Министерство образования и науки Челябинской области

государственное бюджетное профессиональное

образовательное учреждение

«Златоустовский индустриальный колледж им. П.П. Аносова»

**ЗАЩИТА**

Руководитель УП ПМ.03

Преподаватель ГБОУ ЗлатИК

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ю.В.Майер

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ**

По учебной практике

Специальность: 09.02.07 «Информационные системы и программирование»

ПМ.03 «Ревьюирирование программных продуктов»

Выполнил:

Студент группы ИС-32

\_\_\_\_Новиков К.О.

\_\_\_\_\_\_\_\_\_\_\_дата

2021-2022 уч.г.

План прохождения учебной практики

по ПМ.03 «Ревьюирирование программных продуктов»

Студент Новиков К.О.

Группа ИС-32

Специальность 09.02.07 «Информационные системы и программирование»

Количество часов \_\_\_\_\_36\_\_\_\_\_\_\_\_\_\_

Руководитель практики \_\_\_Ю.В.Майер\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Сроки проведения практики\_\_\_\_29.11.2021 - 04.12.2021\_\_\_\_\_\_\_

Таблицу 1 – План прохождения практики

|  |  |  |
| --- | --- | --- |
| Этапы практики | Дата | Освоен  (да/нет) |
| Измерение производительности приложения посредством анализа использования ЦП | 29.11 |  |
| Установка ПО (Linux, компилятор GCC, radare2, iaito) для обратного проектирования | 30.11 |  |
| Обратное проектирование | 01.12 |  |
| Дизассемблирование | 02.12 |  |
| Разработать сопроводительную документацию | 03.12 |  |
| Защита практики | 04.12 |  |

**Отзыв руководителя**

Студент Новиков К.О.

Группа ИС-32

Специальность 09.02.07 «Информационные системы и программирование»

Количество часов \_\_\_\_\_36\_\_\_\_\_\_\_\_\_\_

Руководитель практики \_\_\_Ю.В.Майер\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Сроки проведения практики\_\_\_\_29.11.2021 - 04.12.2021

Место работы ГБПОУ «Златоустовский индустриальный колледж им. П.П. Аносова»

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Показатели и критерии оценивания** | **Баллы** | **Фактический балл** |
| **1** | **Структура отчета по практике** | | |
|  | Структура полностью соответствует заданию | 3 |  |
| Структура имеет несущественное несоответствие заданию | 2 |  |
| Структура существенно не соответствует заданию | 1 |  |
| **2** | **Соответствие содержания рабочей теме,цели,задачам** |  |  |
|  | Полное соответствие | 3 |  |
| Частичное соответствие | 2 |  |
| Низкая степень соответствия | 1 |  |
| **3** | **Полнота раскрытия темы** | | |
|  | Тема раскрыта полностью, приведены необходимые расчеты, пояснения, аргументы, сделаны выводы | 3 |  |
| Тема раскрыта полностью, однако приведены не все необходимые расчеты, пояснения и аргументы | 2 |  |
| Тема раскрыта частично, нет необходимых расчетов, пояснений, аргументов, не сделаны выводы | 1 |  |
| **4** | **Логика изложения материала** | | |
|  | Все структурные элементы организованы в систему, прослеживается логика в раскрытии темы | 3 |  |
| Все структурные элементы организованы в систему, логика в раскрытии темы частично нарушена | 2 |  |
| Все структурные элементы организованы в систему, но нет логики в раскрытии темы | 1 |  |
| **5** | **Соблюдение требований ГОСТ к оформлению ПЗ** | | |
|  | Требования ГОСТ соблюдены полностью | 3 |  |
| Имеются незначительные отклонения от ГОСТ | 2 |  |
| Есть существенные нарушения требований ГОСТ | 1 |  |
| **6** | **Практическая часть** | | |
|  | Выполнена в соответствии с требованиями, без отклонений от нормативов | 3 |  |
| Имеется несущественное отклонение от нормативов | 2 |  |
| Имеется существенное отклонение от нормативов | 1 |  |
| ИТОГО | | |  |

Критерии оценивания:

18-15 б. – «отлично»;

11-14 б. – «хорошо»;

8-10 б. – «удовлетворительно»;

Меньше 8 б. работа не оценивается.

Подпись руководителя:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ю.В.Майер

\_\_\_\_\_\_\_\_\_\_\_\_\_дата

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc88743331)

[1.Термины и определения 5](#_Toc88743332)

[2.Функциональные требования 6](#_Toc88743333)

[3.Измерение производительности приложения посредством анализа использования ЦП 7](#_Toc88743334)

[5.Дизассемблирование 14](#_Toc88743335)

[6.Обратное проектирование 15](#_Toc88743336)

[Заключение 18](#_Toc88743337)

[Литература 19](#_Toc88743338)

**ВВЕДЕНИЕ**

Процессы разработки, приобретения и внедрения сложных систем, к которым относятся в частности программные комплексы, должны находится под жестким управленческим контролем. В настоящее время практически во всех организациях обеспечивается контроль важнейших характеристик, связанных с производством и использованием программных продуктов, таких как время, финансовые средства, ресурсы и т.п. Однако в большинстве случаев вне пределов сферы контроля оказывается наиболее важная характеристика программных продуктов, ради которой, собственно и осуществляются затраты времени, финансовых средств и ресурсов – это качество продукта, поскольку «невозможно контролировать то, что нельзя измерить» (“You cannot control what you cannot measure”).

Дизассемблирование **–** преобразованиепрограммы на машинном языке к ее ассемблерному представлению. Декомпиляция – получение кода языка высокого уровня из программы на машинном языке или ассемблере.

Под *анализом потоков данных* понимают совокупность задач, нацеленных на выяснение некоторых глобальных свойств программы, то есть извлечение информации о поведении тех или иных конструкций в некотором контексте.

Основным результатом деятельности группы разработчиков являются не диаграммы, а программное обеспечение, поэтому модели и основанные на них реализации должны соответствовать друг другу с минимальными затратами по поддержанию синхронизации между ними. Чаще всего разработанные модели преобразуются в программный код. Хотя UML не определяет конкретного способа отображения на какой-либо объектно-ориентированный язык, он проектировался с учетом этого требования. В наибольшей степени это относится к диаграммам классов, содержание которых без труда отображается на такие известные объектно-ориентированные языки программирования, как Java, C++, ObjectPascal, Visual Basic и др.

*Прямым проектированием*(Forward engineering) называется процесс преобразования модели в код путем отображения на некоторый язык реализации.

*Обратным проектированием*(Reverse engineering) называется процесс преобразования в модель кода, записанного на каком-либо языке программирования.

Цель учебной практики: Научиться выполнять прямое и обратное проектирование ПО, овладеть навыками приостановки и возобновления работы с рабочими задачами иинспекцией кода в Visual Studio, совместно работать над проектом в системе контроля версий GIT.

В ходе учебной практики для достижения цели, ставятся задачи:

* измерить производительность приложения посредством анализа использования ЦП,
* установить ПО (Linux, компилятор GCC, radare2, iaito) для обратного проектирования,
* выполнить обратное проектирование,
* выполнить дизассемблирование.

# **1. Список терминов и определений**

**Пользователь**— человек, который имеет, имел, или, возможно, будет иметь доступ в систему для совершения операций.

**Операция** — совокупность действий, составляющих содержание одного акта бизнес-деятельности. Операция должна соответствовать требованиям ACID (Atomicity, Consistency, Isolation, Durability). Совокупность операций одного модуля представляет интерфейс взаимодействия клиент-сервер этого модуля.

**Центральный процессор** (ЦП) – главная часть аппаратного обеспечения компьютера или программируемого логического контроллера. Иногда называют микропроцессором или просто процессором.

**Точка останова** – это преднамеренное прерывание выполнения программы, при котором выполняется вызов отладчика.

**Прокси** – это удаленный компьютер, который, при подключении к нему вашей машины, становится посредником для выхода абонента в интернет. Прокси передает все запросы программ абонента в сеть, и, получив ответ, отправляет его обратно абоненту.

**Репозиторий**(от англ. **repository** – склад, хранилище) – это профессиональный участник рынка ценных бумаг, осуществляющий ведение реестров договоров РЕПО и договоров с деривативами, заключенных на организованных торгах.

**GitHub** — это платформа, хранящая различные Git-репозитории на своих многочисленных серверах. Также GitHub называют крупнейшим веб-сервисом для хостинга и совместной разработки IT-проектов.

**Дизассемблирование** – это получение из исполняемого кода программы код на языке ассемблера.

**Дизассемблер** - программа, осуществляющая дизассемблирова­ние.

**Компилятор** — программа, переводящая текст, написанный на языке программирования, в набор машинных кодов.

**Декомпиляция** – это процесс восстановления исходного кода программы из машинного кода.

**Декомпилятор** – это программа, которая может совершить процесс декомпиляции.

**Обратное проектирование** – исследование некоторого готового устройства или программы, а также документации на него с целью понять принцип его работы; например, чтобы обнаружить недокументированные возможности, сделать изменение или воспроизвести устройство, программу или иной объект с аналогичными функциями, но без прямого копирования.

# **2.Функциональные требования**

ПО, используемое в ходе выполнения практики:

* Visual Studio 2019
* Oracle VM Virtual Box – программа для виртуализации ОС, в которой нужно создать виртуальную ОС Linux с дистрибутивом Ubuntu 64x
* Компилятор GCC - компилятор для различных языков программирования, используемый для свободных Unix-подобных операционных систем.
* Radare2 - свободный кроссплатформенный фреймворк для реверс-инжиниринга, который используется при реверсе, отладке вредоносного ПО и прошивок.
* Iaito - это одна из оболочек Radare2, предназначенная для платформы Linux и устанавливаемая при помощи терминала с веб-сайта GitHub.

