## **Следы использования программного обеспечения в Windows**

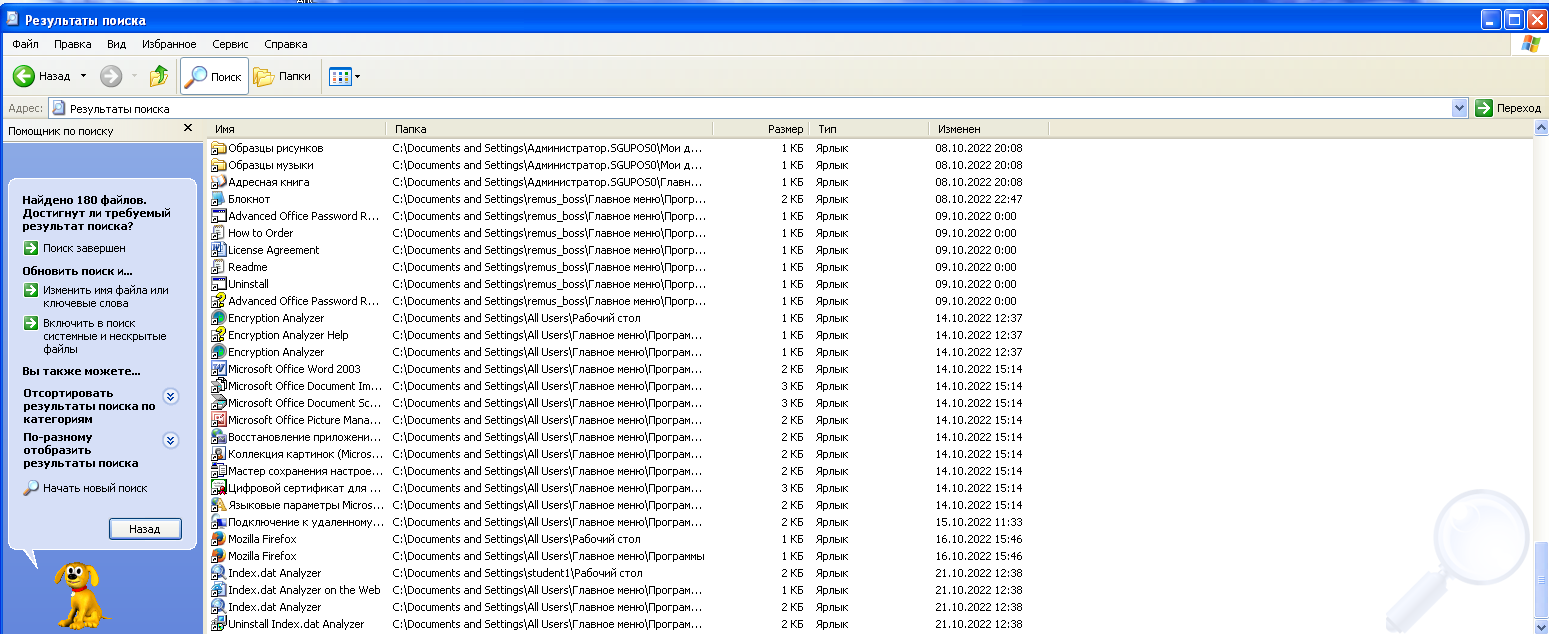
**Работу выполнил студент 431гр. Никитин А.В.**

Ответы на вопросы предваряются опытами по обнаружению и оценке следов, возникающих вследствие действий, указанных в вопросе.

Для диагностического исследования использовать виртуальный диск резервной ОС Windows.

1. Собственно ОС (lnk, MRU в реестре, DR.Watson, \*.evt с помощью EventLog Explorer, Проводник)
   1. Обнаружить хранилища всех файлов LNK на системном диске. Ответить на вопросы для LNK какие файлы-документы открывались последними?

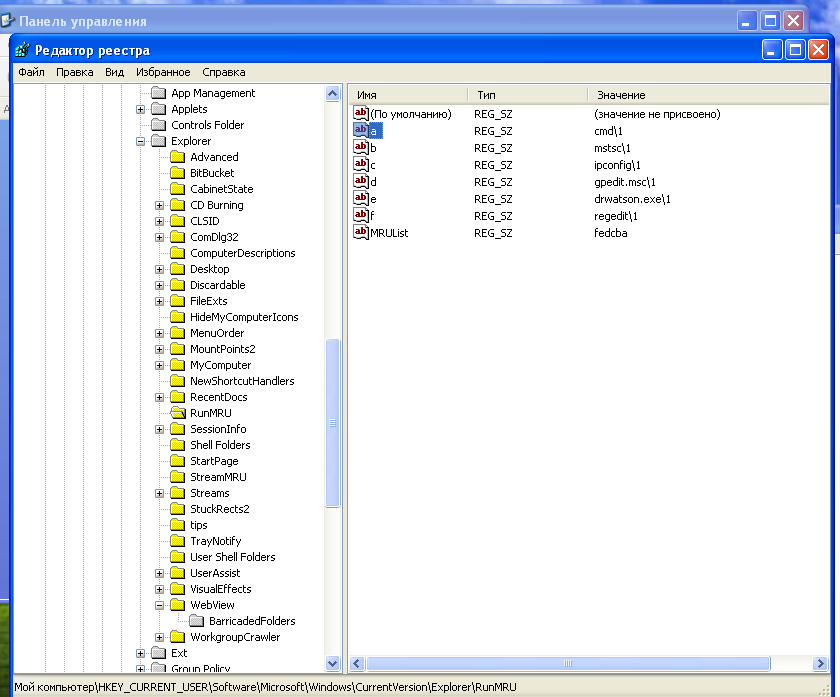
Открыл системный диск и с помощью проводника выполнил поиск по маске **\*.lnk**, а затем отсортировал полученный список по времени использования:



* 1. На основе анализа параметров MRU реестра и содержимого каталога Prefetch ответить на вопрос: какие программы запускались последними?

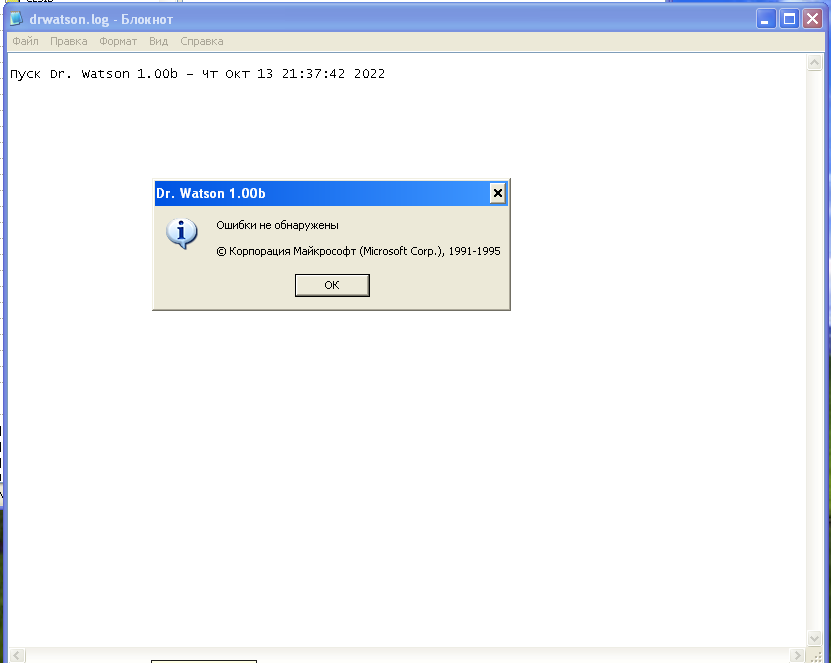
Параметры MRU (most recent used) указывают на то, какие программы были запущены последними. Перешел по ветке реестра **HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU** и увидел, что последние использованные программы это:

**regedit, drwatson, gpedit.msc, ipconfig, mstsc, cmd:**



* 1. Ответить на вопросы для DR.Watson:
     1. Какие программы вызвали ошибку, дата и время возникновения ошибок?

