Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Отчет по Лабораторной работе №1

по курсу “ОС Linux”

Студент \_\_\_\_\_\_\_\_\_\_\_ Красиков И.А.

подпись, дата

Группа ПИ-21-1

Руководитель \_\_\_\_\_\_\_\_\_\_\_ Кургасов В.В.

подпись, дата

Липецк 2023 г.

**Оглавление**

[**Установка ОС Linux на Гипервизор** 3](#_Toc1)

# **Задание**

1) Установить Гипервизор и ОС Ubuntu Server

2) Реализовать одну из задач по варианту в редакторе Vim

Вариант 3:

Переставить все четные элементы в начало массива.

3) Сделать сборку проекта с помощью make

4) Продемонстрировать работу отладчика gdb

# **Установка ОС Linux на Гипервизор**

ОС – Ubuntu Server 64-bit 22.04.3

Гипервизор – VirtualBox

1) Установка iso-образа с официального сайта

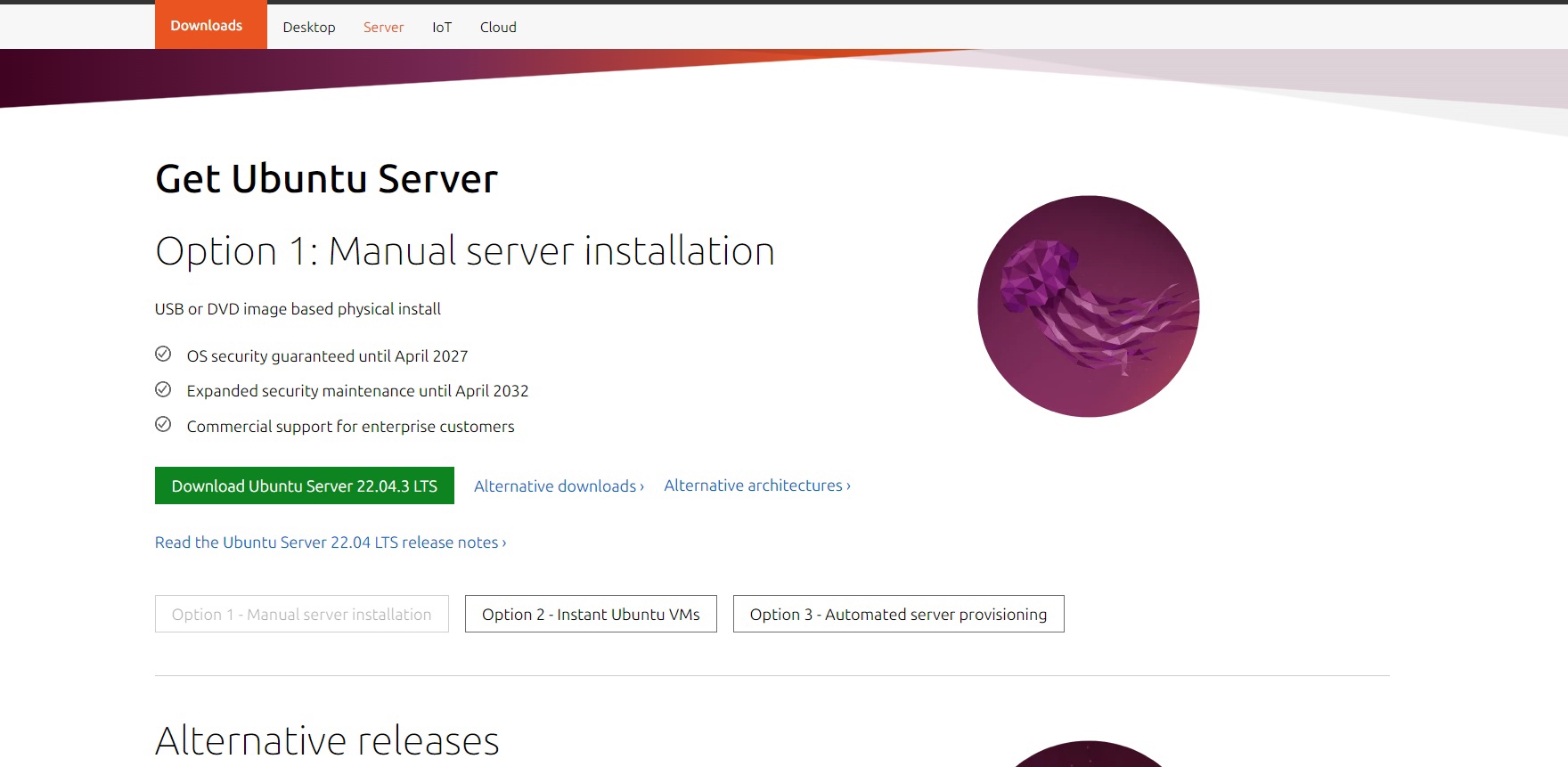


Рис.1 – Официальный сайт Ubuntu

2) Создание виртуальной машины в VirtualBox

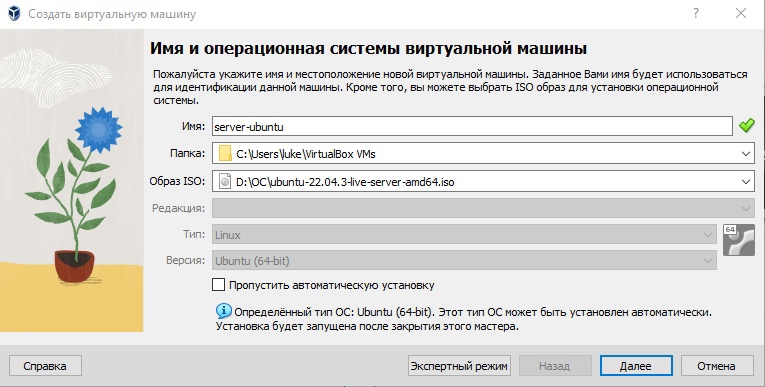


Рис.2 – Выбор iso-файла, папки, в которой находиться виртуальная машина, и имени.

