|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № 9**

|  |  |
| --- | --- |
| **Тема**  “Низкоуровневое программирование под Windows/Linux. Дизассемблирование. Реверс-инжиниринг  **Студент** Жигалкин Д.Р  **Группа** ИУ7-45Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_**  **Преподаватель** Кузнецов Д.А |  |

Москва.

2020 г.

**Цель работы:**

Изучение (и модификация) дизассемблированного кода.

**Техническое задание:**

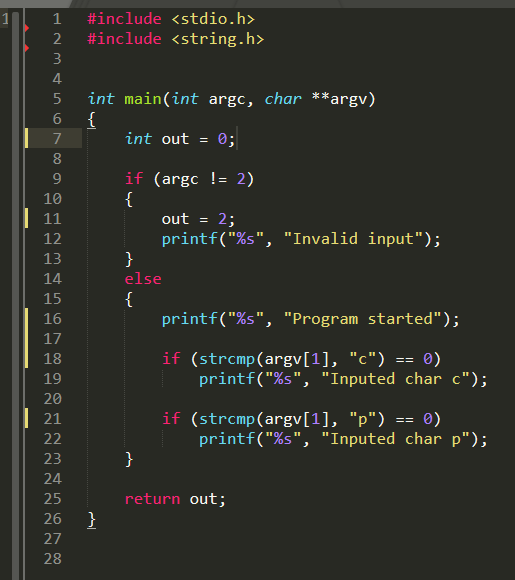
Требуется провести дизассемблирование какого-либо исполняемого файла, например, самостоятельно созданного .exe, и изучить особенности его устройства/получить какую-либо информацию, доступную только при реверс-инжиниринге.

**Теоретические сведения:**

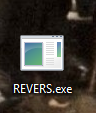
Реверс-инжиниринг (обратная разработка) — исследование готовой программы с целью понять принцип работы, поиска недокументированных возможностей или внесения изменений. Правообладатели могут запретить проведение обратной разработки и использование её результатов, чтобы исключить нарушения их исключительных прав по закону об авторском праве и патентному законодательству.

**Ход выполнения:**

Для исследования был скомпилирован следующий исходный код на языке С.



C:\Users\Dmitry\YandexDisk\Скриншоты\2020-06-01_21-25-28.png

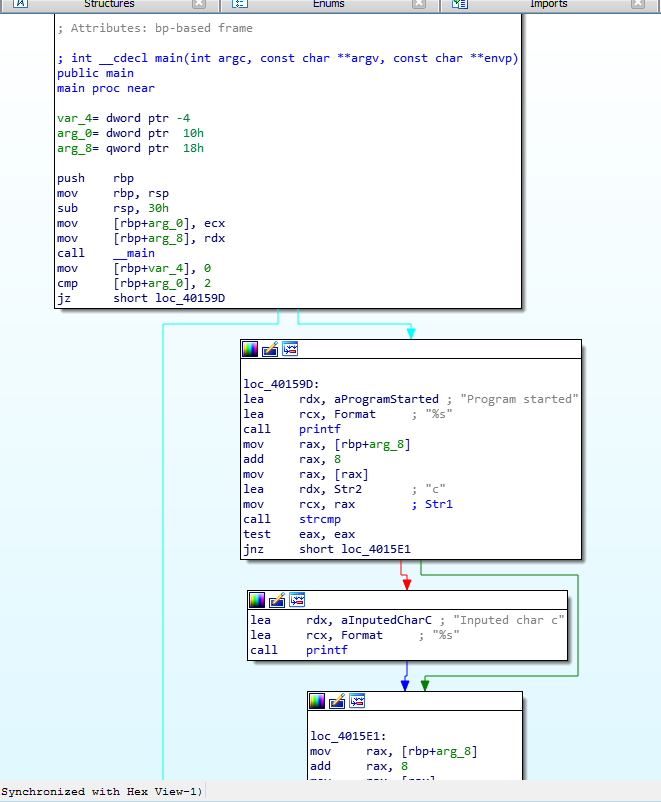


Суть программы в выводе текста, если она была запущена с определенными параметрами, например