Нашел программу drwatson на компьютере и запустил. После запуска программа ошибок не обнаружило. Об этом также свидетельствует файл drwatson.log, в котором отображено время запуска программы и отсутствие ошибок:

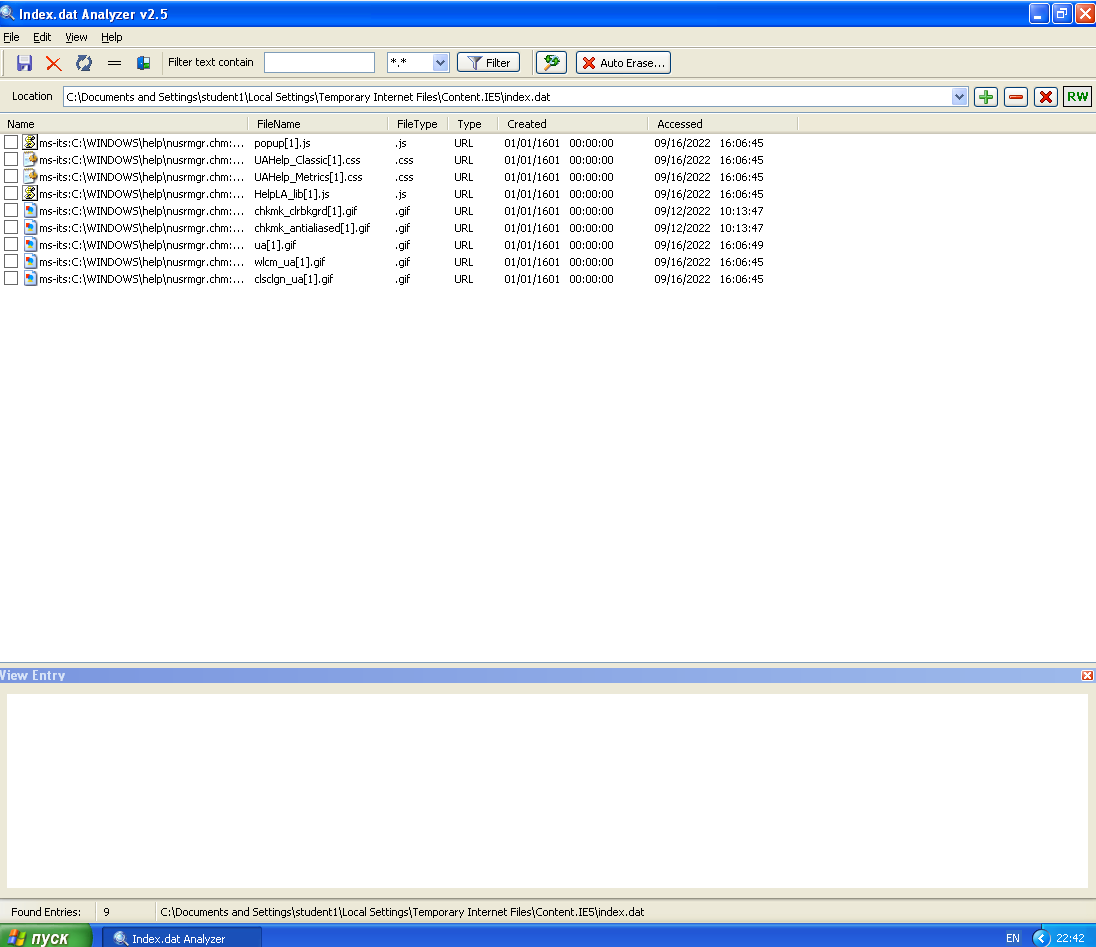


* + 1. – Каковы имя компьютера, имя пользователя, количество процессоров, тип процессора, версия Windows?
    2. – Какие задачи выполнялись в системе во время возникновения ошибок?
    3. – Какие модули загружала каждая программа?

Так как log-файл пуст, то есть ошибок программой обнаружено не было, следовательно, из отчета программы выяснить имя компьютера, пользователя, процессы и типы процессов из-за которых могли возникнуть ошибки не удалось.

* 1. Ответить на вопросы для Проводник (С:\Documents and Settings\[профиль]\Local Settings\Temporary Internet Files\Content.IE5\ index.dat):

Файл index.dat просто так не откроется, поэтому установил программу **Index.dat Analyzer v.2.5** из предложенного архива с полезными программами, а также выбрал интересующий нас для анализа файл:



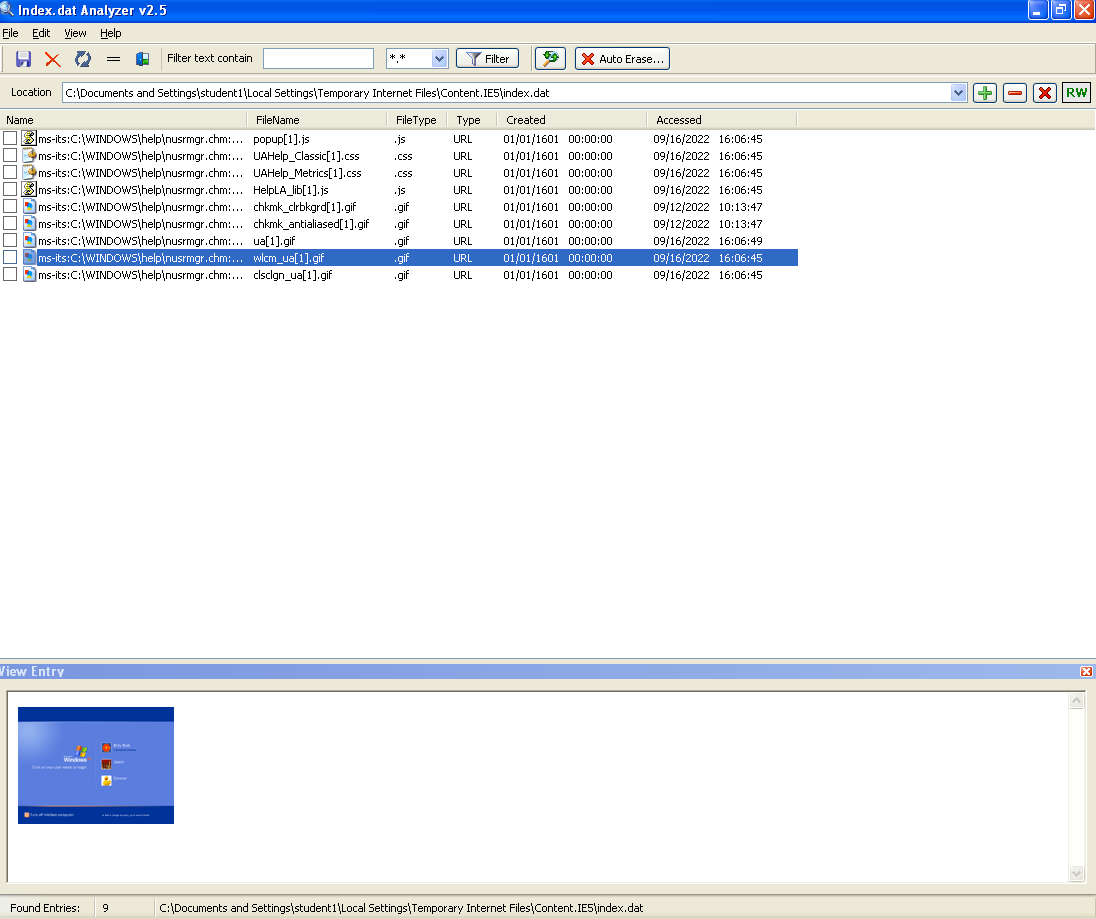
* + 1. – адреса сетевых ресурсов, посещенных пользователем.

