

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
Факультет безопасности информационных технологий**

Дисциплина:

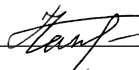
«Технологии и методы программирования»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

Выполнили:

студенты группы N33491

Чапасов Пётр Константинович

_____ 
(подпись)

Проверил:

Ищенко Алексей Петрович

(подпись)

Санкт-Петербург

2022 г.

Техническое Задание

Задание на лабораторную работу №2:

Разработать простейшую программу, запрашивающую ФИО пользователя и заносящую эту информацию в текстовый файл. Если такое ФИО имеется в файле, то выдавать об этом сообщение. После ввода информации программа должна завершать работу и сообщать пользователю о лимитах ее использования (временного или количества запусков). По достижении лимита запусков программа должна предложить пользователю приобрести ее полную версию или деинсталлировать себя. При повторной установке программы, она должна сообщать о своем предыдущем нахождении на этом компьютере и сверяться с прошлыми лимитами пользования (т.е. не давать их суммарно превысить).

Выполняются две версии программы (можно сочетать в одной):

- а) Time-limited (ограничение по времени сделать не более 3 минут, чтобы можно было проследить в момент сдачи достижение лимита).
- б) Start-limited (ограничение на количество запусков тоже должно быть наглядным, например – 4-5).

Выполнение работы

Вариант А

Инструкция по использованию:

Функционал:

Программа запоминает имена, которые вы вводили. Вы можете запустить программу 3 раза. На 4 раз она попросит лиду ввести ключ, либо удалить её. Если установить программу заново, она все равно будет помнить сколько у вас осталось попыток.

Установка:

1. Устанавливаем пакет *lab2_1.deb* указав путь до пакета (Рис. 1)

```
$ sudo apt install /home/petrchapasov/lab2/lab2_1.deb
```

Внимание! Данная инструкция написана под linux-gnu. На других вариантах linux установка выглядит следующим образом:

1. Распаковать архив *source.zip*
2. Закомментировать 65 строку и раскомментировать 66 и 67 строки в *script.cpp*
3. Компилируем *script.cpp*

```
$ g++ script.cpp -o script
```

4. Создаём файл *names.txt*

```
(petrchapasov@kali)-[~/lab2]
$ sudo apt install /home/petrchapasov/lab2/lab2_1.deb
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Заметьте, вместо «/home/petrchapasov/lab2/lab2_1.deb» выбирается «lab2»
Следующие НОВЫЕ пакеты будут установлены:
  lab2
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пак
етов, и 1285 пакетов не обновлено.
Необходимо скачать 0 В/7 368 В архивов.
После данной операции объём занятого дискового пространства возрастёт на 61,4
кВ.
Пол:1 /home/petrchapasov/lab2/lab2_1.deb lab2 all 1.0-3 [7 368 В]
Выбор ранее не выбранного пакета lab2.
(Чтение базы данных ... на данный момент установлено 311160 файлов и каталогов.
)
Подготовка к распаковке .../petrchapasov/lab2/lab2_1.deb ...
Распаковывается lab2 (1.0-3) ...
Настраивается пакет lab2 (1.0-3) ...
Обрабатываются триггеры для kali-menu (2022.3.1) ...
```

Рисунок 1. Установка deb-пакета

Использование:

1. Программа устанавливается в `/usr/bin/lab2` (в случае ручной установки, в нашей папке). Для запуска программы переходим в каталог и запускаем её.

```
$ sudo ./script
```

2. Вводим имена для запоминания (Рис. 2).
3. Для удаления программы использовать (Рис. 6):

```
$ sudo apt-get remove lab2
```

Для установки полной версии, после запроса ключа введите: 19051917

```
(petrchapasov@kali)-[~/lab2]
$ sudo ./script
Loading ...
Welcome!
Input username: Dima
Completed
Remaining trial runs: 2
```

Рисунок 2.1. Ввели новое имя

```
(petrchapasov@kali)-[~/lab2]
$ sudo ./script
Loading ...
Input username: Petr
Name has been already passed
Remaining trial runs: 1
```

Рисунок 2.2. Вводим уже существующее имя

```
(petrchapasov@kali)-[~/lab2]
$ sudo ./script
Loading ...
Buy the full version or uninstall the program
B/D?
D
rm: невозможно удалить 'rm': Нет такого файла или каталога
rm: невозможно удалить 'rm': Нет такого файла или каталога
```

Рисунок 3. Превышение попыток и удаление (Программа установлена вручную)

```
(petrchapasov@kali)-[~/lab2]
$ sudo ./script
Loading ...
Buy the full version or uninstall the program
B/D?
D
rm: невозможно удалить 'rm': Нет такого файла или каталога
rm: невозможно удалить 'rm': Нет такого файла или каталога

(petrchapasov@kali)-[~/lab2]
$ g++ script.cpp -o script

(petrchapasov@kali)-[~/lab2]
$ sudo ./script
Loading ...
Buy the full version or uninstall the program
B/D?
B
Enter the key: 19051917
Successfully! Restart the program
```

