1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт кибербезопасности и защиты информации

**ЛАБОРАТОРНАЯ РАБОТА № 12**

1. «**Система контроля доступа**»
2. по дисциплине «Основы информационной безопасности»
3. Выполнил
4. студент гр.4851003/00001 Маронова К.Д.

<*подпись*>

1. Преподаватель
2. асс. преподавателя Пахомов М.А.

<*подпись*>

1. Санкт-Петербург
2. 2023
3. **Цель работы**

Освоить средства контроля и управления доступом пользователей к ресурсам операционной системы, приобретение навыков распределения прав на примере файловой системы NTFS в среде Windows.

1. **Постановка задачи**

В данной лабораторной работе необходимо изучить управление доступом, изменять параметры и наблюдать за изменениями, а также разработать утилиты которые выполняют операции создания, чтения и изменения атрибутов безопасности для файла. С помощью утилиты подтвердить практике возможность доступа к файлу при разрешении и невозможности доступа – запрете.

1. **Теоретические исследования**

Подсистема контроля и управления доступом в операционной системе Windows отличается высокой степенью гибкости, которая достигается за счет разнообразия защищаемых субъектов и объектов доступа, а также дифференциации видов доступа.

Контроль доступа выполняется централизованно с помощью компонента операционной системы — монитора безопасности (Security Reference Monitor), работающего в привилегированном режиме. Унификация функций контроля доступа повышает эффективность защиты операционной системы.

В системе Windows реализована объектная модель контроля доступа, согласно которой все субъекты и объекты доступа зарегистрированы в системе и обращение к ним регулируется с помощью множества атрибутов безопасности, наиболее важными из которых являются права доступа.

Для подсистемы безопасности операционной системы Windows характерно наличие большого количества различных предопределенных (встроенных) субъектов доступа — учетных записей пользователей и групп. В системе имеются встроенные пользователи, например, Администратор (Administrator), Гость (Guest), System и группы Пользователи (Users), Администраторы (Administrators), Все (Everyone). Встроенные пользователи и группы наделены полномочиями, заданными по умолчанию, что облегчает администрирование системы. Администратор также может создавать новые группы и новых пользователей, устанавливая для них права и принадлежность группам, реализуя тем самым политику информационной безопасности.

Права доступа - множество операций, которые определены для субъектов доступа (например, пользователей и групп) по отношению к объектам доступа (например, файлам, каталогам, принтерам, процессам). Примерами прав доступа являются чтение файла, удаление каталога, печать документа, изменение прав доступа.

Права доступа, установленные группе, автоматически предоставляются всем ее участникам (членам группы), позволяя администратору рассматривать множество пользователей как учетную единицу и минимизировать свои действия по управлению правами.

При входе пользователя в систему для него создается так называемый маркер доступа (ассess token), включающий идентификатор пользователя и идентификаторы всех групп, в которые он включен. В маркере также хранится список пользовательских привилегий, учитываемых при выполнении системных действий.

Всем идентифицируемым объектам доступа, включая файлы, потоки, процессы, объекты ядра, объекты синхронизации процессов, события и пр., при создании присваивается дескриптор (описатель) безопасности. Дескриптор безопасности содержит список контроля доступа (Ассess Control List, АCL).

Список контроля доступа состоит из набора записей контроля доступа (Асcess Control Entry, АCE), каждая из которых указывает идентификатор субъекта доступа (пользователя или группы), тип записи и права доступа. АСЕ может быть либо разрешающей, либо запрещающей, определяя тип прав как разрешения или запреты в маске прав доступа.

Владелец объекта - обычно пользователь, который его создал обладает правом избирательного управления доступом к объекту, всегда может изменять АСL объекта, чтобы разрешить или запретить другим пользователям доступ к объекту.

