1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт компьютерных наук и технологий
5. **Кафедра «Информационная безопасность компьютерных систем»**

**Лабораторная работа 10**

1. «Контроль и управление доступом в операционных системах»
2. по дисциплине «Практикум по информационной безопасности»

Выполнил:

студент гр. 13558/1 Никулкин В. А.

Проверил:

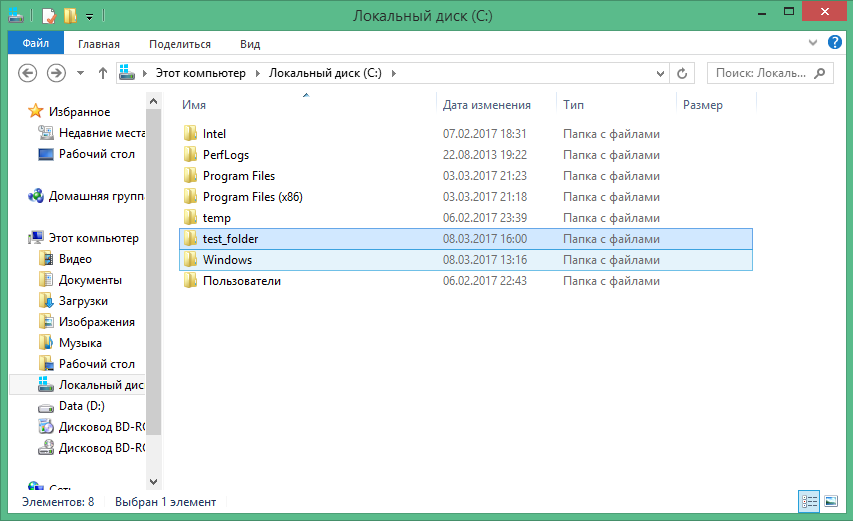
ассистент преподавателя Алексеев. И. В.

1. Санкт-Петербург
2. 2017
3. **Цель работы**

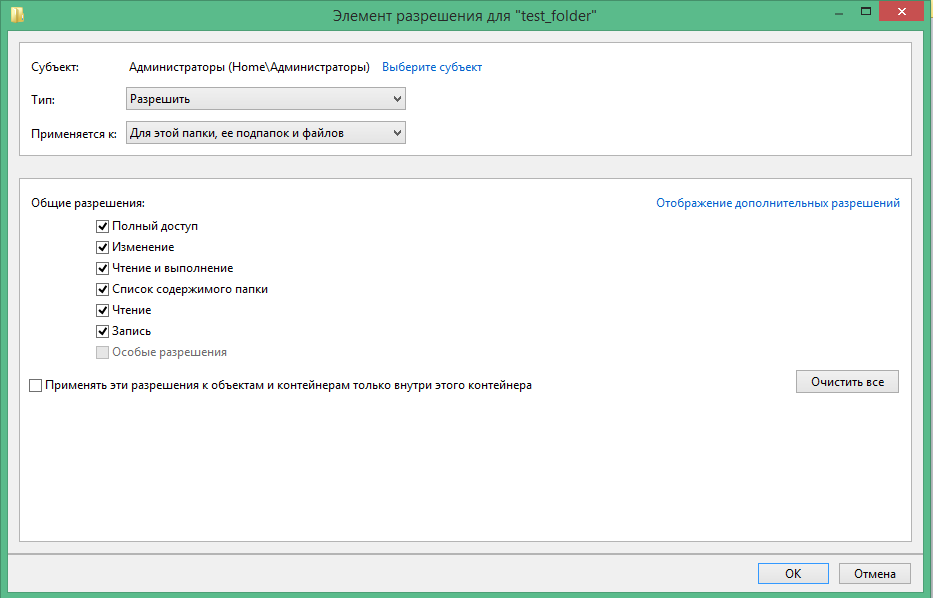
Освоение средств контроля и управления доступом пользователей к ресурсам операционной системы, приобретение навыков распределения прав на примере файловой системы NTFS в среде Windows.