Сетевые ресурсы, которые посещал пользователь, обнаружены не были, поэтому невозможно отследить дату, время посещения и содержание сетевых ресурсов.

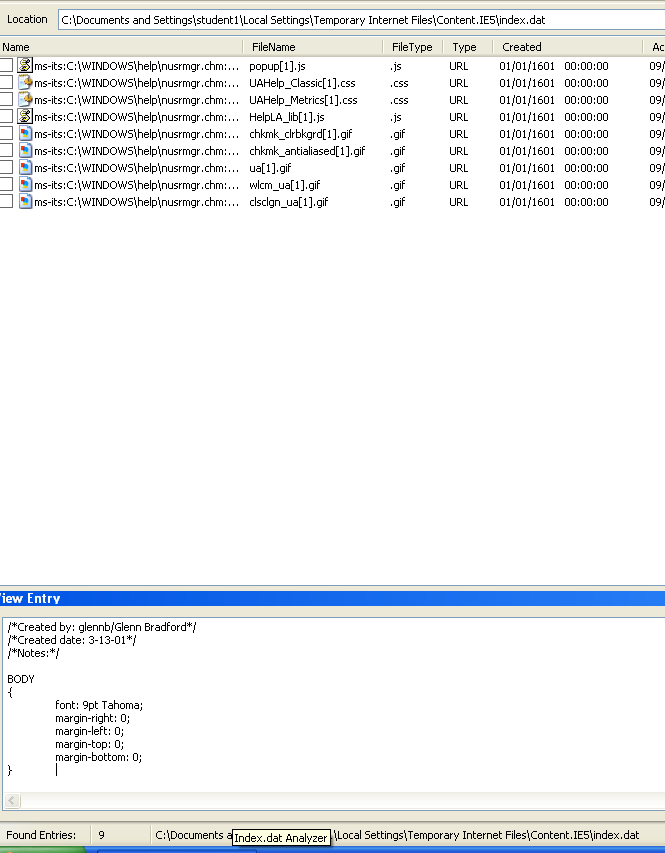
* + 1. – дату и время посещения каждого ресурса (время изменения, время доступа),
    2. – поисковые запросы.
    3. – локальную работу с файлами

А вот локальная работа с файлами по данному адресу представлена:

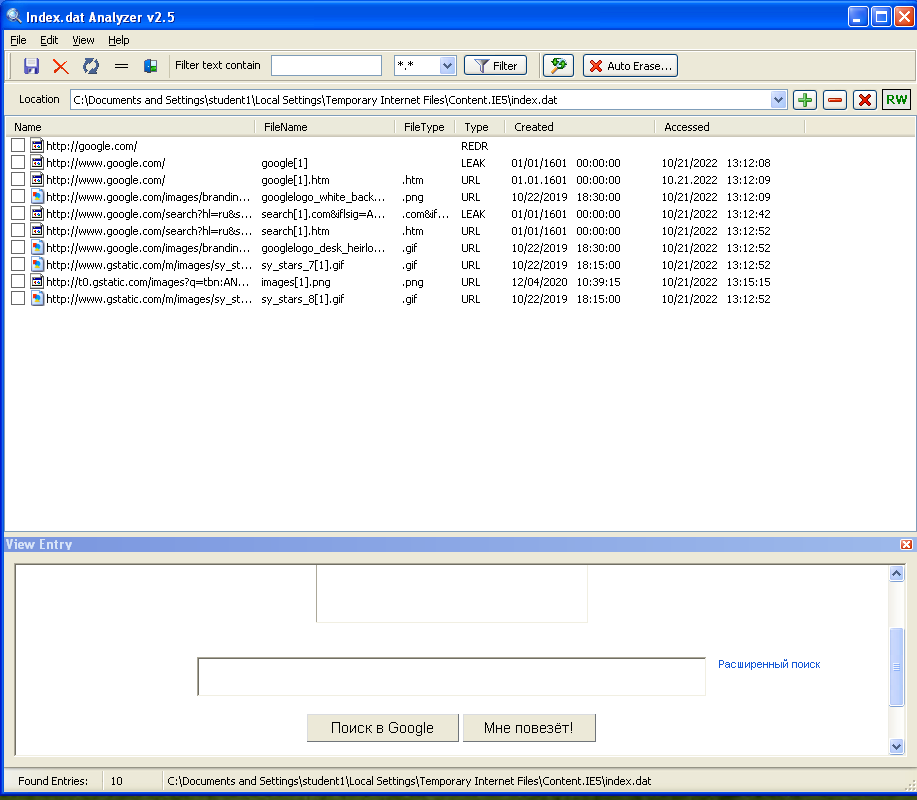
К примеру, нашел сделанный ранее скриншот:

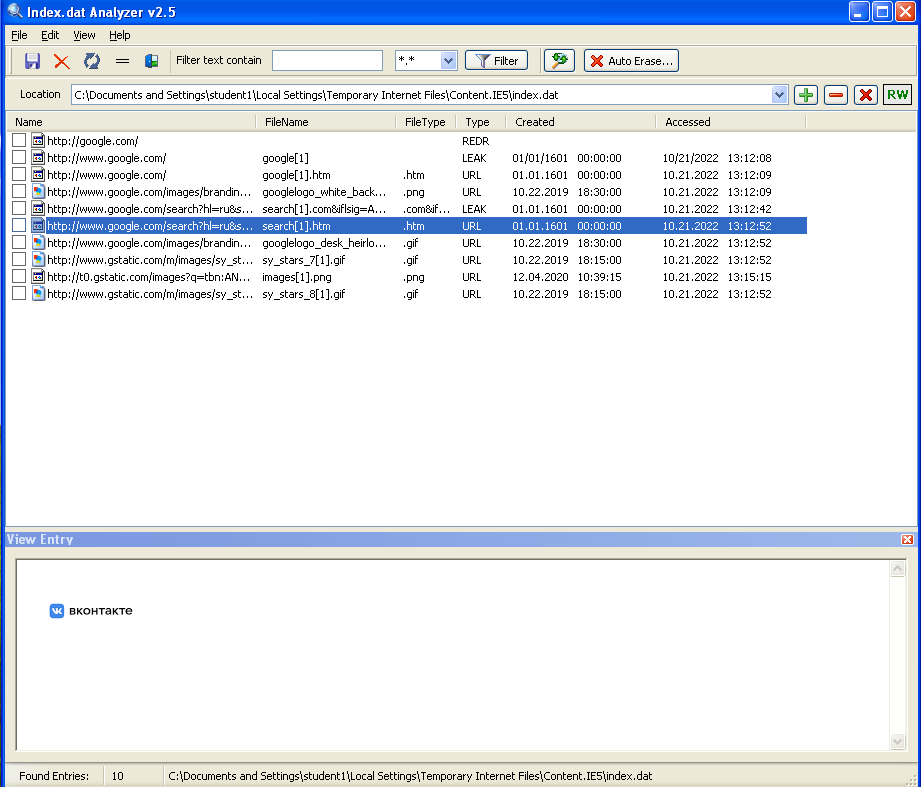


А также код на языке программирования:



Далее самостоятельно ввел несколько поисковых запросов и получил следующие следы в программе:





1. Офисов
   1. Методом мониторинга файловой системы и реестра Windows (ProcMon) определите области следообразования при выполнении следующих операций:
      1. Установка MSOffice2003 – только word (отследите по «Image Path» процесс SETUP, когда запустите установку; отфильтруйте список событий так же и по операции (Operation: CreateFile, RegCreateKey – см рис 1.), обратите внимание на создание файлов с расширениями \*.ini, \*.txt, \*.xml. Действительно ли изменяются dll файлы в системных каталогах windows, указанные ProcMon как создаваемые в процессе инсталляции? Установите это методом сравнения дат создания файлов до и после операции инсталляции. Для этого используйте не измененную версию ОС)

