МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

филиалфедеральногогосударственногобюджетногообразовательногоучреждениявысшегообразования

«Национальныйисследовательскийуниверситет«МЭИ»вг.Смоленске

Кафедра вычислительной техники

Отчет

По лабораторной работе №3

Тема:«Использование виртуальной памяти в приложениях»

покурсу:«Операционные системы»

Студент: Иванов В.В.

Группа: ИВТ 1-18

Преподаватель: Гаврилов А.И.

Вариант:№7

Смоленск,2018 г.

1. **Цель работы**

Целью данной работы является:

* Получение практического опыта работы с виртуальной памятью. Использование механизма работы с виртуальной памятью для решения конкретных прикладных задач;
* Исследование адресного пространства процесса с помощью функций работы с виртуальной памятью.

1. **Задание на лабораторную работу**
2. Введите действительное число x и натуральное число n. Вычислите:
3. Введите диапазон адресов и выведите на экран информацию о состоянии страниц виртуальной памяти в данном регионе (нет доступа, только чтение, чтение и запись).
4. **Анализ задания на лабораторную работу**

Для выполнения первого задания для самостоятельной работы требуется создать кнопку Button1, два элемента Editдля ввода xи n(следует запретить пользователю вводить неверные значения), и элемент Memoдля вывода результата. Для резервирования региона в адресном пространстве следует воспользоваться функцией VirtualAlloc. Начальный адрес нужно вывести в Editдля возможного использования во втором задании.

Для выполнения второго задания для самостоятельной работы требуется создать кнопку Button2, два элемента Editдля ввода начального и конечного адреса(следует запретить пользователю вводить неверные адреса), и элемент Memoдля вывода результата. Для получения информации о состоянии страниц виртуальной памяти следует воспользоваться функцией VirtualQuery.

1. **Модульная структура программы**
2. procedure Button1Click(Sender: TObject);
3. procedure OnActivate(Sender: TObject);
4. procedure Button2Click(Sender: TObject);
5. procedure OnKeyPressEdit(Sender: TObject; var Key: Char);
6. procedure KeyPressPer(Sender: TObject; var Key: Char);
7. **Проектирование (схемы алгоритмов)**