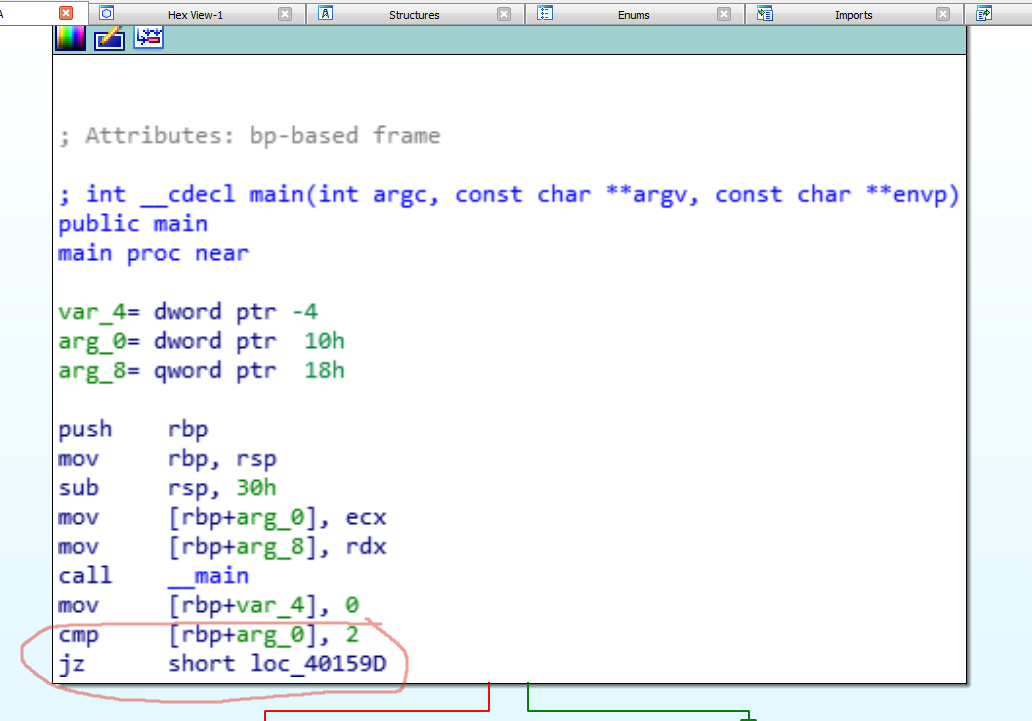
C:\Users\Dmitry\YandexDisk\Скриншоты\2020-06-01_21-31-37.png

C:\Users\Dmitry\YandexDisk\Скриншоты\2020-06-01_21-31-57.png

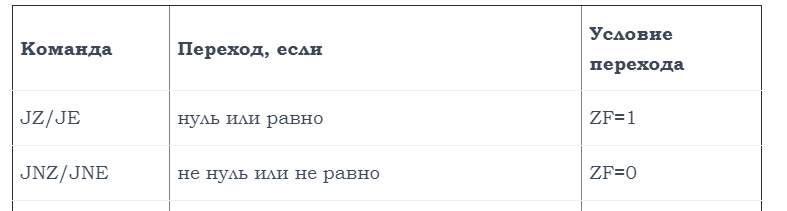
Используя программу IDA я открыл REVERS.exe

