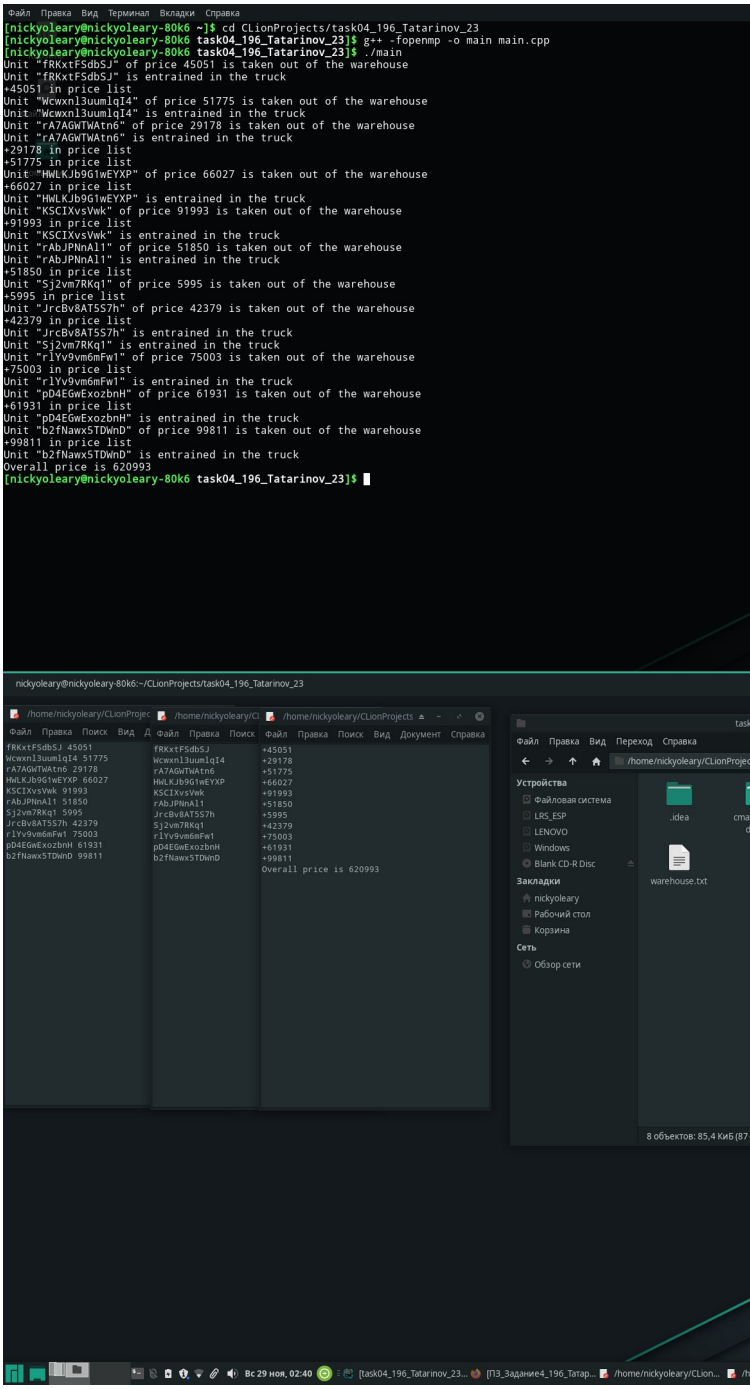
```
private:
    ///Буфер товаров (наименований или стоимостей).
    Buffer<T>* objects;
    ///Файл, в который выводятся товары.
    std::ofstream fout;
    ///Суммарная стоимость товаров (если потребитель
    ///хранит стоимости).
    unsigned long long overall_price;

public:
    Consumer(Buffer<T>* _objects, const std::string& path);
    ~Consumer();

    ///Метод работы с товарами.
    void run();
};

///Метод генерации случайной строки из латинских букв и цифр
///длиной от 10 до 15 символов.
static std::string generate_random_string(std::mt19937& engine);

///Метод создания файла с товарами.
static void create_input_file();

int main() {
    create_input_file();
    auto units = new Buffer<std::string>();
    auto prices = new Buffer<unsigned int>();
    auto Ivanov = new Producer(units, prices, "warehouse.txt");
    auto Petrov = new Consumer<std::string>(units, "truck.txt");
    auto Necheporchuk = new Consumer<unsigned int>(prices, "price_list.txt");

#pragma omp parallel num_threads(4)
    {
        omp_init_lock(&lock);
#pragma omp sections
        {
#pragma omp section
            {
                Ivanov->run();
            }
#pragma omp section
            {
                Petrov->run();
            }
#pragma omp section
            {
                Necheporchuk->run();
            }
        }
        omp_destroy_lock(&lock);
    }

    delete Ivanov;
    delete Petrov;
    delete Necheporchuk;
    delete units;
    delete prices;
    return 0;
}

static std::string generate_random_string(std::mt19937& engine){
    std::uniform_int_distribution<unsigned char> dist1(10, 15);
```

```
    std::uniform_int_distribution<unsigned char> dist2(0, 61);
    std::string chars_to_use =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
    std::string result;
    unsigned char result_length = dist1(engine);
    for(size_t i = 0; i < result_length; i++){
        result += chars_to_use[dist2(engine)];
    }
    return result;
}

static void create_input_file(){
    std::ofstream fout;
    fout.open("warehouse.txt");
    std::random_device rd;
    std::mt19937 engine(rd());
    //Количество товаров на складе.
    std::uniform_int_distribution<unsigned char> dist1(10, 50);
    //Цена каждого товара.
    std::uniform_int_distribution<unsigned int> dist2(1000, 100000);
    unsigned char amount_of_units = dist1(engine);
    for(unsigned char i = 0; i < amount_of_units; i++){
        fout << generate_random_string(engine) << " " << dist2(engine) << "\n";
    }
    fout.close();
}

template<class T>
Buffer<T>::Buffer() {
    st = new std::stack<T>();
}

template<class T>
Buffer<T>::~Buffer(){
    delete st;
}

template<class T>
void Buffer<T>::add(const T& newElement) {
    if(st->size() == st_max_size){
        omp_unset_lock(&lock);
        while(st->size() == st_max_size);
        omp_set_lock(&lock);
    }
    st->push(newElement);
}

template<class T>
T Buffer<T>::remove(){
    if(st->empty()){
        omp_unset_lock(&lock);
        while(st->empty()){
            if(file_is_empty){
                break;
            }
        }
        omp_set_lock(&lock);
    }
    T top = st->top();
    st->pop();
    return top;
}
```