1. **Проверка работоспособности программы**

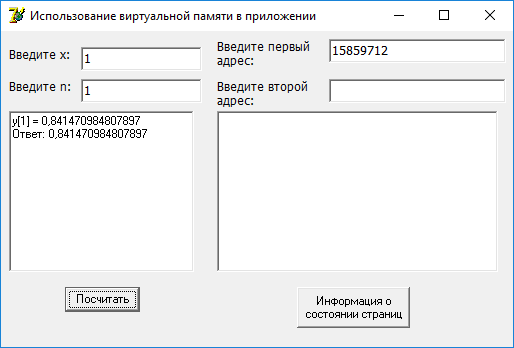


Рисунок 1 – Проверкапервой части №1

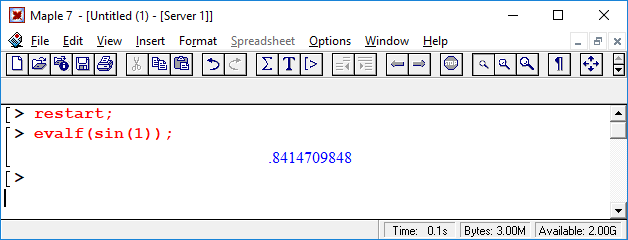


Рисунок 2 – Проверка в Maple №1

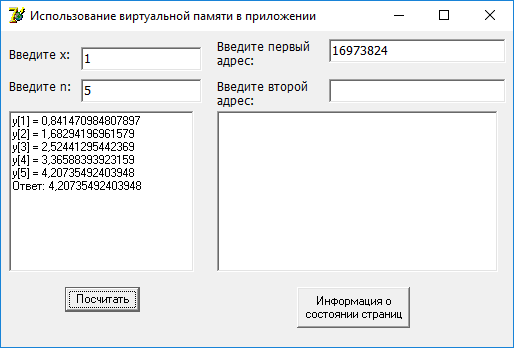


Рисунок 3 – Проверка первой части №2

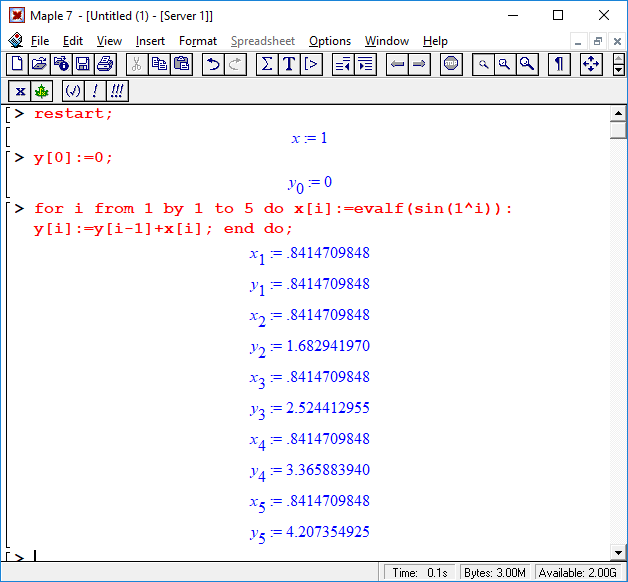


Рисунок 4 – Проверка в Maple №2

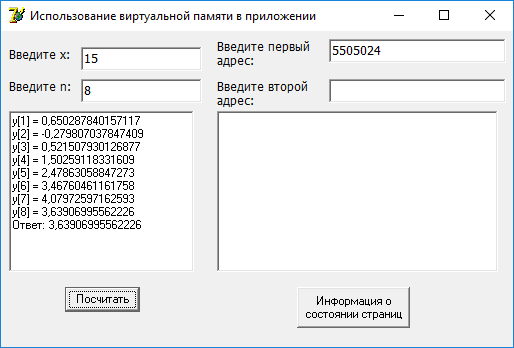


Рисунок 5 – Проверка первой части №3

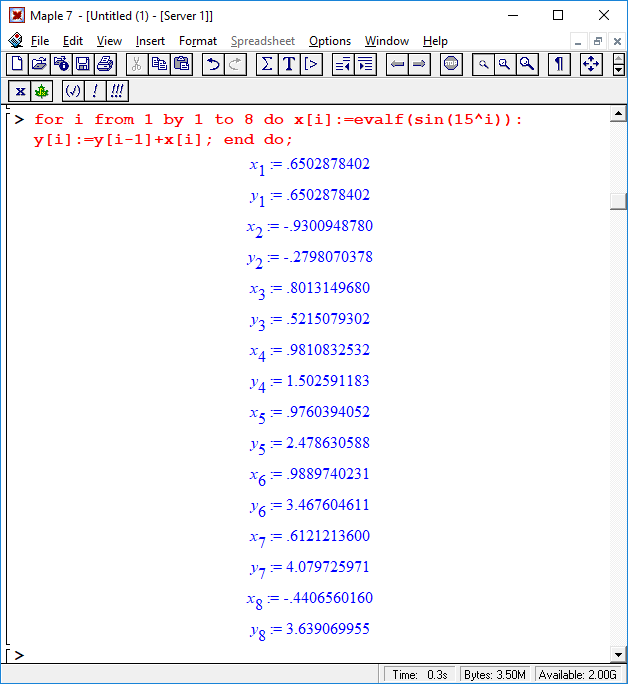


Рисунок 6 – Проверка в Maple№3

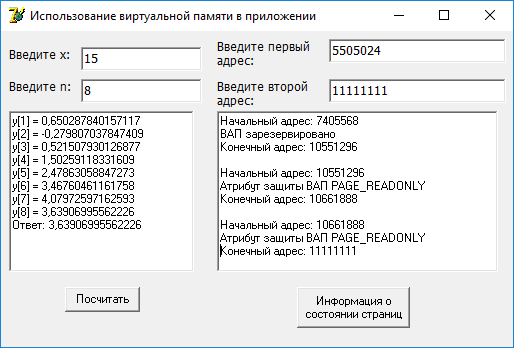


Рисунок 7 – Проверка второй части №1

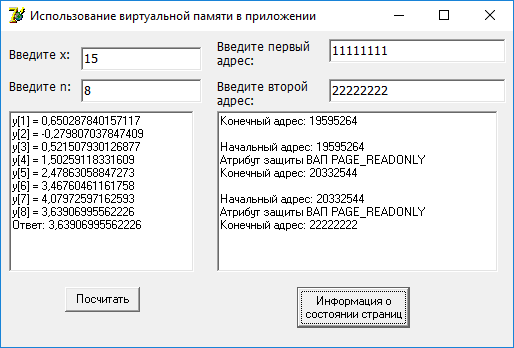


Рисунок 8 – Проверка второй части №2

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

1. **Контрольные вопросы**
2. Объясните различие между такими параметрами, как «Регион адресного пространства», «Блок в регионе адресного пространства» и «страница виртуального адресного пространства»?

Регион адресного пространства − зарезервированный определенный диапазон адресов.

Блок − неразрывная группа страниц с идентичными атрибутами защиты, сопоставленная с одним регионом физической памяти.

Страница виртуального адресного пространства −единица объема памяти, используемая системой при управлении памятью.

1. Объясните алгоритм преобразования виртуального адреса в физический?

Виртуальный адрес делится на номер виртуальной страницы и сдвиг. Номер страницы служит в качестве индекса в таблице страниц. По записи в таблице страниц находится номер физического блока, данный номер присоединяется к старшим разрядам смещения, заменяя, номер виртуальной страницы и формируется физический адрес.