При запросе процессом доступа к объекту управление передается монитору безопасности, который сравнивает идентификаторы пользователя и групп пользователей из маркера доступа идентификаторами, хранящимися в записях АСЕ из АСL объекта. Процесс в течение сеанса работы может осуществлять доступ ко многим объектам, а количество активных процессов и проверяемых АСЕ в каждый момент времени довольно большое, поэтому монитор безопасности проверяет возможность доступа процесса к объекту только при его открытии (получении идентификатора), а не при каждом обращении.

Управление доступом пользователей к файловым объектам в системе Windows возможно только в рамках логических разделов файловой системы NTFS. Доступ к каталогам и файлам контролируется с помощью прав, устанавливаемых для пользователей и групп в ACL соответствующего файлового объекта.

В рамках одного АСL могут быть заданы несколько АСЕ для одного пользователя или группы (например, разрешающая пользователю чтение, разрешающая группе запись и чтение и запрещающая пользователю удаление), которые могут быть противоречивы. Монитор безопасности путем свертки прав вычисляет множество действующих разрешений:

1. Из разрешающих ACE данного (группы) пользователя составляется множество разрешений.

2. Из запрещающих ACE данного пользователя (группы) составляется множество запретов.

3. Результирующее множество разрешений определяется путем вычитания множества запретов из множества разрешений, поскольку запреты обладают большим приоритетом, чем разрешения.

Если для пользователя (группы) не установлено прав, то доступ запрещен. Если АСL не определен, то все виды доступа разрешены.

Для изменения АCL надо либо иметь право доступа "Изменение разрешений" (Change Permissions), либо быть владельцем объекта.

Для управления правами доступа к файловым объектам существует графический интерфейс операционной системы. Выбрав в контекстном меню файла или каталога пункт "Свойства", необходимо перейти на вкладку "Безопасность". На вкладке "Безопасность" в свойствах объекта представлен упрощенный интерфейс управления правами. При нажатии на кнопку "Дополнительно" появляется окно дополнительных параметров безопасности объекта, в котором можно изменить владельца объекта, тонко настроить АСL и наследование прав, рассчитать для конкретного субъекта доступа список действующих разрешений.

1. **Описание решения**

Описание структуры ACL

Таблица 1 – описание структуры ACL

|  |  |  |
| --- | --- | --- |
| ACL | ACE | SID пользователя-разрешающая-полный доступ |
| ACE | SID пользователя-разрешающая-обзор папок/выполнение файлов |
| ACE | SID пользователя-разрешающая-содержимое папки/чтение данных |
| ACE | SID пользователя-разрешающая-чтение атрибутов |
| ACE | SID пользователя-разрешающая-чтение доп. атрибутов |
| ACE | SID пользователя-разрешающая-создание папок/дозапись данных |
| ACE | SID пользователя-разрешающая-запись доп. атрибутов |
| ACE | SID пользователя-разрешающая-удаление папок и файлов |
| ACE | SID пользователя-разрешающая-удаление |
| ACE | SID пользователя-разрешающая-чтение разрешений |
| ACE | SID пользователя-разрешающая-смена разрешений |
| ACE | SID пользователя-разрешающая-смена владельца |
| ACE | SID пользователя-разрешающая-создание файлов/запись данных |