Декомпилятор r2ghidra

# **3.Измерение производительности приложения посредством анализа использования ЦП**

3.1. Сбор данных профилирования

3.1.1. Запускаем Visual Studio и открываем проект через локальную папку

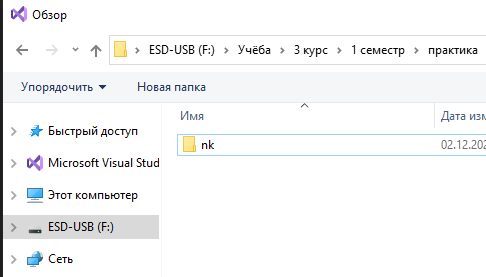


Рис.1. «Выбор проекта»

3.1.2. В коде ставим точки останова, чтобы проверить загрузку ЦП

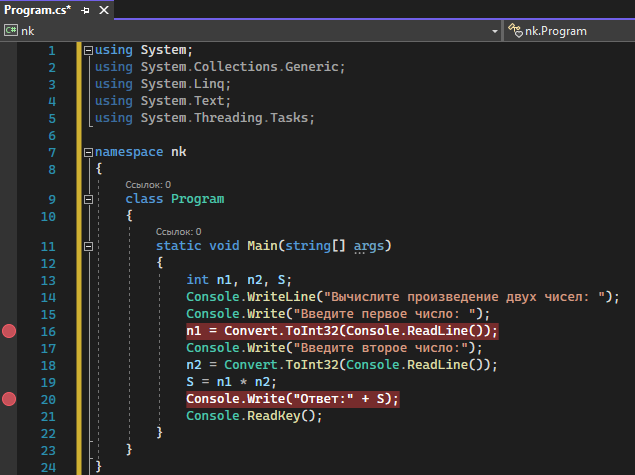


Рис.2. «Точки останова»

3.1.2. Открываем окно диагностики

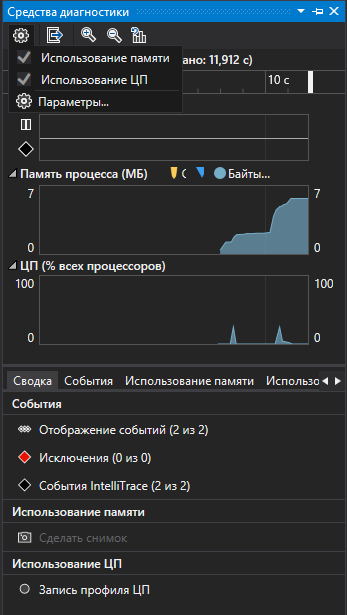


Рис.3. «Окно диагностики»

3.1.3. Начинаем отладку

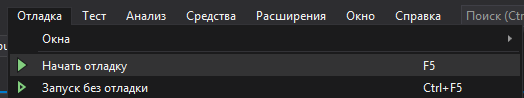


Рис.4. «Начало отладки»

3.1.4. Сбор данных о загрузке ЦП

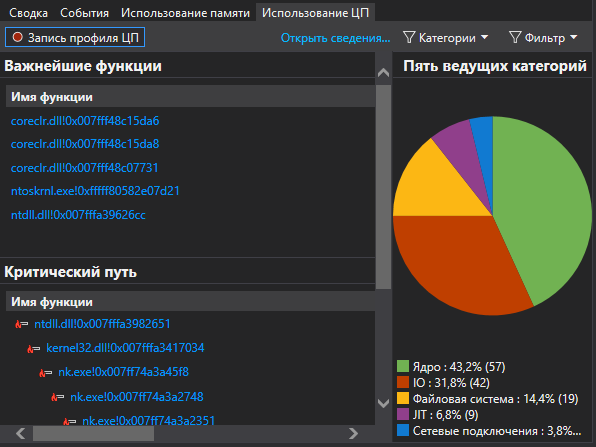


Рис.5. «Загрузка ЦП»

3.1.5. Загрузка ЦП от первой до второй точки останова

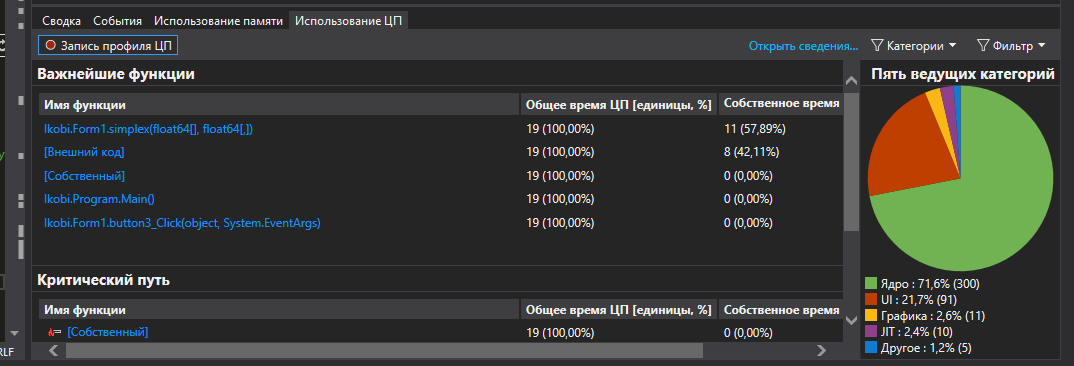


Рис.6. «Загрузка ЦП до точки останова»

**3.2. Анализ данных о загрузке ЦП**

Переход к функции вызывающий/вызываемый



Рис.7. «Функции вызывающий/вызываемый»

3.2.1. Открываем дерево вызовов

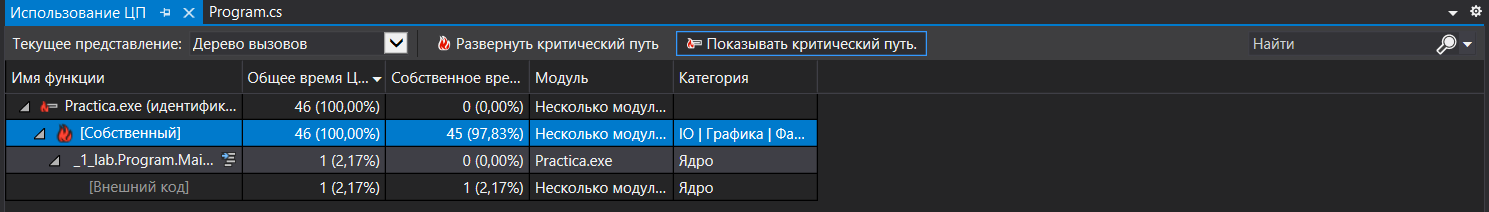


Рис.8. «Дерево вызовов»

3.2.2. Просмотр внешнего кода

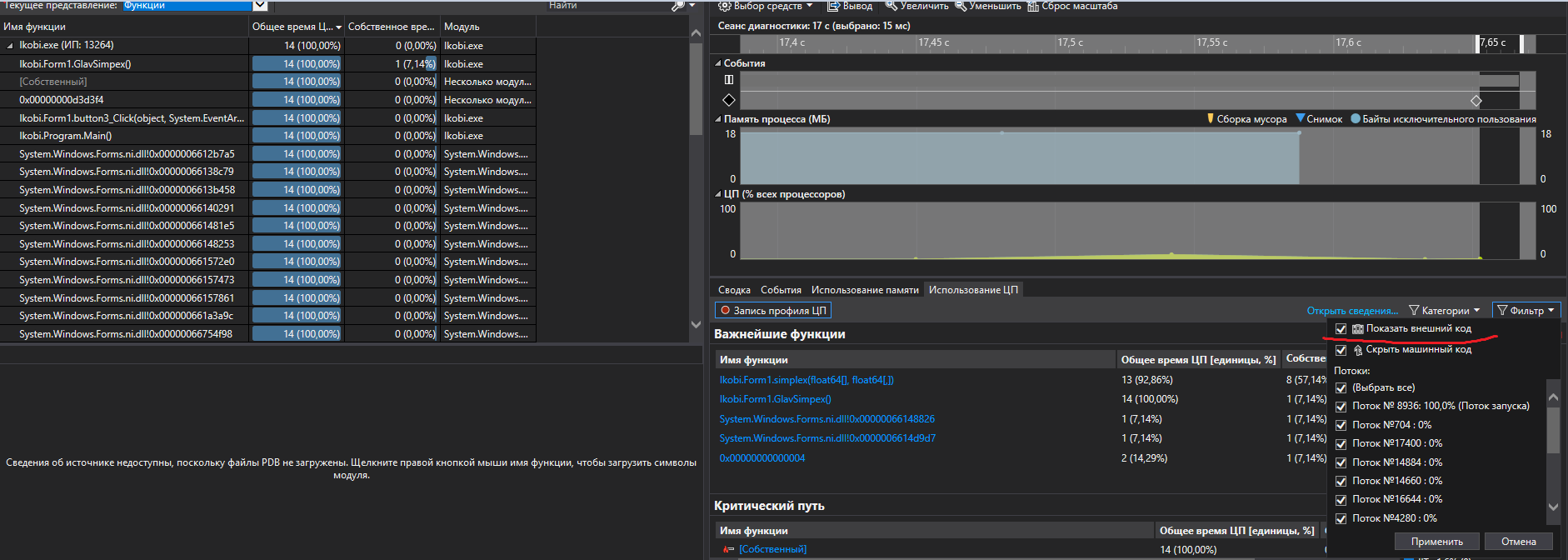


Рис.9. «Внешний код»

3.2.3. Загружаем на GitHub

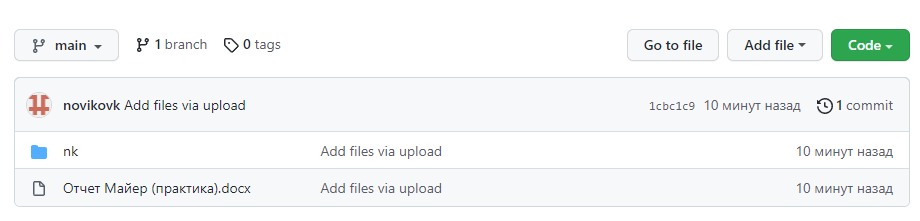


Рис.10 «GitHub»

