­Министерство науки и образования РФ

Федеральное государственное автономное образовательное

учреждение высшего профессионального образования

«Санкт-Петербургский государственный электротехнический

университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»

(СПбГЭТУ «ЛЭТИ»)

Кафедра вычислительной техники

Отчёт

по лабораторной работе № 1

на тему:

“Исследование объектов Windows”

по дисциплине “Операционные системы”

Выполнил студент гр. 4306:

Табаков А.В.

Принял: Тимофеев А.В.

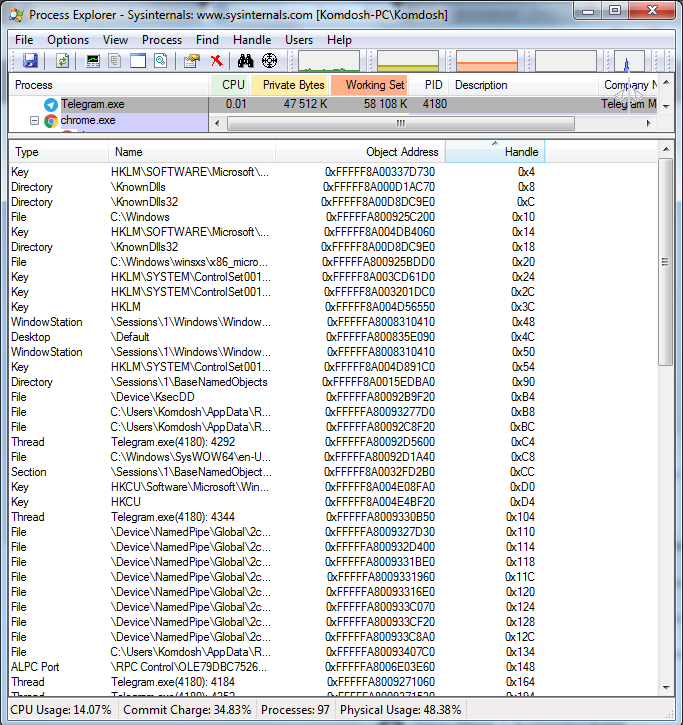
Санкт-Петербург  
2016

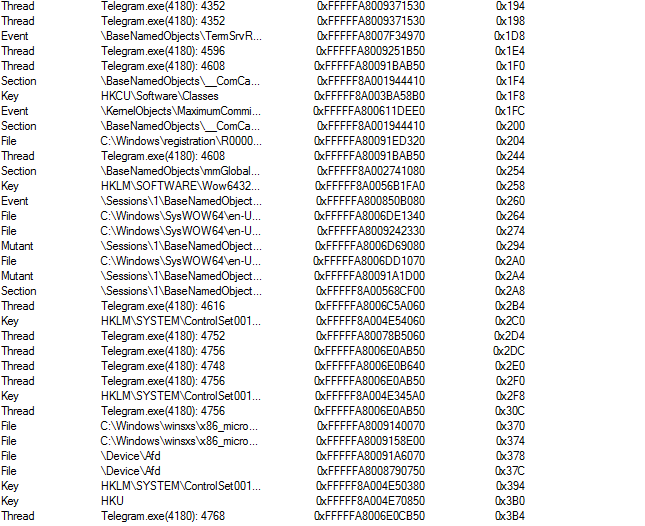
**Цель работы**: исследовать объектные механизмы Win32.

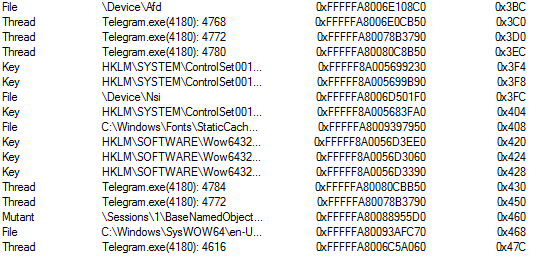
**Задание 1.1**: Получить список открытых объектов и изучить типы объектов