Создаем директорию test\_folder разделе NTFS, с которой и будем работать

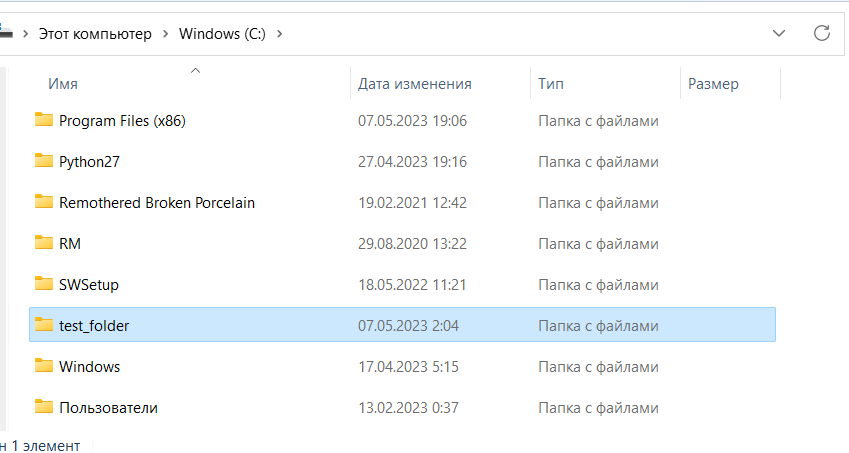


Рисунок 1 – каталог test\_folder

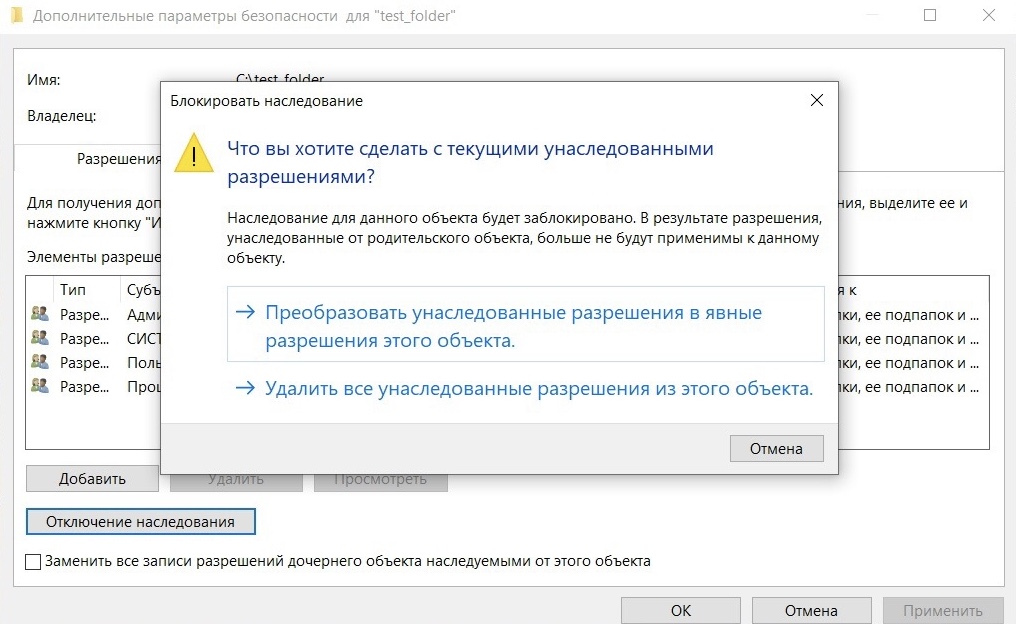
Далее снимаем флаг наследования

Рисунок 2 – снятие флага наследования

После чего убеждаемся, что наш пользователь (“karina”) является владельцем каталога

Далее выполняется проверка действующих разрешений, установленных для активного пользователя. Есть разрешения лишь на чтение и смену разрешений