Рисунок 4. Установка после удаления. Запуск и покупка. (Ручная установка)

```
(petrchapasov@kali)-[~/lab2]
$ sudo ./script
Loading ...
This is full version! Thank for your purchase
Input username: P
Completed
```

Рисунок 5. Запуск после покупки.

```

(petrchapasov@kali)-[/usr/bin/lab2]
$ sudo apt remove lab2
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующие пакеты будут УДАЛЕНЫ:
 lab2
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 1 пак
етов, и 1285 пакетов не обновлено.
После данной операции объём занятого дискового пространства уменьшится на 61,
4 kB.
Хотите продолжить? [Д/н] у
(Чтение базы данных ... на данный момент установлено 311163 файла и каталога.)
Удаляется lab2 (1.0-3) ...
Обрабатываются триггеры для kali-menu (2022.3.1) ...

```

Рисунок 6. Удаление пакета

Исходный код:

script.cpp:

```

#include <iostream>
#include <fstream>
#include <fstream>
#include <string.h>
#include <cstdlib>
#include <cstring>
#include <cmath>

using namespace std;

int checkinst();
void logging(int attempts, int type);
int HashFunc(int k);

int main() {
    string old_n, new_n;
    int tmp = 0;
    int atm;

    cout << "Loading..." << endl;
    atm = checkinst();
    if (atm || atm == -1) {
        ifstream file("names.txt"); //Проверка повтора имен
        cout << "Input username: ";
        cin >> new_n;
        while (getline(file, old_n)) {
            if (old_n.compare(new_n) == 0) {
                cout << "Name has been already passed" <<
endl;

                tmp = 1;
                break;
            }
        }
    }
}

```

```

    }
    file.close();
    if (!tmp) {
        ofstream os("names.txt", ios::out | ios::app);
        os << new_n << endl;
        cout << "Completed" << endl;
        os.close();
    }
    logging(atm, 0);
}
else {
    string des, b="b", B="B", d="d", D="D"; //Выбор варианта
покупки или продажи

    int key_t = 242554996, key_b;

    cout << "Buy the full version or uninstall the program"
<< endl;
    flg:

    cout << "B/D?" << endl;
    cin >> des;
    if (des.compare(B) == 0 or des.compare(b) == 0) {
        cout << "Enter the key: ";
        cin >> des;
        key_b = atoi(des.c_str());
        if (HashFunc(key_b) == key_t) {
            cout << "Successfully! Restart the program"
<< endl;

            logging(atm, 1);
        }
        else {
            cout << "Invalid key!" << endl;
            goto flg;
        }
    }

    else if (des.compare(D) == 0 or des.compare(d) == 0){
        system("sudo apt-get remove lab2");
        //system("sudo rm names.txt");
        //system("sudo rm script");
    }

    else
        goto flg;
}
return 0;
}

int checkinst() {
    string atm1, atm2;
    int attempts, tmp1, tmp2, tmp3;
    string b, del = "Uninstall", bou = "Bought";
    fstream logs("/var/log/lab.log");
    if (!logs.good()){

```

```

        system("sudo touch /var/log/lab.log");
        cout << "Welcome!" << endl;
        logs.close();
        return 3;
    }
    else {
        while (getline(logs, atm2)) {
            tmp2 ++;
            if (atm2.compare(del) == 0) {
                tmp1 = tmp2;
                continue;
            }
            else if (atm2.compare(bou) == 0) {
                tmp3 = 1;
                continue;
            }
            atm1 = atm2;
        }
        if (tmp1 == tmp2) //Проверка наличия программы до этого
            cout << "The program has been already
installed on this computer before" << endl;
        attempts = atoi(atm1.c_str());
        logs.close();
        if (tmp3 == 1) { //Наличие полной версии
            cout << "This is full version! Thank for your
purchase" << endl;

            attempts = -1;
        }
        return attempts;
    }
}

void logging(int attempts, int type) { //Логгирование
    if (!type) {
        if (attempts != -1) {
            ofstream os("/var/log/lab.log", ios::out |
ios::app);

            attempts -= 1;
            cout << "Remaining trial runs: " << attempts <<
endl;

            os << attempts << endl;
            os.close();
        }
        else {
            ofstream os("/var/log/lab.log", ios::out |
ios::app);

            os << "Bought" << endl;
            os.close();
        }
    }
    else if (type) {
        ofstream os("/var/log/lab.log", ios::out | ios::app);
        os << "Bought" << endl;
        os.close();
    }
}

```



```

    }
}

int HashFunc(int k) {
    int n = 1000000000;
    double a = 0.618033988;
    int h = n * fmod (k * a, 1);
    return h;
}

```

Вариант Б

Внимание! Перед запуском второго варианта необходимо удалить log-файл старой программы (Рис. 7).

```
$ sudo rm lab.log
```

В противном случае программа будет считать, что она уже была установленная на этот компьютер.

