1. **Установка ПО**
   1. Установка Linux Ubuntu

Для начала нужно установить Virtual Box что бы на нём поставить Linux.

После нажимаем создать, указываем имя, тип, версию, объем памяти,

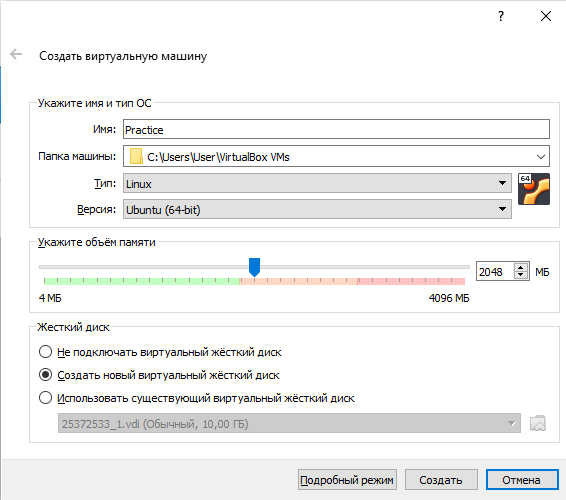


Рис. 1 «Настраиваем машину»

Создаём новый жесткий диск

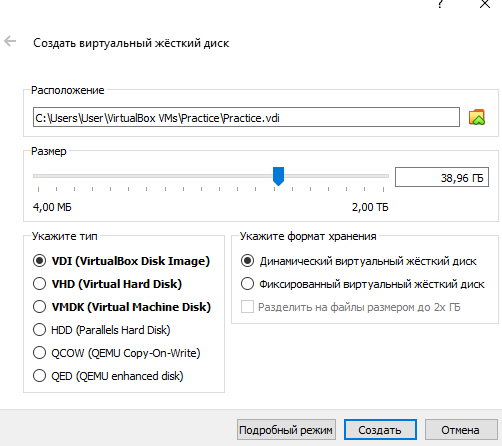


Рис. 2 «Новый жесткий диск»

Затем в настройках нужно поставить контроллер

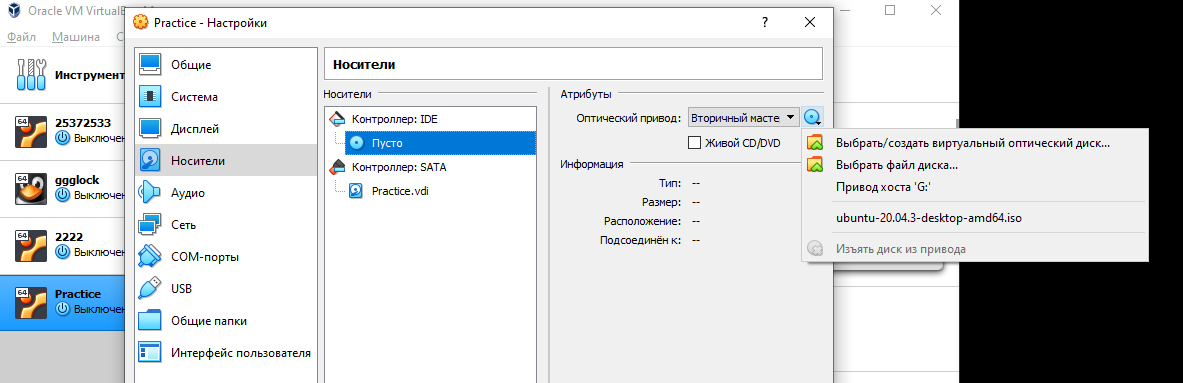


Рис. 3 «Выбираем привод диска»

Запускаем нашу машину

После загрузки можно выбрать язык и нажимаем «Install Ubuntu»