**Ход работы**

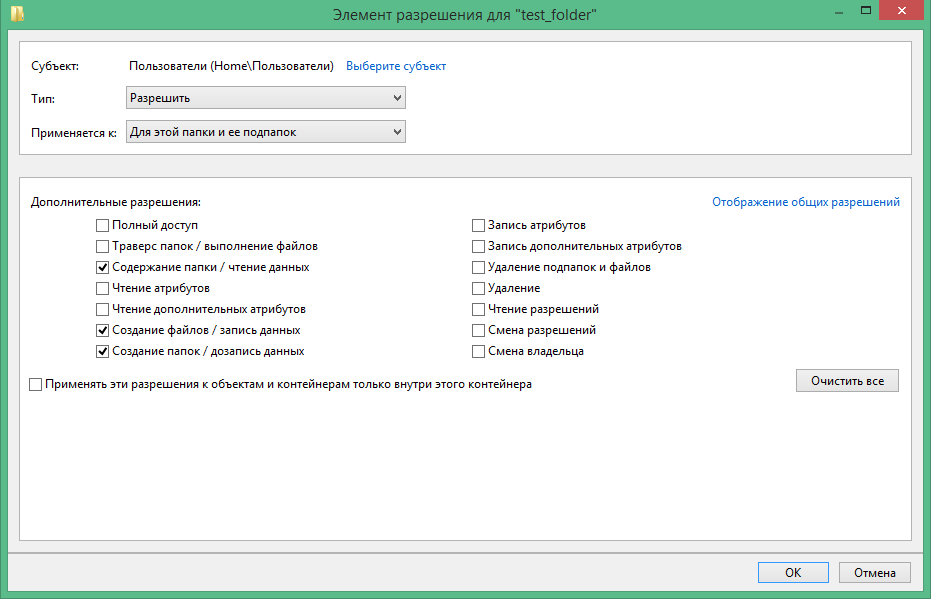
1. Мной был создан каталог test\_folder



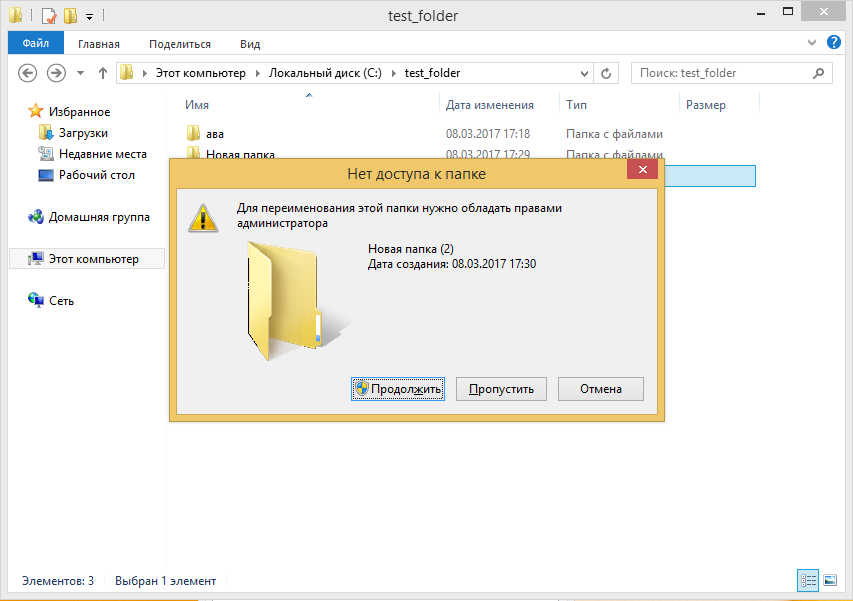
1. Пользователи, находящиеся в группе “Администраторы”, имеют полный доступ



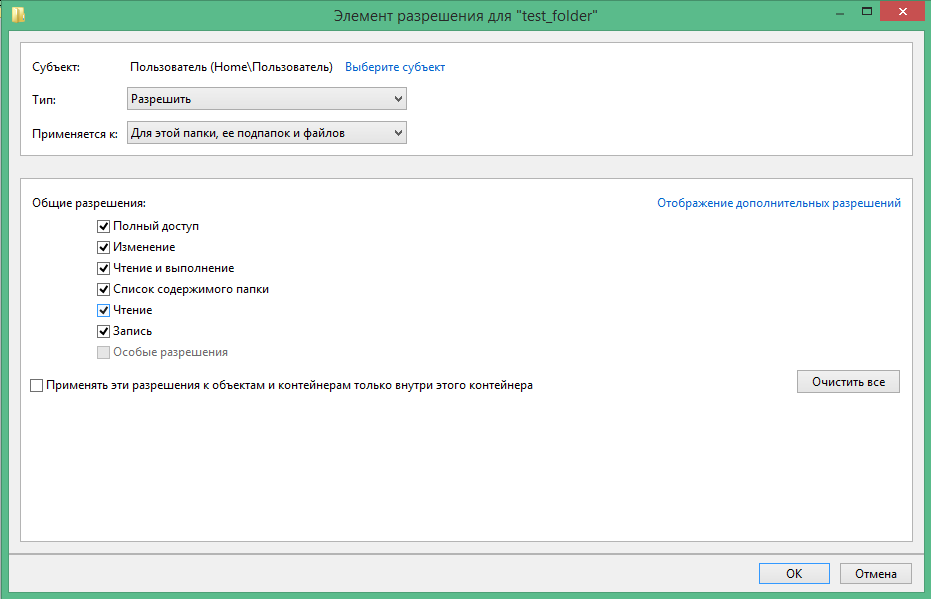
1. Была добавлена группа “Пользователи”, которая обладает правами чтения и записи данных