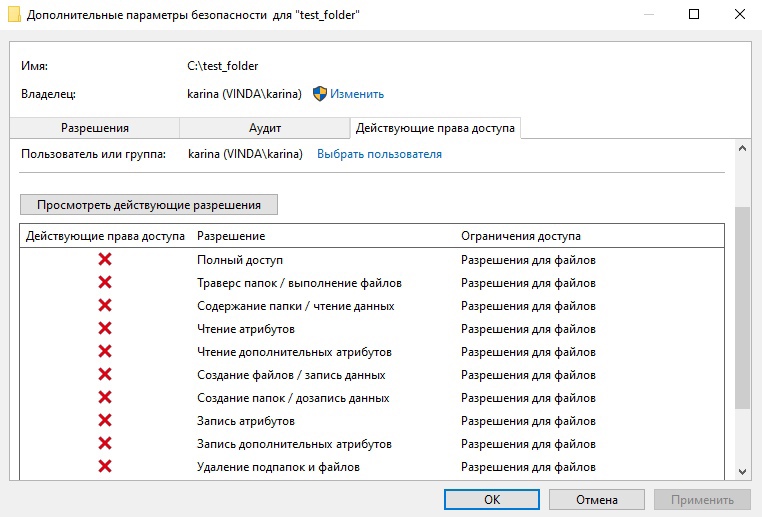
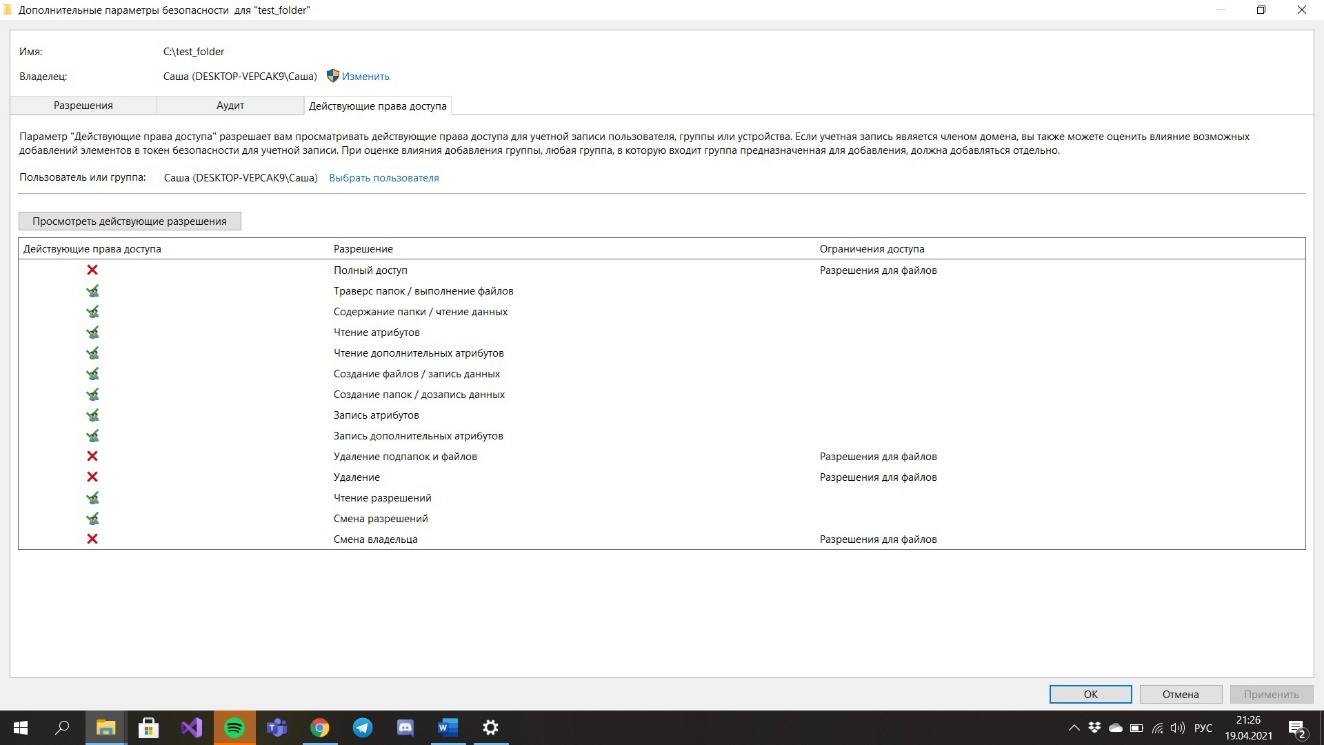


Рисунок 3 – действующие разрешения для пользователя «karina»

Затем продемонстрируем, что групповые права распространяются на всех членов группы. По умолчанию все пользователи находятся в группе “Пользователи”. То есть изменим права доступа для группы “Пользователи”, при этом не изменяя их для конкретного пользователя «karina» и зафиксируем права доступа пользователя «karina» после выполнения данных действий. 