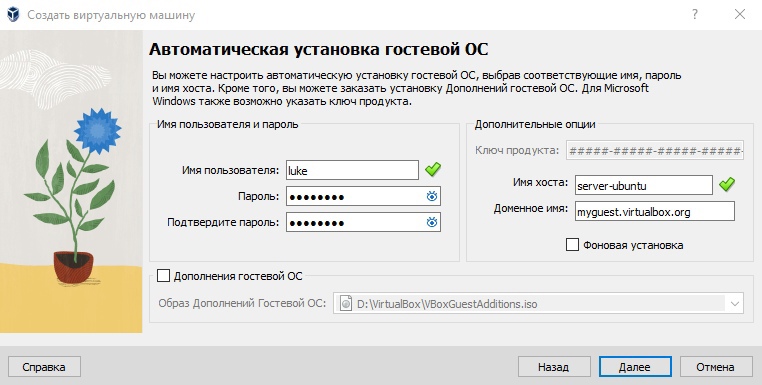


Рис. 3 – Автоматическая установка гостевой ОС

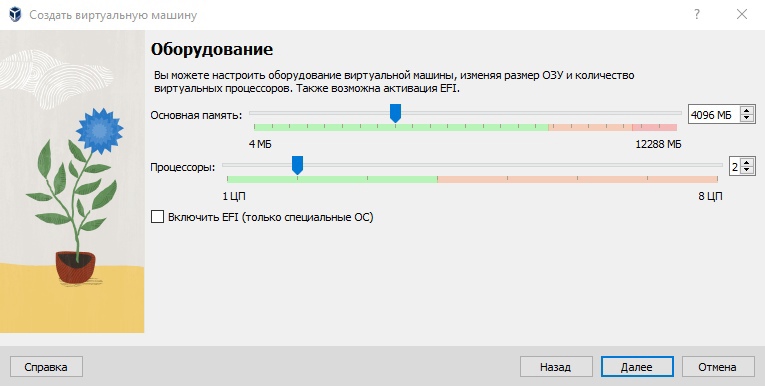


Рис. 4 – Настройка оборудования для виртуальной машины

Выбираем 4 Гб и 2 ЦП для нашей виртуальной машины.