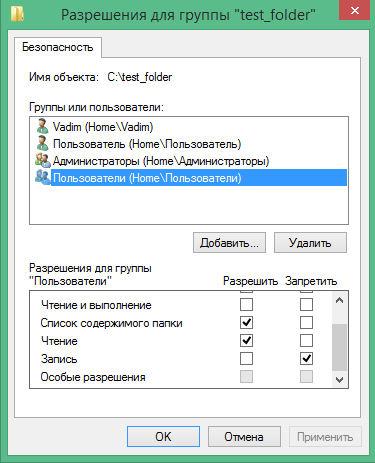


Теперь, например, пользователь, входящий в группу “Пользователи”, может читать и создавать папки, но удалять или переименовывать он их не может.

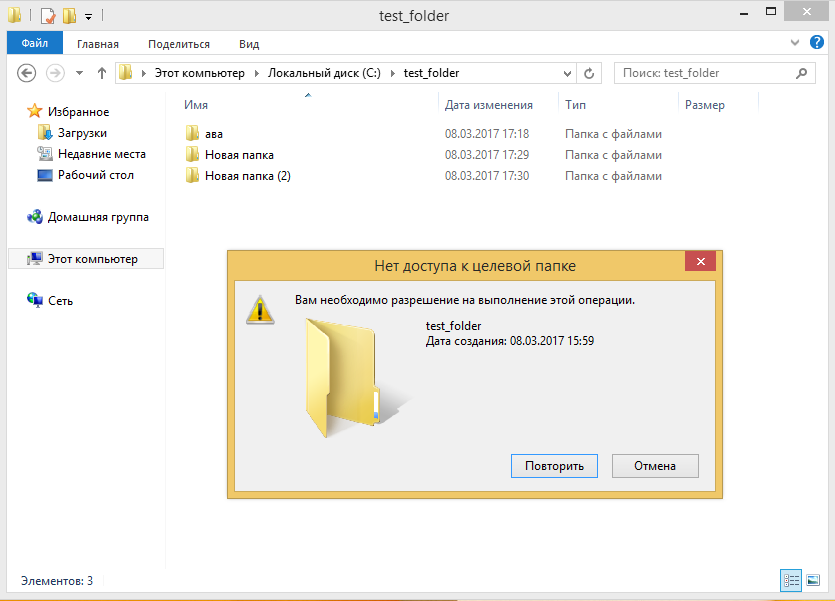


1. Пользователю “Пользователь” был дан полный доступ, а группе пользователи было запрещено записывать файлы