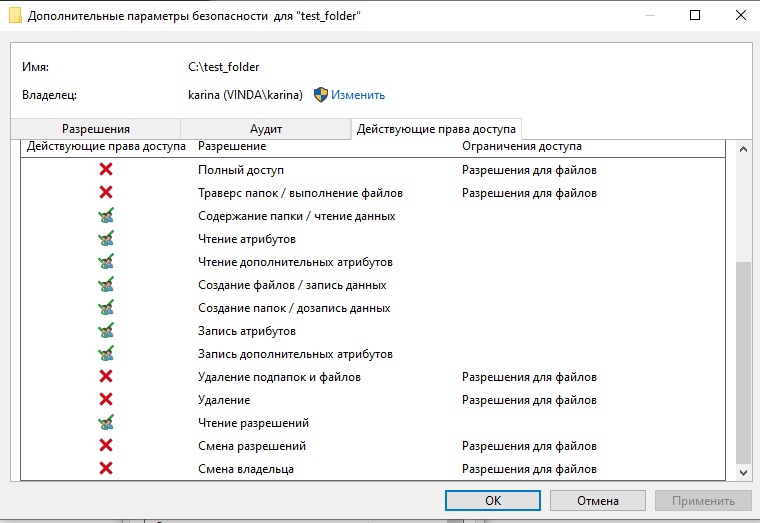


Рисунок 4– действующие разрешения для пользователя «karina»

Далее продемонстрируем приоритет запрещающих прав. Для этого запретим группе “Пользователи” права на запись, а пользователю “karina” разрешим полный доступ.

Мы видим, что у пользователя “karina” все равно нет полных прав, так как запрещающие права имеют больший приоритет.

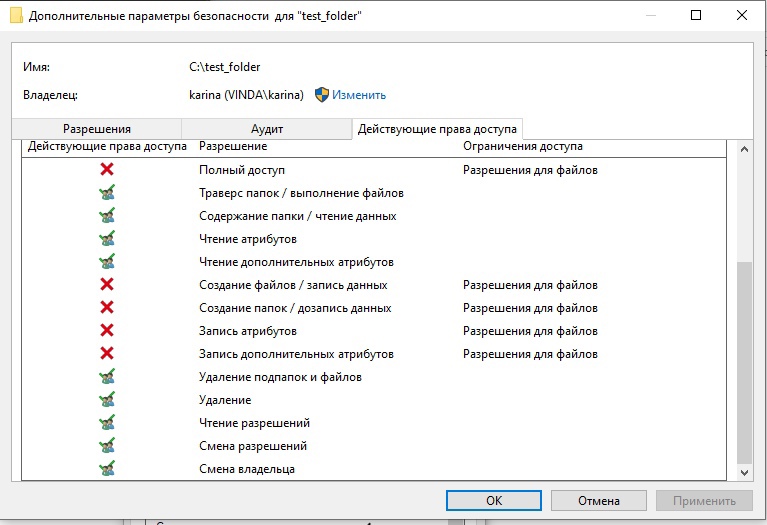


Рисунок 5– действующие разрешения для пользователя “karina”

После этого продемонстрируем суммирование разрешающих прав. Группе “Пользователи” предоставим права на чтение, но не на запись, а пользователю “karina” на запись, но не на чтение.

Мы можем увидеть, что после этих действий у пользователя “karina” есть права и на чтение, и на запись.

