Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования   
«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчет защищен с отметкой

Преподаватель

(подпись)

« » 2020 г.

Отчет

по лабораторной работе №9

по дисциплине «Операционные системы»

Студент группы ПИ-82 Верещагина И.А.

Преподаватель Боровцов Е.Г.

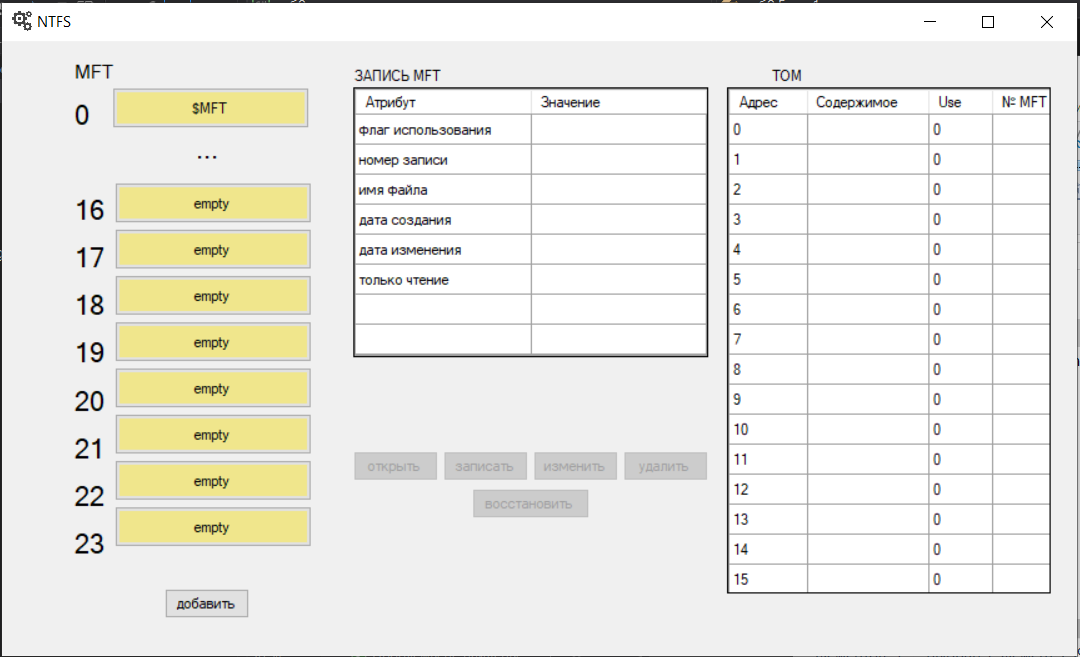
Барнаул 2020 г.

# **Цель:**

Написать программу, визуализирующую основные механизмы работы файловой системы NTFS.

# **Описание.**

Для работы с программой в папке лаб9 запустить исполняемый файл «лаб9.exe».

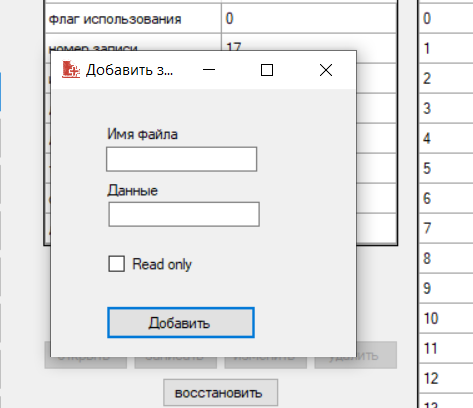


Главный экран содержит 3 таблицы. Первая – MFT таблица, вторая – таблица атрибутов конкретной записи MFT, третья – том.

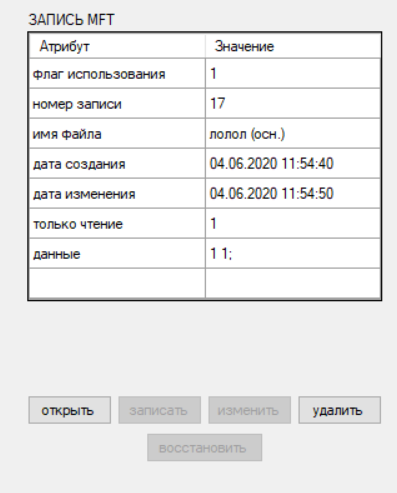
1. MFT - главная структура каждого тома в NTFS, по сути, это последовательность записей фиксированного размера, в моем случае, одна запись может хранить 8 атрибутов (ну правда в коде у файла еще много других флажков). Для упрощения задачи, ограничила количество записей до 8. Первые 16 записей MFT резервируются для файлов метаданных NTFS, поэтому их я пропустила, пользовательские файлы начинаются с 16-й записи.

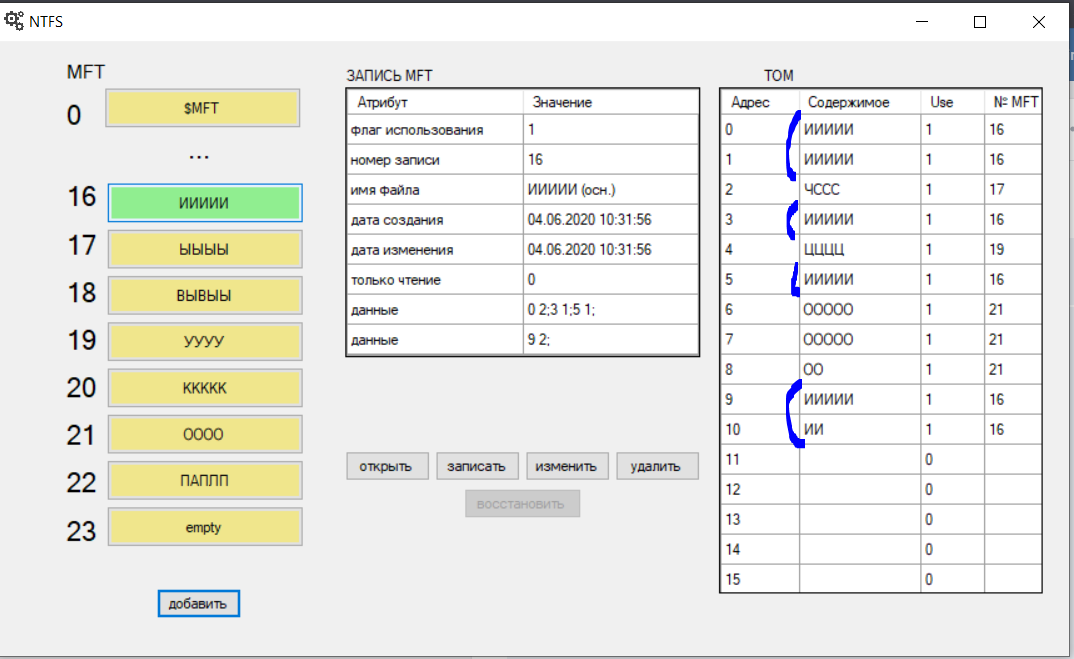
Чтобы увидеть содержимое конкретной записи, нужно на нее нажать, и вся информация выведется во второй таблице.