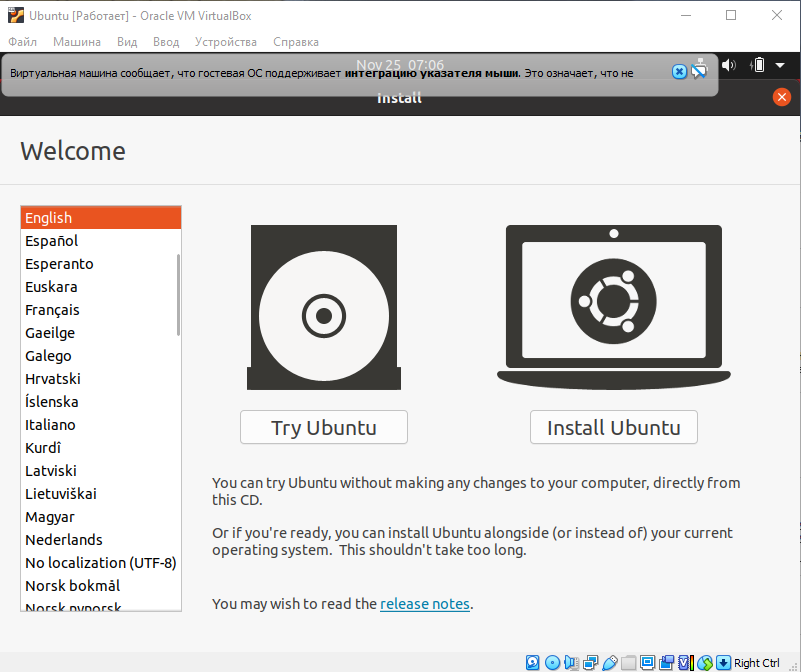


Рис. 4 «Выбираем Install Ubuntu»

Выбираем «Minimal installation» и нажимаем продолжить

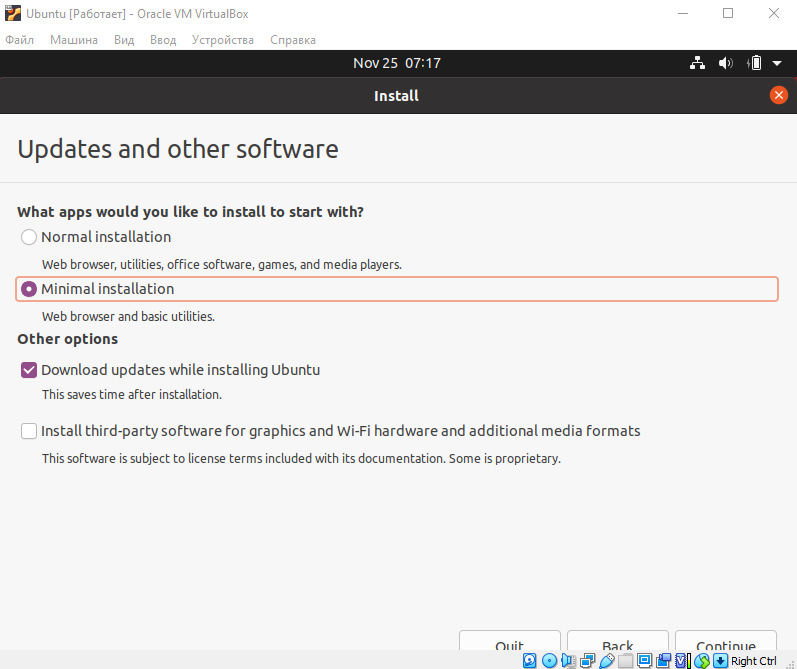


Рис.5 «Выбираем **Minimal installation**»

