Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования «Московский технический университет связи и информатики»

Лабораторная №4

«Архитектура памяти Windows»

Выполнила студентка

группы БВТ1501

Козлова Анна

8 вариант

МОСКВА, 2018

**Цель работы:**

Получение практических навыков по использованию Win32 API для исследования памяти Windows.

**Код:**

#include <windows.h>

#include <iostream>

#include <process.h>

#include <iomanip>

#include <conio.h>

#include <cstdio>

#include <tlhelp32.h>

#define TCHAR wchar\_t

#define cout wcout

#define cin wcin

#define TEXT(p) L##p

using namespace std;

#include <stdio.h>

HANDLE console = GetStdHandle(STD\_OUTPUT\_HANDLE);

void Diagr(int d) {

SetConsoleTextAttribute(console, 11);

for (int i = 0;i < 50;i++) {

if (i == d / 2) SetConsoleTextAttribute(console, 15);

cout << TEXT("|");

}

SetConsoleTextAttribute(console, 15);

cout << " " << d << endl;

}

int main() {

setlocale(LC\_ALL, "Russian");

/// Информация по GlobalMemoryStatus ///

MEMORYSTATUS stat;

ZeroMemory(&stat, sizeof(stat));

stat.dwLength = sizeof(stat);

GlobalMemoryStatus(&stat);

int proc;

SetConsoleTextAttribute(console, 15);

cout << TEXT("MemoryStatus занимает ") << stat.dwLength << TEXT(" байт.") << endl;

cout << TEXT("Память заполнена на ") << stat.dwMemoryLoad << TEXT("%.") << endl;

Diagr(stat.dwMemoryLoad);

cout << TEXT("Физическая память\nВсего - ") << stat.dwTotalPhys / 1024 / 1024 << TEXT(" Мбайт.\n");

cout << TEXT("Свободно - ") << stat.dwAvailPhys / 1024 / 1024 << TEXT(" Мбайт.\n");

double proc1 = stat.dwAvailPhys / 1024;

double proc2 = stat.dwTotalPhys / 1024;

proc = proc1 / proc2 \* 100;

Diagr(proc);

cout << TEXT("Объем, который могут сохранить файлы/файл подкачки\nВсего - ") << stat.dwTotalPageFile / 1024 / 1024 << TEXT(" Мбайт.\n");

cout << TEXT("Свободно - ") << stat.dwAvailPageFile / 1024 / 1024 << TEXT(" Мбайт.\n");

proc1 = stat.dwTotalPageFile / 1024;

proc2 = stat.dwAvailPageFile / 1024;

proc = proc2 / proc1 \* 100;

Diagr(proc);

cout << TEXT("Виртуальная память:\nВсего - ") << stat.dwTotalVirtual / 1024 / 1024 << TEXT(" Мбайт.\n");

cout << TEXT("Свободно - ") << stat.dwAvailVirtual / 1024 / 1024 << TEXT(" Мбайт.\n");

proc1 = stat.dwTotalVirtual / 1024;

proc2 = stat.dwAvailVirtual / 1024;

proc = proc2 / proc1 \* 100;

Diagr(proc);

/// Карта виртуальной памяти ///

system("pause");

system("cls");

SetConsoleTextAttribute(console, 11);

cout << TEXT("Все процессы, потоки, модули и их свойства в системе:\n");

cout << setiosflags(ios::left) << setw(33) << TEXT("Имя процесса")

<< setw(10) << TEXT("PID") << endl;

SetConsoleTextAttribute(console, 15);

/// процессы ///

HANDLE all = CreateToolhelp32Snapshot(TH32CS\_SNAPPROCESS, 0);

if (all == NULL) {

return 0;

cout << TEXT("Всё плохо.");

}

PROCESSENTRY32 proc0;

proc0.dwSize = sizeof(PROCESSENTRY32);

if (Process32First(all, &proc0)) {

do {

cout << setw(33) << proc0.szExeFile;

cout << setw(10) << proc0.th32ProcessID << endl;