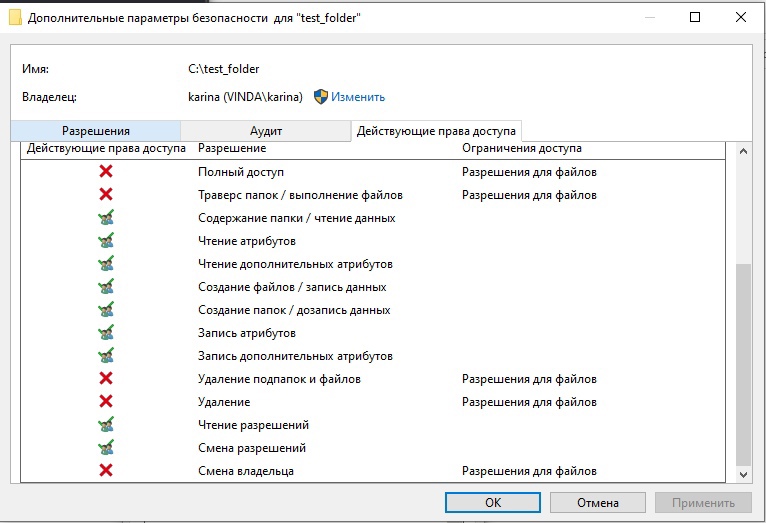


Рисунок 12– права для пользователя “Саша” после произведенных действий

После этого была разработана утилита, выполняющая действия создания, чтения, записи и изменения атрибутов файла. Если файл архивный, то не удается выполнить ни одну из операций чтения, записи и создания файла. Если же файл только для чтения, то не удается записать что-то в файл. Если мы установим наследование, то установленный на каталог атрибут, унаследует и сам файл. Ниже приведены результаты работы программы.

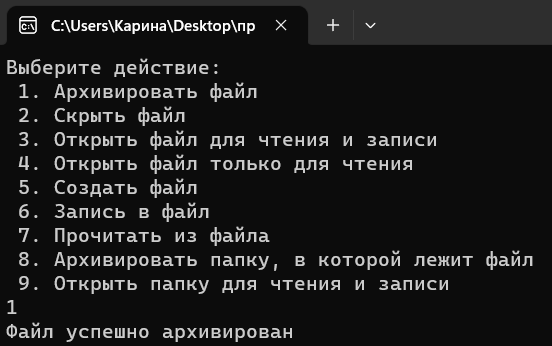


Рисунок 13– установка атрибута Архивный файл

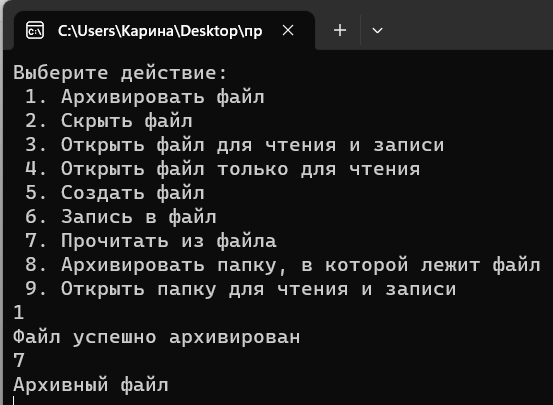


Рисунок 14– невозможно выполнить операцию чтения т.к файл архивный

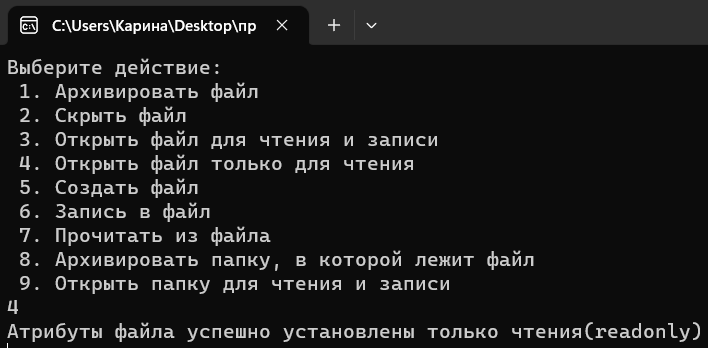


Рисунок 15– установка атрибута «только для чтения»