**4.Установка ПО для обратного проектирования**

Создаем виртуальную машину

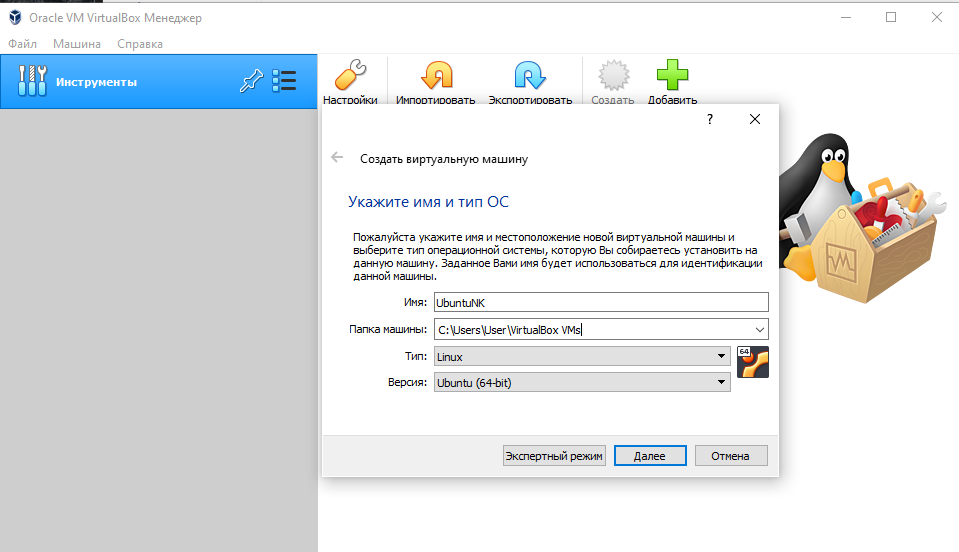


Рис.11. «Виртуальная машина»

Выбираем язык

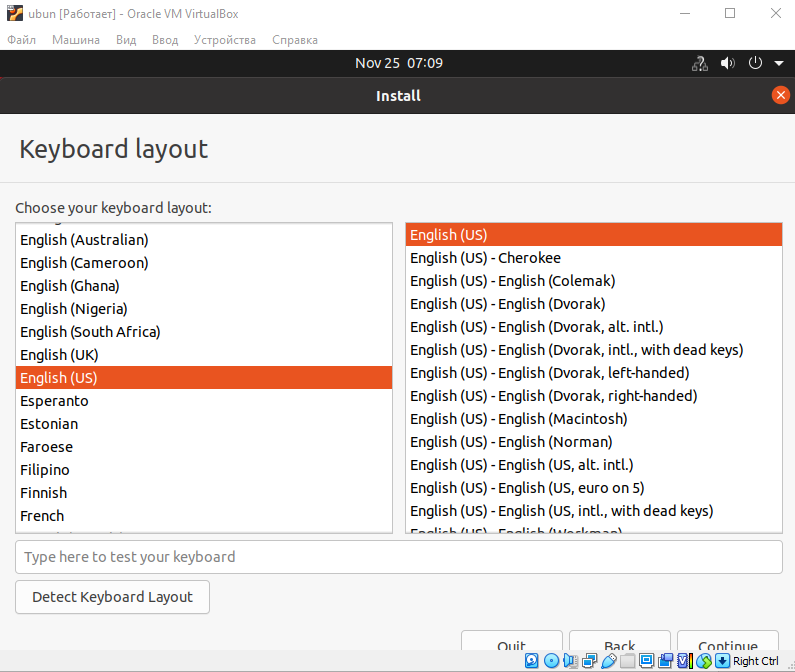


Рис.12. «Язык»

Выбираем какие приложения нужно установить

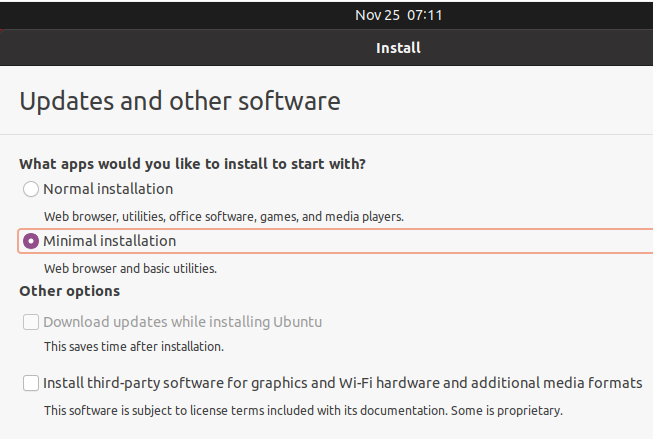


Рис.13. «Установка приложений»

Выбираем тип установки

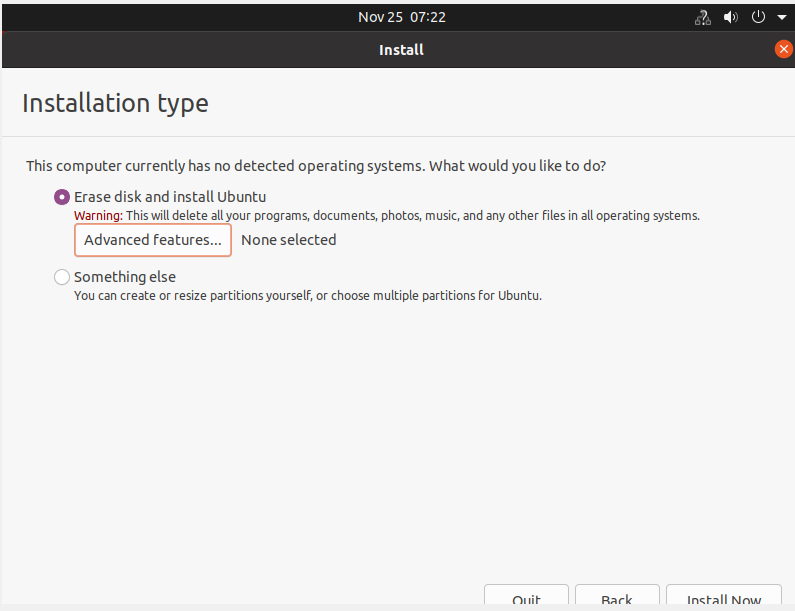


Рис.14. «Тип установки»

Выбираем регион



Рис.15. «Регион»

Авторизируем пользователя

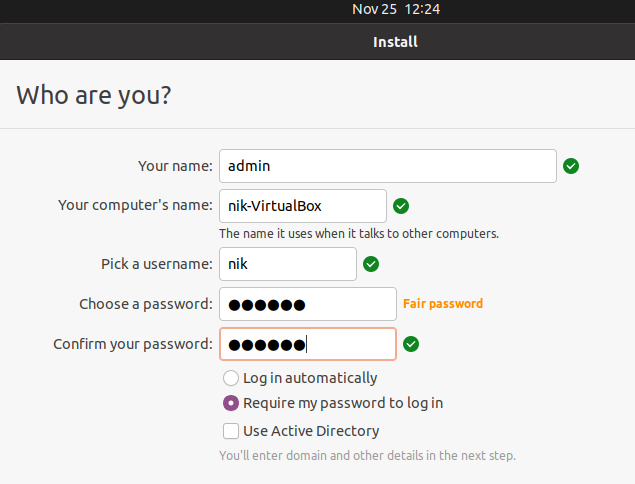


Рис.16. «Авторизация»

Входим в файл для прокси

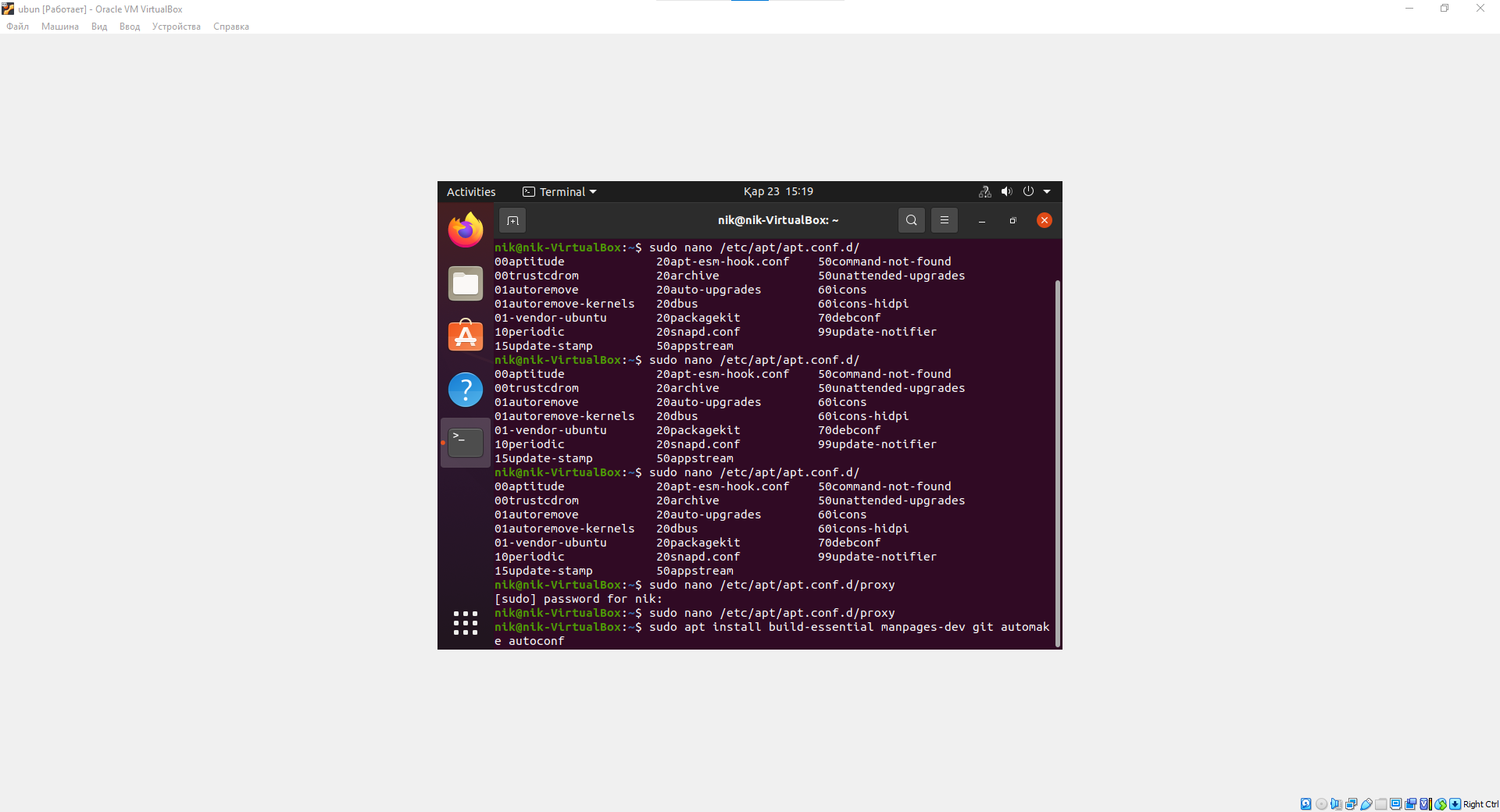


Рис.17. «Прокси файл»