Далее ничего не меняем и продолжаем

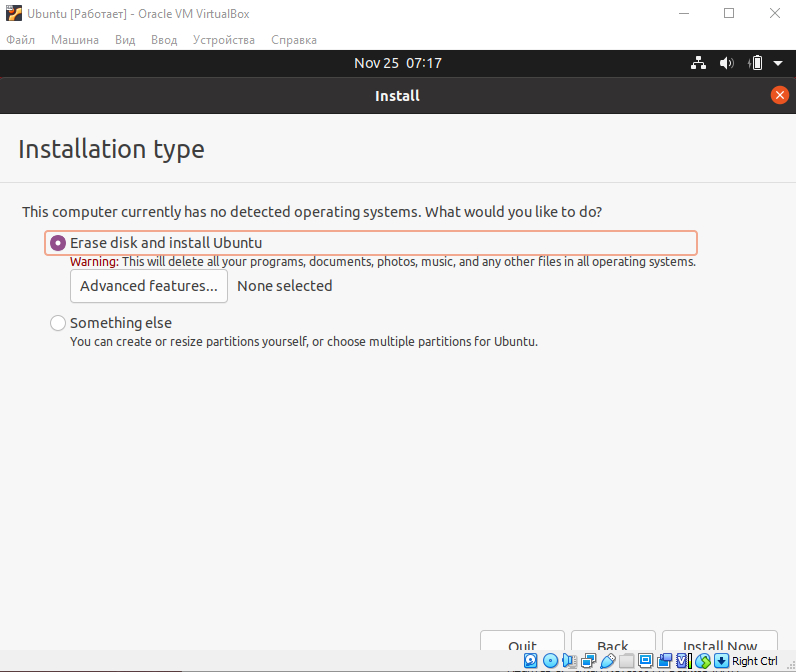


Рис.6 «Installation type»

Выбираем свой регион



Рис. 7 «Регион»

Далее вводим свой логин и пароль

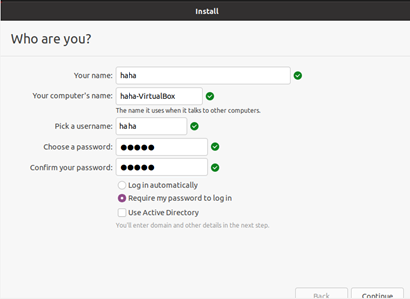


Рис. 8 «Логин и пароль»

Далее происходит установка

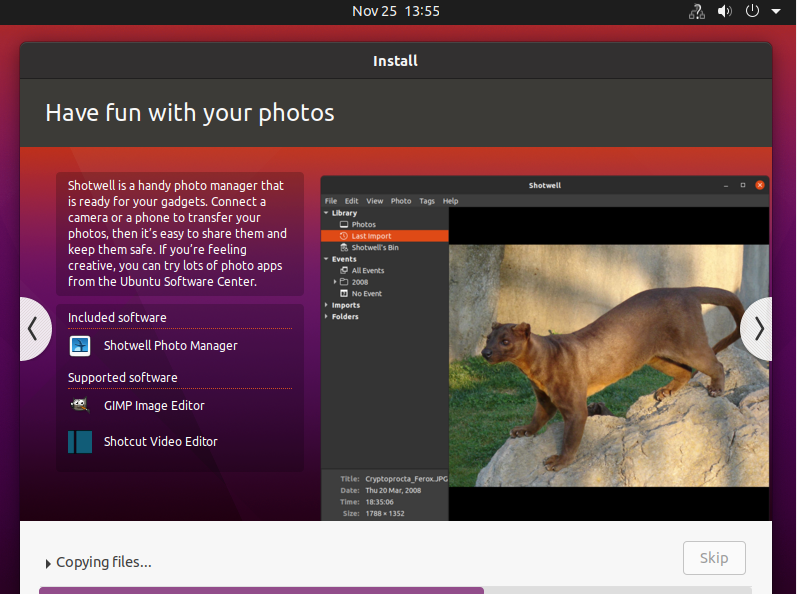
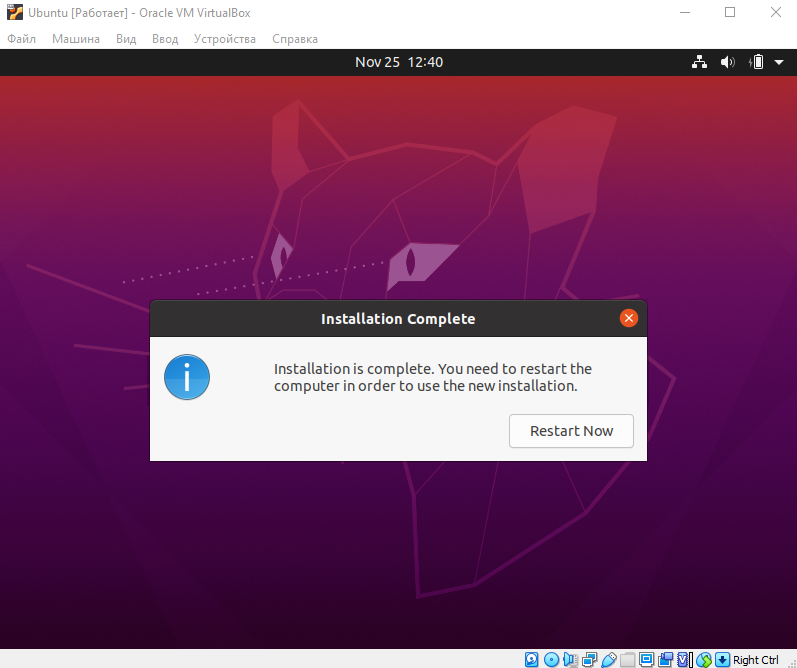