При добавлении записи нужно ввести имя файла (не более 5 символов), данные (ограничение ввода рассчитывается из количества свободных блоков и их вместимости), а также отметить, можно ли изменять файл, все остальное (даты создания, изменения, статус записи и т.д. ставится автоматически).



Если поставить флажок «ReadOnly» , то файл можно только открыть или удалить.





1. У каждой записи всегда есть атрибуты:

«флаг использования» - по ней определяется, свободна ли запись;

«номер записи» - номер в таблице MFT, чисто для себя, чтоб не потеряться;

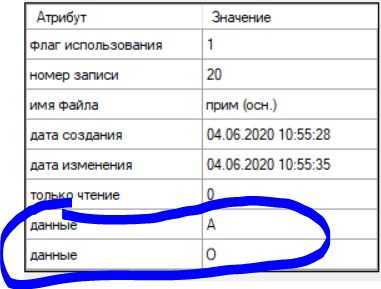
«имя файла» (в скобках указывается, основная или дополнительная запись);

«дата создания» и «дата изменения»;

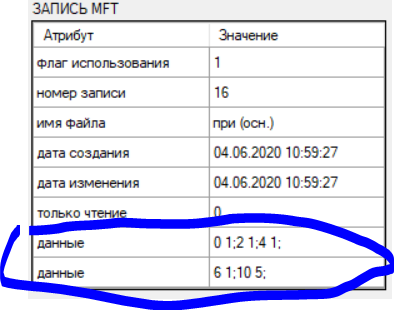
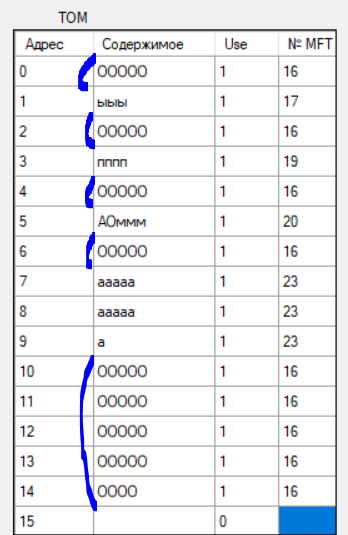
Флаг «только чтение»;

Две оставшиеся клетки свободны, они заполняются либо атрибутом «данные», либо «dop MFT», который указывает номера дополнительных записей.

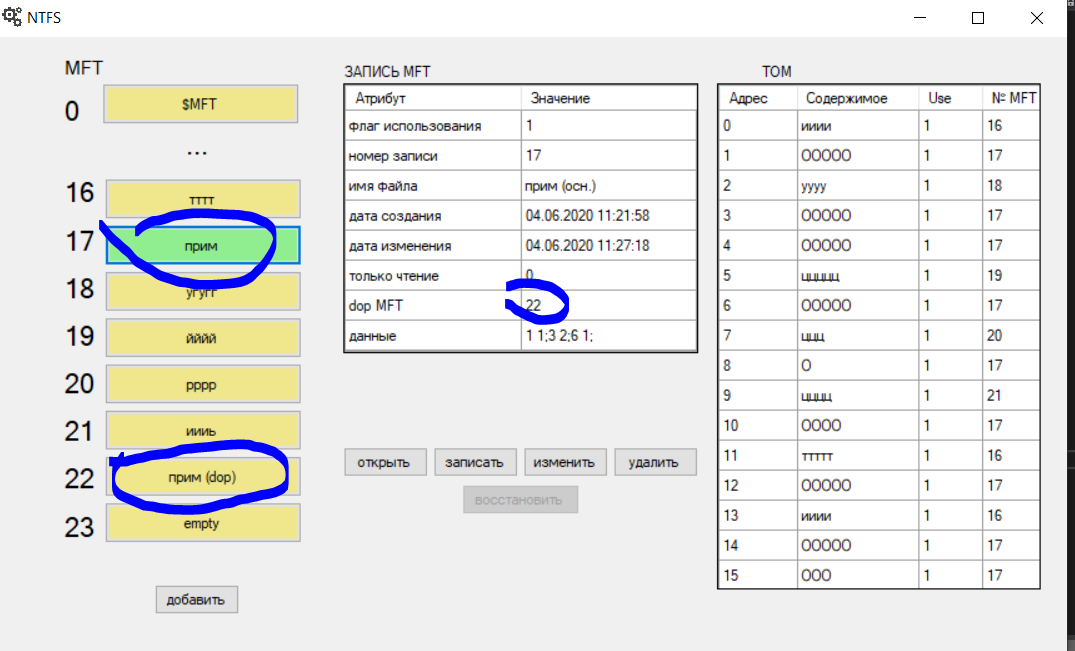
Если атрибут «данные» - резидентный, то он может сохранить один символ, т.е. если содержимое файла состоит из двух символов или меньше, то такой файл можно хранить в записи.

