МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Филиал федерального государственного бюджетного образовательного учреждения высшего образования

«Национальный исследовательский университет «МЭИ» в г. Смоленске

Кафедра вычислительной техники

Отчет

По лабораторной работе №4

Тема: «Файлы, проецируемые в память»

По курсу: «Операционные системы»

Студент: Павлова Ю.В.

Группа: ИВТ-2-18

Преподаватель: Гаврилов А. И.

Вариант: №14

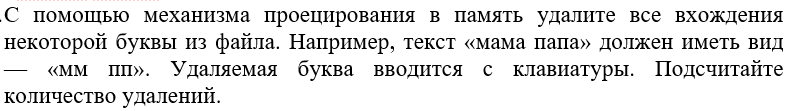
Смоленск,2020 г.

1. **Цель работы**

Цель работы заключается в освоении методов работы с файлами, проецируемыми в память.

Механизм проецирования в память позволяет резервировать регион адресного пространства и передавать ему физическую память. Причем физическая память в этом случае берется из файла, уже находящегося на диске. Данный метод позволяет обойтись без операций файлового ввода-вывода и предварительной буферизации содержимого файла. Проецируемые файлы применяются для загрузки EXE- и DLL-файлов, а также для разделения данных между несколькими процессами, выполняемыми на одной машине.

1. **Задание на лабораторную работу**



1. **Анализ задания на лабораторную работу**

Для выполнения задания для самостоятельной работы требуется создать кнопку Button1, элемент Edit куда будем записывать необходимую букву и Edit1, в котором будет считаться сколько раз была встречена заданная буква в файле.

1. **Модульная структура программы**
2. procedure TForm1.FormCreate(Sender: TObject);
3. procedure TForm1.Button1Click(Sender: TObject);
4. **Проверка работоспособности программы**

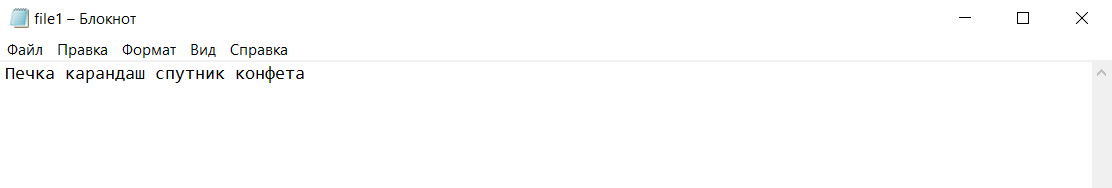


Рисунок 1 – Файл до выполнение программы

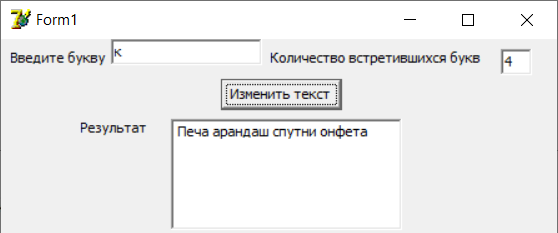


Рисунок 2 – Выполнение программы



Рисунок 3 – Файл после выполнения программы

**Вывод:** Программа работает корректно и выдает ожидаемый результат.

1. **Контрольные вопросы**
2. Для каких целей используется механизм проецирования в память?

Проецируемые файлы применяются для загрузки EXE- и DLL-файлов, а также для разделения данных между несколькими процессами, выполняемыми на одной машине.

1. Чем механизм проецирования в память отличается от работы с виртуальной памятью?

Механизм проецирования в память позволяет резервировать регион адресного пространства и передавать ему физическую память. Причем физическая память в этом случае берется из файла, уже находящегося на диске.

1. Опишите последовательность действий при проецировании в память.

Механизм проецирования в память состоит из следующих этапов:

* Создание или открытие объекта ядра «файла», который будет использоваться как проецируемый в память
* Создание объекта ядра «файл, проецируемый в память».
* Проецирование файловых данных на адресное пространство процесса.
* Работа с данными, содержащимися в фале, спроецированном в память.
* Отмена проецирования на адресное пространство процесса объекта ядра «файл, проецируемый в память».
* Закрытие объект ядра «файл, проецируемый в память».
* Закрытие объект ядра «файл».