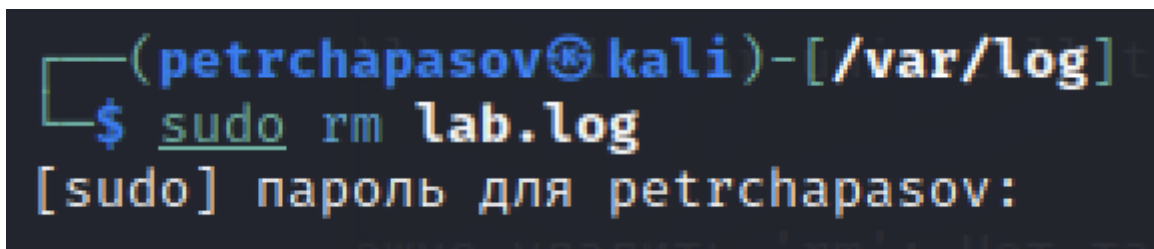


Рисунок 7. Удаление log-файла

Инструкция по использованию:

Функционал:

Программа запоминает имена, которые вы вводили. Вы можете использовать программу 3 минуты. Как только 3 минуты истечет и вы попытаетесь снова ввести имя она попросит лиду ввести ключ, либо удалить её. Если установить программу заново, она все равно будет помнить сколько у вас осталось попыток.

Всё, что касается установки и использования, указано в варианте А.

В случае ручной установки закомментировать 110 строку и раскомментировать 111 и 112 строки в файле *script2.cpp*

```
(petrchapasov@kali)-[~/lab2]
$ sudo ./script
[sudo] пароль для petrchapasov:
Loading ...
Welcome!
Input username or ':q' for exit: Petr
Remaining trial sec: 175
Completed
Input username or ':q' for exit: Dmitriy
Remaining trial sec: 156
Completed
Input username or ':q' for exit: Petr
Name has been already passed
Input username or ':q' for exit: :q
Remaining trial sec: 142
```

Рисунок 8. Использование программы

```
(petrchapasov@kali)-[~/lab2]
$ sudo ./script
Loading ...
Input username or ':q' for exit: Roma
Remaining trial sec: 0
Completed
Trial time is up
Buy the full version or uninstall the program
B/D?
B
Enter the key: 19171905
Invalid key!
B/D?
d
rm: невозможно удалить 'rm': Нет такого файла или каталога
rm: невозможно удалить 'rm': Нет такого файла или каталога
```

Рисунок 9. Ввод неверного ключа по истечению срока

Исходный код:

script.cpp:

```
#include <iostream>
#include <fstream>
#include <fstream>
#include <string.h>
#include <cstdlib>
#include <cstring>
#include <ctime>
#include <cmath>

using namespace std;

int checkinst();
void logging(int attempts, int type);
int HashFunc(int k);

int main() {
    string old_n, new_n, exit = ":q";
    int atm;

    cout << "Loading..." << endl;
    atm = checkinst();
    time_t start = time(0); //В целом все тоже самое, но добавили
    функцию проверки по времени
    if (atm > 0) { //Полная версия
        while(((int) (time(0) - start)) < atm) {

            int tmp = 0;
            ifstream file("names.txt");
            cout << "Input username or ':q' for exit: ";
            cin >> new_n;
            if (exit.compare(new_n) == 0) {
                int atm1;
                atm1 = atm - (int) (time(0) - start);
                if (atm1 < 0)
                    atm1 = 0;
                logging(atm1, 0);
                return 0;
            }
            while (getline(file, old_n)) {
                if (old_n.compare(new_n) == 0) {
                    cout << "Name has been already
passed" << endl;

                    tmp = 1;
                    break;
                }
            }
            file.close();
            if (!tmp) {
                int atm1;
```

```

ios::app);

ofstream os("names.txt", ios::out |

os << new_n << endl;
atm1 = atm - (int) (time(0) - start);
if (atm1 < 0)
    atm1 = 0;
logging(atm1, 0);
cout << "Completed" << endl;
os.close();
}
}
}
else if (atm == -1) { //Триал версия
while (1) {
    int tmp = 0;
    ifstream file("names.txt");
    cout << "Input username or ':q' for exit: ";
    cin >> new_n;
    if (exit.compare(new_n) == 0) {
        logging(atm, 0);
        return 0;
    }
    while (getline(file, old_n)) {
        if (old_n.compare(new_n) == 0) {
            cout << "Name has been already
passed" << endl;

            tmp = 1;
            break;
        }
    }
    file.close();
    if (!tmp) {
        ofstream os("names.txt", ios::out |

ios::app);

os << new_n << endl;
cout << "Completed" << endl;
os.close();
    }
}
}
else {
f1:

    string des, b="b", B="B", d="d", D="D";
    int key_t = 242554996, key_b;

    cout << "Buy the full version or uninstall the program"
<< endl;
    flg:

    cout << "B/D?" << endl;
    cin >> des;
    if (des.compare(B) == 0 or des.compare(b) == 0) {
        cout << "Enter the key: ";
        cin >> des;
        key_b = atoi(des.c_str());

