ЛАБОРАТОРНАЯ РАБОТА № 2

РЕАЛИЗАЦИЯ ПОЛИТИК ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ДИСКРЕЦИОННАЯ МОДЕЛЬ ПОЛИТИКИ БЕЗОПАСНОСТИ ВЫПОЛНИЛ: ст. гр. 4401 Сибгатуллин М. Р.

ВАРИАНТ № 12

**Цель** – изучение проблемам реализации политик информационной безопасности в компьютерных системах на примере дискреционной модели.

КОЛИЧЕСТВО СУБЪЕКТОВ ДОСТУПА = 6

КОЛИЧЕСТВО ОБЪЕКТОВ ДОСТУПА = 4

МАТРИЦА ДОСТУПА СУБЪЕКТОВ К ОБЪЕКТАМ в виде, аналогичном представленному в таблице 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Объект / Субъект** | **Файл\_1** | **Файл\_2** | **Файл\_3** | **Файл\_4** |
| **Администратор** | Полные права | Полные права | Полные права | Полные права |
| **Гость** | Запрет | Чтение | Чтение | Запрет |
| **Пользователь\_1** | Чтение, передача прав | Чтение, запись | Полные права | Полный запрет |
| **Пользователь\_2** | Полный доступ | Чтение, запись | Чтение, Передача прав | Полные права |
| **Пользователь\_3** | Запись, Передача прав | Запись | Передача прав | Полный запрет |
| **Пользователь\_4** | Полный запрет | Полный доступ | Чтение, Запись | Запрет |

ТЕКСТ ПРОГРАММЫ

using System;

namespace ЗИ\_Лаба2

{

class Program

{

static string[] Users = new string[] { "Admin", "User1", "User2", "User3", "User4", "User5"}; //Имена пользователей

static string[] Prava = new string[] { "Полный запрет ", "Передача прав ", "Запись ", "Запись, Передача прав", "Чтение ", "Чтение, Передача прав", "Чтение, Запись ", "Полный доступ " }; //Названия прав доступа

static string[] File = new string[] { "File1", "File2", "File3", "File4"}; //Названия файлов

static int[,] M = new int[6, 4]; //Матрица доступа [Кол-во юзеров, кол-во файлов]

static int numUser = -1; //переменная с номером пользователя

static int numFile = -1; //переменная с номером файла

static void Main(string[] args)

{

string name = ""; //Переменная для ввода имени пользователя

Console.SetWindowSize(150, 40);

//Заполнение матрицы доступа

MatricaFunk();

//Работаем ребята!

while (true)

{

Console.Write("\nВведите имя пользователя или нажмите Enter, чтобы выйти...\nUser: ");

name = Console.ReadLine();

if (name == "") return;

//Тайная команда для печати всей матрицы доступа

if (name == "13") MatricaPechat();

//Проверяем наличие пользователя в БД

numUser = UsersFunc(name);

if (numUser == -1)

{

Console.WriteLine("\nИмя пользователя введено неверно, повторите попытку");

}

else

{

Console.WriteLine("Идентификация прошла успешно, добро пожаловать в систему");

PravaFunc(numUser);

}

}

}

//Проверка наличия пользователя в БД

static int UsersFunc(string name)

{

for (int i = 0; i < Users.Length; i++)

{

if (name == Users[i])

{

return i;

}

}

return -1;

}

static void PravaFunc(int nomer)

{

string comand;

Console.WriteLine("\nПеречень Ваших прав:");

for (int i = 0; i < File.Length; i++)

{

Console.WriteLine("{0}: {1}", File[i], Prava[M[nomer, i]]);

}

while (true)

{

Console.Write("\nЖду ваших указаний (0 - выход, 1 - чтение, 2 - запись, 3 - передача прав) \nВведите команду > ");

comand = Console.ReadLine();

switch (comand)

{

case "0": Console.Write("Пользователь {0} вышел из системы\n", Users[nomer]); return;

case "1": ProverkaPrav(comand); break;

case "2": ProverkaPrav(comand); break;

case "3": ProverkaPrav(comand); break;

}

}

}

static void ProverkaPrav(string com)

{

string str; //переменная для ввода имени файла

string BinaryCode; //переменная для перевода в двоичный код

int BC;

Console.Write("Над каким фалом производится операция?\nНазвание файла: ");

str = Console.ReadLine();

for (int i = 0; i < File.Length; i++)

{

if (str == File[i])

{

numFile = i;

}

}

if (numFile == -1)

{

Console.WriteLine("\nФайл не найден, повторите попытку");

}

else

{

if (com == "1") //чтение

{

BinaryCode = Convert.ToString(M[numUser, numFile], 2);

