**Создание исходного файла**

1. Устанавливаем компилятор GCC и все необходимые для него компоненты

sudo apt install build-essential manpages-dev git automake autoconf



Рис. 1

И пойдёт установка

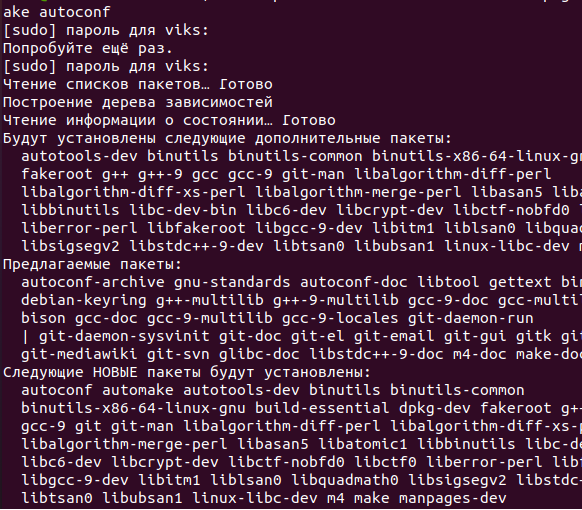


Рис. 2

1. Создаем файл с кодом для компиляции 101

touch program.с



Рис. 3

1. Открываем его и вставляет туда код для теста

[#include](https://vk.com/im?sel=320239488&st=%23include) <stdio.h>  
  
int main (void)  
{  
int x = 0;  
while(x < 5){  
printf ("Hello, World!\n");

x = x + 1;  
}  
return 0;  
}

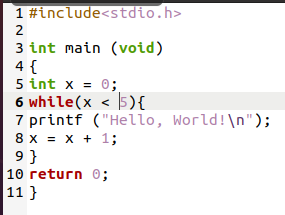


Рис. 4

1. Компилируем

gcc program.c -o program



Рис. 5

1. Открываем и проверяем наш файл

./program

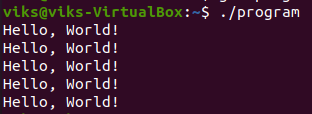


Рис. 6

**Дизассемблирование**

1. Устанавливаем необходимые пакеты для работы с radare2

sudo apt install git build-essential make

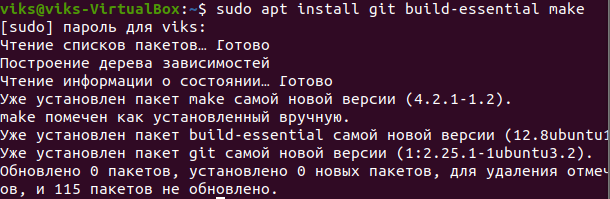


Рис. 7

1. Установка radare2:

git clone https://github.com/radareorg/radare2

radare2/sys/install.sh

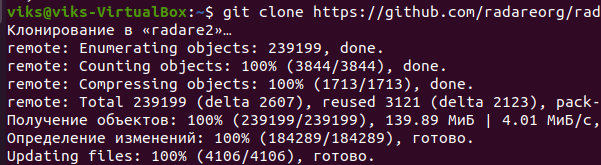


Рис. 8

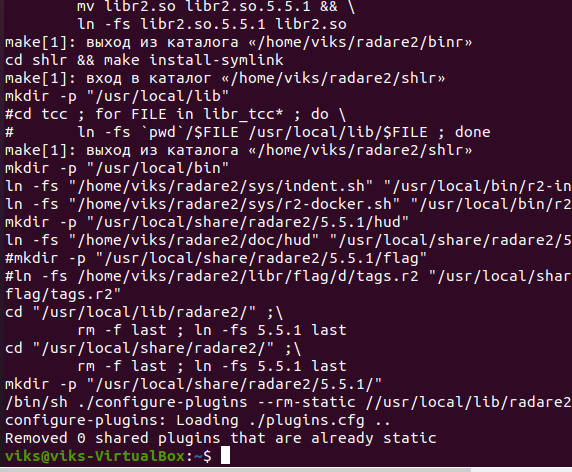


Рис. 9

1. Проверяем и обновляем radare2

r2pm update



Рис. 10

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

sudo apt install qttools5-dev-tools qtbase5-dev qtchooser qt5-qmake qtbase5-dev-tools libqt5svg5-dev make

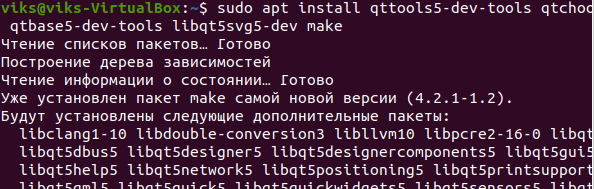


Рис. 11

1. Открываем папку и открываем файл



Рис. 12

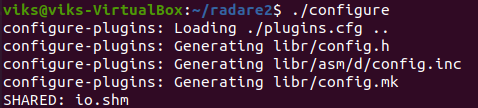


Рис. 13

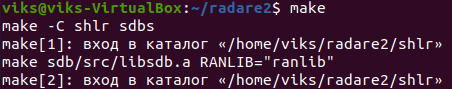


Рис. 14

1. Повторить

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

r2pm install iaito



Рис. 15

1. Устанавливаем декомпилятор «r2ghidra» (установка может выполняться из любого места)

r2pm install r2ghidra



Рис. 16

1. Запускаем «Iaito»