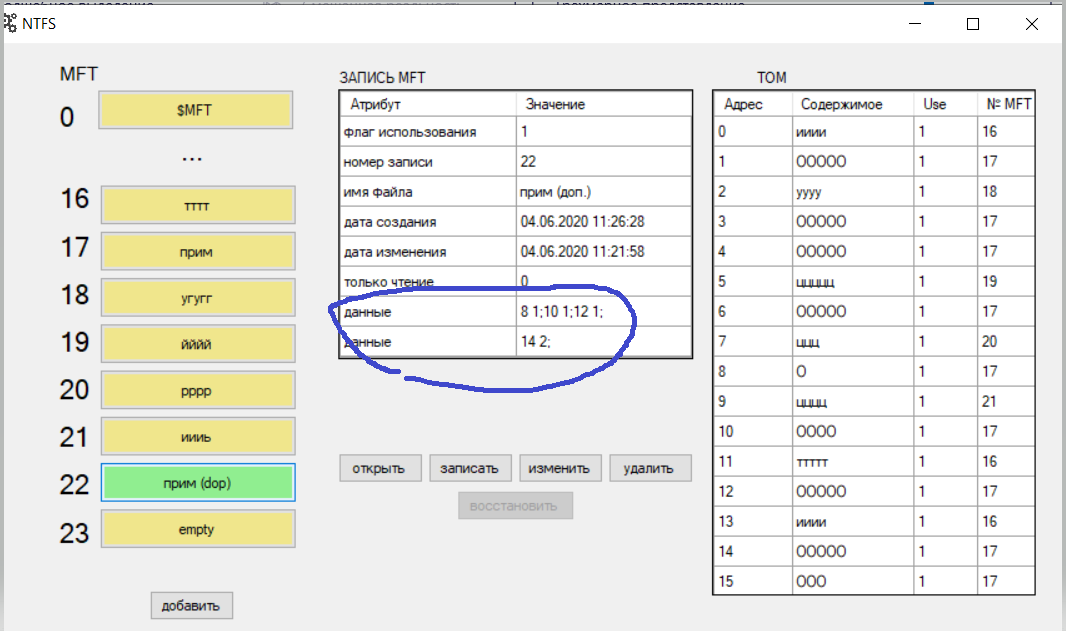


Если же этот атрибут нерезидентный, то он хранит максимум 3 серии последовательных блоков. Серии отделяются «;», каждая серия состоит из пары чисел, где первое – адрес начала последовательности в таблице тома, второе –количество блоков.

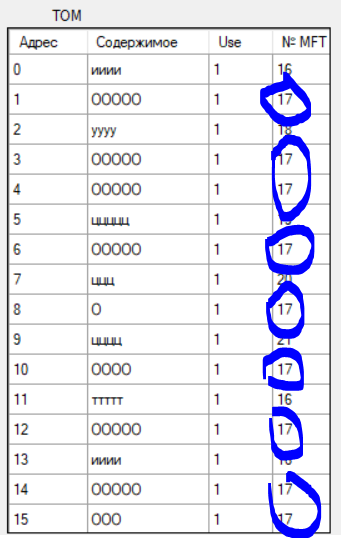
 

Если же файл очень большой, он может занимать несколько записей, обычно это происходит, если данные файла разбросаны по тому и образуется много серий:





Но в томе для удобства указан только номер главной записи:



В коде же было очень много переменных для одной записи

public int useRecord = 0; //используется ли запись

public int numberMFT;//адрес в МФТ

public string name = "";//имя файла

public int datarez;//резидентность данных

public int dopmft=0;//есть ли доп запись

public int bigindex = 1;//индекс, который показывает, основная запись или добавочная

public DateTime DateCreate;//дата создания

public DateTime DateChange;//дата последнего изменения

public int onlyread = 0; //ReadOnly

public ArrayList Ser; //серии хранятся

public ArrayList Rec ;//дополнительные мфт

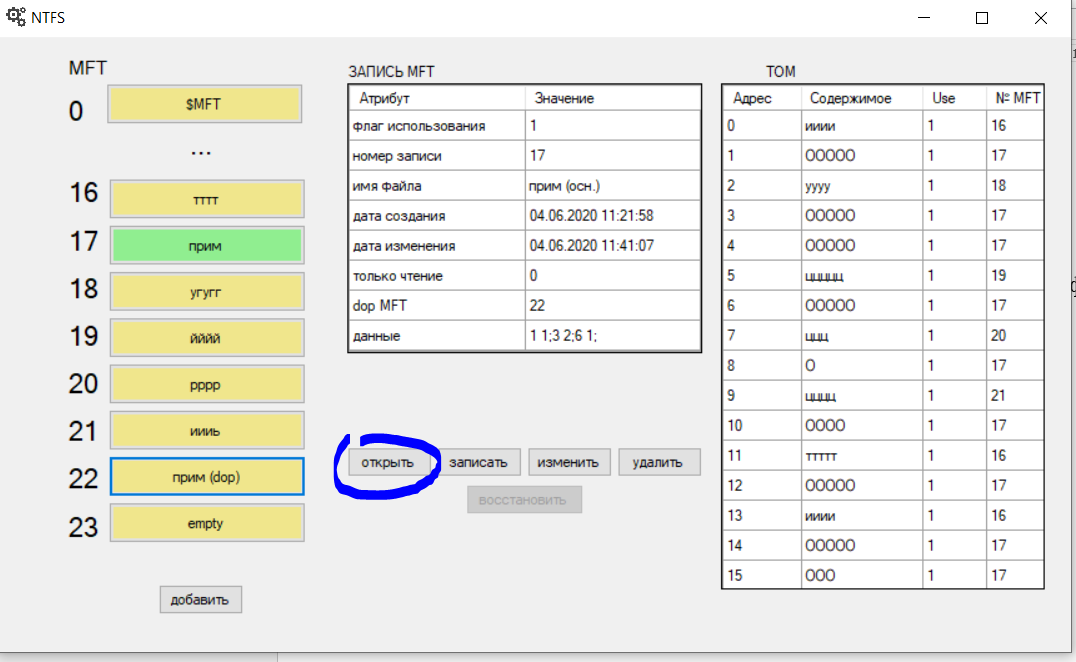
public ArrayList help; для моральной поддержки

1. Том состоит из 16 блоков, каждый блок может хранить 5 символов информации. Каждый блок содержит адрес, данные файла, флаг использования, а также номер основной записи в MFT.

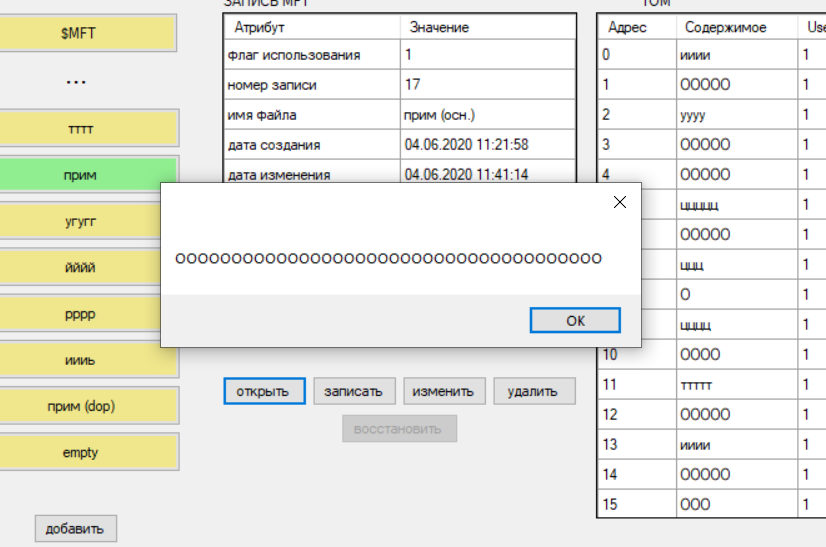
**Описание операций.**

1. **Чтение**

Сначала открывается запись нужного файла, нажимается «открыть».