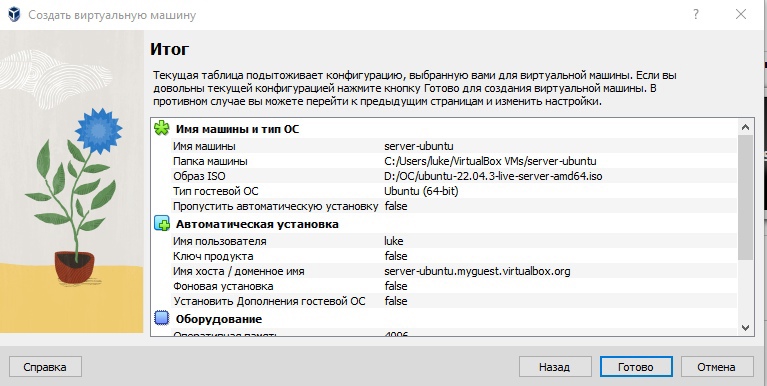


Рис. 5 – Итог

3) Установка и настройка ОС

Запустим виртуальную машину и проведем установку ОС

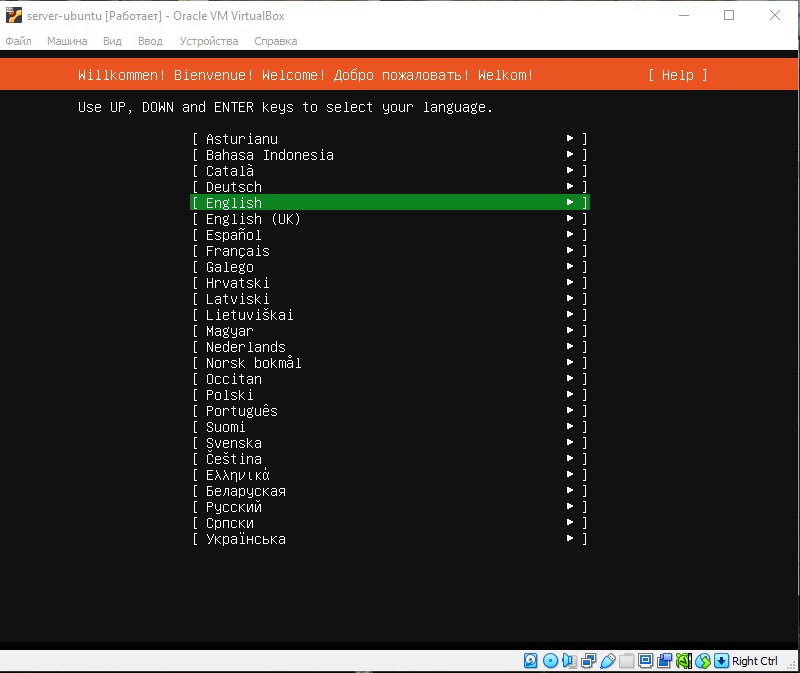


Рис. 6 – Выбор языка

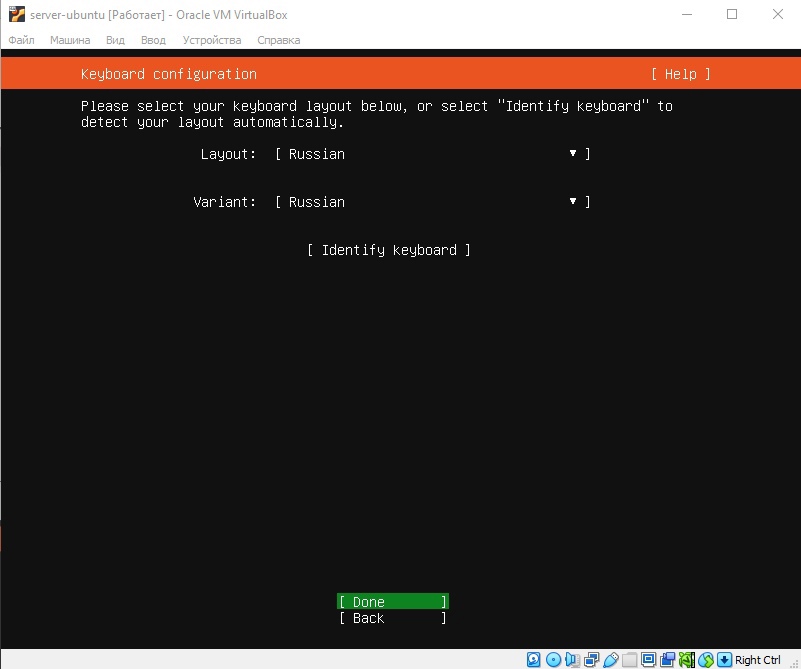


Рис. 7 – Выбор конфигурации клавиатуры

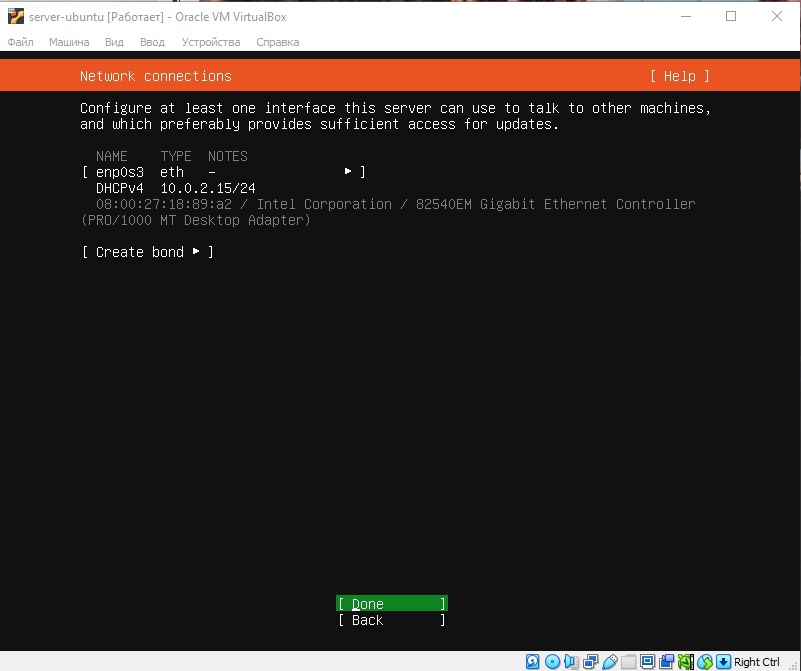


Рис. 8 – Настройка сетевого подключения

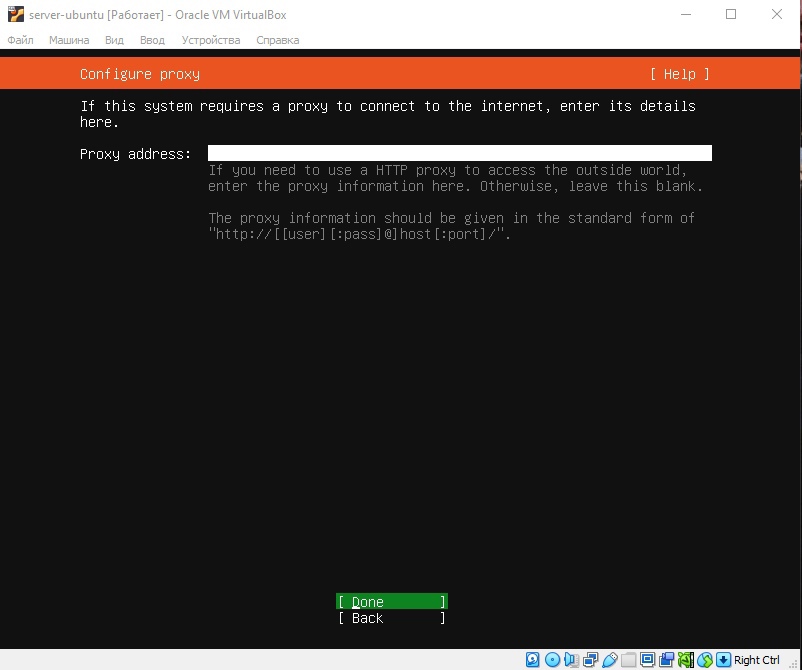


Рис. 9 – Выбор Proxy адреса

Ничего не указываем, потому что Гипервизор автоматически их укажет.