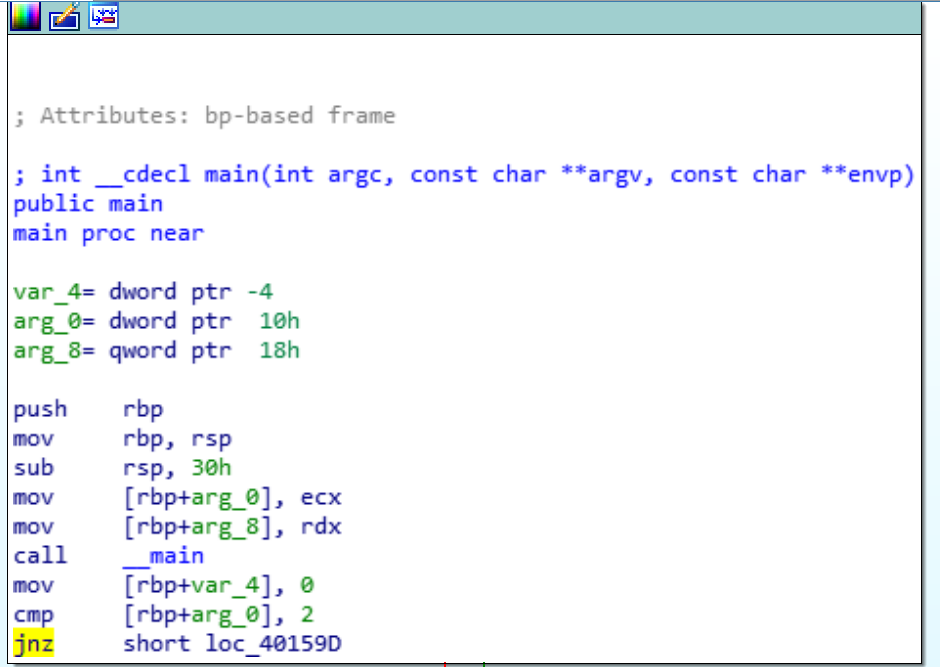


Здесь четко прослеживается функция main., где видна проверка количества переданных аргументов с числом 2 и последующий переход на соответствующую метку.

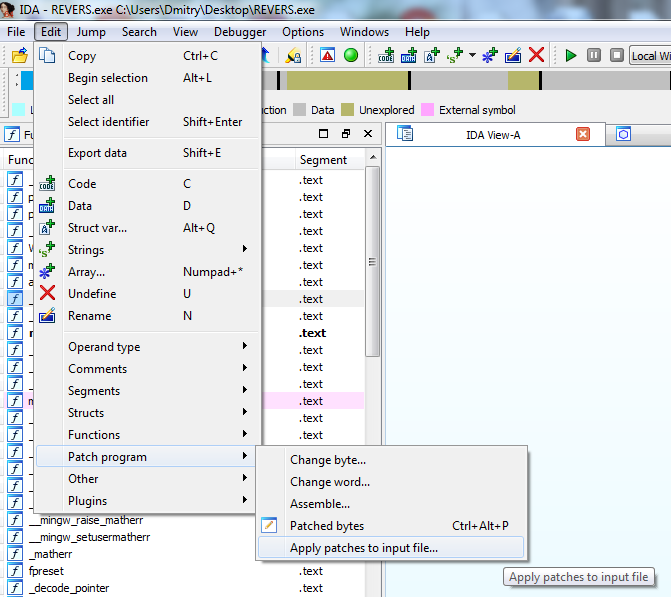


Изменим команду перехода jz на jnz, то есть сделаем противоположный эффект.





Затем применим изменения в ассемблерном коде.



Если раньше результат был такой:

C:\Users\Dmitry\YandexDisk\Скриншоты\2020-06-01_22-00-28.png

То теперь он стал таким:

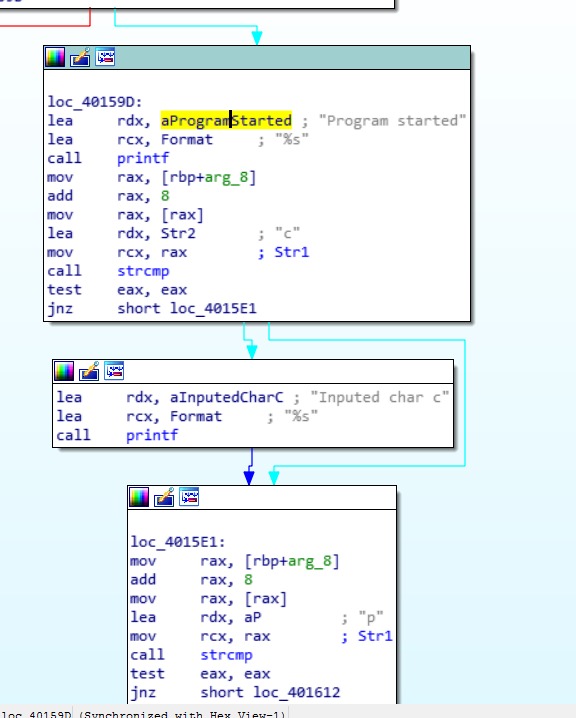
C:\Users\Dmitry\YandexDisk\Скриншоты\2020-06-01_22-00-43.png

C:\Users\Dmitry\YandexDisk\Скриншоты\2020-06-01_22-01-06.png

C:\Users\Dmitry\YandexDisk\Скриншоты\2020-06-01_22-01-39.png

То есть с помощью реверс-инжиниринга удалось изменить поведение программы.

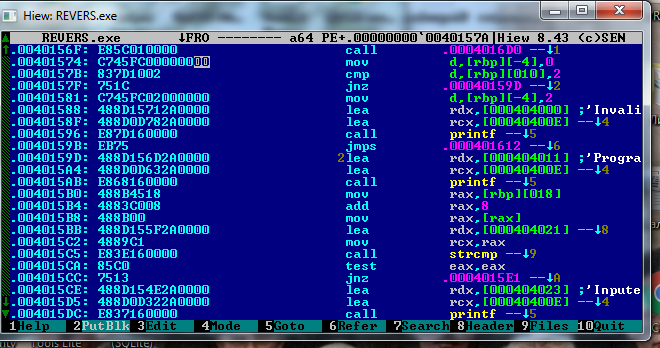
При дальнейшем изучении видно, что можно, например, изменить выводимые программой текстовые сообщения в ходе ее работы.

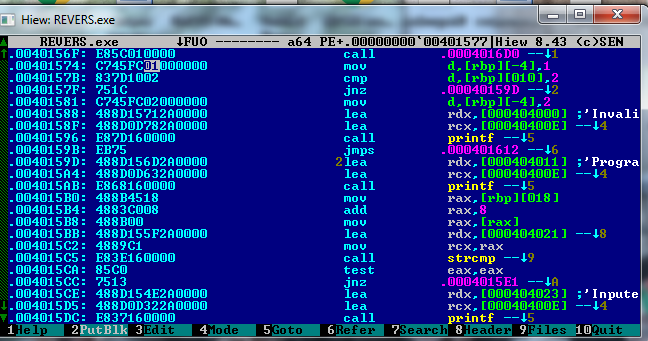


C:\Users\Dmitry\YandexDisk\Скриншоты\2020-06-01_22-04-31.png

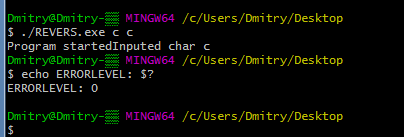
Или изменить код завершения программы с 0 на 1.

Для удобства сейчас я использую программу Hiew.

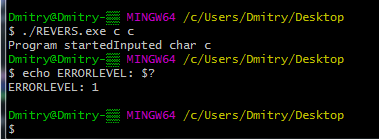




До:



После:



**Вывод**: получены и применены навыки дизассемблирования.

Изначально были планы изменить количество ядер в taskmgr.exe, о чем я узнал из одной книжки, на windows vista, но дело не пошло, но зато я нашел в taskmgr.exe интересную функцию :)