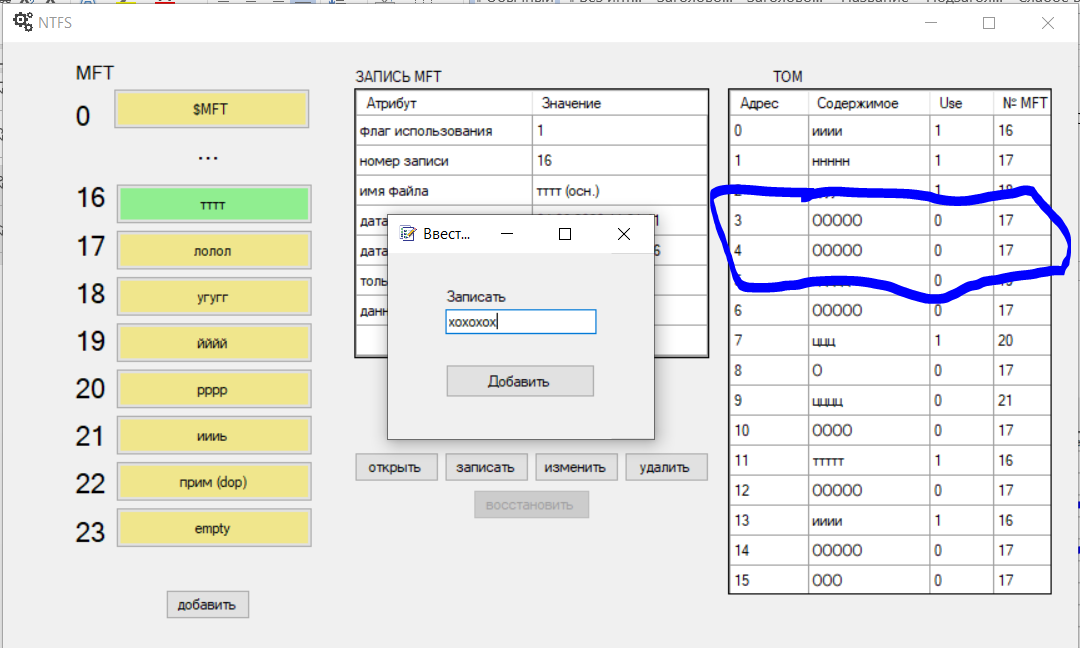


Программа проводит проверку резидентности данных, для этого я завела флаг «datarez», если данные резидентные, то хранятся в самой записи в ячейках атрибута данных и считываются оттуда. Если же данные нерезидентные, то программа обращается к сериям блоков и последовательно считывает информацию с блоков тома.



1. **Запись.**

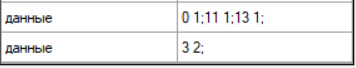
Сначала производится проверка количества заполненных блоков, если еще есть свободные, то открывается окно записи. После ввода данных, производится поиск пустых блоков тома, таковыми являются блоки 3 и 4 (флаг использования обнулен).



Введенное сообщение распределяется по блокам, устанавливается флаг use.

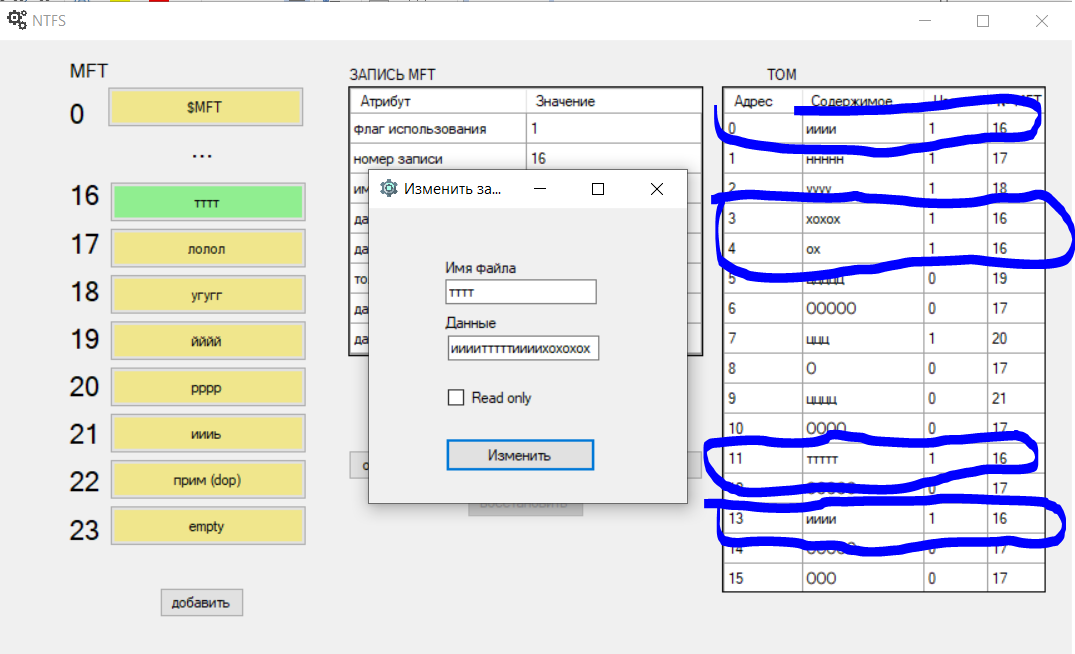


А в атрибуте данных записи файла появляется новая серия «3 2»