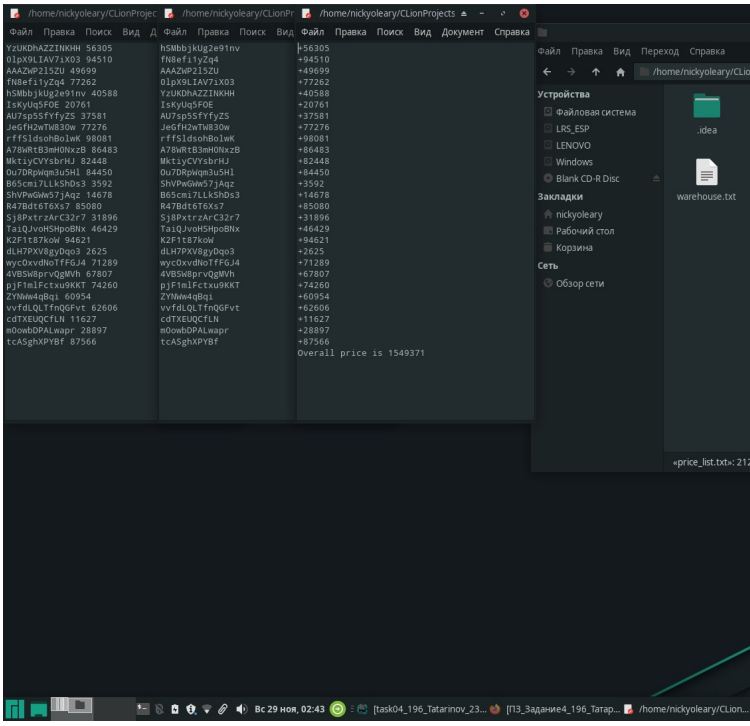
```cpp
Producer::Producer(Buffer<std::string>* _units,
                   Buffer<unsigned int>* _prices,
                   const std::string& path) : units(_units), prices(_prices){
    fin.open(path);
}

Producer::~Producer() {
    units = nullptr;
    prices = nullptr;
    fin.close();
}

void Producer::run() {
    std::string unit;
    unsigned int price;
    while(fin >> unit >> price){
        omp_set_lock(&lock);
        units->add(unit);
        prices->add(price);
        std::cout << "Unit \"" << unit << "\" of price " << price <<
                "" is taken out of the warehouse\n";
        omp_unset_lock(&lock);
    }
    file_is_empty = true;
}

template<class T>
Consumer<T>::Consumer(Buffer<T>* _objects, const std::string& path)
        : objects(_objects), overall_price(0){
    fout.open(path);
}

template<class T>
Consumer<T>::~Consumer() {
    if(typeid(T).name() == typeid(unsigned int).name()) {
        omp_set_lock(&lock);
        std::cout << "Overall price is " << overall_price << "\n";
        fout << "Overall price is " << overall_price << "\n";
        omp_unset_lock(&lock);
    }
    objects = nullptr;
    fout.close();
}

template<class T>
void Consumer<T>::run() {
    while(!file_is_empty) {
        omp_set_lock(&lock);
        T unit = objects->remove();
        if (typeid(T) == typeid(std::string)) {
            fout << unit << "\n";
            std::cout << "Unit \"" << unit << "\" is entrained in the truck\n";
        } else if (typeid(T) == typeid(unsigned int)) {
            overall_price += *reinterpret_cast<unsigned int *>(&unit);
            fout << "+" << unit << "\n";
            std::cout << "+" << unit << " in price list\n";
        }
        omp_unset_lock(&lock);
    }
}
```