Рис. 9 «Установка»

После установки нужно перезапустить машину

Рис. 10 «Перезагрузка»



* 1. Установка остального ПО через терминал
     1. Устанавливаем компилятор GCC и все необходимые для него компоненты

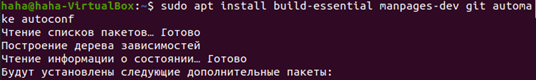


Рис. 11 «Установка GCC»

* + 1. Установка radare2

Устанавливаем необходимые пакеты для работы с radare2

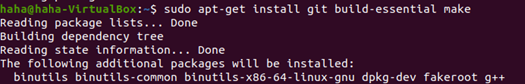


Рис. 12 «Установка пакетов»

Так как используется компьютер в колледже прописываем:

https://cdn.discordapp.com/attachments/890271394011697176/915984742325846026/erer.png

Рис. 13 «Прописываем прокси»

sudo nano /etc/wgetrc

Ищем прокси и меняем на:

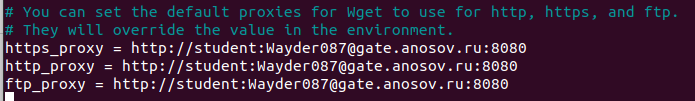
****

Рис. 14 «Меняем прокси»

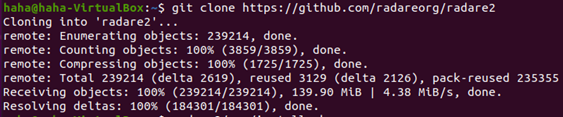


Рис. 15 «Установка radare2»

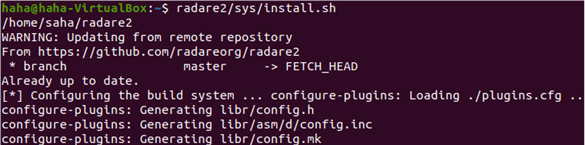


Рис. 16 «Установка radare2»

* + 1. Устанавливаем необходимые компоненты для граф.оболочки «Iaito»

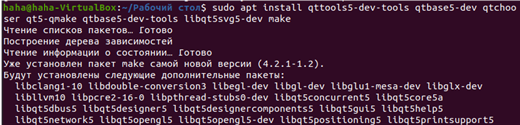


Рис. 17 «Установка компонентов»