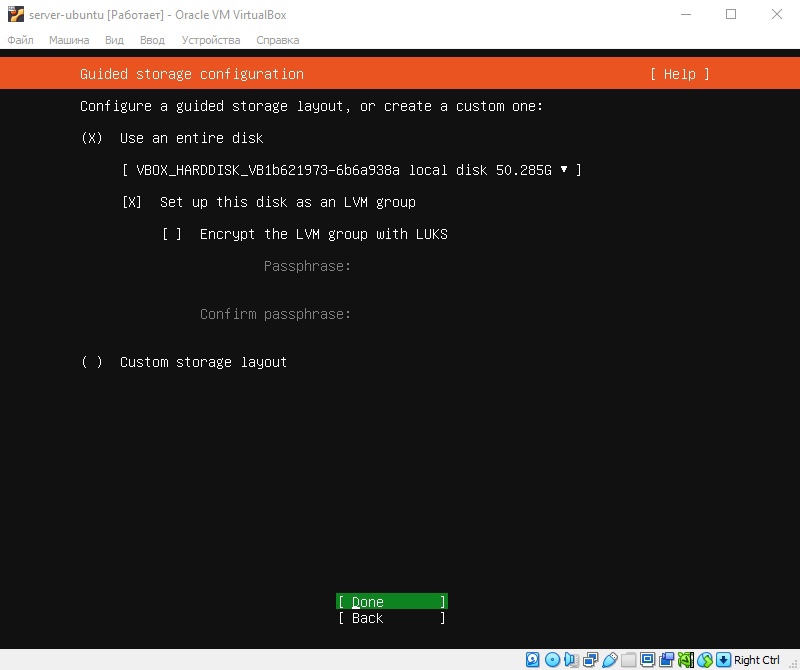


Рис. 10 – Сборка конфигурации памяти

Используем диск созданный VirtualBox

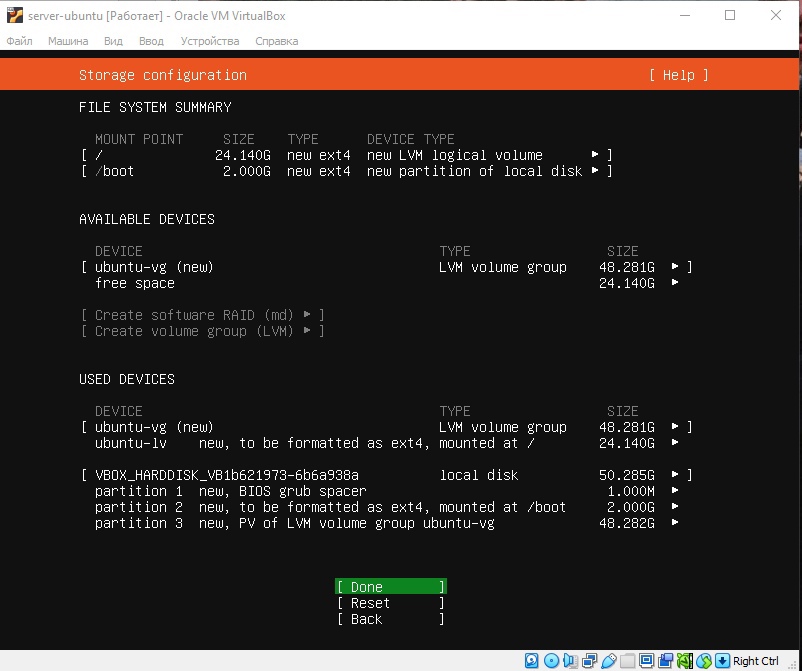


Рис. 11 – Настройка конфигурации памяти

Используем автоматические настройки.