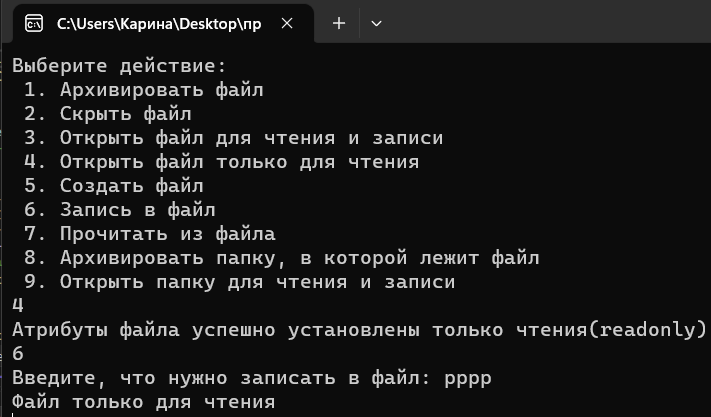


Рисунок 16– невозможно выполнить операцию записи в файл т.к файл только для чтения

1. **Контрольные вопросы**

1)*Что такое множество действующих разрешений?* Это множество действий, которые разрешены данному пользователю (группе) по отношению к файлам, каталогам. Вычисляется исходя из ACL.

*2)Почему проверка прав доступа к файлу осуществляется только при открытии файла, а не при обращениях к нему?* В компьютере достаточно много процессов, которые оперируют какими-то объектами. Каждому объекту присвоен свой ACL, в который входит большое количество записей ACE. Если бы проверка прав доступа к файлу осуществлялась при каждом обращении к файлу, то это бы скорее всего привело бы к снижению производительности и быстродействия.

*3)Можно ли запретить администратору системы доступ к какому-либо файлу? Может ли он обойти это ограничение?* Администратору системы можно запретить доступ к какому-нибудь файлу, но он может просто поменять владельца данного объекта и отменить все запреты, восстановив полный доступ к файлу.

*4)Как изменить владельца объекта в среде Windows?* Для того, чтобы изменить владельца файла нужно открыть окно дополнительных параметров безопасности файла и перейти на вкладку владелец.

*5)Имеет ли владелец какие-либо права к файлу, если существует ACE, запрещающая полный доступ к этому файлу на имя владельца?*

Владелец объекта - обычно пользователь, который его создал обладает правом избирательного управления доступом к объекту, всегда может изменять АСL объекта, чтобы разрешить или запретить другим пользователям доступ к объекту.

1. **Выводы**

В данной работе были изучены права доступа – удобный инструмент в руках администратора системы для обеспечения работы большого числа людей на одном компьютере. Работа над правами доступа при настройке систем, для которых важна информационная безопасность, должна производится с должным вниманием. Так же, были разработаны четыре утилиты для работы с атрибутами безопасности файла.

ПРИЛОЖЕНИЕ А

Листинг программы «Смена атрибутов файла, чтение, создание и запись в файл»

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <Windows.h>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

printf("Выберите действие:\n 1. Архивировать файл\n 2. Скрыть файл\n 3. Открыть файл для чтения и записи\n 4. Открыть файл только для чтения\n\

5. Создать файл\n 6. Запись в файл\n 7. Прочитать из файла\n 8. Архивировать папку, в которой лежит файл\n 9. Открыть папку для чтения и записи\n");

wchar\_t File\_name[] = TEXT("C:/test\_folder/new.txt"); //wchar и TEXT используется для unicode чтобы работать с библиотекой винды