1. Что случится, если при завершении проецирования в память не закрывать дескрипторы объектов с помощью функции CloseHandle?

Объекты ядра следует закрывать после того, как работа с ними закончена, так как операционная система хранит информацию обо всех объектах ядра, используя для этих целей свои ресурсы. Следовательно, во избежание утечки ресурсов, неиспользуемые объекты ядра должны быть закрыты.

1. Можно при проецировании в память увеличить размер файла?

В функции CreateFileMapping можно установить параметры dwMaximumSizeHigh и dwMaximumSizeLow, таким образом можно увеличить размера файла.

1. Каким образом выполняется проецирование в память файлов, размер которых больше 4 Гбайт?

В функции CreateFileMapping установить параметр dwMaximumSizeHigh больше 1.

В функции MapViewOfFile параметры dwFileOffsetHigh,dwFileOffsetLow− определяют смещение в файле. Так как Windows поддерживает файлы и их проекции размером до 16 экзабайтов, то данные параметры используются для определения 64-битового адреса: dwFileOffsetHigh определяет старшие 32-бита, а dwFileOffsetLow –младшие 32 бита. Задаваемое смещение должно быть кратно гранулярности выделения ресурсов.

1. **Заключение**

В результате выполнения лабораторной работы были освоены методы работы с файлами, проецируемыми в память.

1. **Приложение А. Текст программы**

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls;

type

TForm1 = class(TForm)

Label2: TLabel;

LetterField: TEdit;

MemoTextWin: TMemo;

Label3: TLabel;

Button1: TButton;

Edit1: TEdit;

Label1: TLabel;

procedure FormCreate(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

begin

MemoTextWin.Clear();

end;

procedure TForm1.Button1Click(Sender: TObject);

var

hFile, hFileMap: THandle;

pFile, Str: PChar;

Size, i, j, z: Integer;

Letter: Char;

TextStr: string;

counter: integer;

begin

hFile := CreateFile('file1.txt', GENERIC\_READ or GENERIC\_WRITE,

FILE\_SHARE\_READ or FILE\_SHARE\_WRITE, nil,

OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, 0);

if hFile = INVALID\_HANDLE\_VALUE then

begin

ShowMessageFmt('Ошибка %d при открытии файла ', [GetLastError]);

exit;

end;

Size := GetFileSize(hFile, nil);

hFileMap := CreateFileMapping(hFile, nil, PAGE\_READWRITE, 0, Size, nil);

if hFileMap = INVALID\_HANDLE\_VALUE then

begin

ShowMessageFmt('Ошибка %d при создании объекта "файл, проецируемый в память"', [GetlastError]);

CloseHandle(hFile);

exit;

end;

pFile := MapViewOfFile(hFileMap, FILE\_MAP\_WRITE, 0, 0, 0);

if pFile = nil then

begin

ShowMessageFmt('Ошибка %d при проецировании в память', [GetlastError]);

CloseHandle(hFile);

CloseHandle(hFileMap);

exit;

end;

if LetterField.GetTextLen <> 1 then

ShowMessage('Введите одну символ!')

else

begin

counter:=0;

TextStr := LetterField.Text;

Letter := TextStr[1];

i := 0;

for z:=0 to Size do

begin

if pFile[z] = Letter then

begin

Inc(counter);

end;

end;

Edit1.Text := IntToStr(counter);

while (i <= Size) do

begin

while pFile[i] = Letter do

begin

j := i;

while j <= Size - 1 do

begin

pFile[j] := pFile[j + 1];

Inc(j);

end;

end;

Inc(i);

end;

MemoTextWin.Lines.Add(pFile);

UnMapViewOfFile(pFile);

CloseHandle(hFile);

CloseHandle(hFileMap);

end;

end;

end.