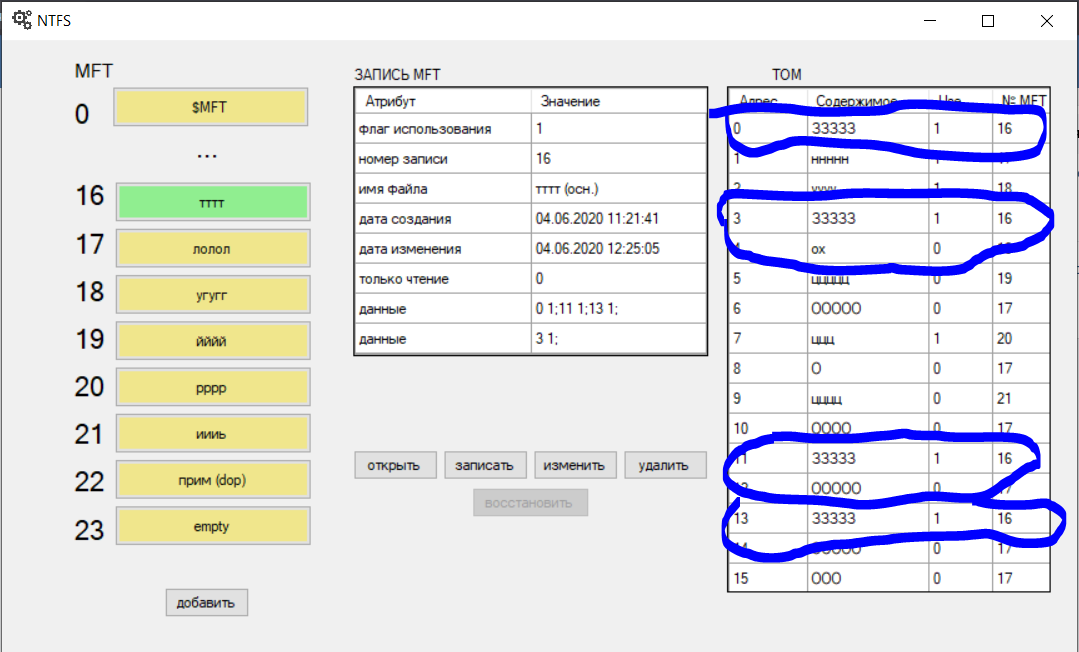


1. **Модификация**

При изменении файла в новом окне выводятся данные введенные пользователем (пригодилась операция считывания). Можно изменить имя, уровень доступа, содержимое. При модификации данных я ограничила ввод новых данных, т.е. объем данных либо останется прежним, либо уменьшится. Я буду издеваться над файлом под номером 16.



После модификации происходит перезапись измененных атрибутов, в том числе и данных, в следствие чего становится возможным сжатие данных. Например, в этом случае я просто заменила все символы на «З», после перезаписи освободился блок 4, в котором было 2 символа, а все остальные блоки заполнены по 5 символов.

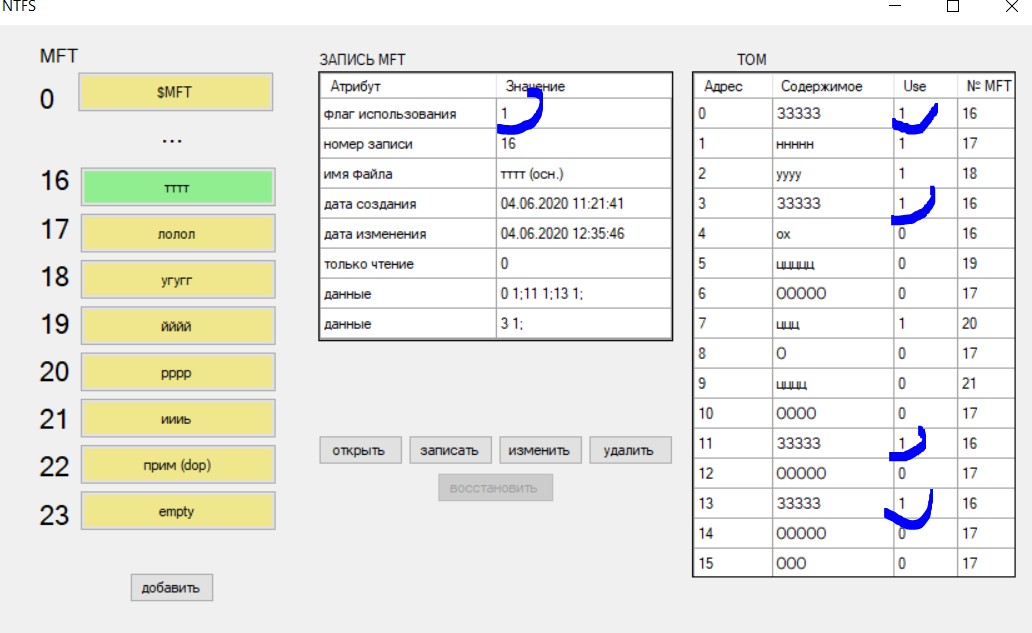


Правда косячно, что не сделала еще уплотнение, чтоб данные не были разбросаны, но времени мало.

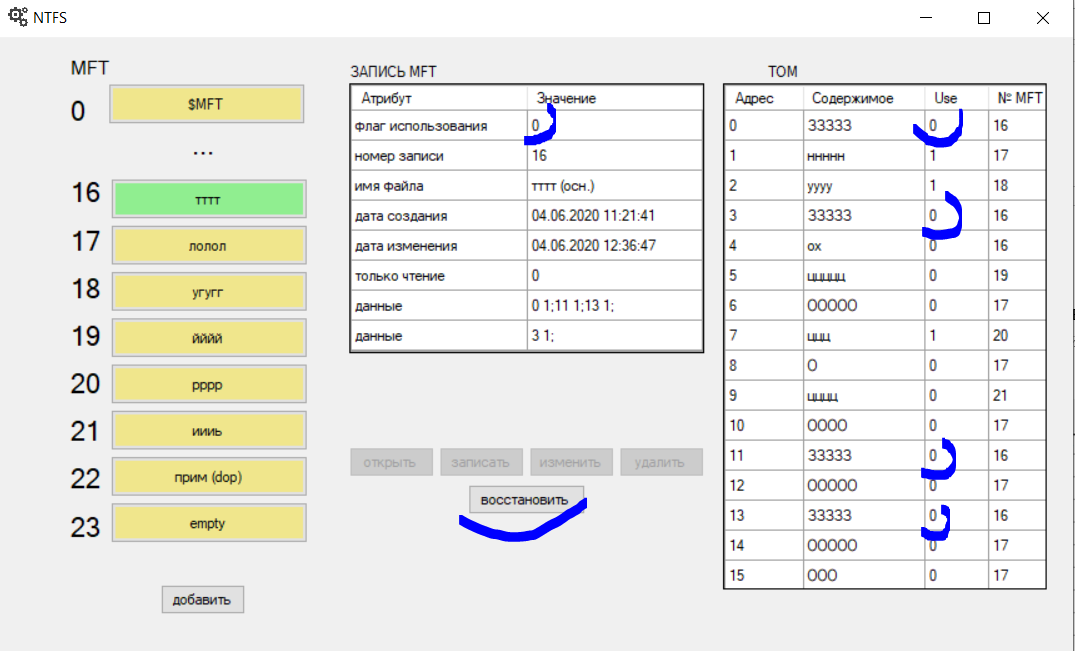
1. **Удаление**

Удаление реализуется максимально просто, нужно изменить значение флажков использования, а все остальные данные остаются на своих местах и удаляются только после записи новых данных, т.е. появляется возможность восстановления данных, если на их место не было записано других файлов.