4. ТЕСТИРОВАНИЕ

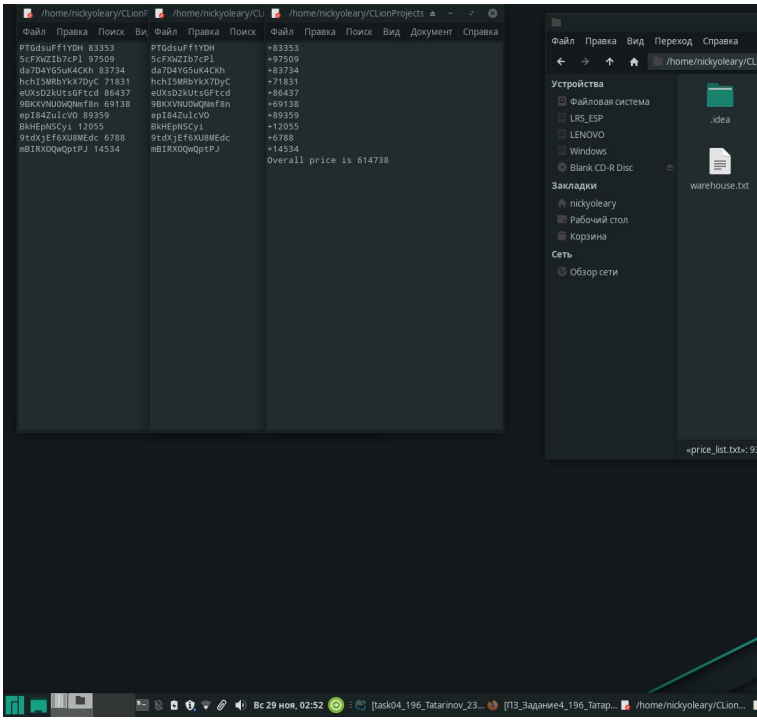
Производится через командную строку. Для компиляции файла “main.cpp” (Linux Manjaro): “g++ -fopenmp –o main main.cpp”. Для запуска исполняющего файла “main.exe”: “./main”,