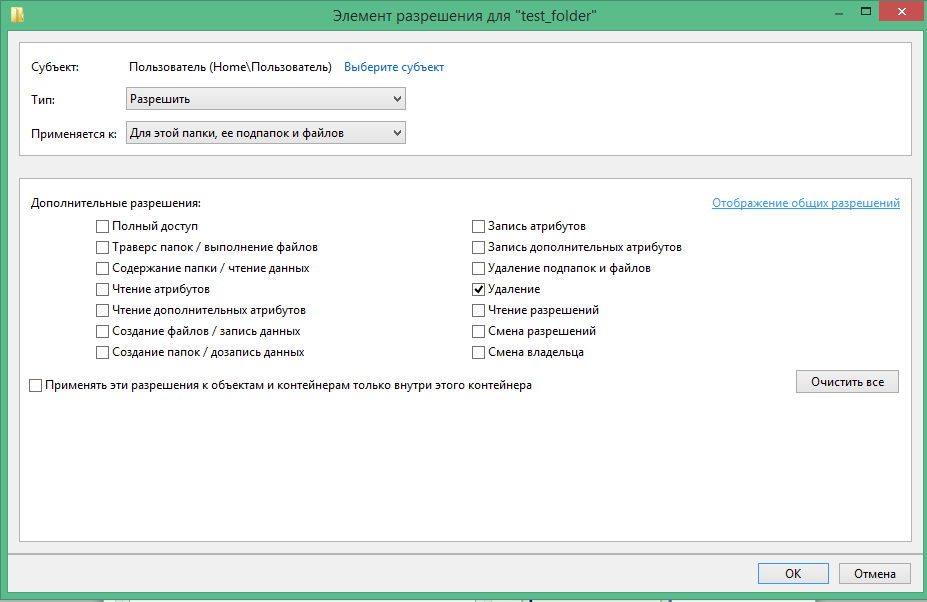




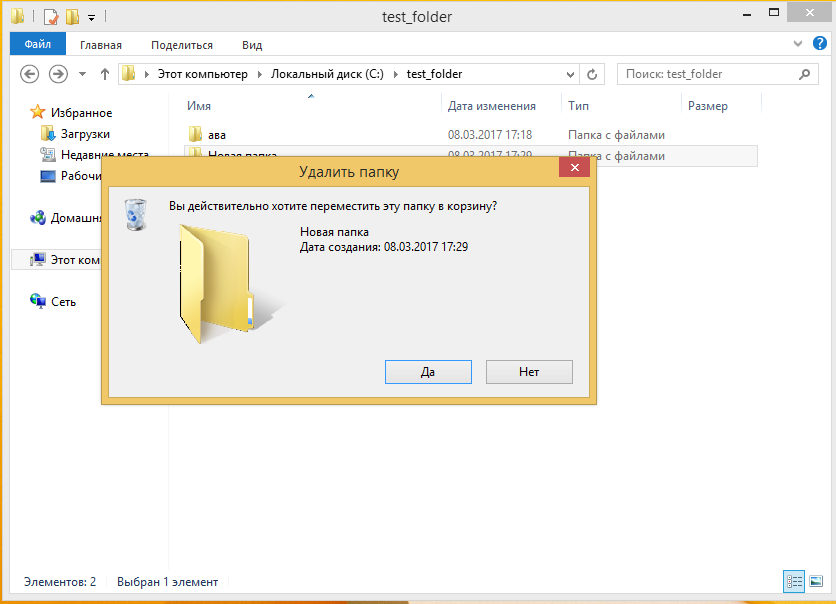
При попытке создать файла, был получен отказ



1. Группе “Пользователи” было дано право чтения и записи, а пользователю “Пользователь” было дано только право удаления.



Пользователь “Пользователь” теперь обладает как правами, которые были даны ему, так и правами всей группы “Пользователи”.



1. Листинги утилит

Утилита для создания файла

#include <stdio.h>

#include <stdlib.h>

int main()

{

FILE \*in = fopen("test.txt", "w");

fclose(in);

return 0;

}

Утилита для чтения файла

#include <stdio.h>

#include <stdlib.h>

int main()

{

FILE \*in = fopen("test.txt", "r");

char c;

while(1)

{

c = fgetc(in);

if(c == EOF)

break;

}

return 0;

}

Утилита для записи файла

#include <stdio.h>

#include <stdlib.h>

int main()

{

FILE \*in = fopen("test.txt", "r");

char \*s;

long length;

fseek(in, 0, SEEK\_END);

length = ftell(in);

fseek(in, 0, SEEK\_SET);

s = (char\*) malloc(sizeof(char) \* length);

fread (s, 1, length, in);

for(int i = 0; i < length; ++i)

printf("%c", s[i]);

fclose(in);

free(s);

return 0;

}

Утилита для изменения атрибутов безопасности

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "Russian");

int i = 0;

printf("Какой атрибут применить?\n");

scanf("%d", &i);

if (i == 1)

system("attrib +R test.txt");// только для чтения

if (i == 2)

system("attrib +H test.txt");// не выводится при показе тех или иных списков файлов

if (i == 3)

system("attrib +S test.txt");// атрибут «системный»

if(i == 4)

system("attrib +A test.txt");// программа, осуществляющая резервное копирование файла на жестком диске

return 0;

}

1. Контрольные вопросы
2. Владелец не будет иметь полный доступ к файлу, если существует ACE, который имеет более высокий приоритет, чем разрешающий.
3. Множество допустимых прав – это множество разрешенных действий для данной группы или пользователя. Рассчитать можно путём вычитания из множества разрешающих ACE множество запрещающих ACE.
4. Проверка права доступа осуществляется только при открытии файла потому, что если ACE будет проверяться многократно, то производительность будет сильно снижена.
5. Администратору можно запретить доступ к файлу, но администратор может обойти запрет, если станет владельцем файла.
6. Свойства->Безопасность->Дополнительно->Владелец, далее просто выбираем того пользователя, которого хотим сделать владельцем.

**Вывод**

Список контроля доступа помогает настраивать права для определённого объекта. Запрещающие права имеют приоритет над разрешающими, а групповые права распространяются на всех членов группы. Если у пользователя есть какие-то права, он обладает, а остальные члены группы, в которую он входит, не обладают, то его права и права группы суммируются, то есть он обладает и теми, и теми правами. Владелец файла обладает всеми правами на действия над объектом и всегда может изменять ACL объекта.