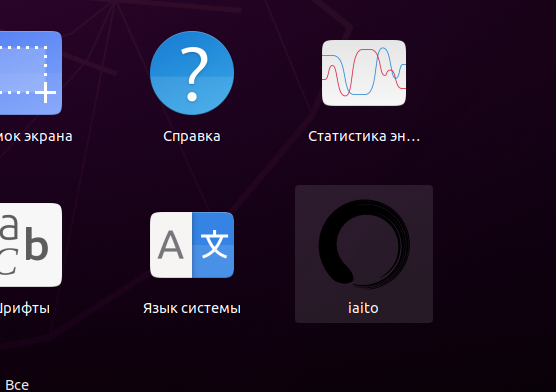


Рис. 17

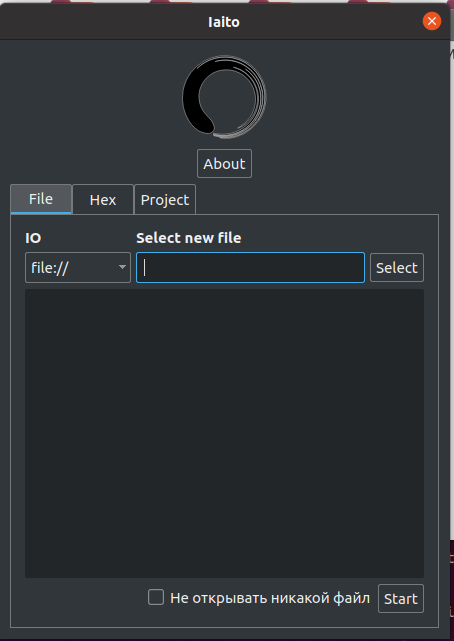


Рис. 18

1. Запускаем и проверяем тестовый файл (после можно закрыть)

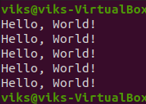


Рис. 19

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

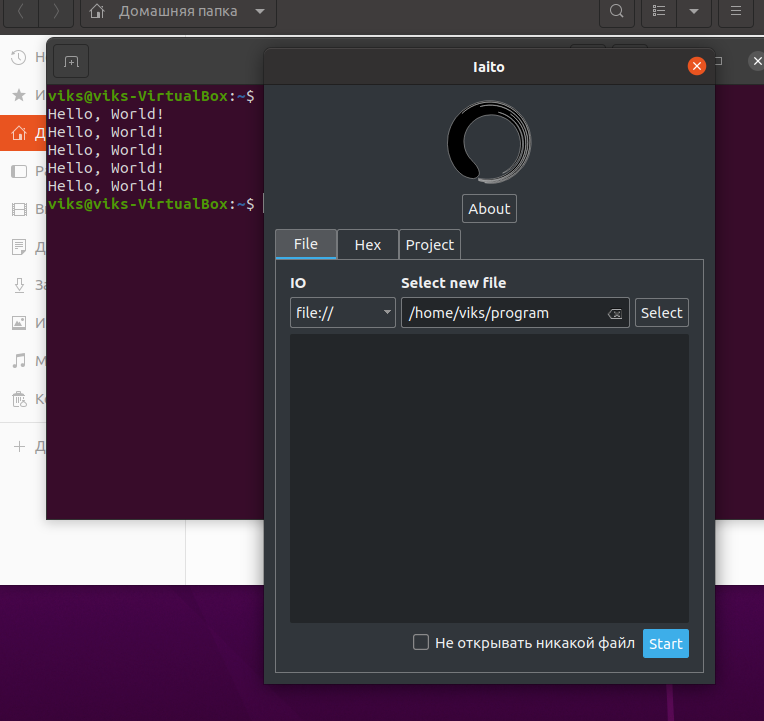


Рис. 20

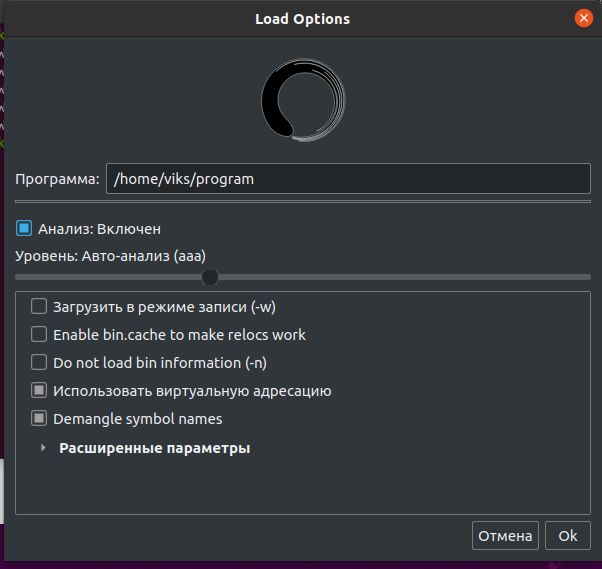


Рис. 21

1. Открываем слева «main»