Файл Правка Вид Терминал Вкладки Справка
[nickyoleary@nickyoleary-80k6 ~]$ cd CLionProjects/task04_196_Tatarinov_23
[nickyoleary@nickyoleary-80k6 task04_196_Tatarinov_23]$ g++ -fopenmp -o main main.cpp
[nickyoleary@nickyoleary-80k6 task04_196_Tatarinov_23]$ ./main
Unit "fRKxtFSdbSJ" of price 45051 is taken out of the warehouse
Unit "fRKxtFSdbSJ" is entrained in the truck
+45051 in price list
Unit "Wcwxnl3uumlqI4" of price 51775 is taken out of the warehouse
Unit "Wcwxnl3uumlqI4" is entrained in the truck
Unit "rA7AGWTWAtn6" of price 29178 is taken out of the warehouse
Unit "rA7AGWTWAtn6" is entrained in the truck
+29178 in price list
+51775 in price list
Unit "HWLKJb9G1wEYXP" of price 66027 is taken out of the warehouse
+66027 in price list
Unit "HWLKJb9G1wEYXP" is entrained in the truck
Unit "KSCIXvsVwk" of price 91993 is taken out of the warehouse
+91993 in price list
Unit "KSCIXvsVwk" is entrained in the truck
Unit "rAbJPNnAl1" of price 51850 is taken out of the warehouse
Unit "rAbJPNnAl1" is entrained in the truck
+51850 in price list
Unit "Sj2vm7RKq1" of price 5995 is taken out of the warehouse
+5995 in price list
Unit "JrcBv8AT5S7h" of price 42379 is taken out of the warehouse
+42379 in price list
Unit "JrcBv8AT5S7h" is entrained in the truck
Unit "Sj2vm7RKq1" is entrained in the truck
Unit "rlYv9vm6mFw1" of price 75003 is taken out of the warehouse
+75003 in price list
Unit "rlYv9vm6mFw1" is entrained in the truck
Unit "pD4EGwExozbnH" of price 61931 is taken out of the warehouse
+61931 in price list
Unit "pD4EGwExozbnH" is entrained in the truck
Unit "b2fNawx5TDWnD" of price 99811 is taken out of the warehouse
+99811 in price list
Unit "b2fNawx5TDWnD" is entrained in the truck
Overall price is 620993
[nickyoleary@nickyoleary-80k6 task04_196_Tatarinov_23]$
nickyoleary@nickyoleary-80k6:~/CLionProjects/task04_196_Tatarinov_23
Устройства
Файловая система
LRS_ESP
LENOVO
Windows
Blank CD-R Disc
Закладки
nickyoleary
Рабочий стол
Корзина
Сеть
Обзор сети
.idea
warehouse.txt
Вс 29 ноя, 02:40


nickyoleary@nickyoleary-80k6:~/CLionProjects/task04_196_Tatarinov_23
nickyoleary@nickyoleary-80k6:~/CLionProjects/task04_196_Tatarinov_23


/home/nickyoleary/CLionProjects
Файл Правка Поиск Вид Документ Справка
YzUKDhAZZINKHH 56305
OlpX9LIAV7iXO3 94510
AAAZWP215ZU 49699
fN8efi1yZq4 77262
hSNbbjkUg2e9inv 40588
IsKyUqSFOE 20761
AU7sp5SfYfyZS 37581
JeGfH2WTW830w 77276
rffSldsohBolwK 98081
A78WRtB3mHONxzB 86483
MktiyCVYsbrHJ 82448
Ou7DRpWqm3u5Hl 84450
B65cmi7LLkShDs3 3592
ShVPwGWw57jAqz 14678
R47Bdt6T6Xs7 85080
Sj8PxtrzArC32r7 31896
TaiQJvoHSHpoBNx 46429
K2F1t87koW 94621
dLH7PXV8gyDqo3 2625
wycOxvdNoTfF6J4 71289
4VBSW8prvQgNVh 67807
pjF1mlFctxu9KKT 74260
ZYNWw4qBqi 60954
vvfdLQLTfnQGFvt 62606
cdTXEUQCfLN 11627
mOowbDPALwapr 28897
tcASghXPYBf 87566
+56305
+94510
+49699
+77262
+40588
+20761
+37581
+77276
+98081
+86483
+82448
+84450
+3592
+14678
+85080
+31896
+46429
+94621
+2625
+71289
+67807
+74260
+60954
+62606
+11627
+28897
+87566
Overall price is 1549371
Устройства
Файловая система
LRS_ESP
LENOVO
Windows
Blank CD-R Disc
Закладки
nickyoleary
Рабочий стол
Корзина
Сеть
Обзор сети
.idea
warehouse.txt
«price_list.txt»: 212
Вс 29 ноя, 02:43


nickyoleary@nickyoleary-80k6:~/CLionProjects/task04_196_Tatarinov_23



СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) Параллельное программирование на OpenMP [Электронный ресурс] //URL: http://ccfit.nsu.ru/arom/data/openmp.pdf (режим доступа: свободный, дата обращения: октябрь 2020)