1. Как подсчитатьколичество регионов в адресном пространстве процесса?

Для подсчета количества регионов а адресном пространстве процесса можно воспользоваться функцией VirtualQueryв цикле.

1. Как подсчитать количество блоков в регионе, имеющих один и тот же тип физической памяти?

Для этого можно воспользоваться функций VirtualQueryв цикле и обратиться к полю Type структуры MEMORY\_BASIC\_INFORMATION.

1. **Заключение**

В результате выполнения лабораторной работы был получен практический опыт работы с виртуальной памятью для решения конкретных прикладных задач; изучено исследование адресного пространства процесса с помощью функций работы с виртуальной памятью.

1. **Приложение А. Текст программы**

unit Unit1;

interface

uses

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

Dialogs, StdCtrls, Math;

type

TForm1 = class(TForm)

Edit1: TEdit;

Label1: TLabel;

Label2: TLabel;

Edit2: TEdit;

Button1: TButton;

Memo1: TMemo;

Edit3: TEdit;

Label3: TLabel;

Label4: TLabel;

Edit4: TEdit;

Memo2: TMemo;

Button2: TButton;

procedure Button1Click(Sender: TObject);

procedure OnActivate(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure OnKeyPressEdit(Sender: TObject; var Key: Char);

procedure KeyPressPer(Sender: TObject; var Key: Char);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

VAR

i,n:Integer;

y,x:Double;

p1,p:^Double;

begin

Memo1.Clear();

y:=0;

x:=StrToFloat(Edit2.Text);

n:=StrToInt(Edit1.Text);

p:=VirtualAlloc(nil,4096,MEM\_COMMIT,PAGE\_READWRITE);

Edit3.Text:=IntToStr(integer(p));

p1:=p;

for i:=1 to n do

begin

y:=y+sin(Power(x,i));

p^:=y;