//TEXT определяет строковой литерал как строку unicode

int p;

scanf("%d", &p);

while (p != 12) {

if (p == 1) {

if (SetFileAttributes(File\_name, FILE\_ATTRIBUTE\_ARCHIVE)){ //функция для изменения атрибутов. Возвращает true и false. На вход строка и сименем файла

//и атрибут, который ему передаём

printf("Файл успешно архивирован\n");

}

else{

printf("ОШИБКА архивации файла\n");

}

}

if (p == 2) {

int result;

result= SetFileAttributes(File\_name, FILE\_ATTRIBUTE\_HIDDEN);

if (result) {

printf("Файл успешно скрыт!\n");

}

else {

printf("ОШИБКА скрытия файла\n");

}

}

if (p == 3) {

if (SetFileAttributes(File\_name, FILE\_ATTRIBUTE\_NORMAL)) {

printf("Файл открыт для записи и чтения(normal)\n");

}

else {

printf("ОШИБКА выполнения пункта 3\n");

}

}

if (p == 4) {

if (SetFileAttributes(File\_name, FILE\_ATTRIBUTE\_READONLY)) {

printf("Атрибуты файла успешно установлены только чтения(readonly)\n");

}

else {

printf("ОШИБКА выполнения пункта 4\n");

}

}

if (p == 5) {

DWORD attributes = GetFileAttributesA("C:\\test\_folder");

/\* DWORD - это стандартный беззнаковый 32-битный тип данных в Windows.h, который используется

для манипулирования двоичными данными, целыми числами и указателями в операционной системе Windows. \*/

if (attributes & FILE\_ATTRIBUTE\_ARCHIVE)

printf("Папка архивная\n");

else {

FILE\* file1;

file1 = fopen("C:\\test\_folder\\new.txt", "w");

printf("Файл создан\n");

fclose(file1);

}

}

if(p == 6) {

DWORD attributes = GetFileAttributesA("C:\\test\_folder\\new.txt");

char fileContent[128] = { 0 };

printf("Введите, что нужно записать в файл: ");

scanf("%s", fileContent);

if ((attributes) & FILE\_ATTRIBUTE\_READONLY)

printf("Файл только для чтения\n");

else if ((attributes) & FILE\_ATTRIBUTE\_ARCHIVE)

printf("Архивный файл\n");

else {

FILE\* file2 = fopen("C:/test\_folder/new.txt", "w+");

fputs(fileContent, file2);

fclose(file2);

printf("Данные были успешно записаны в файл!\n");

}

SetFileAttributes(File\_name, FILE\_ATTRIBUTE\_NORMAL);

}

if (p == 7) {

char fileContent[1208] = { 0 };

DWORD attributes = GetFileAttributesA("C:\\test\_folder\\new.txt");

if (attributes & FILE\_ATTRIBUTE\_ARCHIVE)

printf("Архивный файл\n");

else {

FILE\* file3 = fopen("C:/test\_folder/new.txt", "r");

int i = 0;

while (!feof(file3)) {

fileContent[i] = fgetc(file3);

i++;

}

fileContent[i-1] = '\0';

printf("Текст из файла: %s\n", fileContent);

}

}

if (p == 8) {

if (SetFileAttributes(L"C:/test\_folder", FILE\_ATTRIBUTE\_ARCHIVE)) { //функция для изменения атрибутов. Возвращает true и false. На вход строка и сименем файла

//и атрибут, который ему передаём

printf("Папка успешно архивирована\n");

}

else {

printf("ОШИБКА архивации папки\n");

}

}

if (p == 9) {

if (SetFileAttributes(L"C:/test\_folder", FILE\_ATTRIBUTE\_NORMAL)) { //функция для изменения атрибутов. Возвращает true и false. На вход строка и сименем файла

//и атрибут, который ему передаём

printf("Папка успешно открыта для чтения и записи\n");

}

else {

printf("ОШИБКА открытия папки в пункте 9\n");

}

}

scanf("%d", &p);

}

return 0;

}