**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Информационной безопасности**

отчет

**по лабораторной работе №2**

**по дисциплине «Программирование»**

Тема: Разработка хэш-функции

(с использованием односвязного списка)

|  |  |  |
| --- | --- | --- |
| Студент гр. 6361 |  | Нерсисян А. |
| Преподаватель |  | Халиуллин Р.А. |

Санкт-Петербург

2017

**Постановка задачи**

Реализовать хэш-функцию в программе. Написать программу для аутентификации пользователя с помощью пароля. Программа должна работать в двух режимах:

1. Создание пароля 2. Проверка пароля

Пароль не должна сохранится в программе. Для этого нужно использовать хэширование\* (хэш-функцию). Во время выполнения программы пароль будет записан в односвязный список.

**\*Хэширование** — преобразование массива входных данных произвольной длины в (выходную) битовую строку фиксированной длины, выполняемое определённым алгоритмом.

**Исходный код**

#include <cstdio>

#include <iostream>

#include <cstdlib>

#include <conio.h> // для getch()

using namespace std;

struct Node

{

char element;

Node \*next;

};

Node\* InPassword()

{

char element;

Node \*delta, \*Head = NULL, \*Back;

while (1)

{

element = getch();

if (element == 13)

{

Back->next = NULL;

break;

}

else

{

delta = new Node;

delta->element = element;

if (!(Head)) Head = delta;

else Back->next = delta;

Back = delta;

}

}

return Head;

}

int HashFunc(Node\* pass)

{

int pow = 1, i = 1;

unsigned long long hash = 0;

while (pass != NULL)

{

hash += ((((((pass->element - 'a') << 5) \* 24) ^ (pass->element))\*pass->element)\*i) ^ i;

i++;

Node\* password = pass;

pass = pass->next;

delete password;

}

return hash;

}

int main()

{

char mode;

FILE \*file;

Node\* pass;

start:

printf("Please choose the programm mode\n");

printf("Enter 1 to enter new password\n");

printf("Enter 2 for cheking password\n ");

cin >> mode;

if (mode == 1)

{

printf("Enter the new password and press ENTER\n");

pass = InPassword();

file = fopen("hash.txt", "wb");

if (fopen == NULL)

{

printf("Error, file does not open!");

return -1;

}

fprintf(file, "%d", HashFunc(pass));

fclose(file);

printf("Done");

return 0;

}

if (mode == 2)

{

long oldHash;

file = fopen("hash.txt", "rb");

if (fopen == NULL)

{

printf("Error, file does not open!");

return -1;

}

fscanf(file, "%d", &oldHash);

fclose(file);

printf("Enter password:");

pass = InPassword();

if (HashFunc(pass) == oldHash) printf("ACCESS GRATED!");

else printf("ACCESS DENIED!");

cin.get();

return 0;

}

else

{

printf("%sError!");

goto start;

}

return 0;

}

**Результаты тестирования, рисунки**

Программа реализована следующим образом:

* Запуск программы
* Выбор режима

Режим 1:

* Ввод нового пароля
* Хеширование пароля
* Сохранение пароля в текстовой файл (hash.txt)

Режим 2:

* Ввод пароля
* Хеширование пароля
* Сравнение нового и старого хеша

Программу нужно запустить в командной стоке. После запуска программы нужно ввести 1 или 2 для выбора режима работы программы (рис. 1). После выбора режима нужно вводить новый пароль (рис. 1)

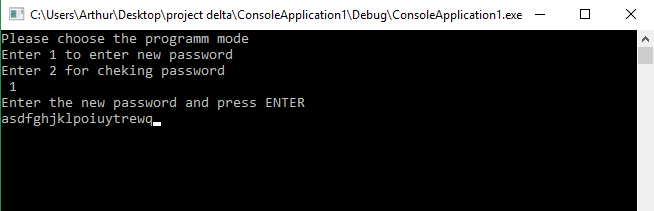
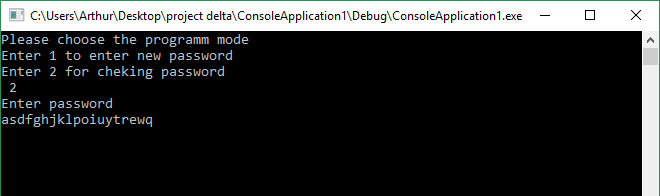


рисунок 1

Потом нужно программу заново запустить и выбрать режим 2 (рис. 2).

Ввести пароль.

рисунок 2

В случае, если пароль введен правильно программа выводит сообщение “Access Grated!” (если введен неправильно, то “Access Denied!”)