Inc(p);

end;

p:=p1;

y:=0;

For i:=1 to n do

begin

y:=p^;

memo1.Lines.Add('y['+ IntToStr(i)+'] = '+ FloatToStr(y));

Inc(p);

end;

memo1.Lines.Add('Ответ: ' + FloatToStr(y));

end;

procedure TForm1.OnActivate(Sender: TObject);

begin

memo1.Clear();

memo2.Clear();

end;

procedure TForm1.Button2Click(Sender: TObject);

var

mbi:TMemoryBasicInformation;

dwRez:DWORD;

firstAddress: Pointer;

lastAddress: longword;

temp1, temp2: int64;

error: boolean;

begin

if Edit3.GetTextLen = 0

thenShowMessage('Введите первый адрес')

else if Edit4.GetTextLen = 0

thenShowMessage('Введите второй адрес')

else

begin

Memo2.Clear();

error:=false;

temp1:=StrToInt64(Edit3.Text);

if temp1 > 4294967295 then

begin

error:=true;

ShowMessage('Выход за пределы диапазона для первого адреса!');

end;

temp2:=StrToInt64(Edit4.Text);

if temp2 > 4294967295 then

begin

error:=true;

ShowMessage('Выход за пределы диапазона для второго адреса!');

end;

if temp1 >= temp2 then

begin

error:=true;

ShowMessage('Первый адрес должен быть меньше второго!');

end;

if not error then

begin

firstAddress:= pointer(StrToInt64(Edit3.Text));

lastAddress:= StrToInt64(Edit4.Text);

while Integer(firstAddress) <lastAddress do

begin

dwRez:=VirtualQuery(firstAddress, mbi, sizeof(MEMORY\_BASIC\_INFORMATION));

Memo2.Lines.Add('Начальныйадрес: '+IntToStr(integer(mbi.BaseAddress)));

case mbi.AllocationProtect of

0: Memo2.Lines.Add('ВАП зарезервировано');

PAGE\_NOACCESS:Memo2.Lines.Add('Атрибутзащиты ВАП PAGE\_NOACCESS');

PAGE\_READONLY:Memo2.Lines.Add('Атрибутзащиты ВАП PAGE\_READONLY');

PAGE\_READWRITE:Memo2.Lines.Add('Атрибутзащиты ВАП PAGE\_READWRITE');

PAGE\_EXECUTE:Memo2.Lines.Add('Атрибутзащиты ВАП PAGE\_EXECUTE');

PAGE\_EXECUTE\_READ:Memo2.Lines.Add('Атрибутзащиты ВАП PAGE\_EXECUTE\_READ');

PAGE\_EXECUTE\_READWRITE:Memo2.Lines.Add('Атрибутзащиты ВАП PAGE\_EXECUTE\_READWRITE');

PAGE\_WRITECOPY:Memo2.Lines.Add('Атрибутзащиты ВАП PAGE\_WRITECOPY');

PAGE\_EXECUTE\_WRITECOPY:Memo2.Lines.Add('Атрибутзащиты ВАП PAGE\_EXECUTE\_WRITECOPY');

end;

firstAddress:= pointer(Integer(mbi.BaseAddress) + mbi.RegionSize);

if Integer(firstAddress) <lastAddress

then Memo2.Lines.Add('Конечныйадрес: '+IntToStr(Integer(firstAddress)))

else Memo2.Lines.Add('Конечныйадрес: '+IntToStr(lastAddress));

Memo2.Lines.Add('');

end;

end;

end;

end;

procedure TForm1.OnKeyPressEdit(Sender: TObject; var Key: Char);

begin

if (not (Key in ['0','1'..'9'])) and (Key<>#08) then

Key:=#0;

end;

procedure TForm1.KeyPressPer(Sender: TObject; var Key: Char);

begin

if (not (Key in ['0','1'..'9'])) and (Key<>#08) and (Key<>',') then

Key:=#0;

end;

end.