Вводим в прокси колледжа

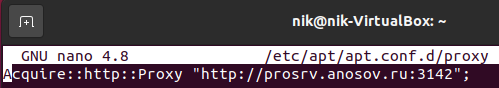


Рис.18. «Прокси колледжа»

Устанавливаем компилятор gcc



Рис.19. «Компилятор gcc»

Устанавливаем пакеты для работы с radare2

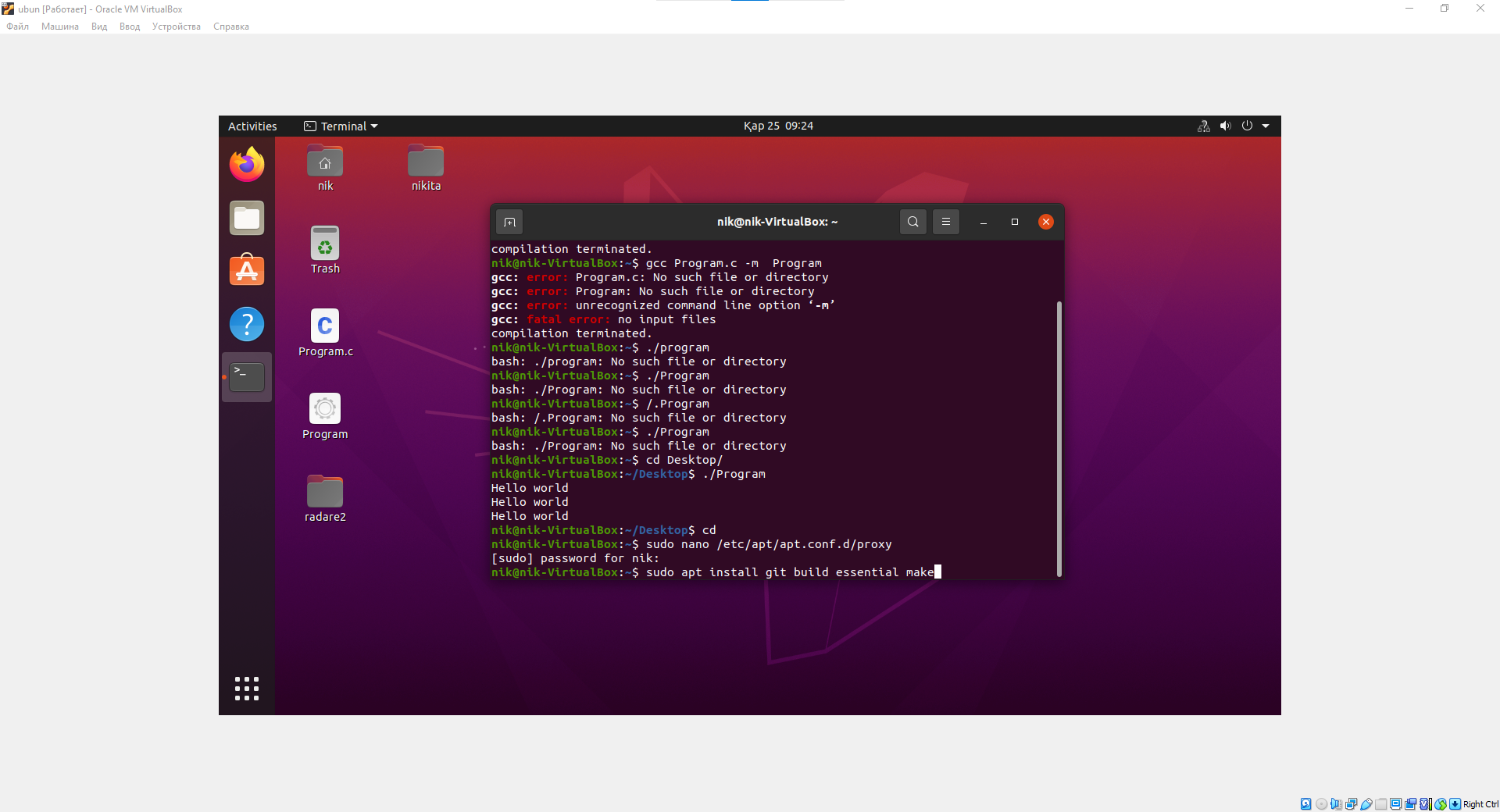


Рис.20. «Пакеты radare2»

Используем прокси колледжа

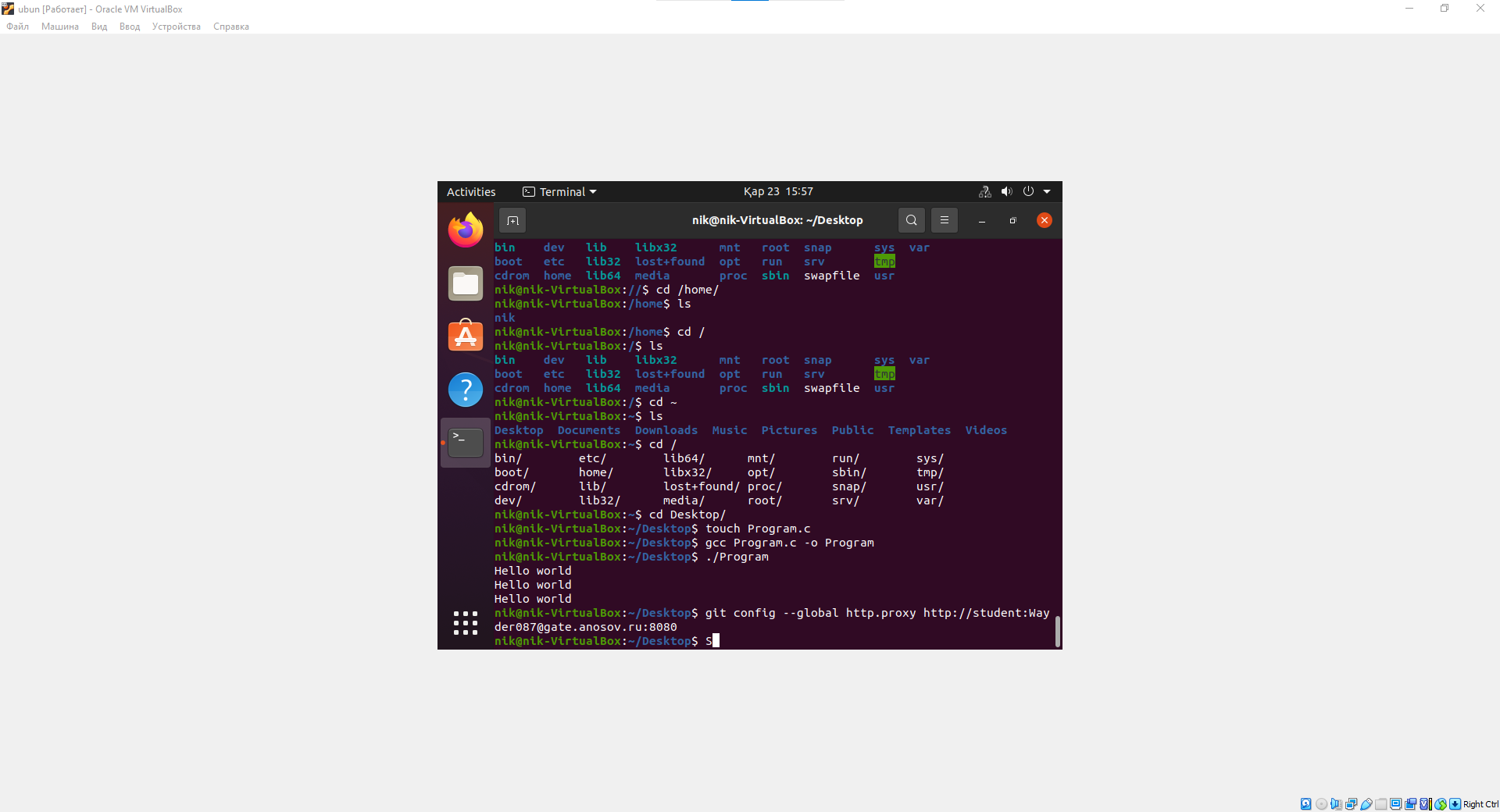


Рис.21. «Прокси колледжа»