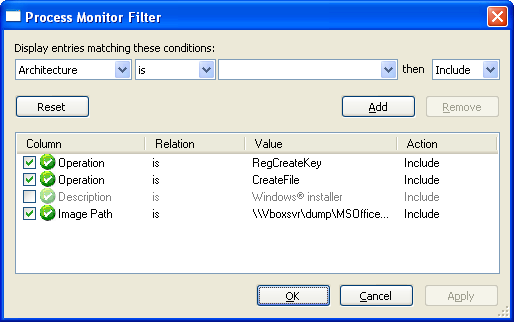
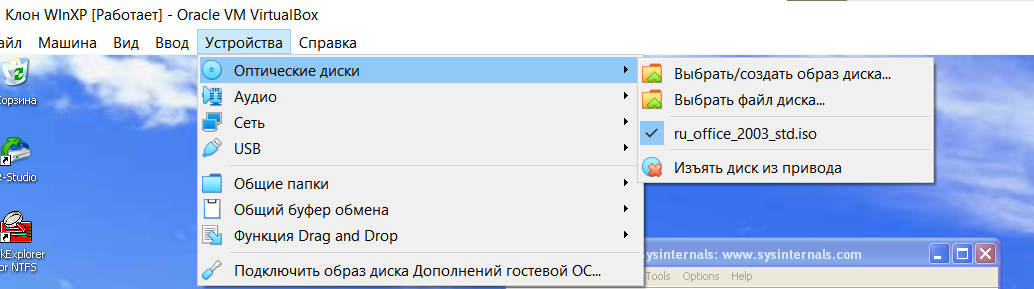
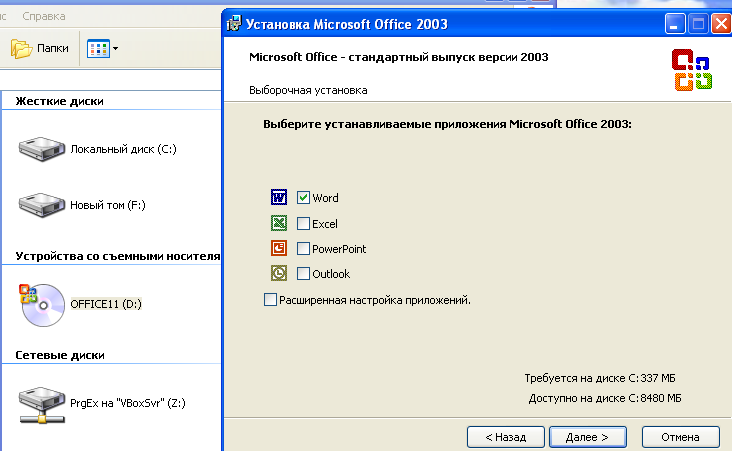


Рис. Предпочитаемые фильтры имеют установленную галочку.

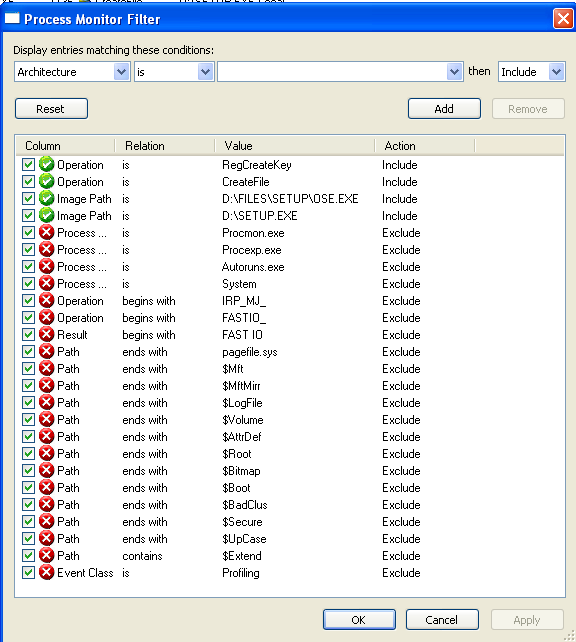
Добавил образ оптического диска .iso для того чтобы можно было установить программу word:



Запустил на клонированной системе установку Office (word). Для этого также потребовалось ввести ключ продукта (**QQ6HT-FRFQ6-FKPXT-BKKMP-DGYQ6).** Затем, соответственно, выбрал установку только программы Word:

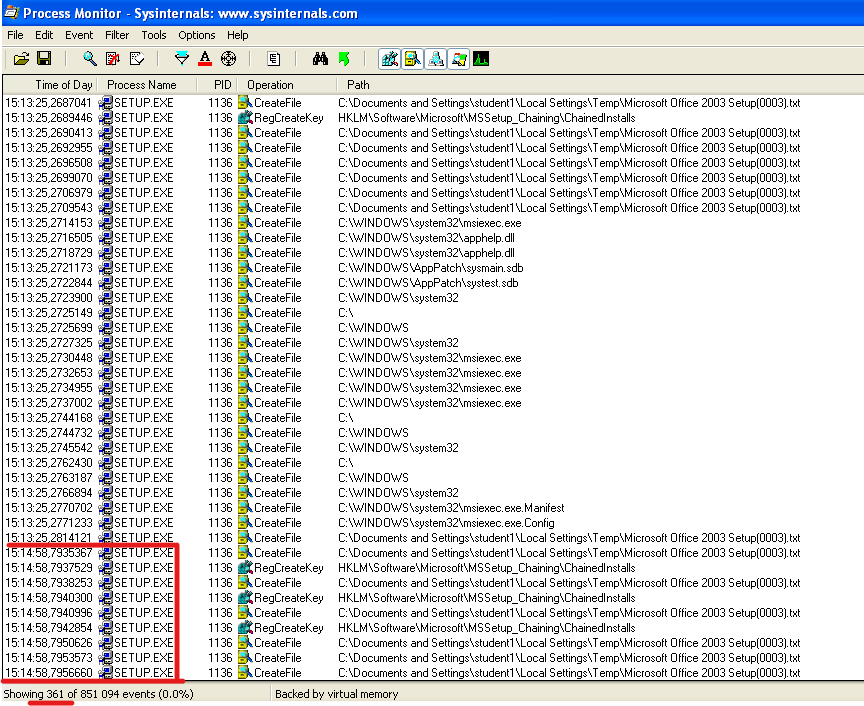


Итак, выполнил установку, с отслеживанием процессов с помощью программы ProcMon. Для этого перед установкой программы Word заранее открыл ProcMon и выставил определенные фильтры.

Image Path указал путь к установщику, то есть [D:\SETUP.EXE](file:///D:/SETUP.EXE), а также выбрал файл OSE.EXE, который также находится на установочном носителе D:

В итоге, до запуска непосредственно процесса установки Word, ProcMon смог зафиксировать

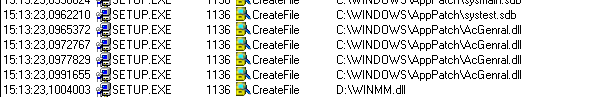
**352 события,** а в процессе установки появилось еще **9** новых записей и получилось **361** запись:

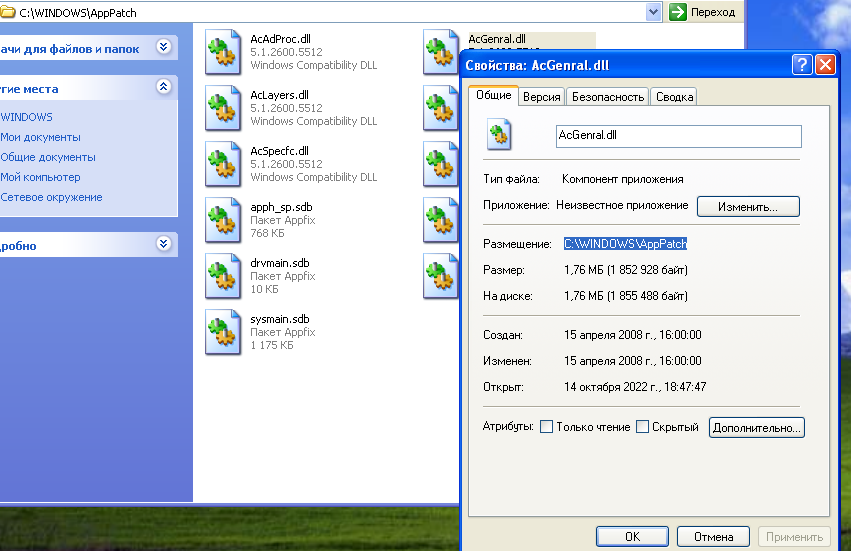


Далее найдем файлы с расширением dll и посмотрим на время создания в программе ProcMon, а затем сравним с временем их создания на неклонированной машине:

Для файла C:\WINDOWS\AppPatch\AcGeneral.dll в программе видим время создания **15:13:23,**

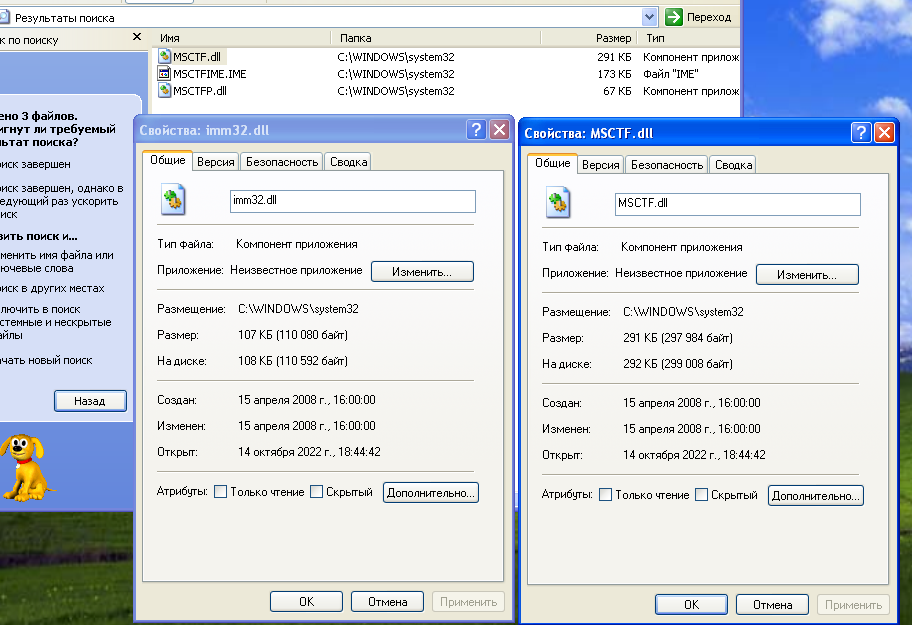
а в исходной машине тот же файл создан далеко не в это время:



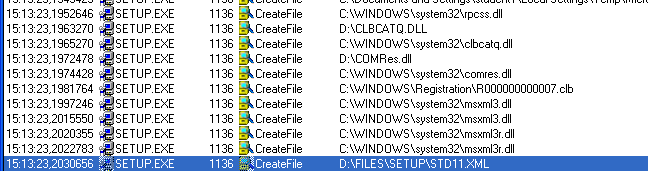


Аналогично и для следующих файлов:

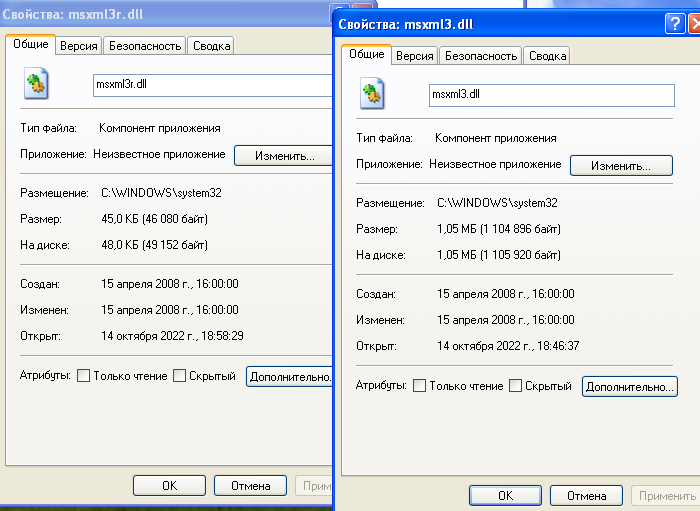




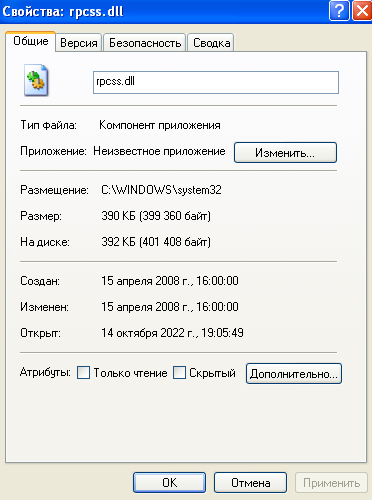
Также нашел еще системные файлы .dll, а также единственный файл с расширением .XML (внизу скриншота), который создается в процессе открытия программы установки и найти в системе его не возможно, собственно так как расположен он на диске-установщике:



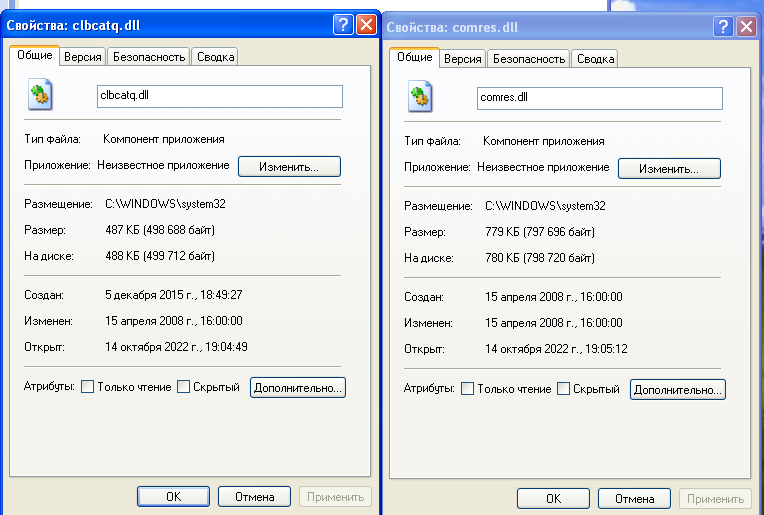
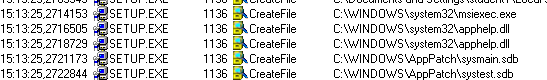
Теперь просмотрим .dll файлы в исходной системе:



Ни один из файлов изменен или создан в это время не был:

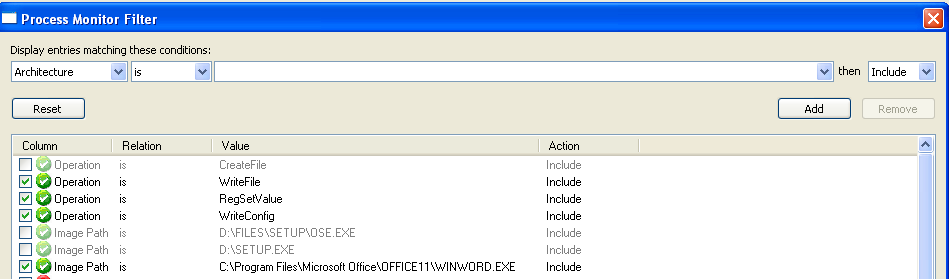


Последние записи о системных файлах, которые удалось найти в программе:

 Аналогично просмотрим файлы в проводнике:

1. Создание и редактирование файла (Обратите внимание на создание в реестре в ветвях с именем MRU новых параметров и файлов \*.lnk, \*.doc, \*.tmp в файловой системе). С этой целью установите в ProcMon фильтры событий Image Path:WinWord, Operation: WriteFile, Write, RegSetValue.

