**Настройка прокси на компьютерах колледжа**

**sudo nano /etc/apt/apt.conf.d/proxy**

**Прописываем**

**Acquire::http::Proxy “http://prosrv.anosov.ru:3142”;**

**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

1. Создаем файл с кодом для компиляции

* touch program.c



Рис.2

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;  
}

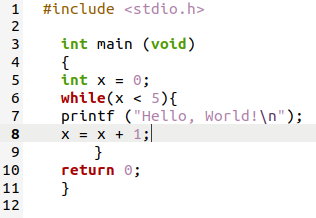


Рис.3

1. Компилируем

* gcc program.c -o program



Рис.4

1. Открываем и проверяем наш файл

* ./program

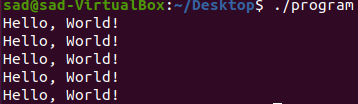


Рис.5

**Дизассемблирование**

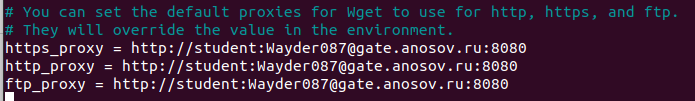
1. Устанавливаем необходимые пакеты для работы с radare2
   * sudo apt install git build-essential make
2. Установка radare2:

***Если, через компьютер в колледже, то прописываем***

**git config --global http.proxy** [**http://student:Wayder087@gate.anosov.ru:8080**](http://student:Wayder087@gate.anosov.ru:8080)

**sudo nano /etc/wgetrc**

**Ищем прокси и меняем на**

****

* + git clone https://github.com/radareorg/radare2
  + radare2/sys/install.sh

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.6

1. Открываем папку и открываем файл



Рис.7



Рис.8

1. Устанавливаем графический интерфейс «iaito» (установка может выполняться из любого места)
   * r2pm install iaito



Рис.9

1. Устанавливаем декомпилятор «r2ghidra» (установка может выполняться из любого места)
   * r2pm install r2ghidra



Рис.10

1. Запускаем «Iaito»