Открываем файл для ввода прокси

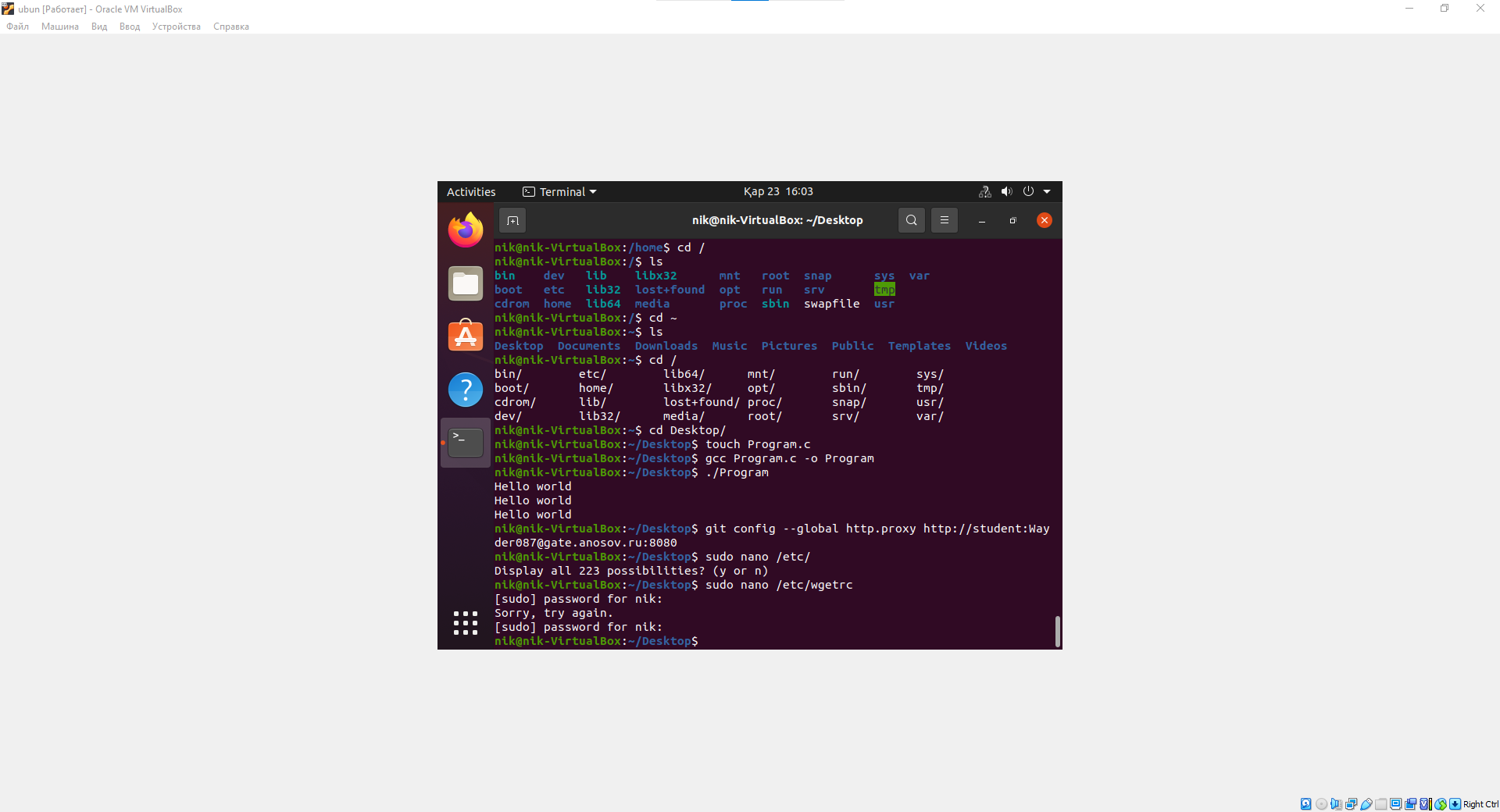


Рис.22. «Открытие файла для прокси»

Ввод прокси колледжа для http, https и ftp прокси

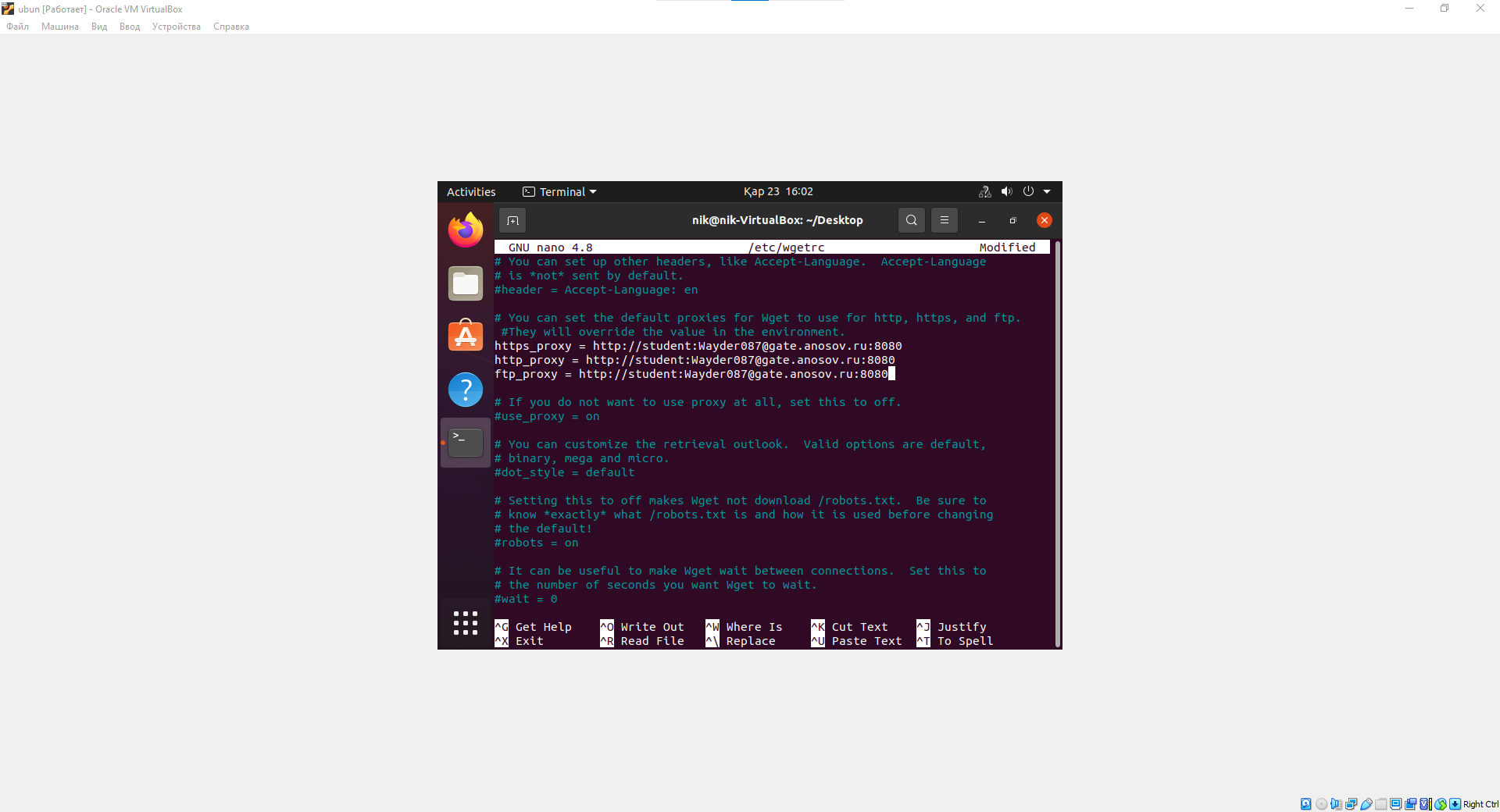


Рис.23. «Измененные прокси»

Клонированние radare2 с репозитория Git

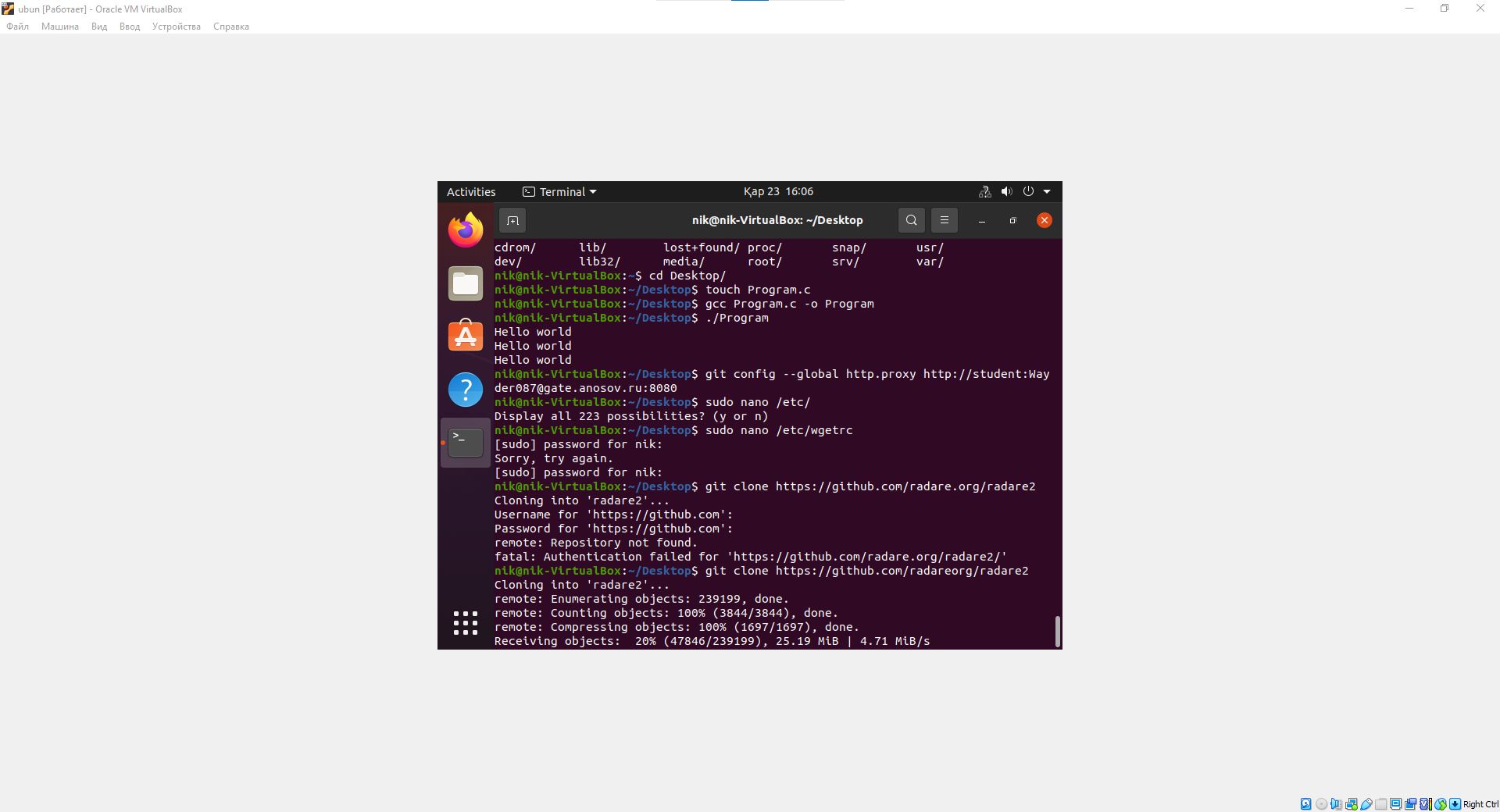


Рис.24. «Клонирование radare2»

