Министерство цифрового развития, связи и массовых коммуникаций

Российской Федерации

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

(МТУСИ)

Лабораторная работа №1 по теме:

«РЕАЛИЗАЦИЯ ПРОСТЕЙШЕГО ГЕНЕРАТОРА ПАРОЛЕЙ»

Вариант №1

Выполнил студент группы БФИ2202  
Сидорук Д.В.

**Цель работы**

Получение основных теоретических сведений и практических навыков по оценке стойкости парольной защиты

**Ход работы**

1. Ознакомиться с теоретической частью данной работы.

2. Составить программу-генератор паролей.

3. Составить отчет по проделанной работе.

4. Защитить работу.

**Постановка задачи**

Реализовать простейший генератор паролей, обладающий основными требованиями к парольным генераторам.

Программа должна выполнять следующие действия.

1. Ввод идентификатора пользователя с клавиатуры. Данный идентификатор представляет собой последовательность символов a1, a2, ..., aN, где N — количество символов идентификатора (может быть любым), ai — i-й символ идентификатора пользователя.

2. Формирование пароля пользователя b1, b2, ..., bM для данного

идентификатора, где M — количество символов пароля, соответствующее вашему варианту и вывод его на экран. Алгоритм получения символов пароля bi указан в перечне требований для вашего варианта (таблица 1).

|  |  |  |
| --- | --- | --- |
| Вариант | М | Перечень требований |
| 1 | 6 | *b1, b2 — случайные заглавные буквы английского алфавита; b3 = N2mod10 (где mod 10 — остаток от деления числа на 10); b4 — случайная цифра; b5 — случайный символ из множества {!, ”, #, $, %, &, ’, (, ), \*}; b6 — случайная малая буква английского алфавита* |

**Листинг программы**

#include <iostream>  
#include <random>  
#include <span>  
#include <string>  
#include <string\_view>  
  
class PassGen {  
public:  
 PassGen() : r\_device(), gen(r\_device()) {}  
  
 std::string generate(std::string\_view ident) {  
 std::string res;  
 res.push\_back(getRandomFromSet(uppers));  
 res.push\_back(getRandomFromSet(uppers));  
 res.push\_back(((ident.size() \* ident.size()) % 10) + '0');  
 res.push\_back(getRandomFromSet(digits));  
 res.push\_back(getRandomFromSet(symbols));  
 res.push\_back(getRandomFromSet(lowers));  
 return res;  
 }  
  
private:  
 static std::vector<char> genSetFromRange(char from, char to) {  
 std::vector<char> res;  
 for (char c = from; c != to; ++c)  
 res.push\_back(c);  
 return res;  
 }  
  
 const std::vector<char> uppers = genSetFromRange('A', 'Z');  
 const std::vector<char> lowers = genSetFromRange('a', 'z');  
 const std::vector<char> digits = genSetFromRange('0', '9');  
 const std::vector<char> symbols = {'!', '"', '#', '$', '%',  
 '&', '\'', '(', ')', '\*'};  
  
 char getRandomFromSet(std::span<const char> set) {  
 std::uniform\_int\_distribution<int> dist(0, set.size() - 1);  
 return set[dist(gen)];  
 }  
  
 std::random\_device r\_device;  
 std::mt19937 gen;  
};  
  
int main() {  
 PassGen passGen;  
  
 std::string ident;  
 std::cin >> ident;  
 std::cout << passGen.generate(ident) << '\n';  
}

**Результат выполнения программы**

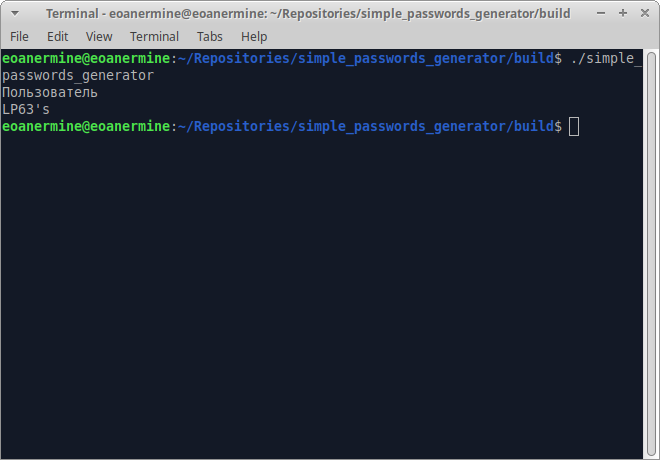


Рисунок 1 - формирование случайного пароля

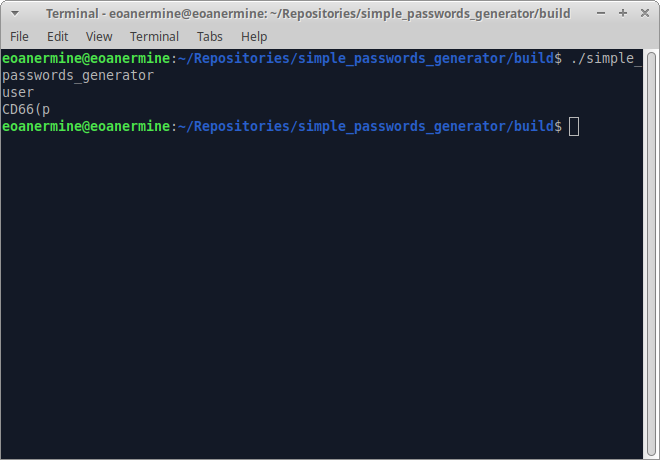


Рисунок 2 - формирование случайного пароля 2