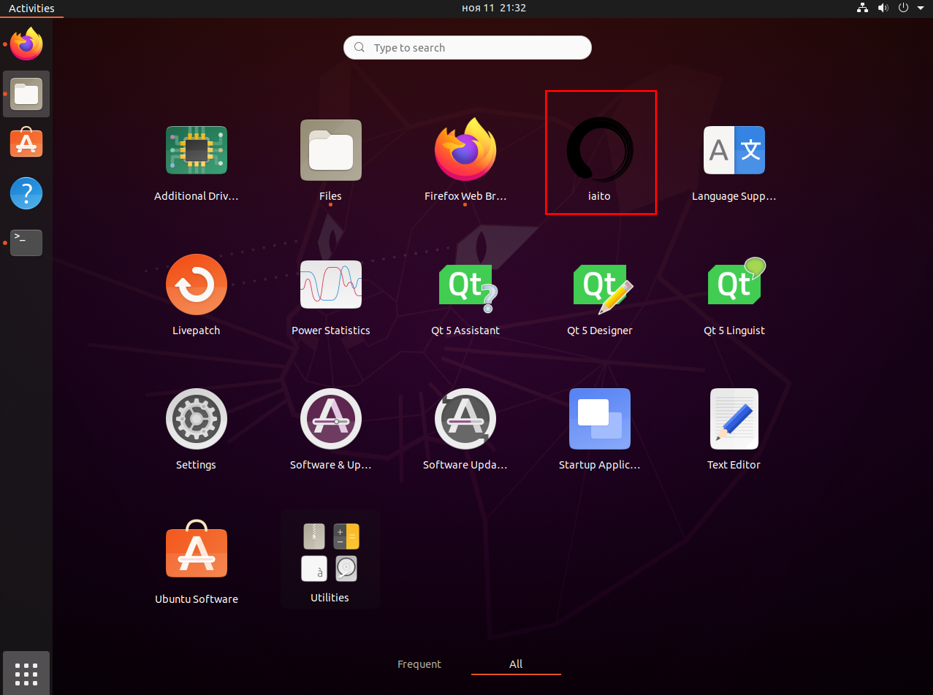


Рис.11

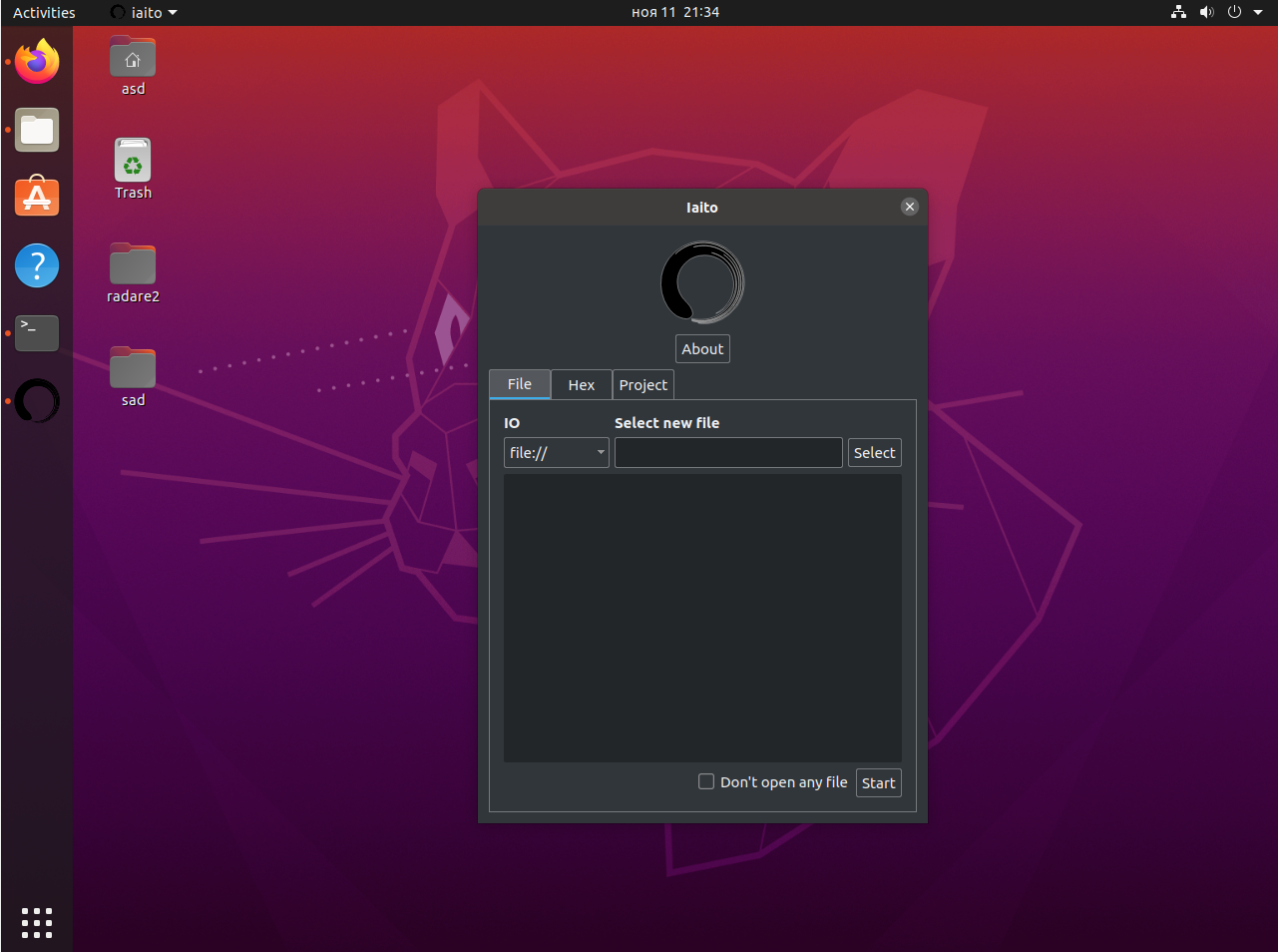


Рис.12

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

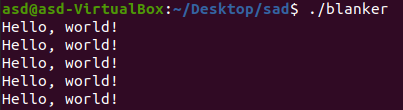