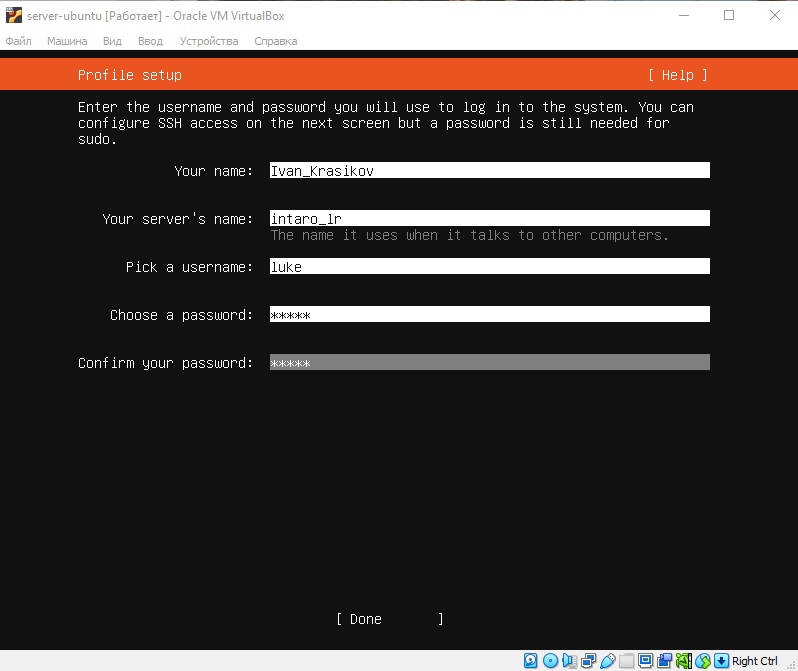


Рис. 12 – Задаем данные пользователя, название сервера, пароль

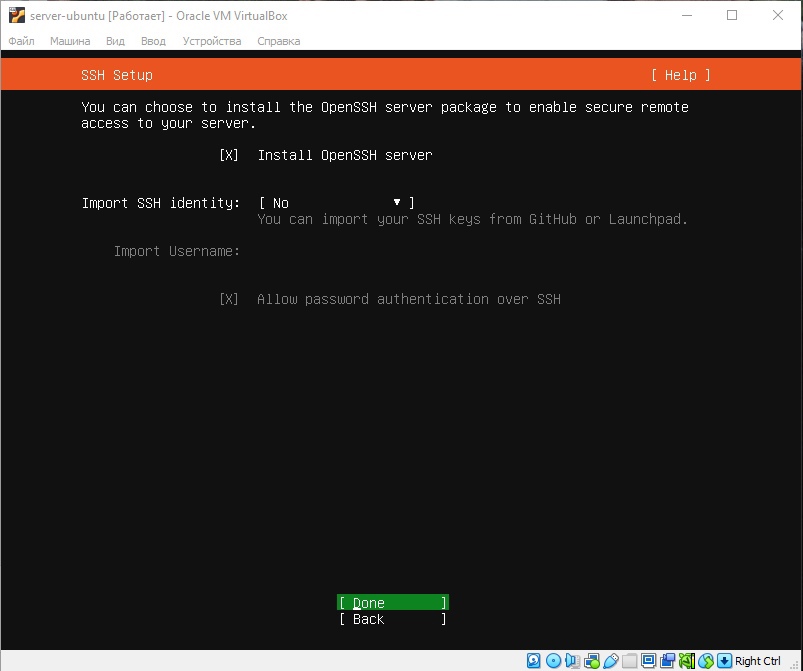


Рис. 13 – Настройка SSH

Ставим крестик на установке OpenSSH сервера, остальное оставляем по умолчанию.