BC = Convert.ToInt32(BinaryCode);

if (BC / 100 == 1)

{

Console.WriteLine("Операция прошла успешно!");

}

else

{

Console.WriteLine("У вас нет прав на эту операцию!");

}

}

if (com == "2") //запись

{

BinaryCode = Convert.ToString(M[numUser, numFile], 2);

BC = Convert.ToInt32(BinaryCode);

if (BC % 100 >= 10)

{

Console.WriteLine("Операция прошла успешно!");

}

else

{

Console.WriteLine("У вас нет прав на эту операцию!");

}

}

if (com == "3") //передача прав

{

BinaryCode = Convert.ToString(M[numUser, numFile], 2);

BC = Convert.ToInt32(BinaryCode);

if (BC % 10 == 1)

{

PeredachaPrav(BC);

}

else

{

Console.WriteLine("У вас нет прав на эту операцию!");

}

}

return;

}

}

static void PeredachaPrav(int cod)

{

string BinaryCode; //переменная для перевода в двоичный код

int BC;

string[] Text = new string[] { "0 - чтение,", "1 - запись,", "2 - передача прав,", "3 - полные права" };

string comand = "";

int usnum = -1;

if (cod / 100 == 0) Text[0] = "";

if (cod % 100 < 10) Text[1] = "";

if (cod % 10 == 0) Text[2] = "";

if (cod != 111) Text[3] = "";

Console.Write("Какое право передаётся? ({0}{1}{2}{3})\nВведи номер операции: ", Text[0], Text[1], Text[2], Text[3]);

comand = Console.ReadLine();

if (comand == "0" && Text[0] == "")

{

Console.WriteLine("Ошибка команды, повторите попытку!");

return;

}

if (comand == "1" && Text[1] == "")

{

Console.WriteLine("Ошибка команды, повторите попытку!");

return;

}

if (comand == "2" && Text[2] == "")

{

Console.WriteLine("Ошибка команды, повторите попытку!");

return;

}

if (comand == "3" && Text[3] == "")

{

Console.WriteLine("Ошибка команды, повторите попытку!");

return;

}

Console.Write("Какому пользователю передается право?\nВведи имя пользователя: ");

usnum = UsersFunc(Console.ReadLine());

if (usnum == -1)

{

Console.WriteLine("\nИмя пользователя введено неверно, повторите попытку");

}

else

{

BinaryCode = Convert.ToString(M[usnum, numFile], 2);

BC = Convert.ToInt32(BinaryCode);

switch (comand)

{

case "0": //передача чтения

if (BC < 100) BC += 100;

if (BC == 100) M[usnum, numFile] = 4;

if (BC == 101) M[usnum, numFile] = 5;

if (BC == 110) M[usnum, numFile] = 6;

if (BC == 111) M[usnum, numFile] = 7;

break;

case "1": //передача записи

if (BC % 100 < 10) BC += 10;

if (BC == 010) M[usnum, numFile] = 2;

if (BC == 011) M[usnum, numFile] = 3;

if (BC == 110) M[usnum, numFile] = 6;

if (BC == 111) M[usnum, numFile] = 7;

break;

case "2": //передача передачи прав

if (BC % 10 == 0) BC += 1;

if (BC == 001) M[usnum, numFile] = 1;

if (BC == 011) M[usnum, numFile] = 3;

if (BC == 101) M[usnum, numFile] = 5;

if (BC == 111) M[usnum, numFile] = 7;

break;

case "3": //передача полного доступа

M[numUser, numFile] = 0;

M[usnum, numFile] = 7;

break;

default:

Console.WriteLine("Ошибка команды, повторите попытку!");

break;

}

Console.WriteLine("Операция прошла успешно!");

}

}

static void MatricaFunk()

{

//Заполняем права админа

for (int i = 0; i < File.Length; i++)

{

M[0, i] = 7;

}

Random rnd = new Random();

//Заполняем права остальных

for (int i = 1; i < Users.Length; i++)

for (int j = 0; j < File.Length; j++)

{

M[i, j] = rnd.Next(0, 7);

}

}

static void MatricaPechat()

{

Console.WriteLine("{0} | {1} | {2} | {3} | {4} ", "Объект / Субъект", "Файл 1", "Файл 2", "Файл 3", "Файл 4");

for (int i = 0; i < Users.Length; i++)

{

Console.WriteLine(" {0} | {1} | {2} | {3} | {4} ", Users[i], Prava[M[i, 0]], Prava[M[i, 1]], Prava[M[i, 2]], Prava[M[i, 3]]);

}

}

}

}