Рис.12

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

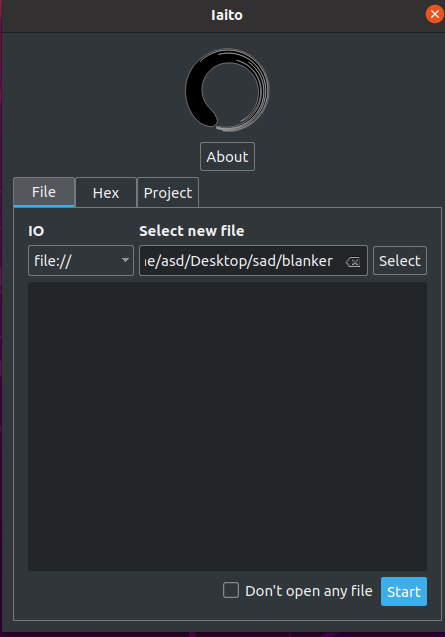


Рис.13

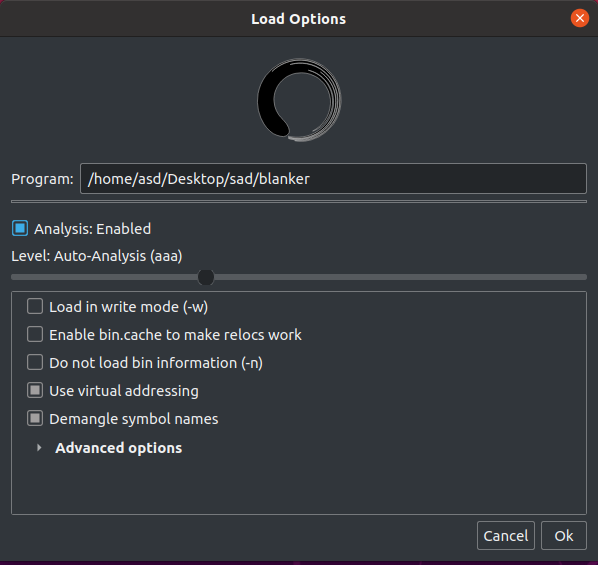


Рис.14

1. Открываем слева «main»