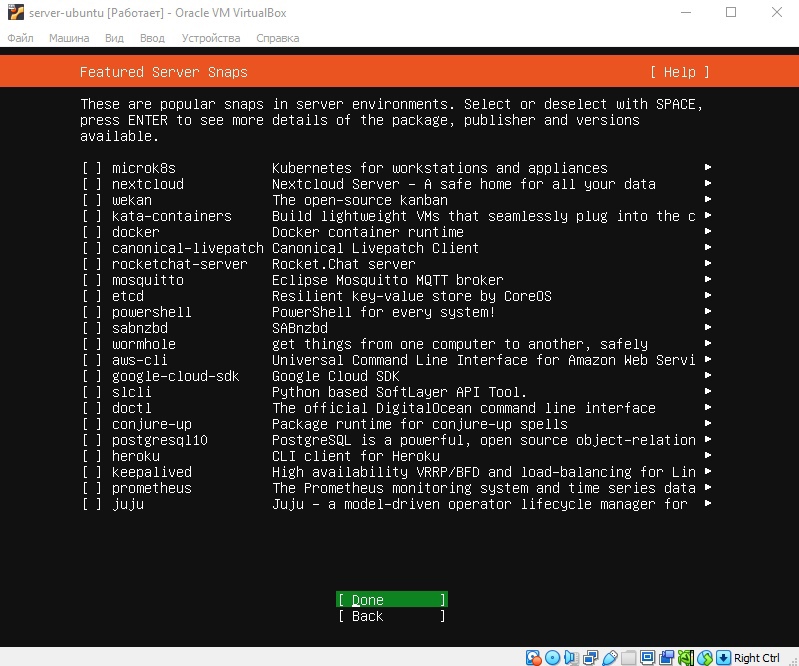


Рис. 14 – Выбор дополнительных пакетов для установки

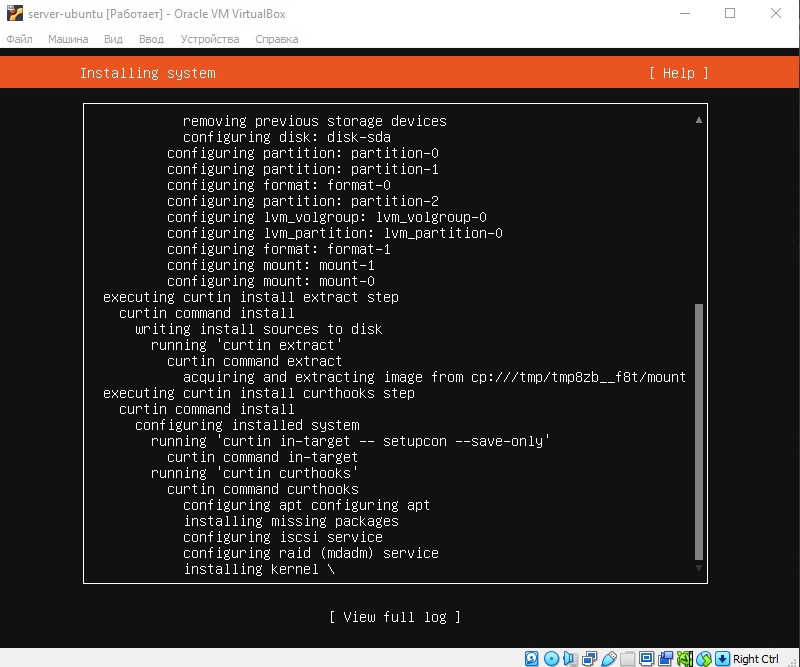


Рис. 15 – Установка ОС

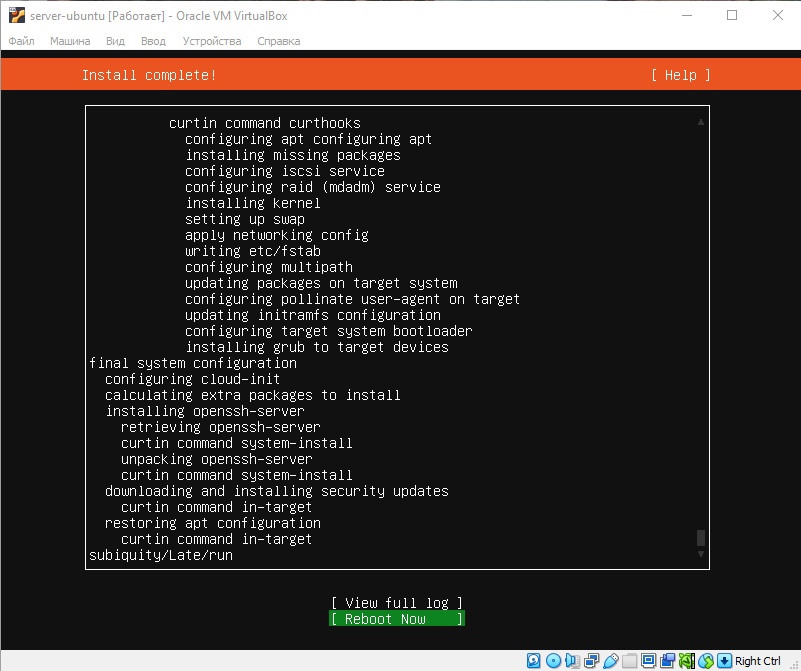


Рис. 16 – Конец установки

Установка закончена, после чего выбираем перезагрузку.

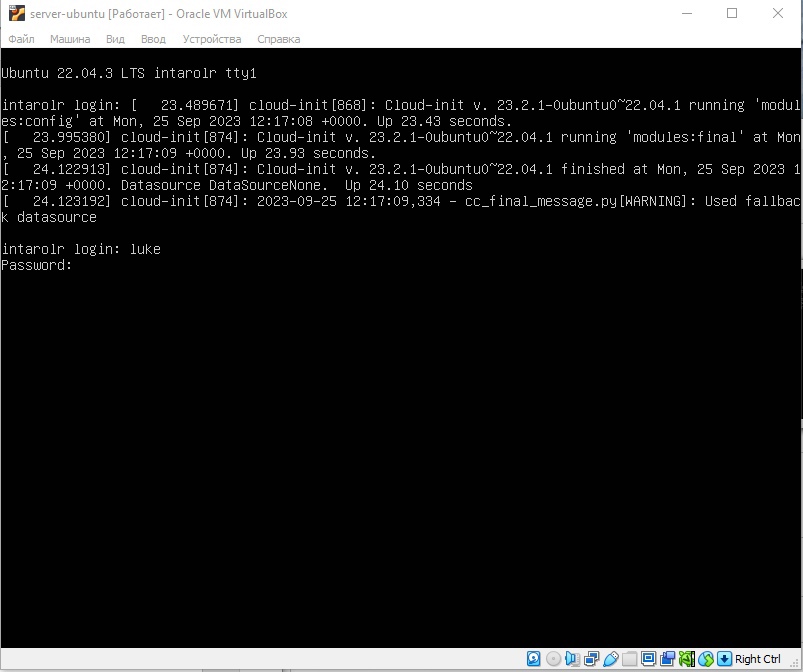


Рис. 17 – Вход в систему

После чего меняем разрешение экрана. Заходим в /etc/default и открываем файл grub, и записываем строчку GRUB\_CMDLINE\_LINUX\_DEFAULT = “quiet vga=795”, после вписываем команду sudo upgrate-grub и перезапускаем систему