Установка radare2 на рабочий стол

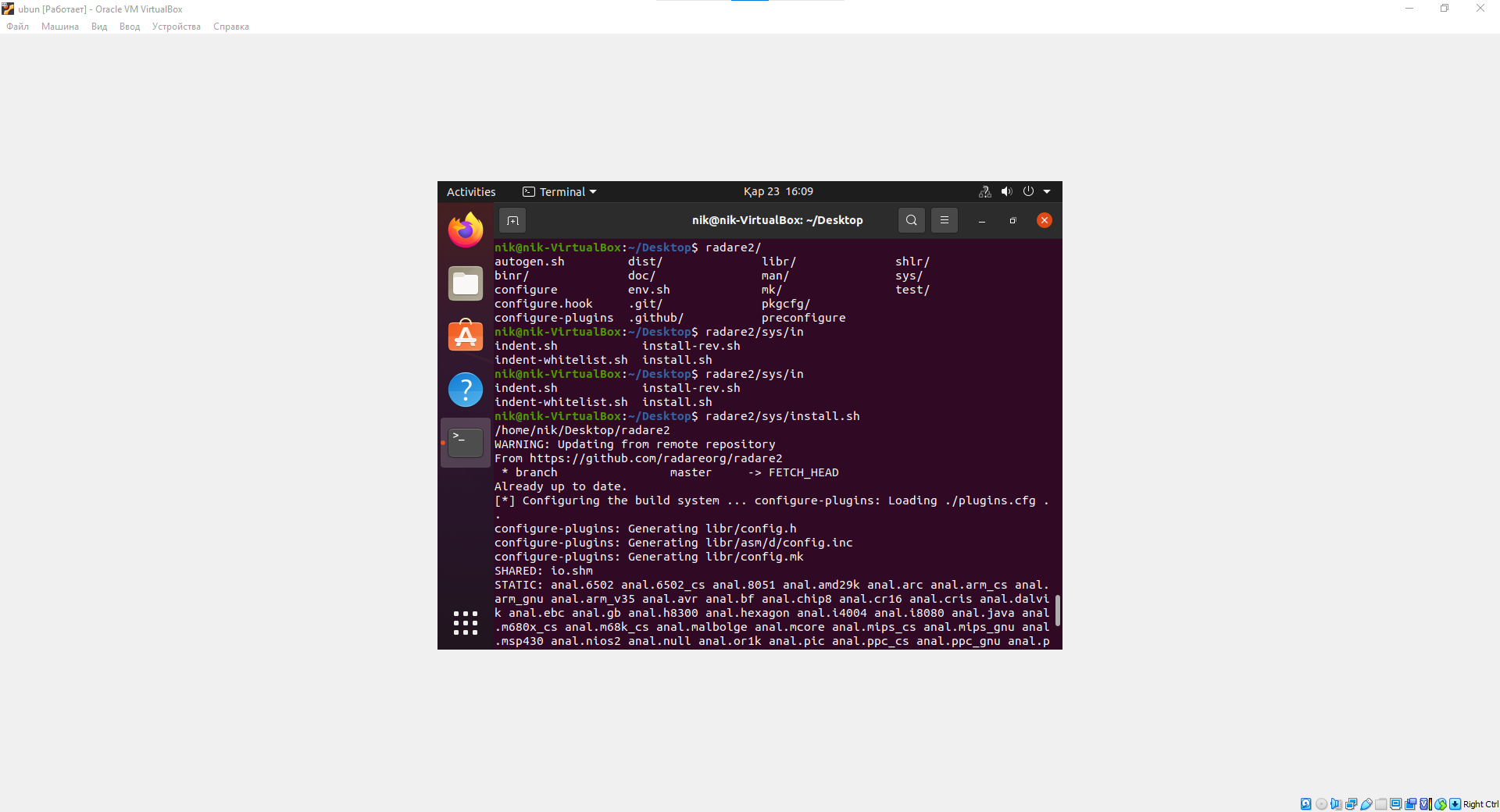


Рис.25. «Установка radare2»

Установка компонентов графической оболочки iaito



Рис.26. «Установка оболочки iaito»

Переход в папку radare2 и открытие файла конфигурации

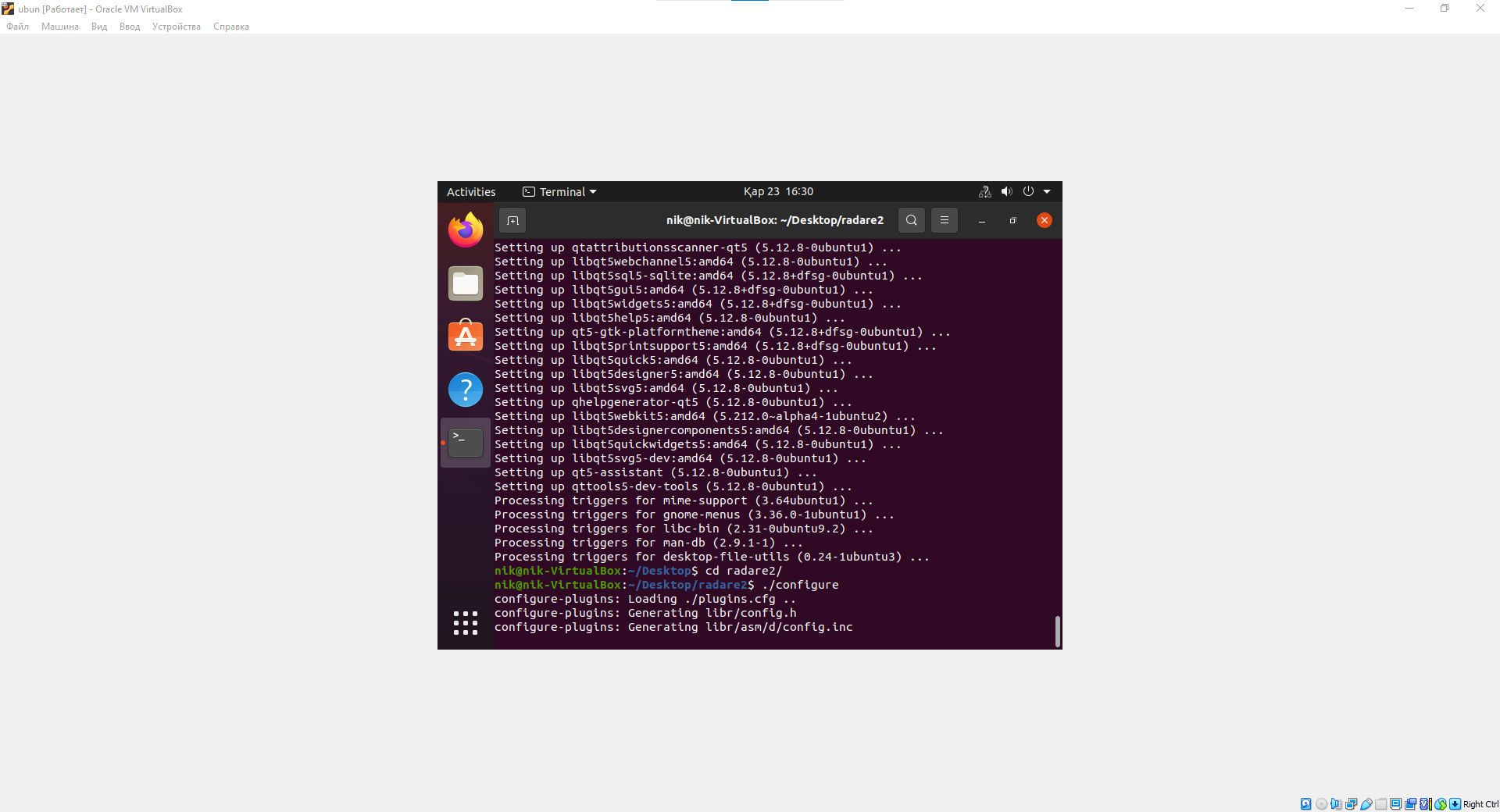


Рис.27. «Конфигурация radare2»

Распаковка radare2



Рис.28. «Распаковка radare2»