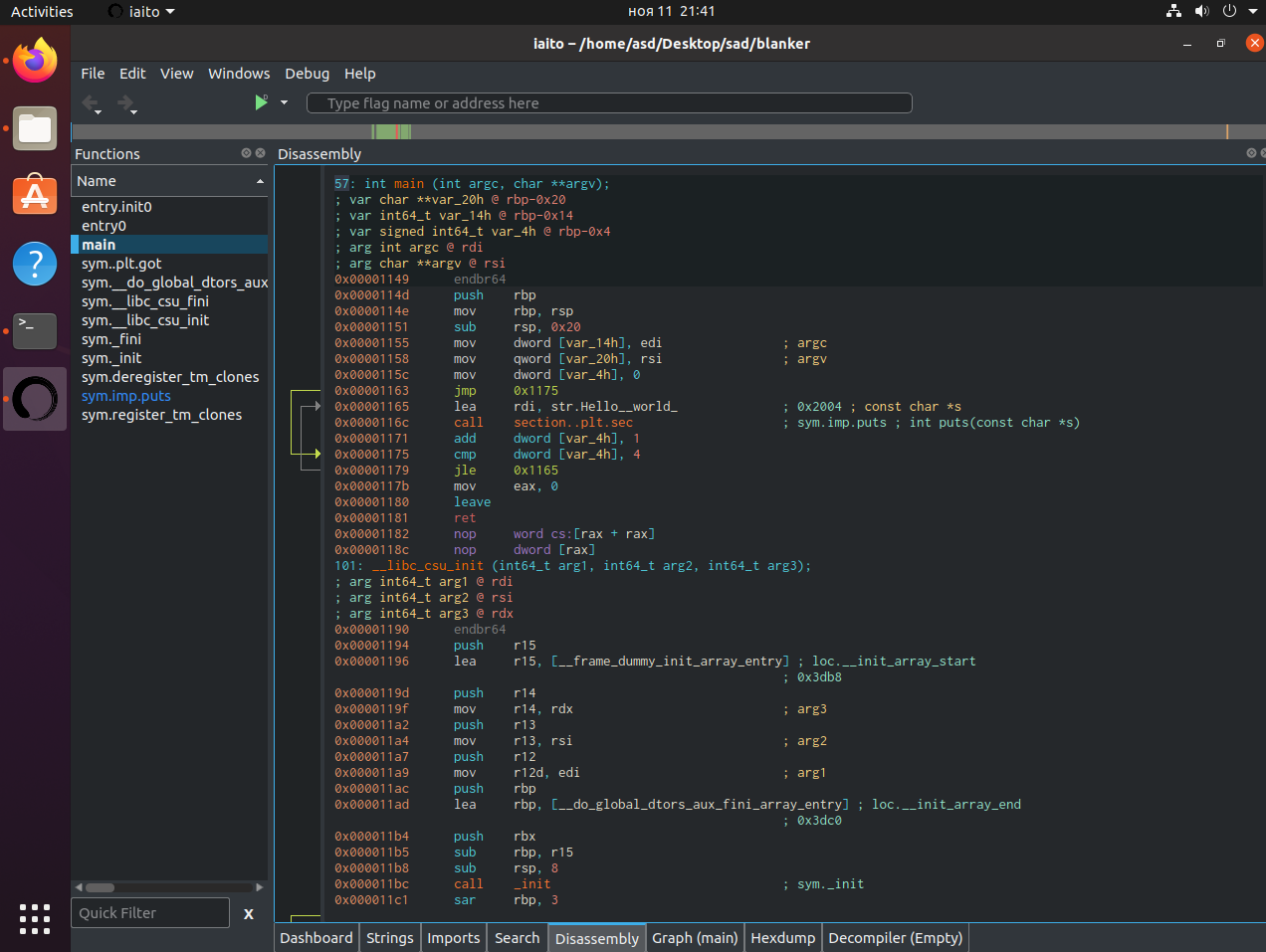


Рис.15

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

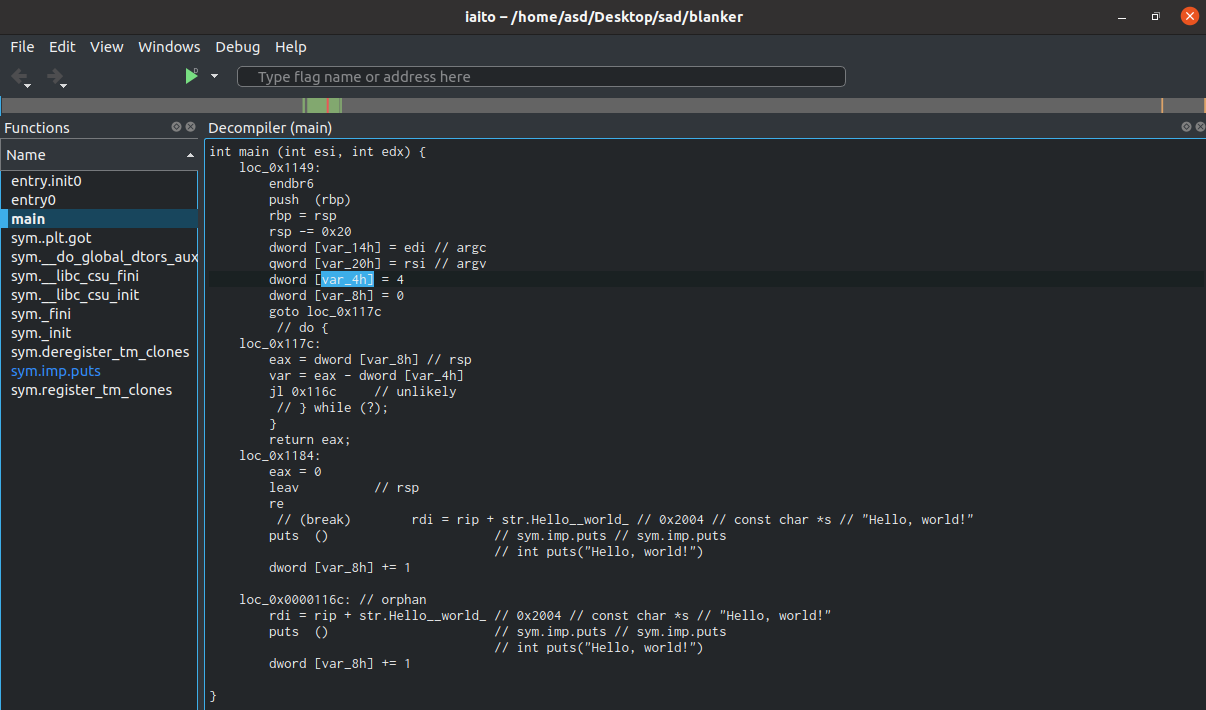


Рис.16

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

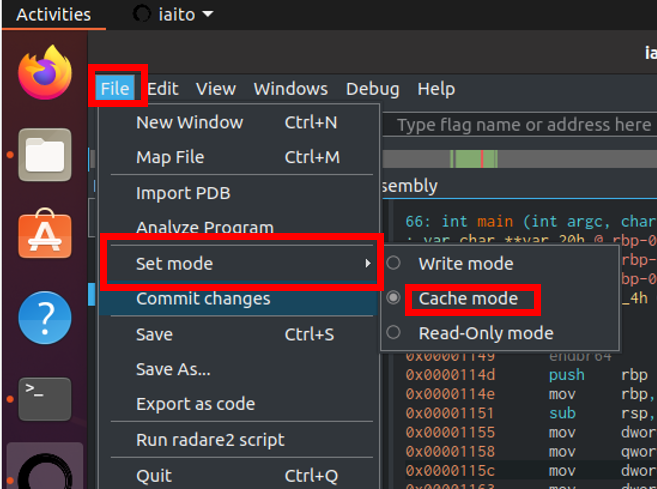


Рис.17

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.