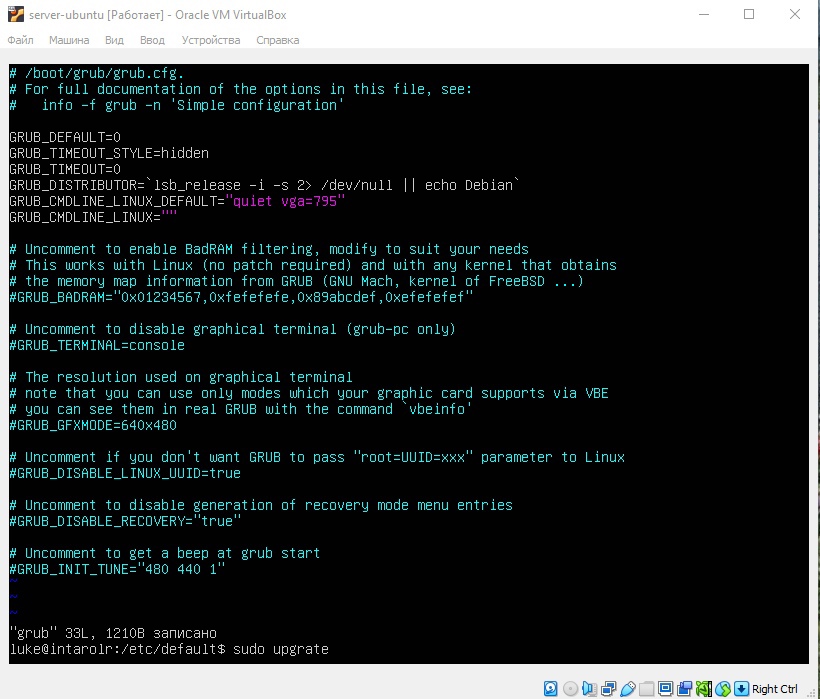


Рис. 18 – Файл grub

Устанавливаем gcc, через команду sudo apt install gcc.

Устанавливаем make, через команду sudo apt install make.

Устанавливаем gdb, через команду sudo apt install gdb.

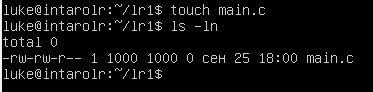
# **Реализация, сборка и отладка программы по варианту**

Задача:

Переставить все четные элементы в начало массива.

## **1) Реализация**

Для начала создадим директорию и файл main.c



После чего откроем с помощью vim файл main.c и напишем код программы

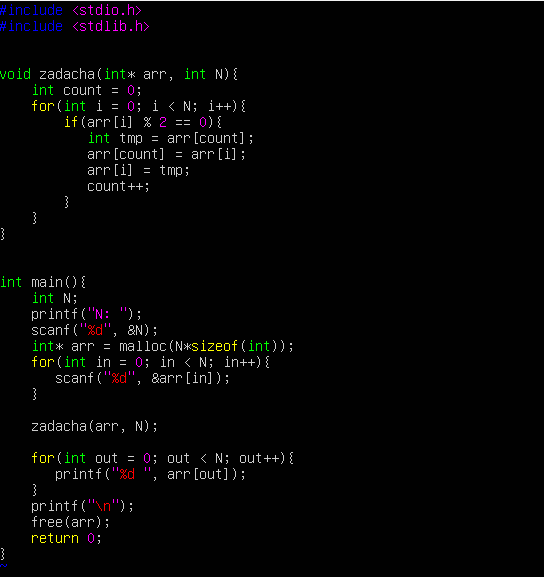
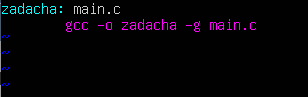
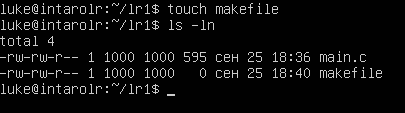


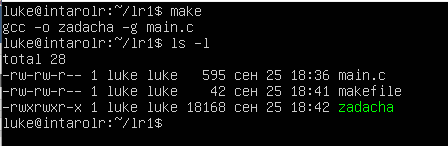
Рис. 19 – Код программы в Vim

## **2) Сборка**

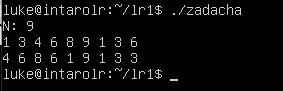
После чего сделаем сборку через make, для этого нужно создать файл makefile и вписать туда правила сборки.



Теперь мы можем написать команду make и наша программа соберется.



Запускаем программу через команду ./zadacha



## **3) Отладка**

С помощью команды gdb zadacha, мы можем запустить процесс отладки.

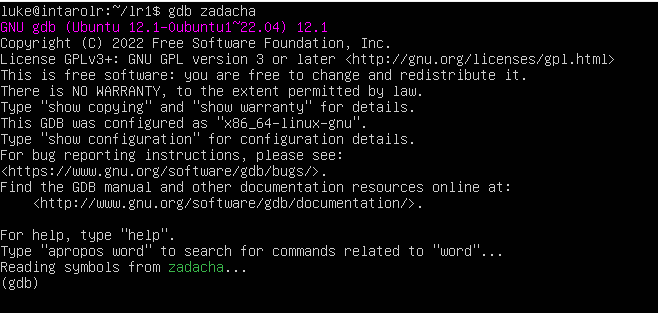
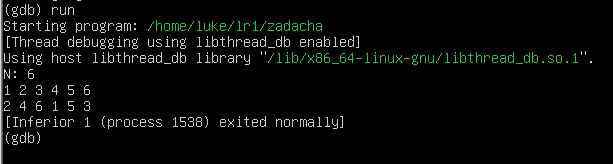


Рис. 20 – Процесс отладки

С помощью команды run можем запустить программу



С помощью команды list можно вывести код программы, но по умолчанию выводит только 10 строк, для снятия ограничения нужно вписать команду set list unlimited.