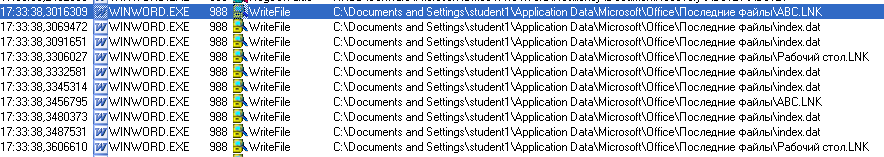
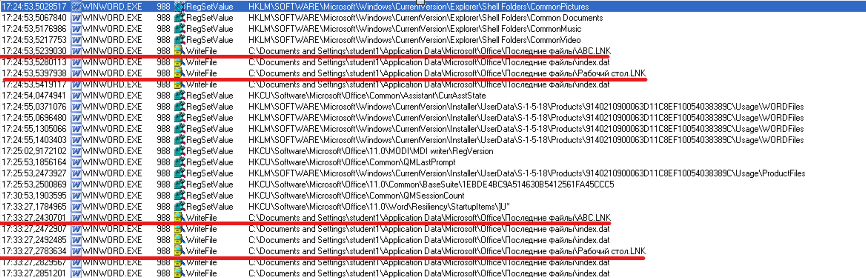
Итак, открыл программу ProcMon и выставил следующие фильтры:



А затем создал пустой word-файл, получил следующие события, отображенные в программе:

**.lnk-файлы:**

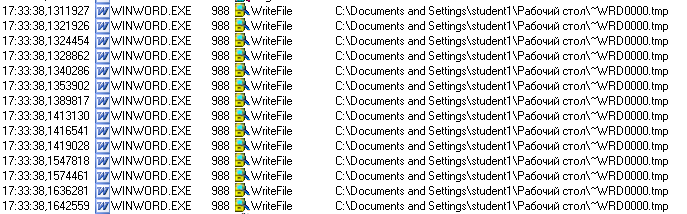
Отмечу, что для разделения событий, связанных с созданием файла, а также с его редактированием, что и можно видеть на скриншоте ниже. В 17:24:53 файл был создан, а в 15:33:27 отредактирован:



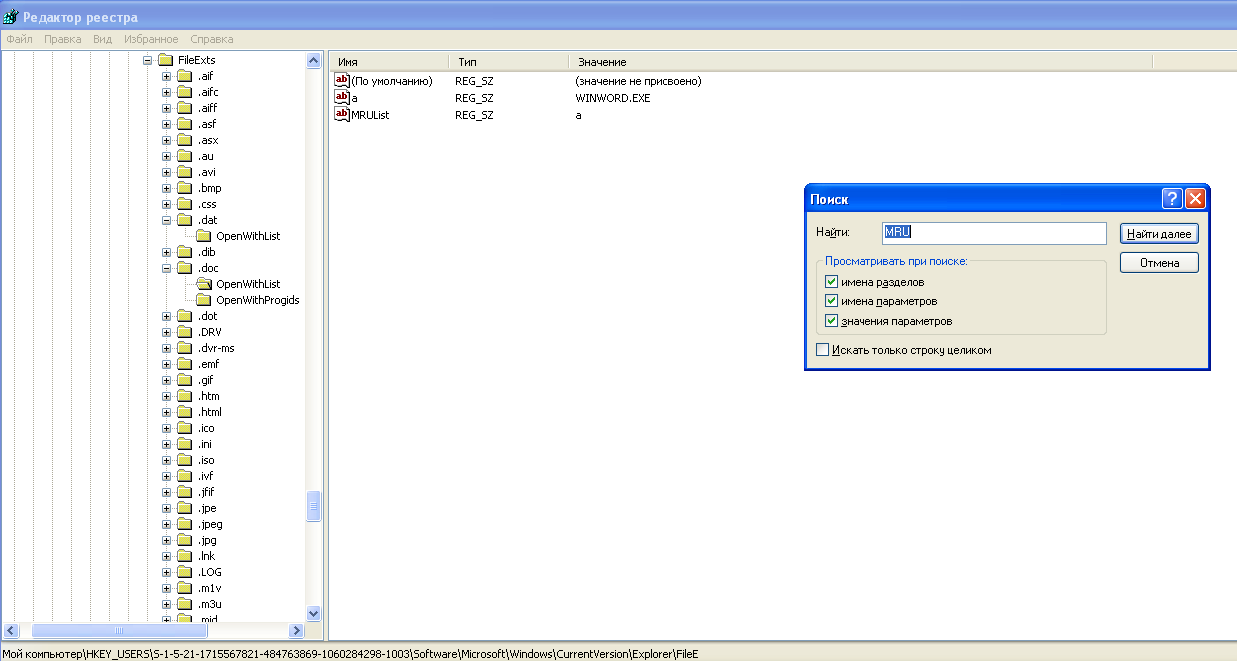
**.doc** файлы имеются в событиях только на момент создания файла, то есть в 17:24:33:



**.tmp** файлы же, в отличие от файлов doc, создаются только во время редактирования документа, о чем свидетельствует время:

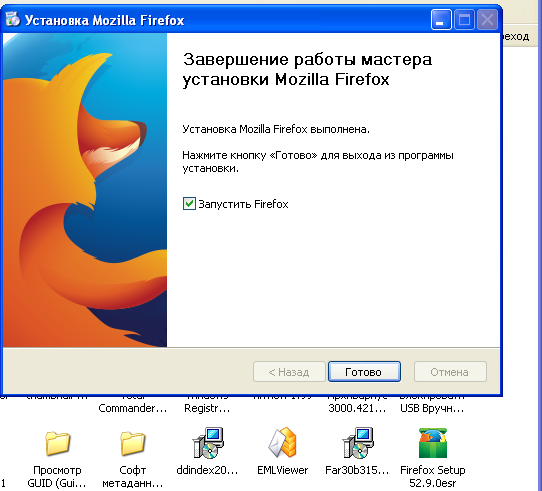


Видим в реестре открытие программы word:



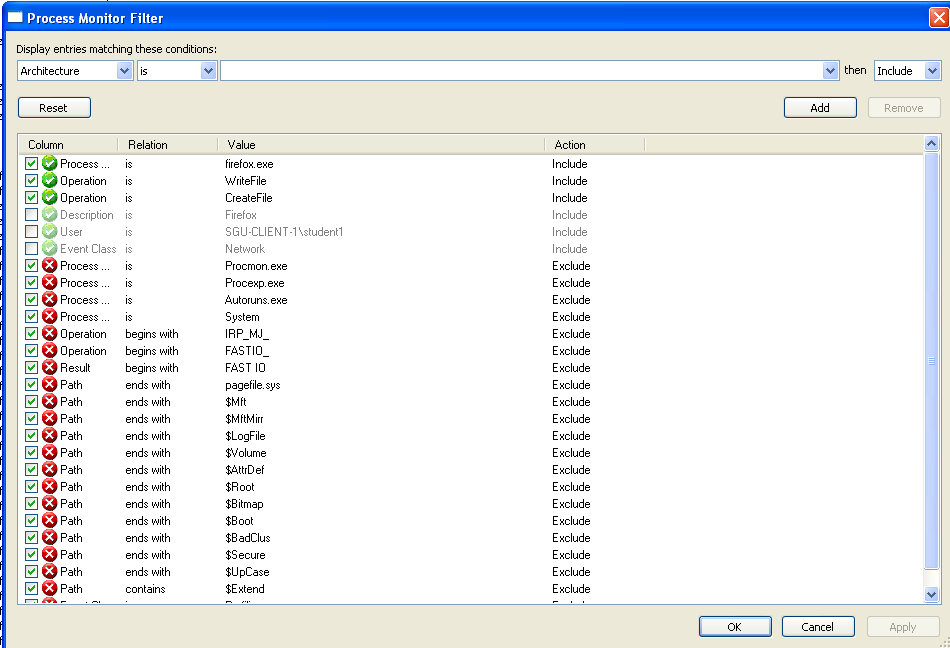
1. Браузера FireFox
   1. Установите этот браузер