```

```

        if (HashFunc(key_b) == key_t) {
            cout << "Successfully! Restart the program"
<< endl;

            logging(atm, 1);
            return 0;
        }
        else {
            cout << "Invalid key!" << endl;
            goto flg;
        }
    }

    else if (des.compare(D) == 0 or des.compare(d) == 0) {
        system("sudo apt-get remove lab2");
        //system("sudo rm rm names.txt");
        //system("sudo rm rm script");
        return 0;
    }

    else
        goto flg;
}
cout << "Trial time is up" << endl;
goto f1;
}

```

```

int checkinst() {
    string atm1, atm2;
    int attempts, tmp1, tmp2, tmp3;
    string b, del = "Uninstall", bou = "Bought";
    fstream logs("/var/log/lab.log");
    if (!logs.good()){
        system("sudo touch /var/log/lab.log");
        cout << "Welcome!" << endl;
        logs.close();
        return 180;
    }
    else {
        while (getline(logs, atm2)) {
            tmp2 ++;
            if (atm2.compare(del) == 0) {
                tmp1 = tmp2;
                continue;
            }
            else if (atm2.compare(bou) == 0) {
                tmp3 = 1;
                continue;
            }
            atm1 = atm2;
        }
        if (tmp1 == tmp2)
            cout << "The program has been already
installed on this computer before" << endl;
        attempts = atoi(atm1.c_str());
    }
}

```

```

        logs.close();
        if (tmp3 == 1) {
            cout << "This is full version! Thank for your
purchase" << endl;
            attempts = -1;
        }
        return attempts;
    }

}

void logging(int attempts, int type) {
    if (!type) {
        if (attempts != -1) {
            ofstream os("/var/log/lab.log", ios::out |
ios::app);

            cout << "Remaining trial sec: " << attempts <<
endl;

            os << attempts << endl;
            os.close();
        }
        else {
            ofstream os("/var/log/lab.log", ios::out |
ios::app);

            os << "Bought" << endl;
            os.close();
        }
    }
    else if (type) {
        ofstream os("/var/log/lab.log", ios::out | ios::app);
        os << "Bought" << endl;
        os.close();
    }
}

int HashFunc(int k) {
    int n = 1000000000;
    double a = 0.618033988;
    int h = n * fmod (k * a, 1);
    return h;
}

```

Дополнительно:

Чтобы обойти проверку, достаточно перейти в каталог */var/log* и в файле *lab.log* прописать слово “*Bought*”. Таким образом программа будет думать, что у вас куплена полная версия.

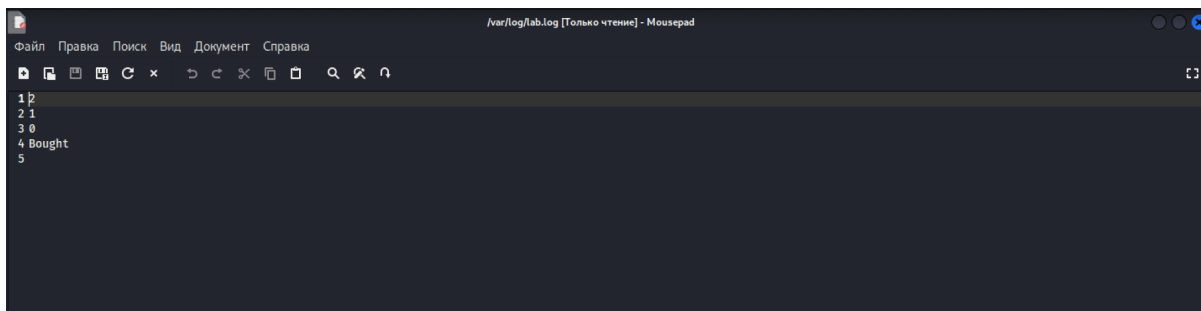


Рисунок 10. Обход защиты