До удаления файла №16:

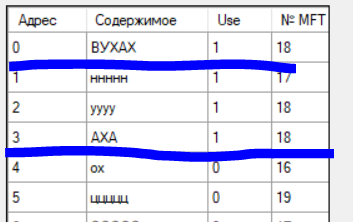


После:

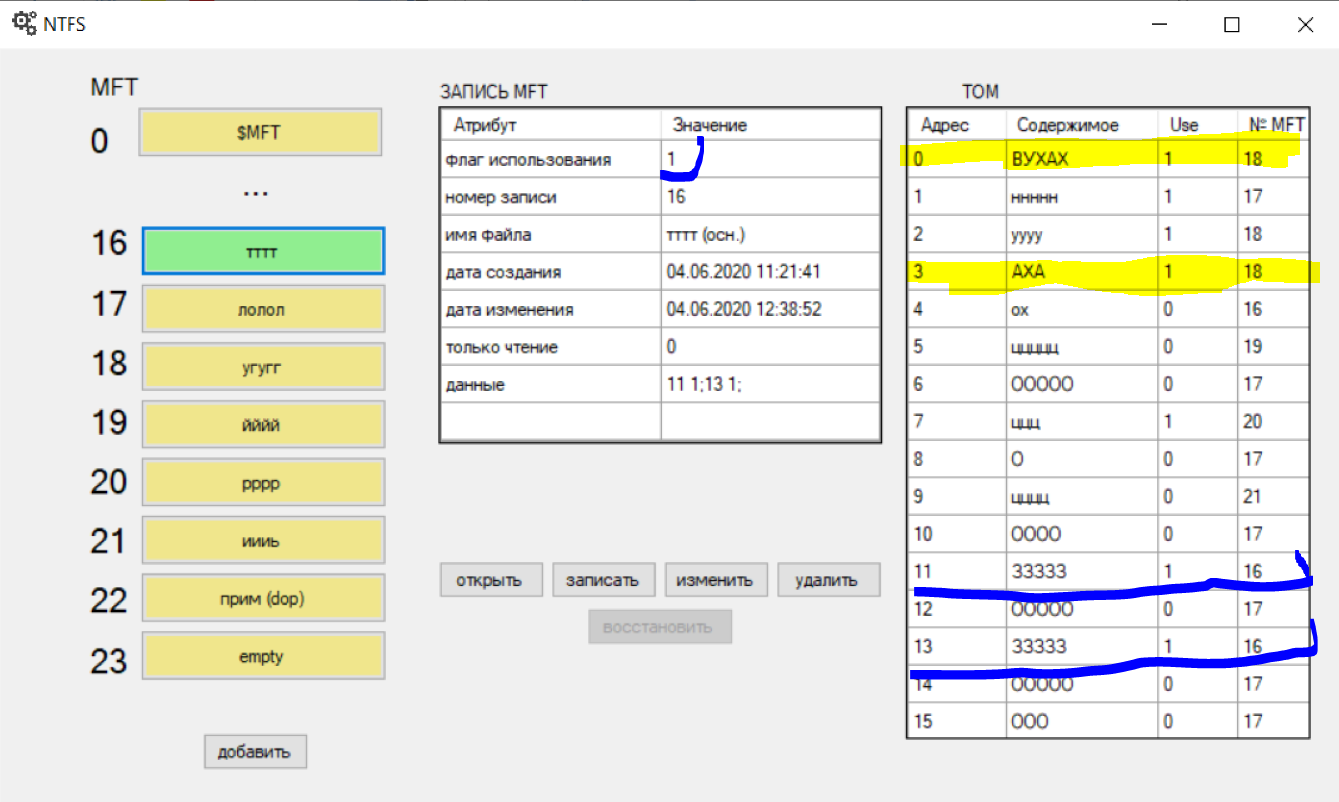


Флажки обнулились, кнопки управления скрылись, но появилась кнопка восстановления, по ее нажатию происходит проверка блоков, в которых хранились данные, если они все еще там, то флажок снова становится единицей.

Попробую заполнить ячейки 0 и 3 данными другого файла:



Теперь вернусь к файлу 16 и восстановлю его. Как видно, ячейки 11 и 13 восстановились, а 0 и 3 были нет.



Вывод: NTFS – очень интересная система с большим количеством фишек. Было интересно ее изучать. Основная сложность написания состояла в выборе того, что стоит упустить, а что все-таки описать. Что-то не удалось, есть косяки, но в целом было весело, но мало времени.

# Код:

Главная форма:

namespace лаб9

{

public partial class Form1 : Form

{

public Button[] mftrec = new Button[9];

public static ListRecord mftlist = new ListRecord();

public int rec = 0;//выбранная запись

public int countblock=0;//количество заполненных блоков

public int countrecord = 1;//количество заполненных записей

public int SizeBlock = 5;//размер блока, ну т.е. количество символов

public int lastadd = 0; //последний добавленный

public Form1()

{

InitializeComponent();

}

//видимость кнопок для файла

private void active()

{

read.Enabled = true;

delete.Enabled = true;

change.Enabled = true;

write.Enabled = true;

}

private void unactive()

{

read.Enabled = false;

delete.Enabled = false;

change.Enabled = false;

write.Enabled = false;

live.Enabled = false;

}

//раскраска записей

private void recordclick()

{

for (int i = 0; i < 9; i++)

{

if (rec != i)

mftrec[i].BackColor = Color.Khaki;

else mftrec[i].BackColor = Color.LightGreen;

}

recordshow();

}

//запуск формы

private void Form1\_Load(object sender, EventArgs e)

{

unactive();

for (int i = 0; i < 9; i++)

{

mftrec[i].BackColor = Color.Khaki;

if (i != 0)

mftrec[i].Text = "empty";