} while (Process32Next(all, &proc0));

}

CloseHandle(all);

HANDLE hProcess;

int id;

SetConsoleTextAttribute(console, 11);

cout << TEXT("Какой процесс рассмотреть? ") << endl;

SetConsoleTextAttribute(console, 15);

cin >> id;

/// поток ///

hProcess = OpenProcess(PROCESS\_ALL\_ACCESS, FALSE, id);

MEMORY\_BASIC\_INFORMATION meminfo;

UINT adr = 0;

ZeroMemory(&meminfo, sizeof(meminfo));

cout << TEXT("Базовый адрес") << TEXT("\t") << TEXT("AllocationProtect") << TEXT("\t")

<< TEXT("Размер региона") << TEXT("\t") << TEXT("State") << endl;

while (adr < 0x7FFFFFF) {

VirtualQueryEx(hProcess, (void\*)adr, &meminfo, sizeof(meminfo));

cout << TEXT("0x") << hex << adr << dec

<< TEXT("\t") << meminfo.AllocationProtect << TEXT("\t\t\t") << meminfo.RegionSize << TEXT("\t\t");

switch (meminfo.State) {

case MEM\_FREE:

cout << TEXT("free");

break;

case MEM\_RESERVE:

cout << TEXT("reserve");

break;

case MEM\_COMMIT:

cout << TEXT("commit");

break;

}

cout << endl;

adr = adr + meminfo.RegionSize;

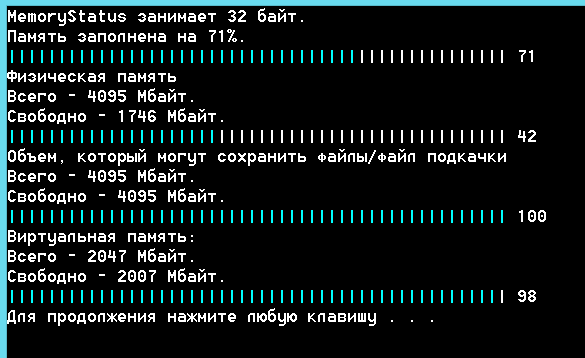
}

system("pause");

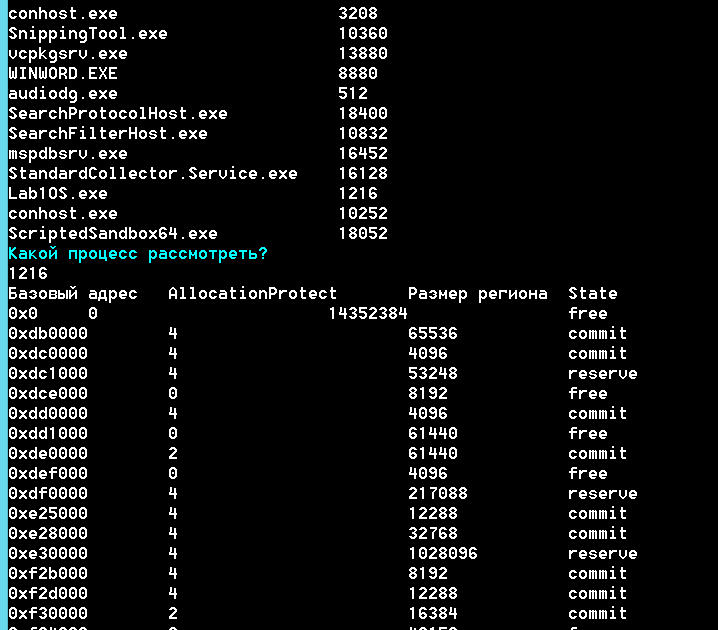
return 0;

}

**Пример работы:**



На скрине ниже показана часть процессов и их id и часть виртуальной карты памяти.



**Вывод:**

В результате проделанной работы исследовали архитектуру памяти Windows и научились ее строить.