Установка графического интерфейса Iaito

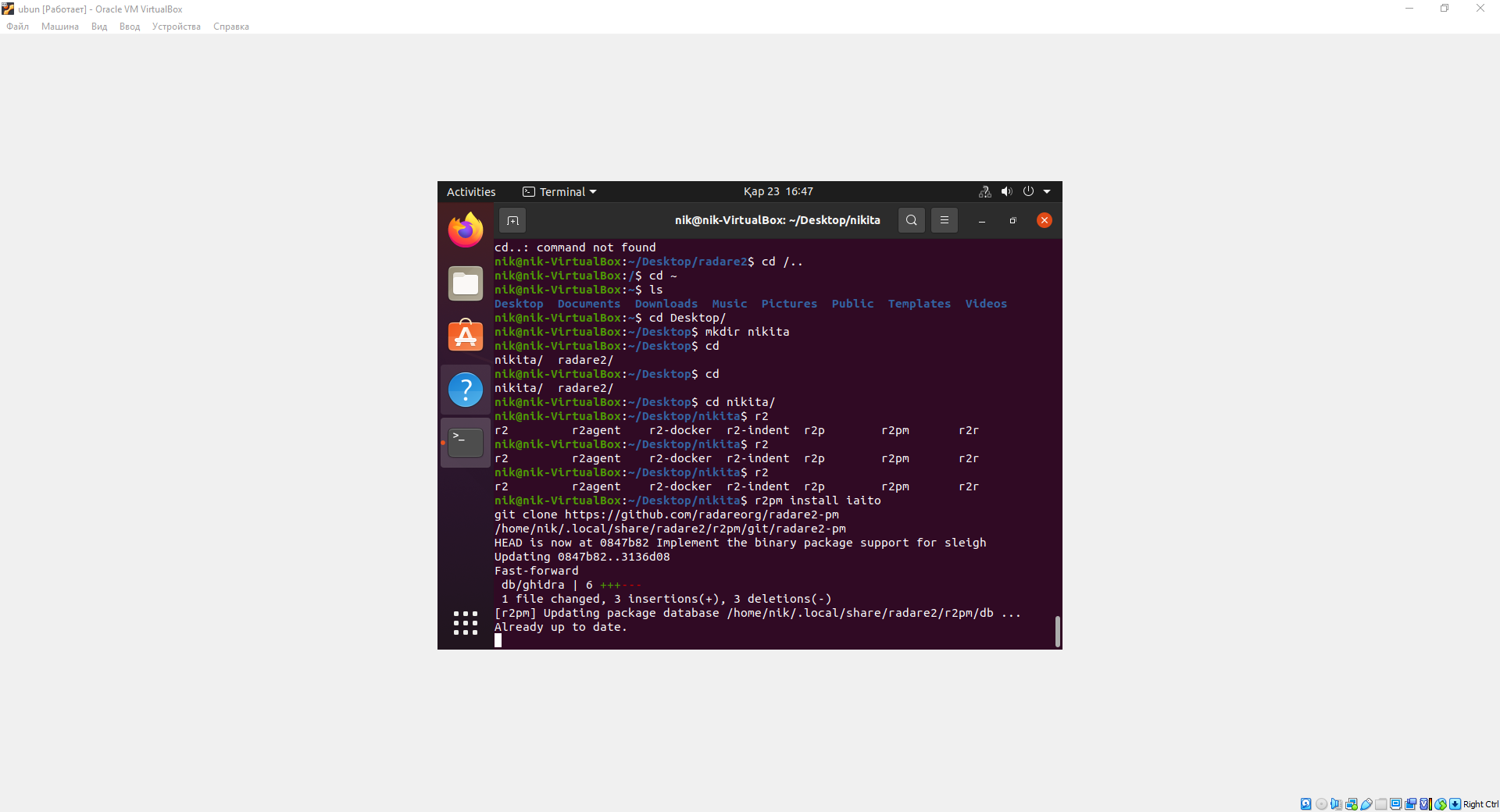


Рис.29. «Графический интерфейс iaito»

Установка декомпилятора r2ghidra



Рис.30. «Декомпилятор r2ghidra»

# **5. Дизассемблирование**

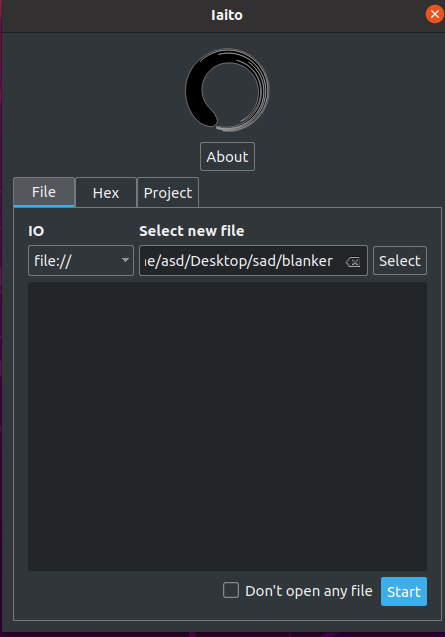


Рис.31. «Вход в iaito»

Запускаем iaito и переходим во вкладку main

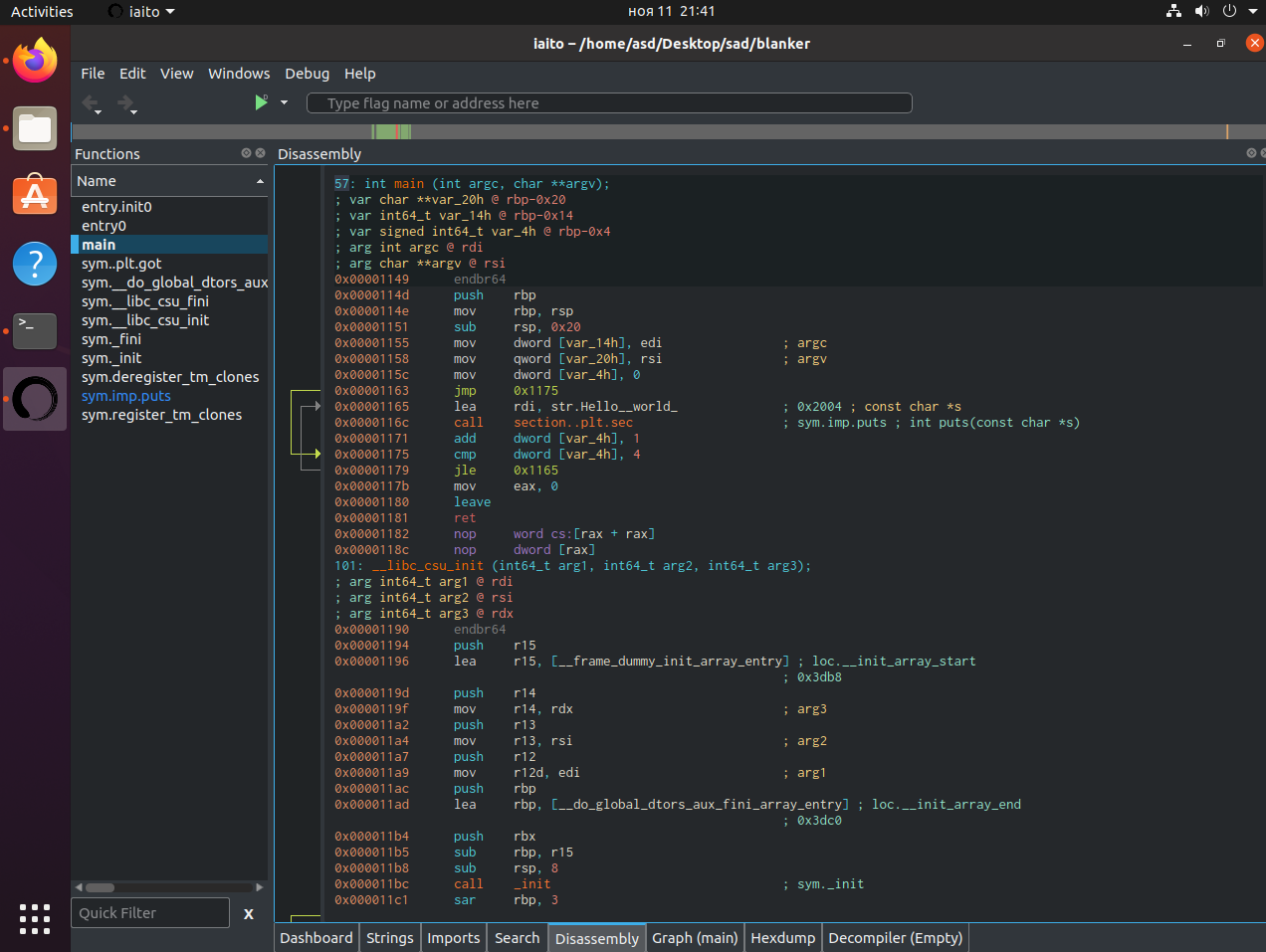


Рис.32. «Iaito»

Выбор элемента для изменения и его изменение

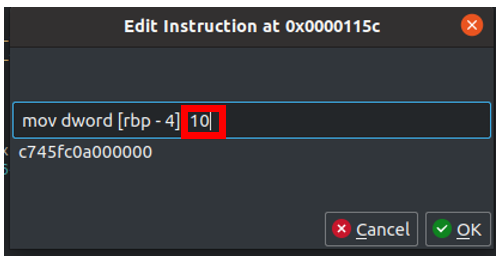


Рис.33. «Изменения в коде»

Проверка изменений

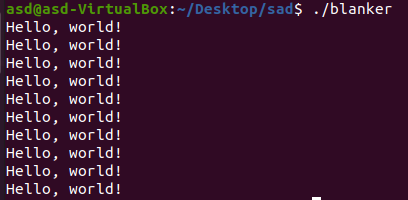


Рис.34. «Проверка изменений»

# **6.Обратное проектирование**

Создание файла с кодом для компиляции на рабочем столе



Рис35. «Файл для компиляции»