Выбран процесс Telegram.exe c PID = 4180

С объекты программы можно посмотреть с помощью Process Explorer

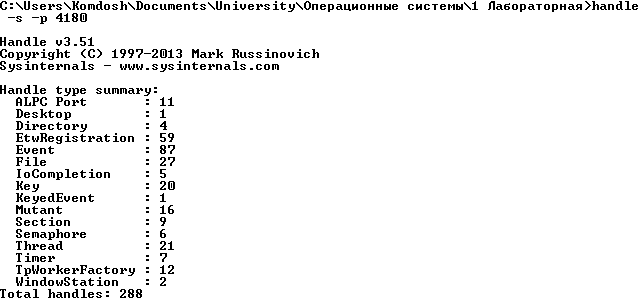






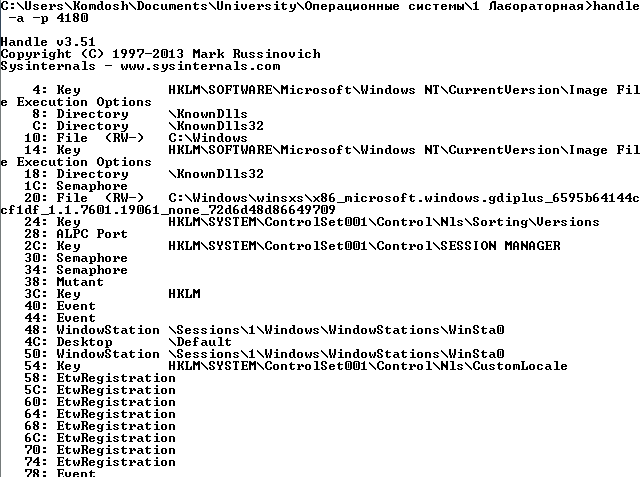
А также с помощью утилиты handle.exe

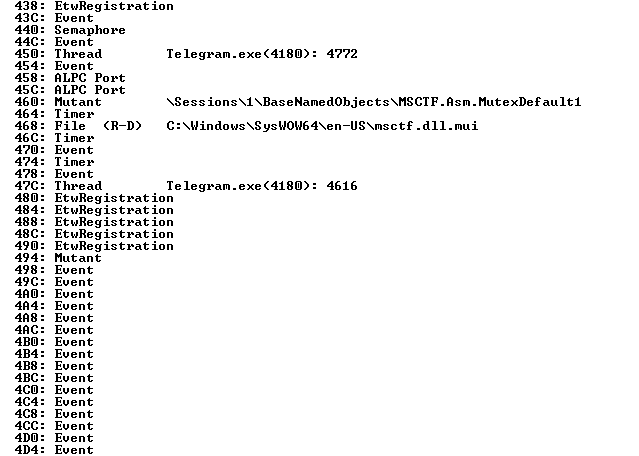
Вывод используемых типов объектов программой Telegram.exe с помощью ключа -s



Вывод всех объектов программы Telegram.exe с помощью ключа –a

Приведён не полный список, он совпадает с выводом программы Process Explorer





Описание типов объектов:

*ALPC port* (advanced local procedure call) – порт усовершенствованного вызова локальных процедур.

desktop – объект, содержащий logical display surface и элементы пользовательского интерфейса, может создавать и управлять ими.

window station – объект, содержащий свои экземпляры буфера обмена, atom table(список пар строка-идентификатор), и один или несколько объектов desktop

file – наименьшая единица хранения данных, к которой пользователь может получить доступ

*event* – *это* synchronization object(объект, дескриптор которого может быть аргументом функции wait); используется для оповещения ожидающих потоков о произошедшем событии

thread – объект внутри процесса, который может быть поставлен в очередь на выполнение

process – содержит ресурсы, необходимые для выполнения программы: виртуальное адресное пространство, исполняемый код, открытые дескрипторы системных объектов, контекст безопасности, уникальный идентификатор процесса, переменные окружения, приоритет исполнения и один или несколько потоков

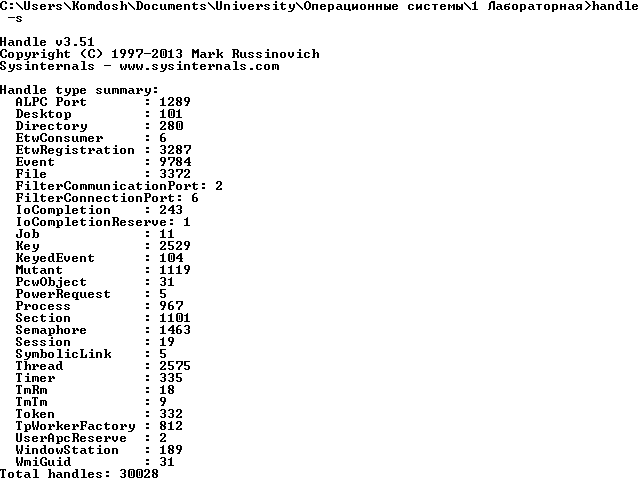
mutant – мьютекс, аналог одноместного семафора

semaphore - объект, ограничивающий количество потоков, которые могут войти в заданный участок кода

*key –* ключ регистра*.*

*section –* область памяти, которая доступна для общего доступа.