Устанавливаем графический интерфейс «iaito» (установка может выполняться из любого места)

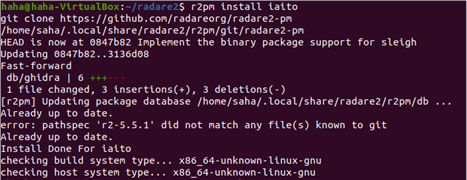


Рис. 18 «Установка iaito»

* + 1. Устанавливаем декомпилятор «r2ghidra»

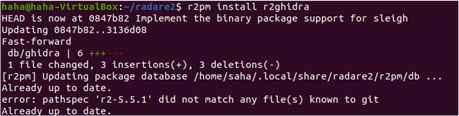


Рис. 19 «Установка r2ghidra»

# **Обратное проектирование**

* 1. Создаем файл с кодом для компиляции

https://cdn.discordapp.com/attachments/890271394011697176/915990109965393940/ererer.png

Рис. 20 «Создаём файл»

* 1. Открываем его и вставляет туда код для теста

include <stdio.h>

int main (void)

{

int x = 0;

while(x < 4){

printf ("Guten Morgen!\n");

x = x + 1;

}

return 0;

}

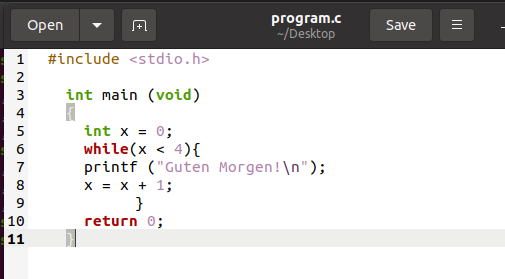
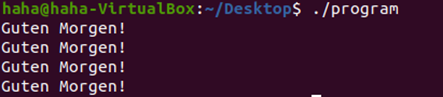


Рис. 21 «Код»

* 1. Компилируем

https://cdn.discordapp.com/attachments/890271394011697176/915990248960458862/trtrt.png

Рис. 22 «Компиляция»

* 1. Открываем и проверяем наш файл

# **2. Дизассемблирование**

Рис. 23 «Проверка файла»

2. 1. Открываем папку и открываем наш файл

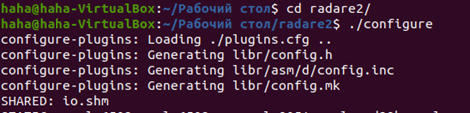


Рис. 24 «Открываем файл»

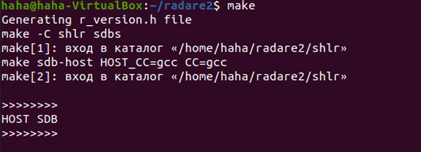


Рис. 25 «Открываем файл»

2.2. Запускаем «Iaito»

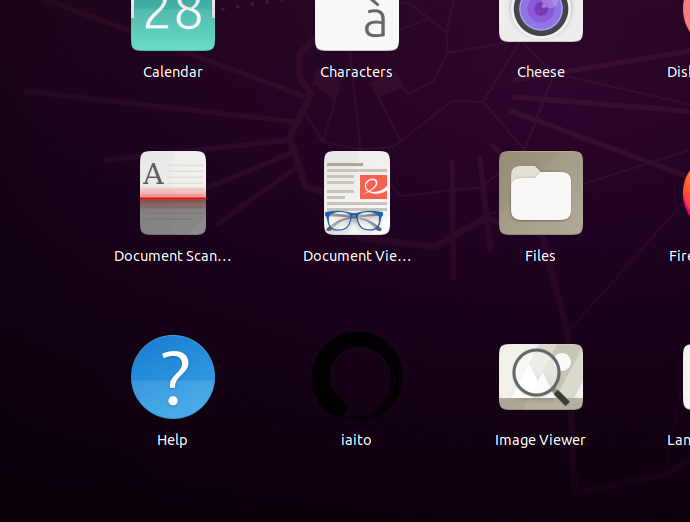


Рис. 26 «Приложение Iaito»