Файл для компиляции на рабочем столе

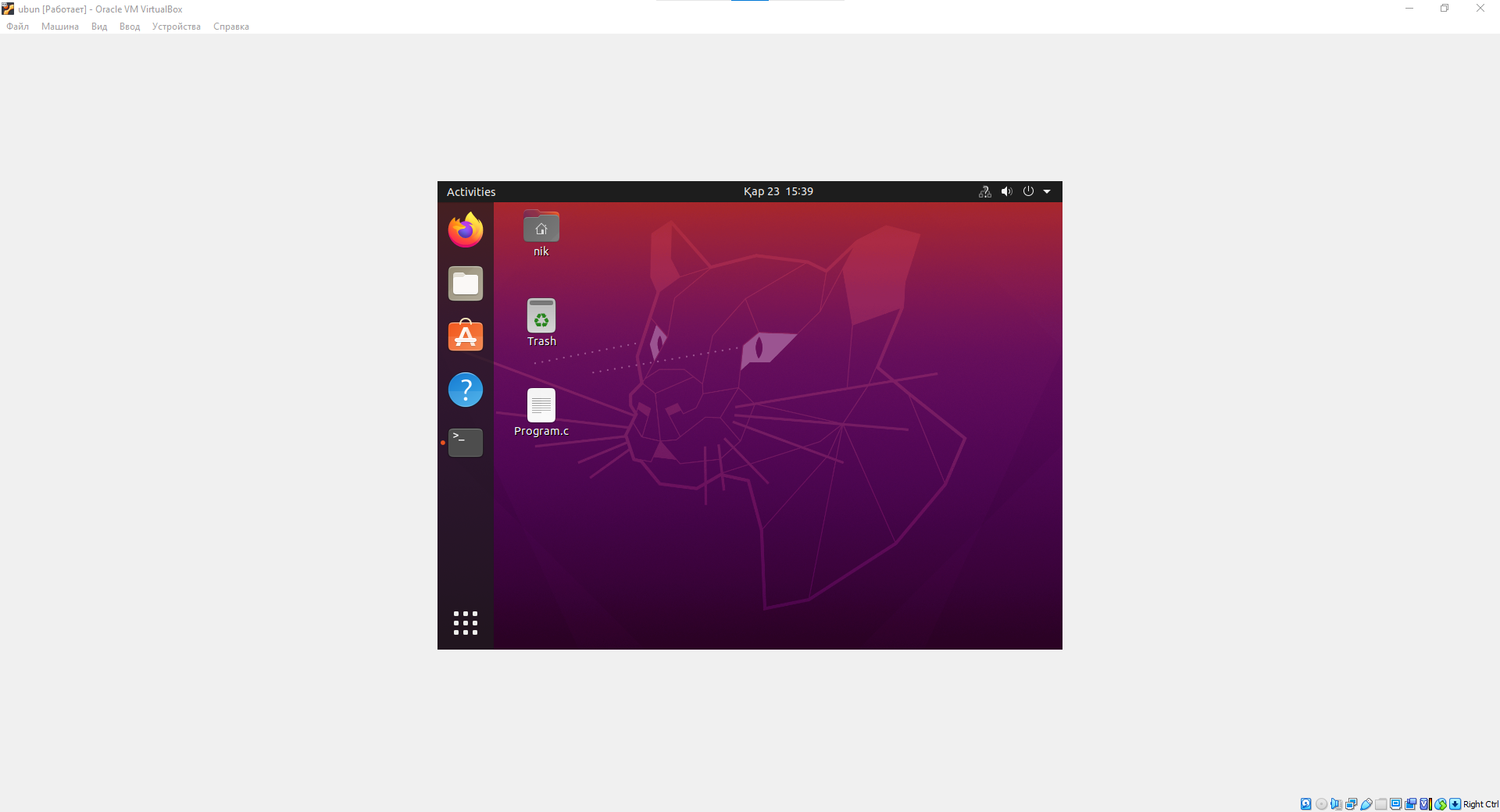


Рис.36. «Файл на рабочем столе»

Исходный код

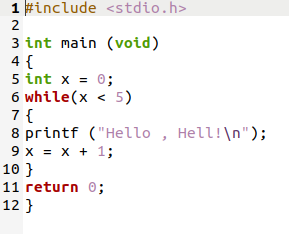


Рис.37. «Исходный код»

Код в файле для компиляции

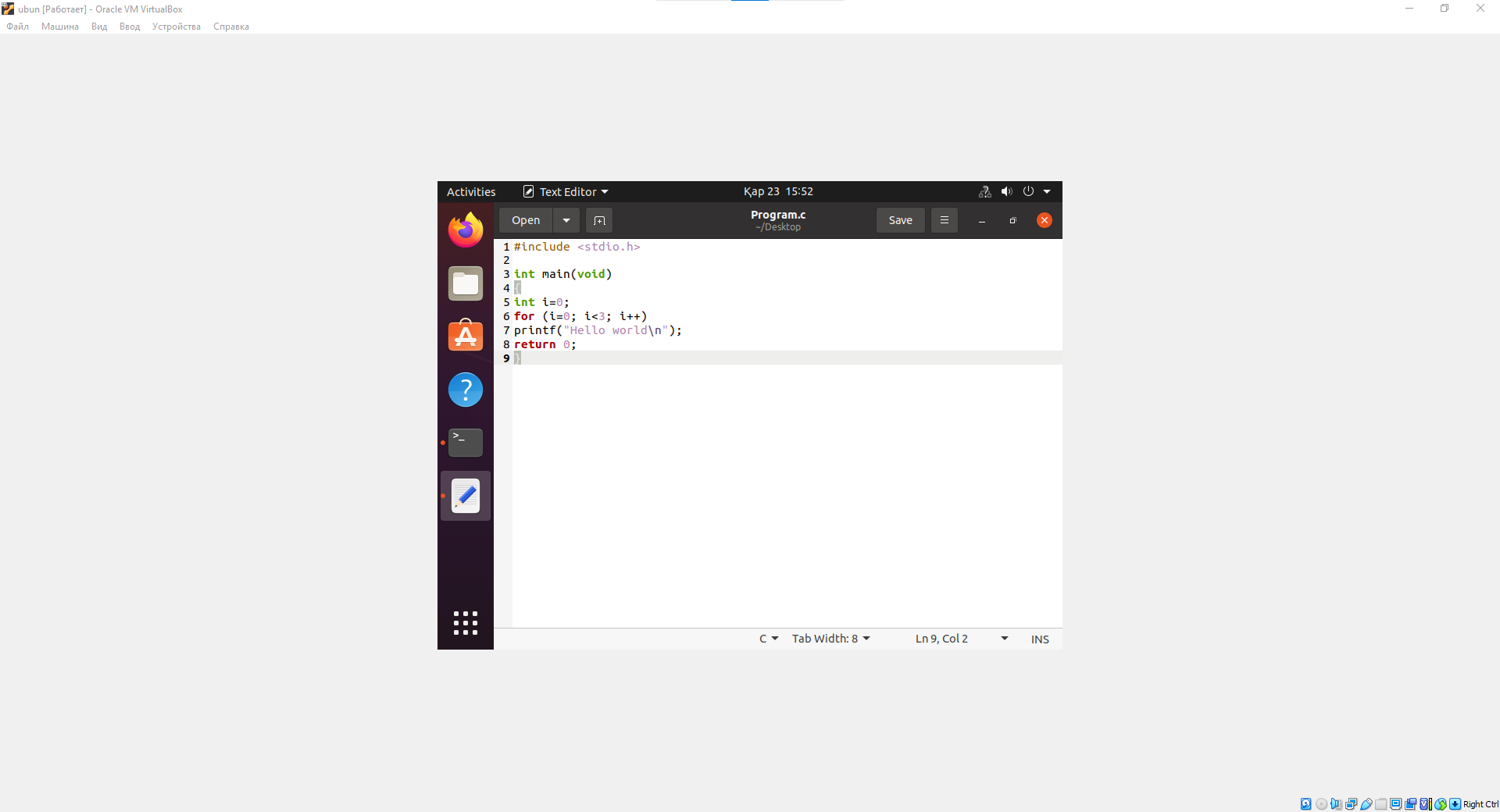


Рис.38. «Код в файле»

Компиляция файла с кодом



Рис.39. «Компиляция файла»

Запуск скомпилированного файла



Рис.40. «Скомпилированный файл»

# **Заключение**

Цель учебной практики по ПМ.03 «Ревьюирование программных продуктов» являлось, научиться выполнять прямое и обратное проектирование ПО, овладеть навыками приостановки и возобновления работы с рабочими задачами иинспекцией кода в Visual Studio, совместно работать над проектом в системе контроля версий GIT.

В ходе выполнения заданий на учебой практике было выполнение прямо и обратное проектирование ПО, получение навыков приостановки и возобновления работы с рабочими задачами и инспекцией кода в Visual Studio, совместно работать над проектом в системе контроля версий GIT.

В ходе учебной практики для достижения цели, ставятся задачи:

* измерить производительность приложения посредством анализа использования ЦП,
* установить ПО (Linux, компилятор GCC, radare2, iaito) для обратного проектирования,
* выполнить обратное проектирование,
* выполнить дизассемблирование.

# **Литература**

1. 3.Золотов, С. Ю. Проектирование информационных систем [Электронный ресурс] / С.Ю. Золотов. - Томск: Эль Контент, 2013. - 88 с.
2. Карпенков, С. Х. Технические средства информационных технологий [Электронный ресурс] / С.Х. Карпенков. - 3-е изд., испр. и доп. - М.|Берлин: Директ-Медиа, 2015. - 376 с.
3. Коноплева, И. А. Информационные технологии [Электронный ресурс] / И.А. Коноплева. - 2-е изд., перераб. и доп. - Москва: Проспект, 2014. - 328 с.
4. Корячко, В. П. Процессы и задачи управления проектами информационных систем [Электронный ресурс] / В.П. Корячко. - Москва: Горячая линия - Телеком, 2014. - 376 с.
5. Ланских, Юрий Владимирович Предметно-ориентированные информационные системы [Электронный ресурс] : учеб. пособие для студентов направления 09.03.02, 10.03.01,
6. Проектирование информационных систем. Лекция 1. Презентация [Электронный ресурс]. - Москва: Национальный Открытый Университет «ИНТУИТ», 2014. - 27 с.
7. Советов, Борис Яковлевич. Информационные технологии [Электронный ресурс]: учебник / Б. Я. Советов, В. В. Цехановский. - 6-е изд. - Москва: Юрайт, 2015. - х эл. опт. диск (CD-ROM)
8. Советов, Борис Яковлевич. Информационные технологии [Электронный ресурс]: учебник / Б. Я. Советов, В. В. Цехановский. - 6-е изд. - Москва: Юрайт, 2015. - х эл. опт. диск (CD-ROM)
9. Страбыкин, Дмитрий Алексеевич. Организация ЭВМ: лабораторный практикум на компьютерах: учеб. пособие для студентов направления подготовки 09.03.01 (230100.62) / Д. А. Страбыкин; ВятГУ, ФАВТ, каф. ЭВМ. - 3-е изд., перераб. и доп. - Киров: [б. и.], 2013. - 62 с.