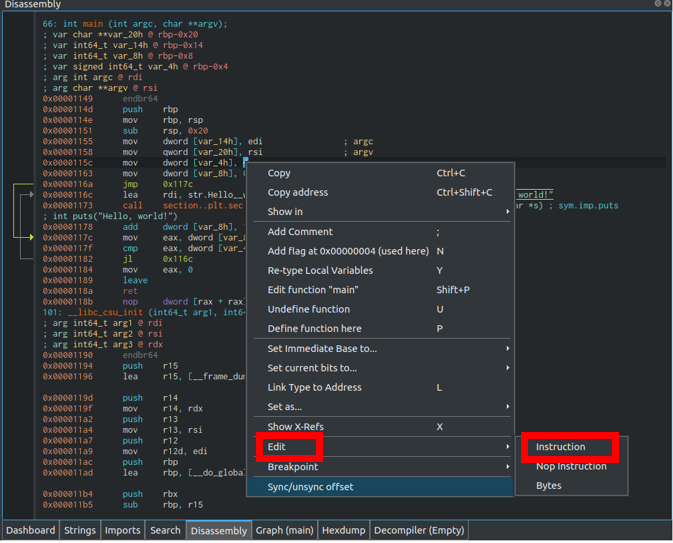


Рис.18

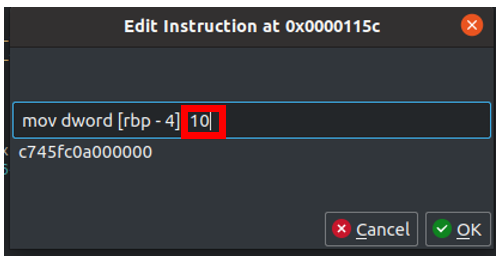


Рис.19

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

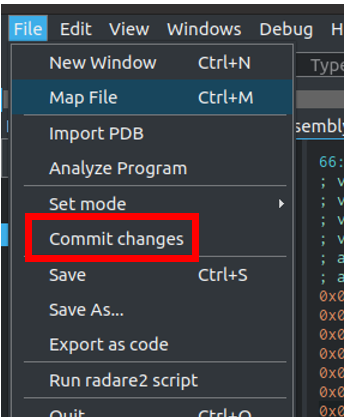


Рис.20

1. Запускаем наш файл для проверки

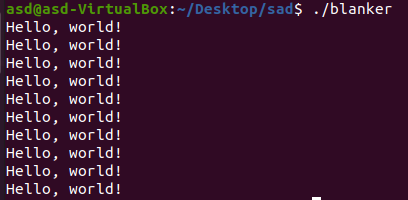


Рис.21