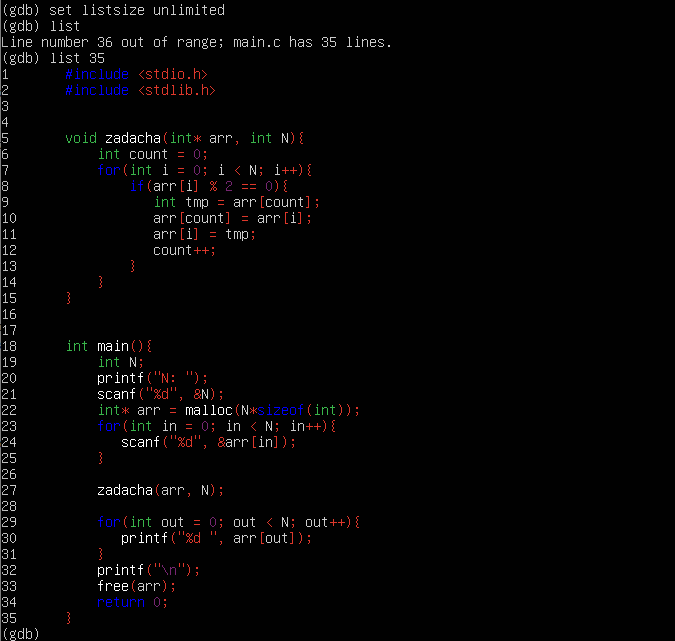
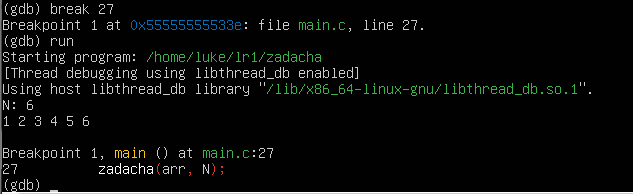
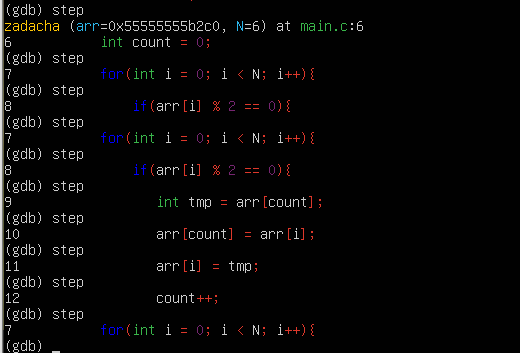


Рис. 21 – Вывод кода программы через gdb

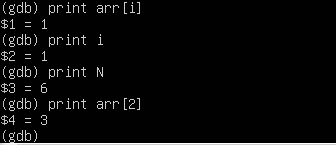
С помощью команды break можно задать точку останова, программа буде выполнять до заданной строчки.



C помощью команды step можно перейти на следующую строчку



С помощью команды print можно вывести значение переменной



С помощью команды delete можно удалить точку останова, указав номер точки.

С помощью команды quit можно выйти из процесса отладки gdb.

# **Ответы на контрольные вопросы**

1) Что такое IDE?

Ответ: Это программное обеспечение, которое помогает программистам писать программы, оно включает текстовый редактор, сборщик и отладчик.

2) Что такое API?

Ответ: Описание способов взаимодействия одной программы с другими.

3) Что такое библиотека в программировании?

Ответ: Это готовый набор функция и объект для какого-либо языка программирования.

4) Понятия Статической и Динамической библиотек.

Ответ: Код статических библиотек полностью входит в код программы, а код динамических библиотек не входит в код программы, код программы содержит только ссылку на библиотеку.

5) Что такое плагин?

Ответ: Независимо компилируемый программный модуль, динамически подключаемый к основной и предназначенный для расширения или использования ее возможностей.

6) Назовите несколько текстовых консольных редакторов для Linux?

Ответ: Vi, Nano, Vim

7) Что делает команда gcc?

Ответ: Компилирует программу.

8) Что делает команда make?

Ответ: Собирает программу в исполняемый файл, сокращая ввод команд через gcc, собирая их в файле makefile

9) Что делает команда gdb?

Ответ: Запускает процесс отладки программы.

10) Дайте определение заголовочного файла и файла реализации.

Ответ: Заголовочный файл – это файл, содержимое которого автоматически добавляется в текст программы препроцессором. Пример: stdio.h. Файл реализации – это файл с исходным кодом программы. Пример: main.c.

11) Что означает единица трансляции? В чем особенность разработки программ из нескольких единиц трансляции?

Ответ: Единица трансляции – это максимальный блок исходного текста, который физически можно оттранслировать (преобразовать во внутреннее машинное представление; в частности, откомпилировать). Особенность разработки с использованием нескольких единиц трансляции в языке C заключается в том, что программа делиться на несколько файлов и подключается через #include.

12) Дайте краткую характеристику каждому этапу трансляции программ, написанных на языке C.

Ответ: Сначала код программы обрабатывает препроцессор, он объединяет заголовочный файл и файл реализации, после чего все это обрабатывает компилятор, он переводит код высокого уровня в машинный код, после это все переходит в линкер.