С помощью утилиты handle.exe можно вывести список всех типов объектов, используемых в системе. Всего в системе используется 30028 объектов.



Описание типов объектов (начало см. выше):

job object – позволяет группировать несколько процессов и управлять ими как одним целым.

*directory* – объект, который содержит другие объекты.

*EtwConsumer* (ETW - Event Tracing for Windows) – потребляет события из ETW

*EtwRegistration* – регистрирует события для ETW

*FilterCommunicationPort* – менеджер порта коммуникации процессов

*FilterConnectionPort* – менеджер порта соединения процессов

*IoCompletion* – порт завершения, средство, повышающее производительность приложений, часто использующих операции ввода/вывода.

*IoCompletionReserve* – резервирует место для I/O completion

*KeyedEvent* – объекты ключевых событий системы

*PcwObject(Performance Counters for Windows)* – объект счётчика производительности

*PowerRequest* – объект запроса питания

*Session* – контролируют процесс установок.

*SymbolLink* – ссылка на другие объекты

*Timer* – объект извещающий процессы с определённым промежутком

*TmRm* – менеджер транзакций

*TmIm* – менеджер транзакций

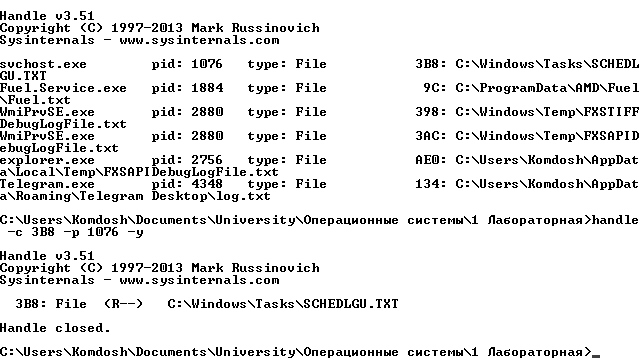
*Token* – доступные права для объекта

*TpWorkerFactory* – делает системные вызовы для фабрик и их работников, содержит имя, атрибуты, желаемую маску доступа и дескриптор безопасности, для объектов.

*UserApcReserve* – резервирует место для APC (Asynchronous Procedure Calls) асинхронных вызовов

*WmiGuide(*Windows Management Instrumentation globally unique identifier*)* – уникальный id управления инструментами Windows.

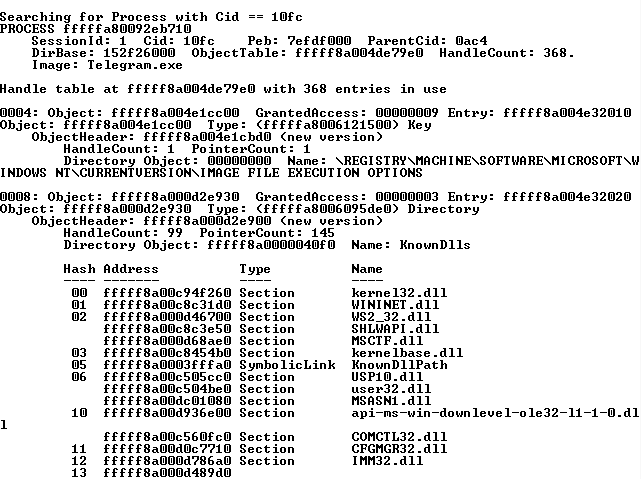
Открытые файлы в системе с расширением .txt



**Вывод**: в операционной системе Windows, каждый объект имеет свой дескриптор, однозначно определяющий тип объекта. Действия над объектами совершаются с использованием их дескрипторов.

**Задание 1.2**: Изучить хранение информации об объектах процесса

Процесс: Telegram.exe в момент написания этого задания pid=434810=10FC16. Заметим, что PID изменился, раньше он был 4180, делаем вывод, что PID не закрепляется за процессом.

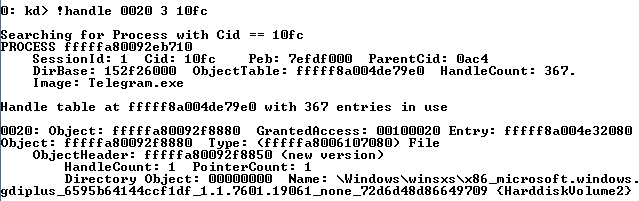


…

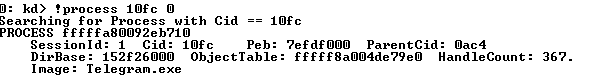


Всего 367 вхождений, что отличается от 288 в handle.exe, однако данные handle.exe были получены пару дней назад, из чего следует вывод, что количество объектов процесса, не фиксированы.