}

for (int i = 0; i < 16; i++)

{

//зарание заполняю таблицы строками

Tom.Rows.Add();

Tom.Rows[i].Cells[0].Value = i;

Tom.Rows[i].Cells[2].Value = 0;

// Task.Delay(1000).GetAwaiter().GetResult();

}

Tom.ClearSelection();

//заполняю таблицу записи

for (int i = 0; i < 8; i++)

{

//зарание заполняю таблицы строками

record.Rows.Add();

}

record.ClearSelection();

record.Rows[0].Cells[0].Value = "флаг использования";

record.Rows[1].Cells[0].Value = "номер записи";

record.Rows[2].Cells[0].Value = "имя файла";

record.Rows[3].Cells[0].Value = "дата создания";

record.Rows[4].Cells[0].Value = "дата изменения";

record.Rows[5].Cells[0].Value = "только чтение";

}

private add z = new add(); //для окна добавления

private write y = new write(); //для окна записи

//добавление нового файла

private void add\_Click(object sender, EventArgs e)

{

if (9 != countrecord)

{

if (countblock != 16)

{

lastadd = 0;

if (z.IsDisposed)

z = new add();

z.Show();

}

else

MessageBox.Show("Том заполнен =(", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else

MessageBox.Show("Максимальное количество записей достигнуто =(", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

//поиск пустого блока

public static int findEmptyBlock()

{

int find = -1;

for (int i = 0; i < 16; i++)

{

if (Tom.Rows[i].Cells[2].Value.ToString() == "0")

{ find = i;

return find;}

}

return find;

}

private void Form1\_Activated(object sender, EventArgs e)

{

for (int i = 0; i < 8; i++)

{

record.Rows[i].Cells[1].Value = "";

if (i > 5)

record.Rows[i].Cells[0].Value = "";

}

if (lastadd!=0)

{//очистка таблицы

rec = lastadd;

for (int i = 0; i < 9; i++)

{

if (rec != i)

mftrec[i].BackColor = Color.Khaki;

else mftrec[i].BackColor = Color.LightGreen;

}

recordshow();

record node = mftlist.findMFTnum(rec);

mftrec[node.numberMFT].Text = node.name;

if(node.dopmft==1)

mftrec[node.Rec].Text = node.name+" (dop)";

node.DateChange = DateTime.Now;

}

}

//вывод записи файла

public void recordshow()

{

for (int i = 0; i < 8; i++)

{

record.Rows[i].Cells[1].Value = "";

if (i > 5)

record.Rows[i].Cells[0].Value = "";

}

record node = mftlist.findMFTnum(rec);

unactive();

if (node.name != "")

{

if (node.useRecord == 0) live.Enabled = true;

else

if (node.onlyread == 1)

{

read.Enabled = true;

delete.Enabled = true;

}

else

active();

record.Rows[0].Cells[1].Value = node.useRecord;

if (node.bigindex==1)

record.Rows[2].Cells[1].Value = node.name+" (осн.)";

else

record.Rows[2].Cells[1].Value = node.name + " (доп.)";

record.Rows[3].Cells[1].Value = node.DateCreate;

record.Rows[4].Cells[1].Value = node.DateChange;

record.Rows[5].Cells[1].Value = node.onlyread;

if (rec == 0)

{

record.Rows[1].Cells[1].Value = 0;

record.Rows[6].Cells[0].Value = "данные";

record.Rows[6].Cells[1].Value = "где-то в начале";

unactive();

}

else

{

record.Rows[1].Cells[1].Value = node.numberMFT + 15;

int countser = node.Ser.Count;

int inl;

if (node.dopmft == 1)

{

record.Rows[6].Cells[0].Value = "dop MFT";

record.Rows[6].Cells[1].Value = node.Rec+15;

countser = 6;

inl = 7;

}

else inl = 6;

string ser = "";

for (int i = 0; i < countser; i+=2)

{

if (i!=0 && i % 3 == 0)

{

record.Rows[inl].Cells[0].Value = "данные";

record.Rows[inl++].Cells[1].Value = ser;

ser = "";

}

ser += node.Ser[i];

ser += " ";

ser += node.Ser[i + 1];

ser += ";";

}

if (inl < 8)

{

record.Rows[inl].Cells[0].Value = "данные";

record.Rows[inl].Cells[1].Value = ser;

}

if (node.datarez == 1)

{

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

{ record.Rows[6 + i].Cells[0].Value = "данные";

record.Rows[6 + i].Cells[1].Value = node.data[i];

}}}}}

//удаление записи

private void delete\_Click(object sender, EventArgs e)

{

record node = mftlist.findMFTnum(rec);

//если это доп запись, переходим с основную

if (node.bigindex == 0)

{

rec = node.Rec;

node = mftlist.findMFTnum(rec);

}

record.Rows[0].Cells[1].Value = 0;

//обнуляем флаг использования

node.useRecord = 0;

countrecord--;

if (node.dopmft == 1)//если есть дополнительные записи, обнуляем для них флаги использования

{

rec = node.Rec;

node = mftlist.findMFTnum(rec);

node.useRecord = 0;

countrecord--;

rec = node.Rec;

node = mftlist.findMFTnum(rec);

}

int count = node.help.Count;

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

{

Tom.Rows[Convert.ToInt32(node.help[i])].Cells[2].Value = 0;

countblock--;

}

unactive();

live.Enabled = true;

}

//восстановление

private void live\_Click(object sender, EventArgs e)

{

record node = mftlist.findMFTnum(rec);

string name1 = node.name;

//если это доп запись, переходим с основную

if (node.bigindex == 0)

{

rec = node.Rec;

node = mftlist.findMFTnum(rec);

if (node.name != name1)

{

MessageBox.Show("Восстанволение не возможно, так как основная запись файла была удалена =(", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

}

record.Rows[0].Cells[1].Value = 1;

//обнуляем флаг использования

node.useRecord = 1;

countrecord++;

if (node.dopmft == 1)//если есть дополнительные записи, обнуляем для них флаги использования

{

rec = node.Rec;

node = mftlist.findMFTnum(rec);

if (node.name != name1)

{

MessageBox.Show("Восстанволение допрлнительной страницы не возможно, так как запись файла была удалена =(", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

node.useRecord = 1;

countrecord++;

rec = node.Rec;

node = mftlist.findMFTnum(rec);

}

int count = node.help.Count;

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

{

if (Convert.ToInt32(Tom.Rows[Convert.ToInt32(node.help[i])].Cells[3].Value) == node.numberMFT + 15)