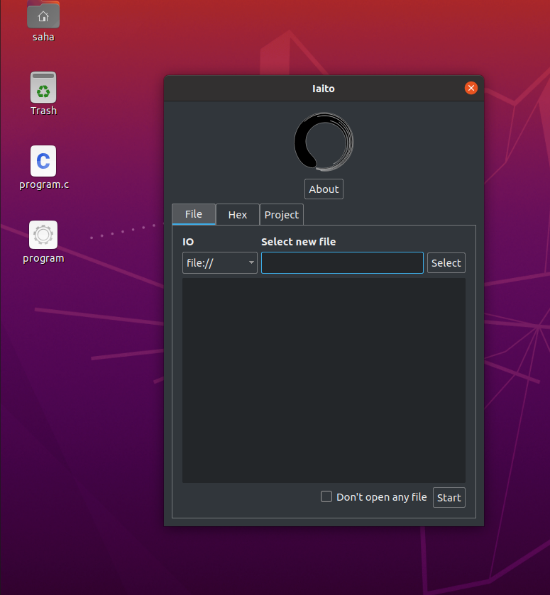


Рис. 27 «Интерфейс»

2.3. Открываем тестовый файл в «radare2». Настройки оставляем стандартные.

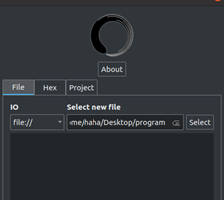


Рис. 28 «Выбираем файл»

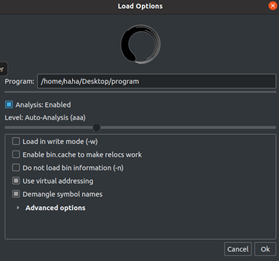


Рис. 29 «Настройки»

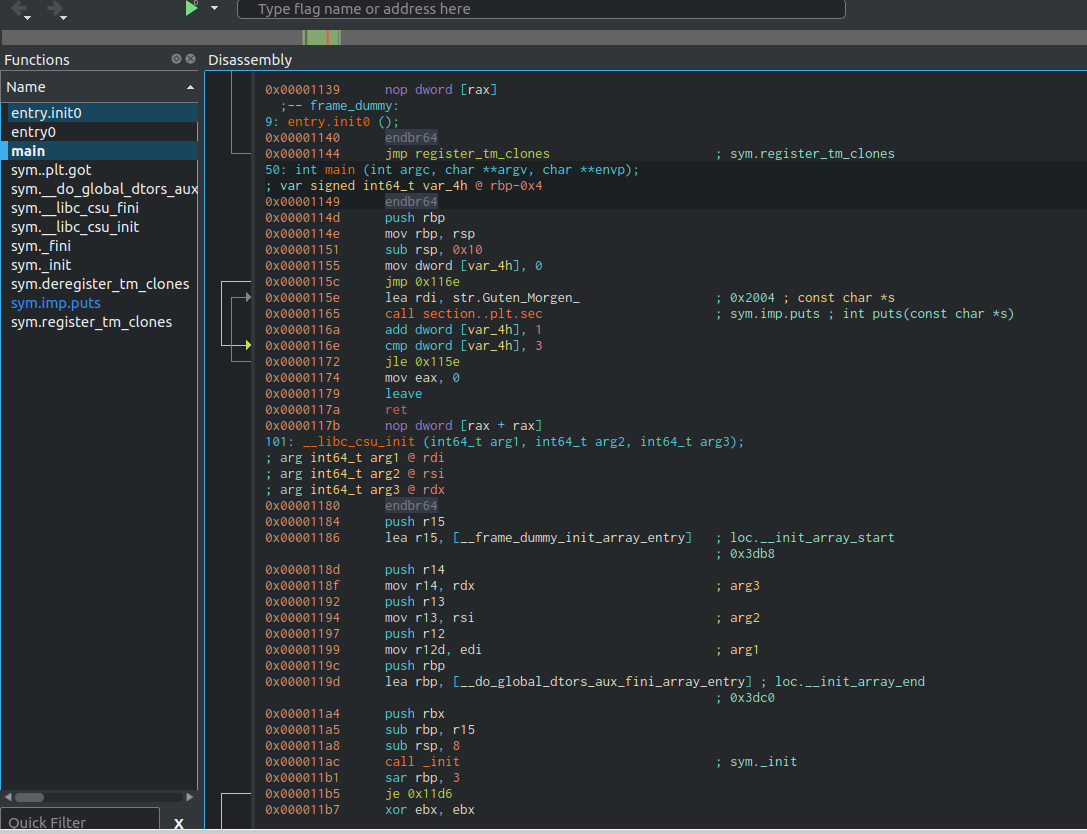
2.4. Открываем слева «main»