Последняя версия Mozilla FireFox, которая поддерживалась компанией была версия 52.9.0.esr, ее и установим:



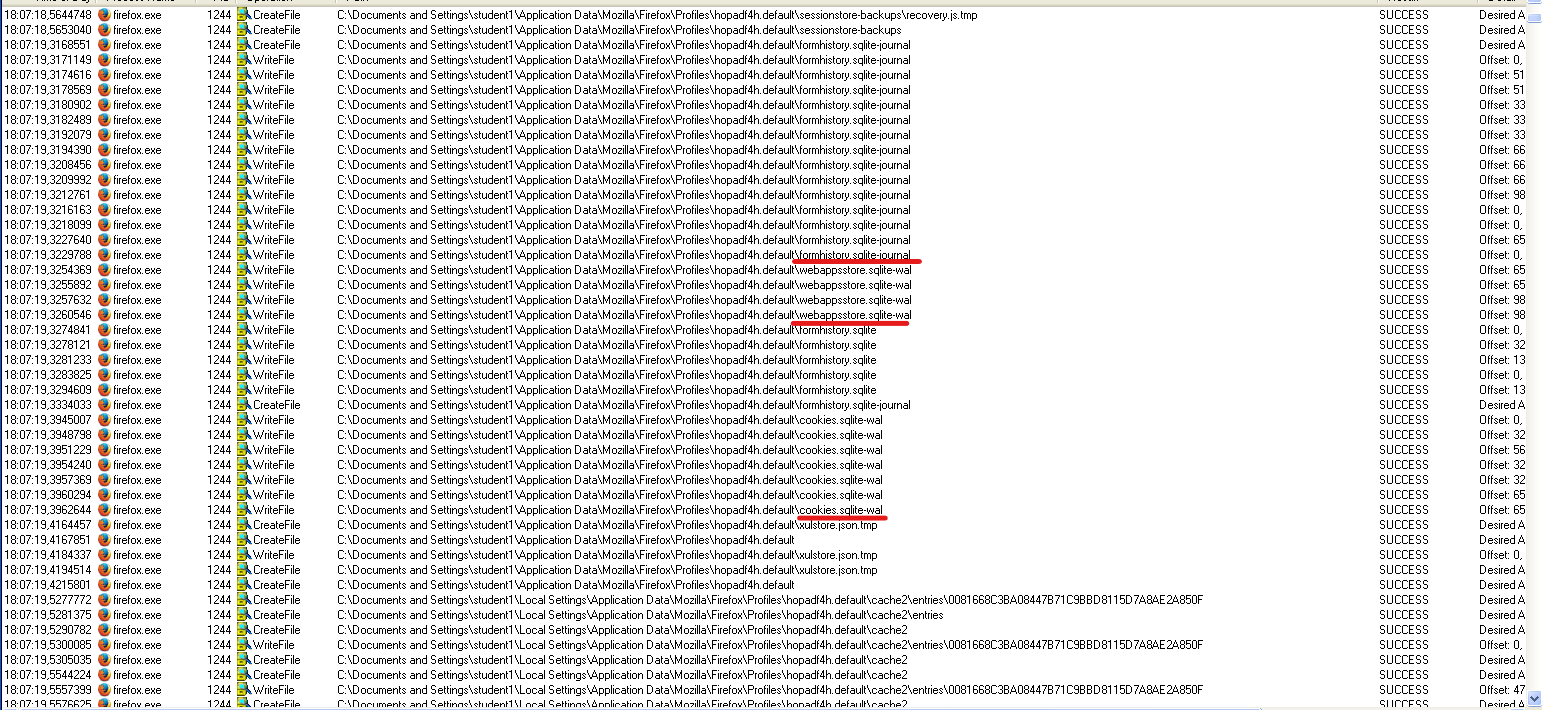
* 1. Установите в Virtual Box эмуляцию сети NAT, адаптер установите PCNet-Fast  III .
  2. Методом мониторинга файловой системы и реестра Windows (ProcMon) определите области следообразования при выполнении следующих операций:
     1. Осуществите заход из виртуальной машины на сайты по протоколам http и https. Найдите области следообразования, указывающие на факт посещения сайтов.

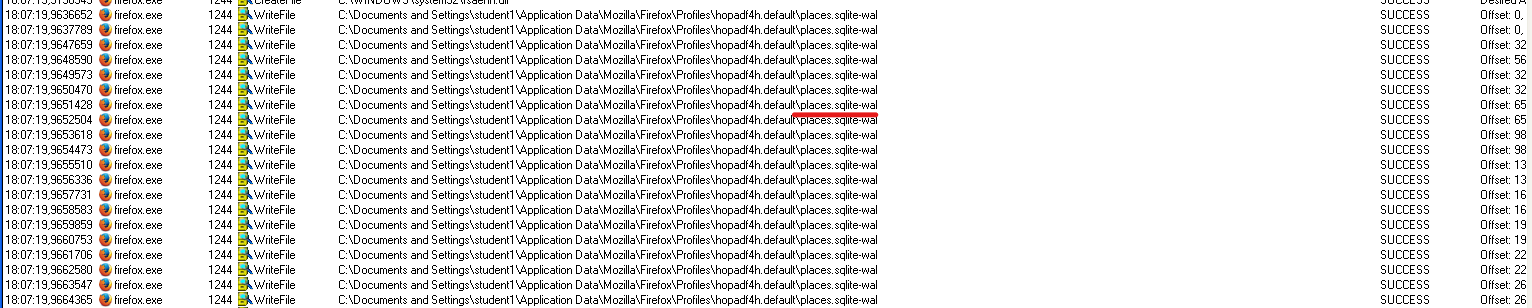
Выставил фильтры в программе ProcMon:



Затем осуществил заход на сайты vk.com, а также youtube.com и получил следующие следы в ProcMon:

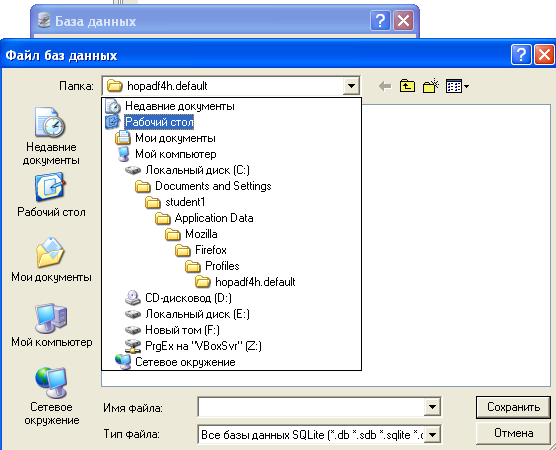
Получил предполагаемые базы данных, к которым будем подключаться с помощью SQLite 3 (formhistory, cookies, places):