{

Tom.Rows[Convert.ToInt32(node.help[i])].Cells[2].Value = 1;

countblock++;

}

else { node.help.RemoveAt(i--); count--; }

}

active();

live.Enabled = false;

liveHelp();

recordshow();

}

//добавление

public partial class add : Form

{

public add()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

label3.Visible = false;

label5.Visible = false;

label6.Visible = false;

//пустое поле

if (textBoxName.Text == String.Empty )

label3.Visible = true;

else

if (Form1.mftlist.findName(textBoxName.Text) != null)//если имя уже существует

label6.Visible = true;

else

{ record node = Form1.mftlist.findEmptyElement(); //пустая запись

node.useRecord = 1; //используется ли запись

record node1 = Form1.mftlist.findEmptyElement(); //пустая запись

node.help.Clear();

node.Ser.Clear();

if (textBoxPrior.Text == String.Empty || textBoxPrior.Text.Length < 3) //если данных файла нет или они помещаются в запись

{

node.datarez = 1;

node.data = textBoxPrior.Text;

node.dopmft = 0;

node.bigindex = 1;

}

else

{

node.bigindex = 1;

node.datarez = 0;

string str = textBoxPrior.Text;

ArrayList res = new ArrayList();

int k = 0;

while (str.Length > 0)

{

if (str.Length > 4)

{

res.Add(str.Substring(0, 5));

str = str.Remove(0, 5);

}

else

{

res.Add(str);

str = "";

}

}

// str.Length;

int countser = res.Count;

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

{

k = Form1.findEmptyBlock();

if (k > -1)

{

Form1.countblock++;

node.help.Add(k);

Form1.Tom.Rows[k].Cells[2].Value = "1";

Form1.Tom.Rows[k].Cells[3].Value = node.numberMFT + 15;

Form1.Tom.Rows[k].Cells[1].Value = res[i];

}

else

{

MessageBox.Show("ых, места мало, но сохраню, сколько смогу", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);

break;

}

}

int index = Convert.ToInt32(node.help[0]);

node.Ser.Add(index);

index -= 1;

int sum = 0;

countser = node.help.Count;

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

{

if (index + 1 == Convert.ToInt32(node.help[i]))

{

sum++;

index++;

if (i + 1 == countser)

node.Ser.Add(sum);

}

else

{

node.Ser.Add(sum);

sum = 1;

index = Convert.ToInt32(node.help[i]);

node.Ser.Add(index);

if (i + 1 == countser)

node.Ser.Add(sum);

}

}

int countse = node.Ser.Count / 2;

if (countse <= 6)

node.dopmft = 0;

else

{

if (node1 == null)

{

node.Ser.RemoveRange(6, node.Ser.Count - 6);

MessageBox.Show("не удалось все сохранить");

}

else

{

node.dopmft = 1;

node1.help.Clear();

node1.Ser.Clear();

for (int i = 6; i < node.Ser.Count; i++)

{

node1.Ser.Add(node.Ser[i]);

// node.Ser.RemoveAt(6);

}

node1.useRecord = 1; //используется ли запись

node1.bigindex = 0;

node.Rec = node1.numberMFT;

node1.Rec = node.numberMFT;

} } }

Form1.lastadd = node.numberMFT;

node.name = textBoxName.Text;//имя файла

node.DateCreate = DateTime.Now;//дата создания

node.DateChange = node.DateCreate;

if(checkBox1.Checked==true)

node.onlyread = 1;

else node.onlyread = 0;

Form1.countrecord++;

if (node.dopmft == 1)

{

Form1.countrecord++;

node1.name = textBoxName.Text;//имя файла

node1.DateCreate = DateTime.Now;//дата создания

node1.DateChange = node.DateCreate;

if (checkBox1.Checked == true)

node1.onlyread = 1;

else node1.onlyread = 0; }

this.Close();

}

}

private void add\_Load(object sender, EventArgs e)

{

if (16 == Form1.countblock)

{

textBoxPrior.MaxLength = 2;

}

else

textBoxPrior.MaxLength = (16 - Form1.countblock) \* 5;

}

}

//модификация

public partial class change : Form

{

public change()

{

InitializeComponent();

}

private void change\_Load(object sender, EventArgs e)

{

record node = Form1.mftlist.findMFTnum(Form1.rec);

textBoxName.Text = node.name;

if (node.onlyread == 1)

checkBox1.Checked = true;

else checkBox1.Checked = false;

if (node.datarez == 1)

{

textBoxPrior1.Text = node.data;

textBoxPrior1.MaxLength = 2;

}

else {

int h = node.help.Count;

string str = "";

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

str += Form1.Tom.Rows[Convert.ToInt32(node.help[i])].Cells[1].Value.ToString();

textBoxPrior1.Text = str;

textBoxPrior1.MaxLength = str.Length;

}

}

private void button1\_Click(object sender, EventArgs e)

{

label3.Visible = false;

label5.Visible = false;

label6.Visible = false;

record node = Form1.mftlist.findMFTnum(Form1.rec);

//пустое поле

if (textBoxName.Text == String.Empty)

label3.Visible = true;

else

if (textBoxName.Text!= node.name && Form1.mftlist.findName(textBoxName.Text) != null)//если имя уже существует

label6.Visible = true;

else

{

if (textBoxPrior1.Text == String.Empty || textBoxPrior1.Text.Length < 3) //если данных файла нет или они помещаются в запись

{

node.datarez = 1;

node.data = textBoxPrior1.Text;

node.dopmft = 0;

}

else

{

node.datarez = 0;

string str = textBoxPrior1.Text;

ArrayList res = new ArrayList();

int k = 0;

while (str.Length > 0)

{

if (str.Length > 4)

{

res.Add(str.Substring(0, 5));

str = str.Remove(0, 5);

}

else

{

res.Add(str);

str = ""; } }

int countser = res.Count;

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

Form1.Tom.Rows[Convert.ToInt32(node.help[i])].Cells[1].Value = res[i];

if (countser < node.help.Count)

{

for (int i = countser; i < node.help.Count; i++)

Form1.Tom.Rows[Convert.ToInt32(node.help[i])].Cells[2].Value = 0;

node.help.RemoveRange(countser, node.help.Count - countser);

Form1.countblock = Form1.countblock - node.help.Count - countser;

}

} }

node.name = textBoxName.Text;//имя файла

node.DateChange = DateTime.Now;

if (checkBox1.Checked == true)

node.onlyread = 1;

else node.onlyread = 0;

Form1.liveHelp();

this.Close();

}