Рассмотрим подробную информацию об объекте 0x20:



Подробнее о процессе:



Продолжение работы в другой день, Telegram система дала другой PID = A5C16

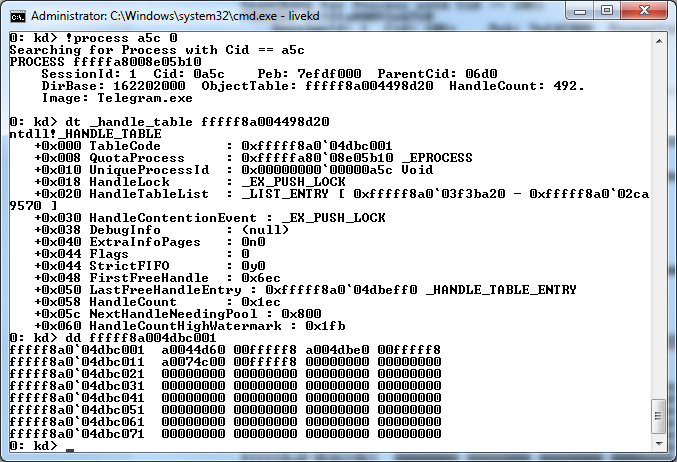
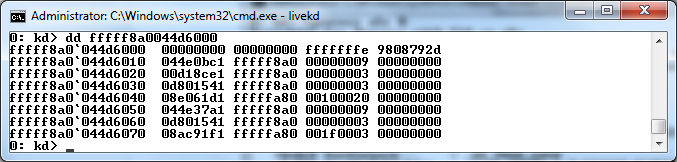
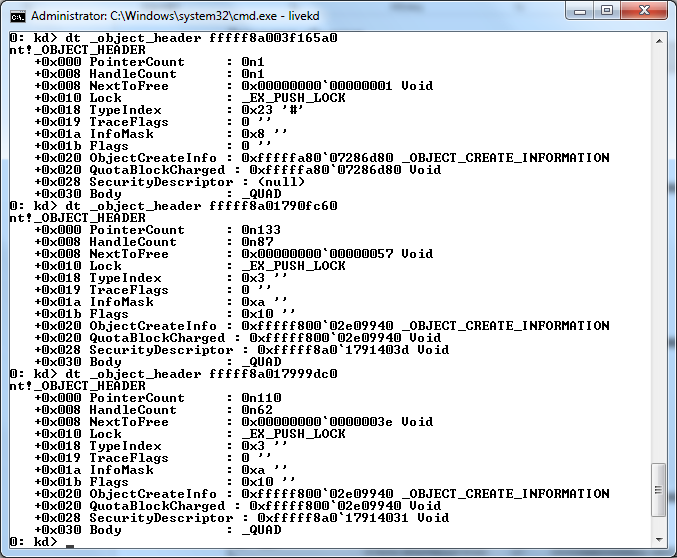


Таблица имеет два уровня, перейдём в таблицу fffff8a0044d6000:

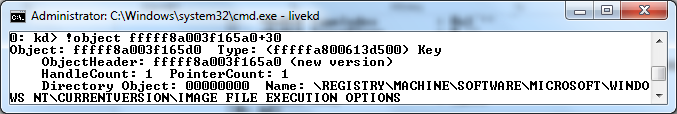


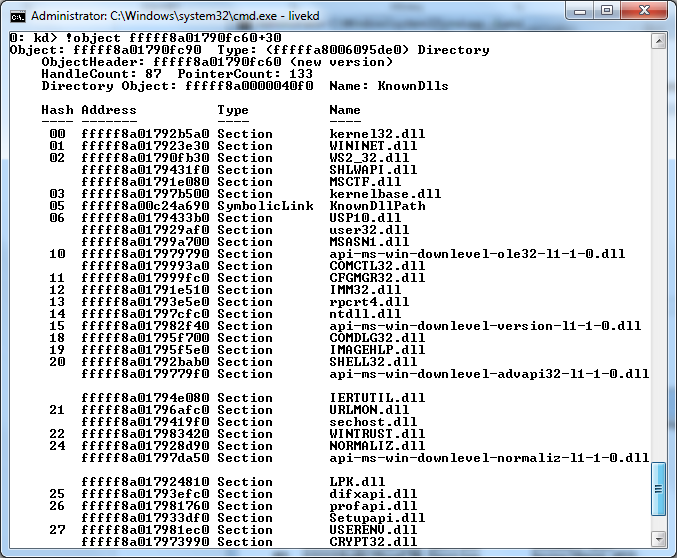
Компьютер перезагружен, адреса изменены:

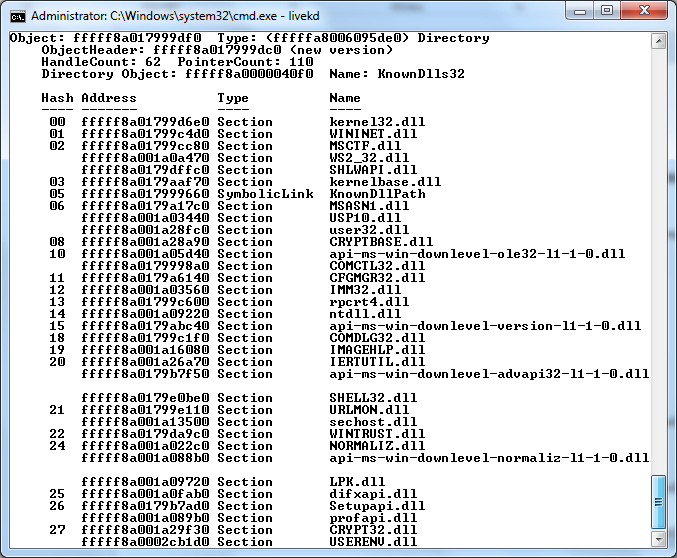
Заголовки объектов



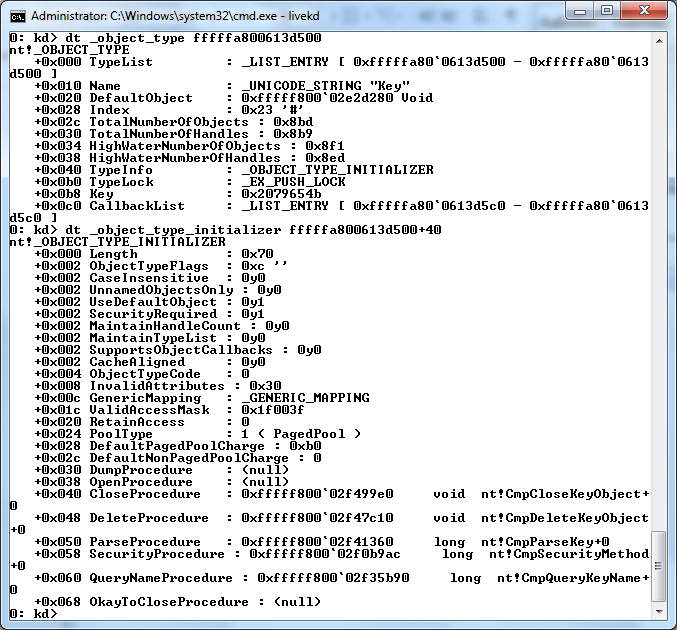
Описания объектов:

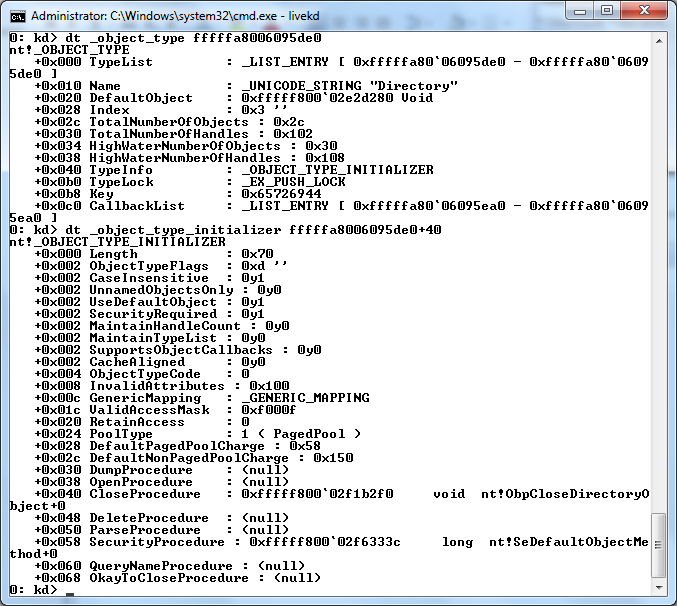






Проанализируем тип и структуру данных объектов:





Вывод: количество объектов, достаточно велико, даже для простых процессов. Действия над объектом выполняются с помощью его дескриптора, независимо от самого объекта.