Рис .30 «Код»

2.5. Открываем вкладку декомпилятора. Видим, что код практически читаем, кроме переменных

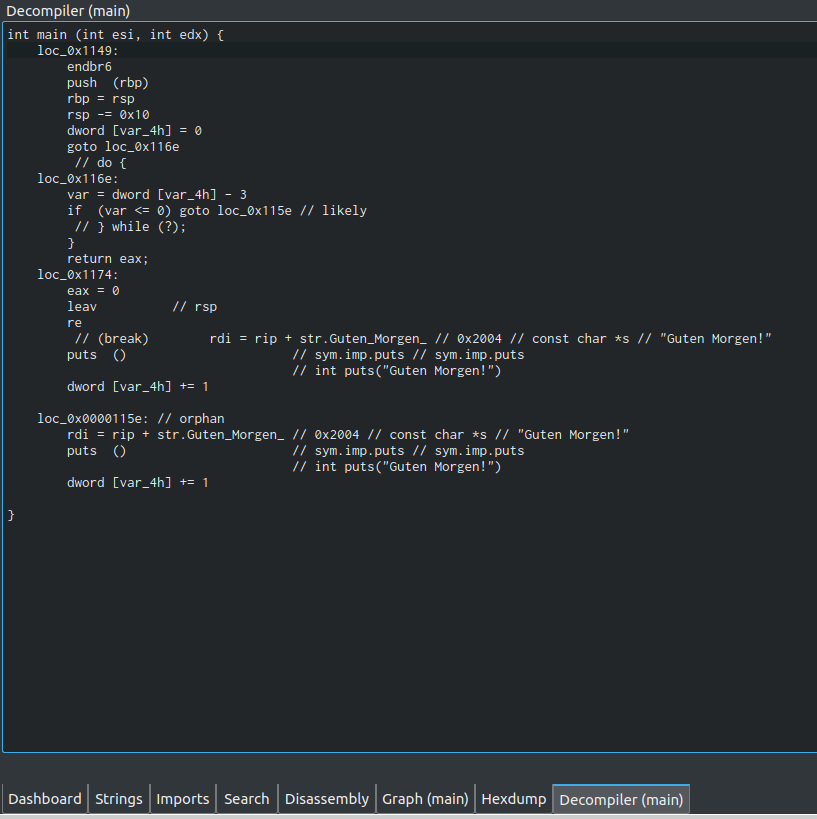


Рис. 31 «Декомпилятор»

2.6. Меняем режим работы программы на Cache mode

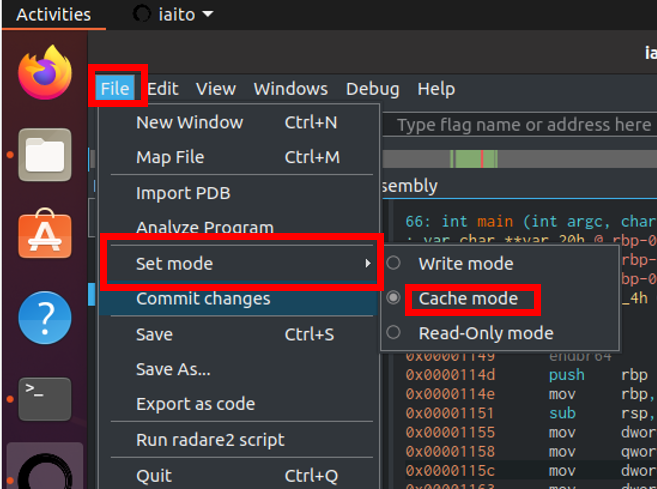


Рис. 32 «Cache mode»

2.7. Так как фраза «Guten Morgen!» выводится 4 раза, а нам необходимо увеличить вывод до 10 раз, то смотря на код, можно увидеть, что используется цикл «while» по значению переменной [var4\_h] равной 3 (т.к. While <4, а цикл начинается с 0). Выделяем цифру 3 и нажимаем «Edit» затем «Instruction» меняя значение переменное с 3 до 9.

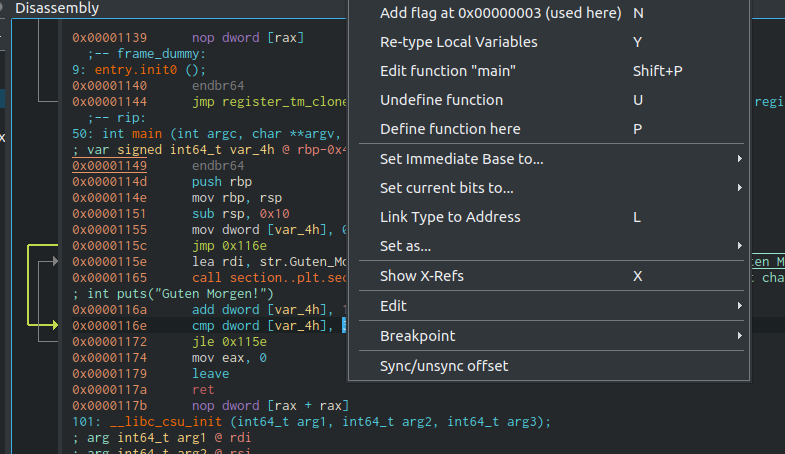


Рис. 33 «Edit>Instruction»

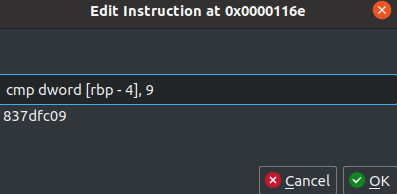
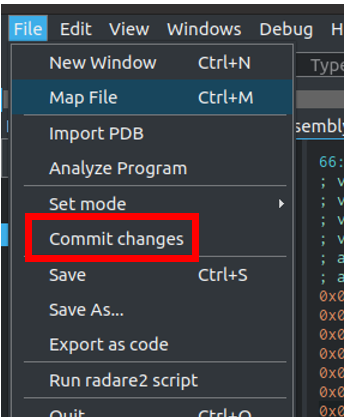


Рис. 34 «Изменяем значение»

2.8. Сохраняем наши изменения, нажав на кнопку «Commit changes»

Рис. 35 «Применяем изменения»



2.9. Запускаем файл для проверки

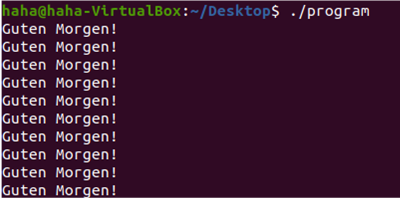


Рис. 36 «Проверка»