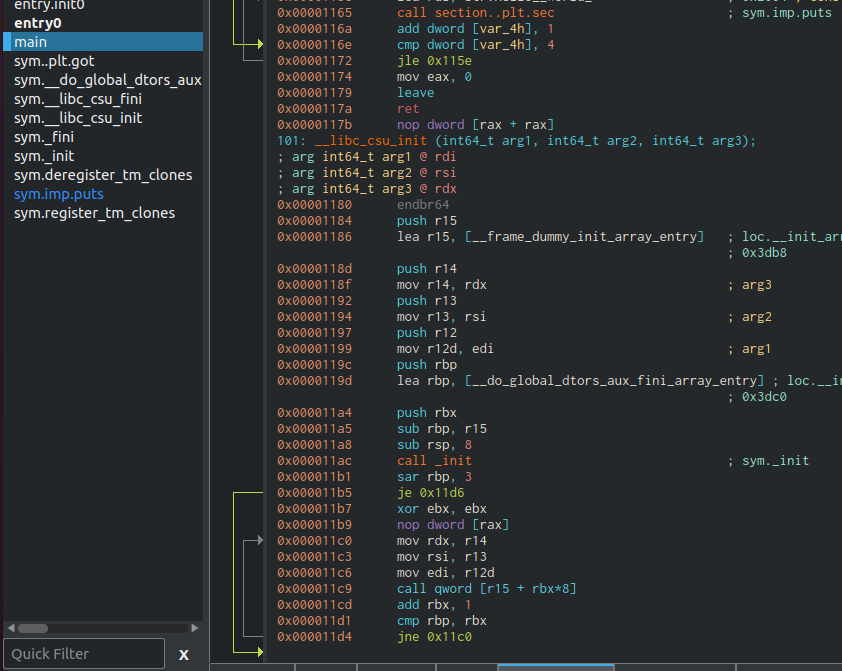


Рис. 22

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

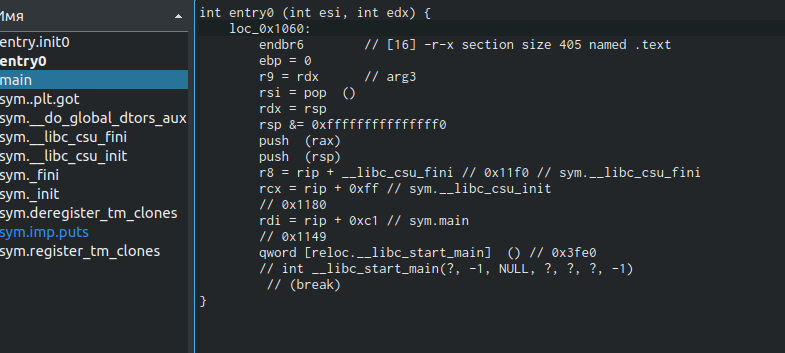


Рис. 23

1. Меняем режим работы программы (File > Set mode > Cache mode)

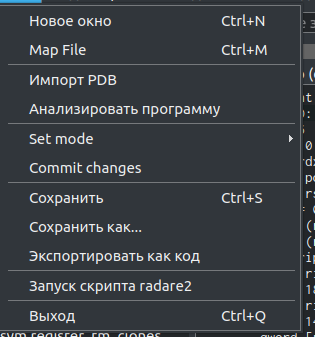


Рис. 24

1. Так как фраза «Hello world!» выводится 5 раз, а нам необходимо увеличить вывод до 10 раз, то смотря на код, можно увидеть, что используется цикл «while» по значение переменной [var4\_h] равной 4 (т.к While <=4, а цикл начинается с 0). Выделяем цифру 4 и нажимаем «Edit» меняя значение переменное с 4 до 10.



Рис. 25

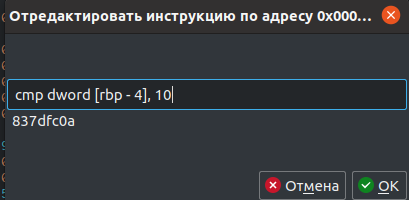


Рис. 26

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

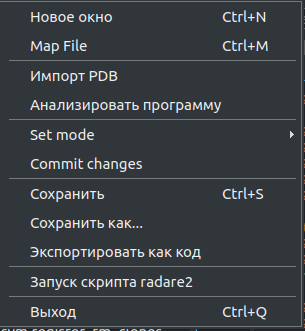


Рис. 27

1. Запускаем наш файл для проверки

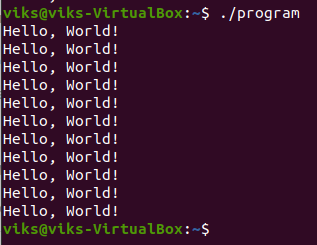


Рис. 28