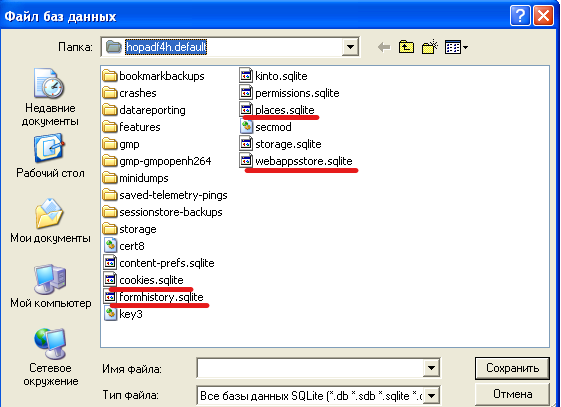


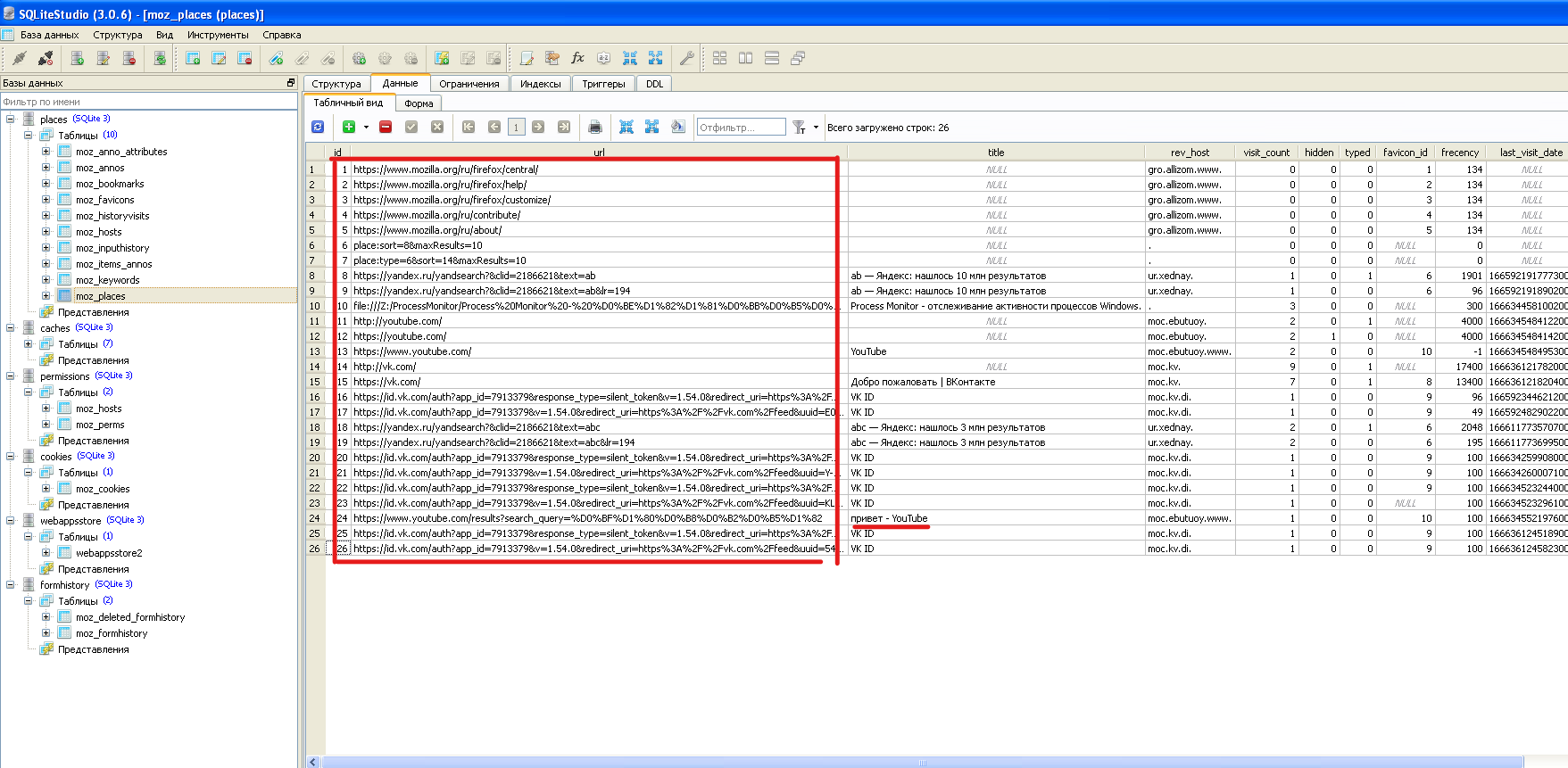


* + 1. Осуществите заход на сайты, требующие ввода данных в поля (GET). Найдите сведения об этом в областях следообразования. Для просмотра сведений используйте программу sqlitestudio-2.1.2.exe

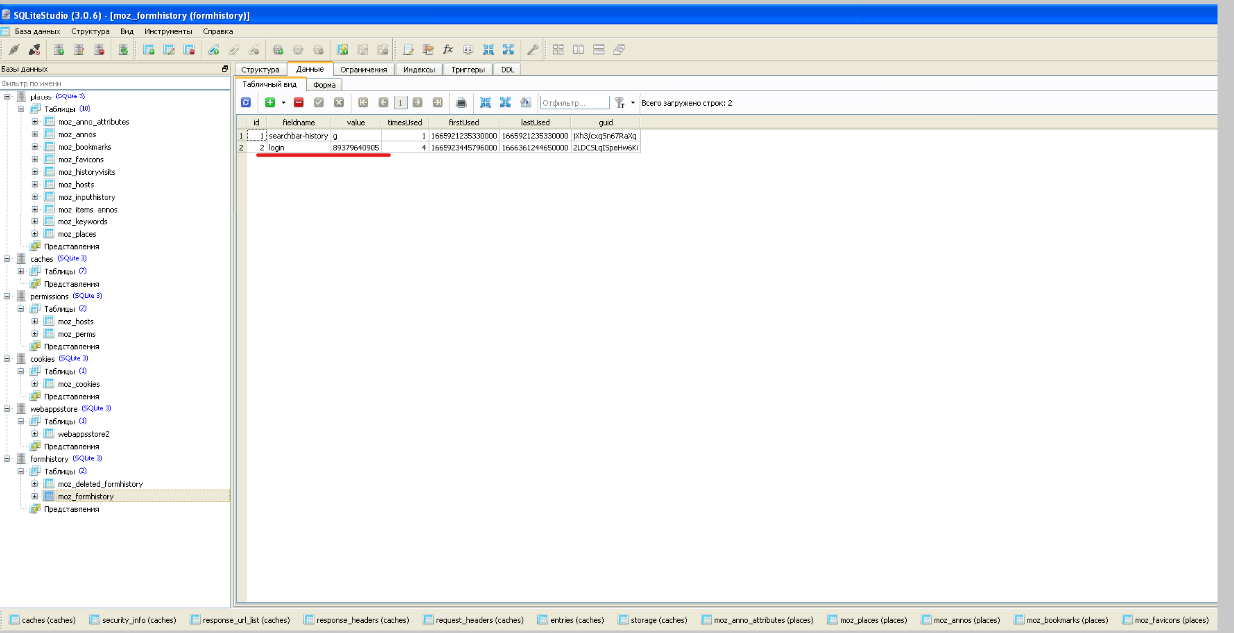
Теперь откроем полученные базы данных в приложении и просмотрим образованные следы:



Зашел на сайт YouTube и ввел там в поиске слово «Привет», данный след оказался в базе данных places, а конкретно в таблице moz\_places. Также там оказались и все остальные места которые я посетил:

Из скриншота выше видно, что ссылка по которой я переходил хранится в атрибуте url, а вот запрос внутри YouTube хранится по атрибуту title. Важно, что информацию, которую получил сайт Вконтакте в данной таблице найти не удалось.

Информация же, которая была введена на сайте Вконтакте была найдена в таблице moz\_formhistory уже из базы данных formhistory:



Также нашел информацию по хостам:

