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

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

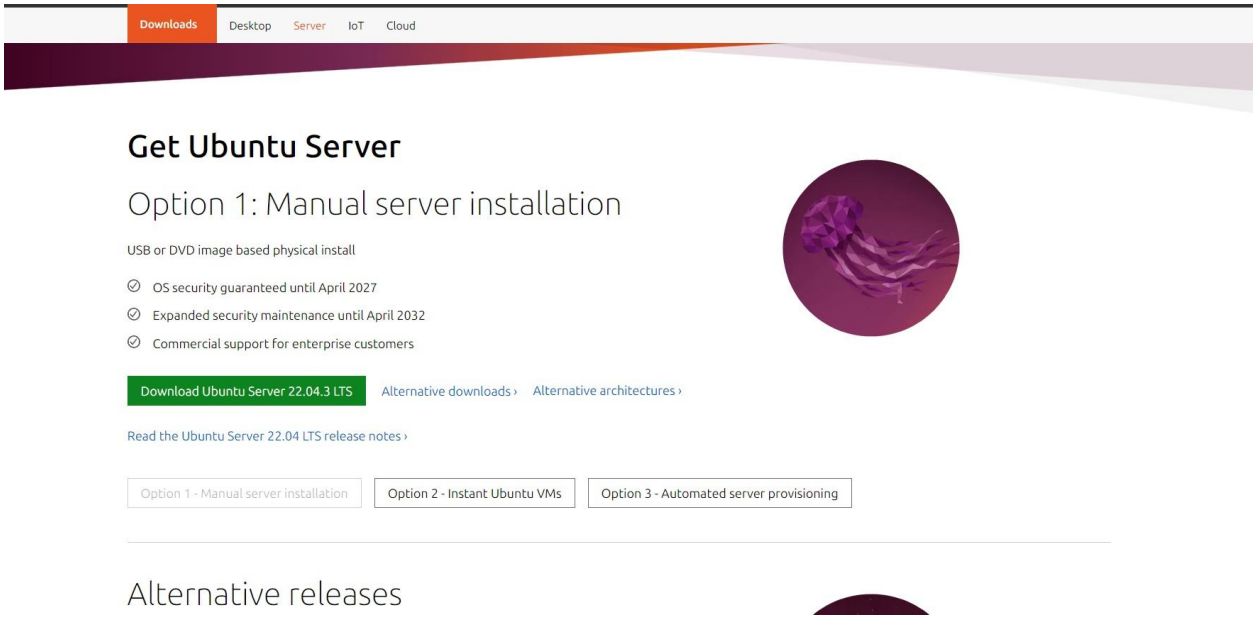


Рис.1 – Официальный сайт Ubuntu

2) Создание виртуальной машины в VirtualBox

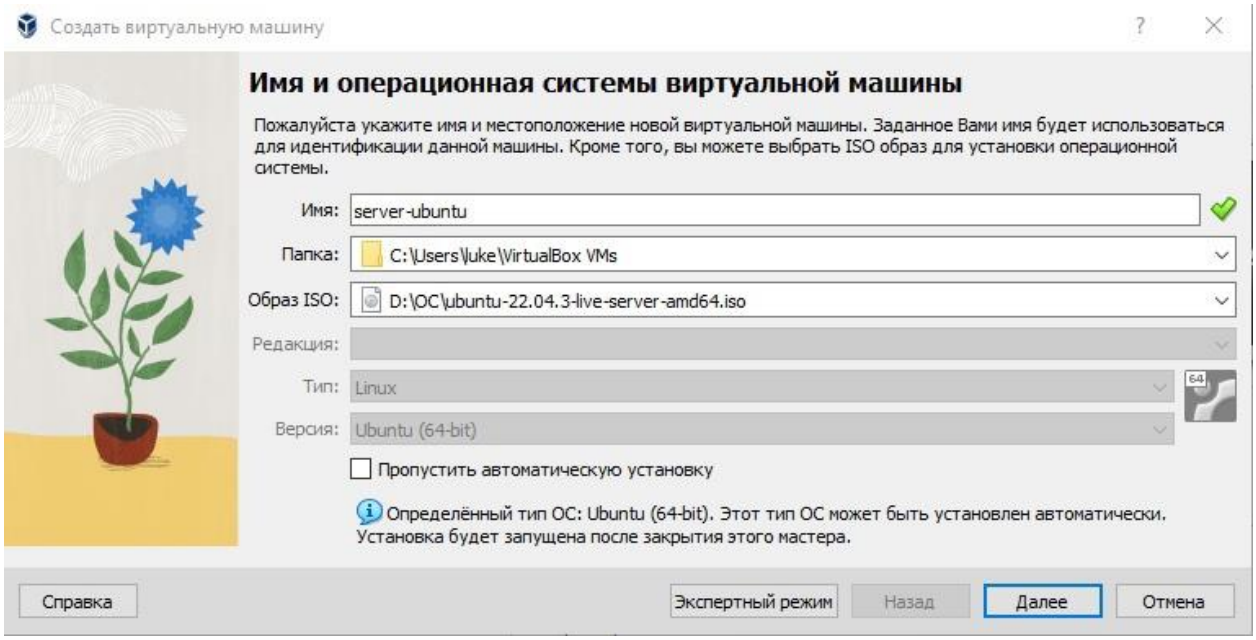


Рис.2 – Выбор iso-файла, папки, в которой находиться виртуальная машина, и имени.

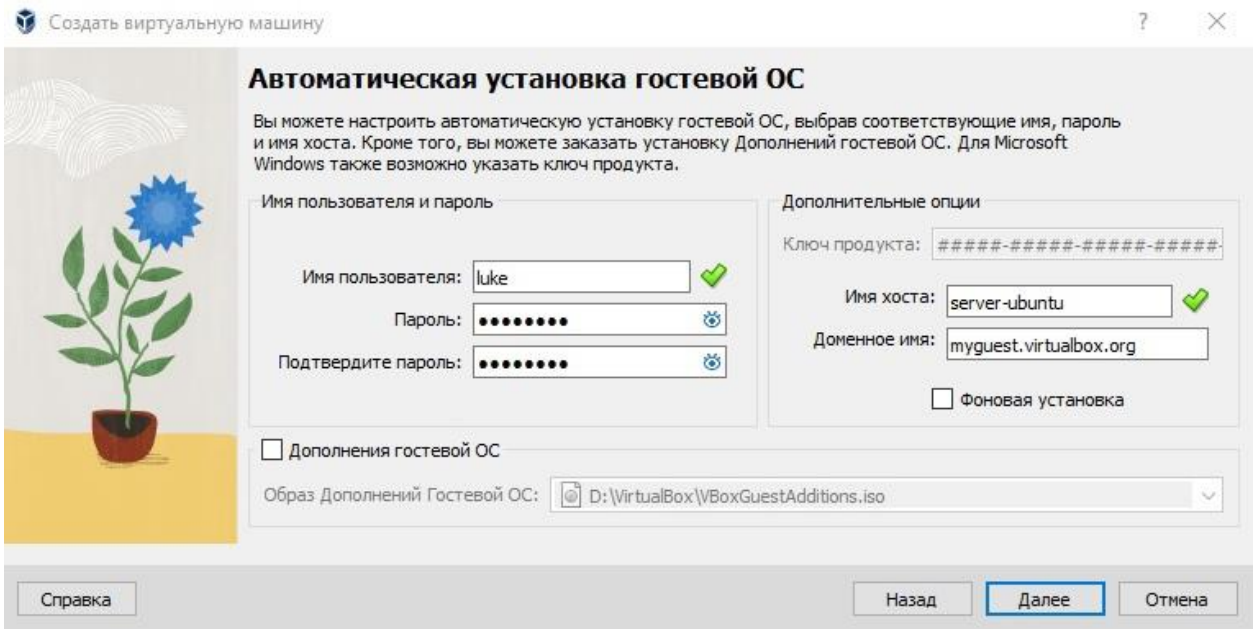


Рис. 3 – Автоматическая установка гостевой ОС

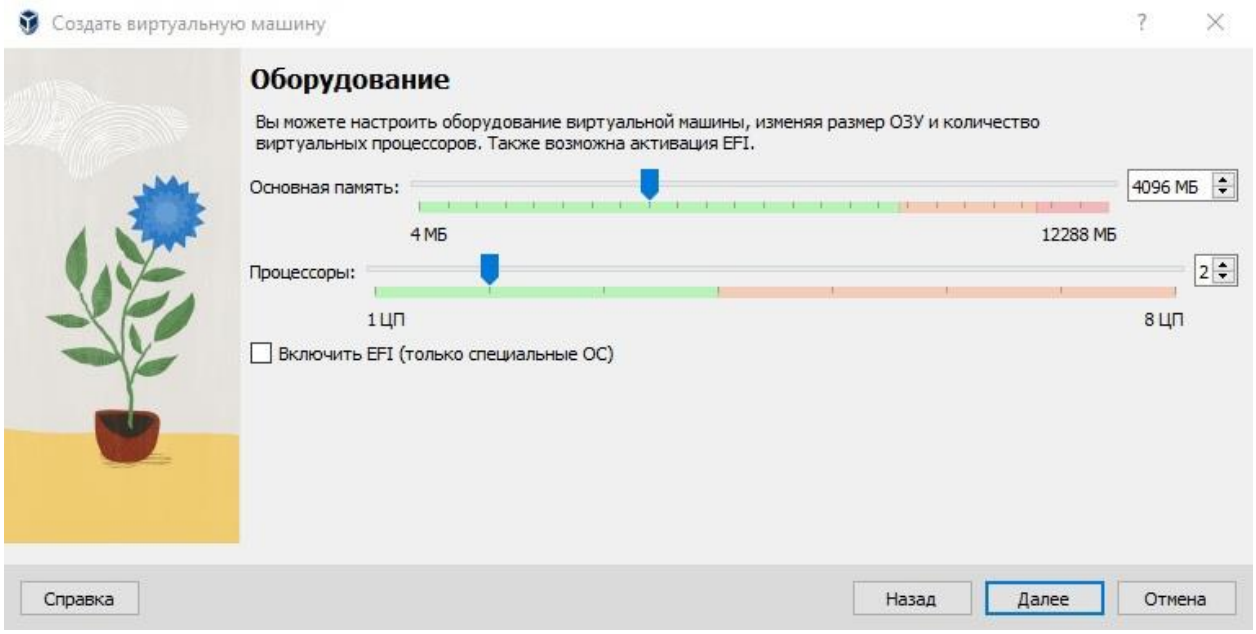


Рис. 4 – Настройка оборудования для виртуальной машины

Выбираем 4 Гб и 2 ЦП для нашей виртуальной машины.

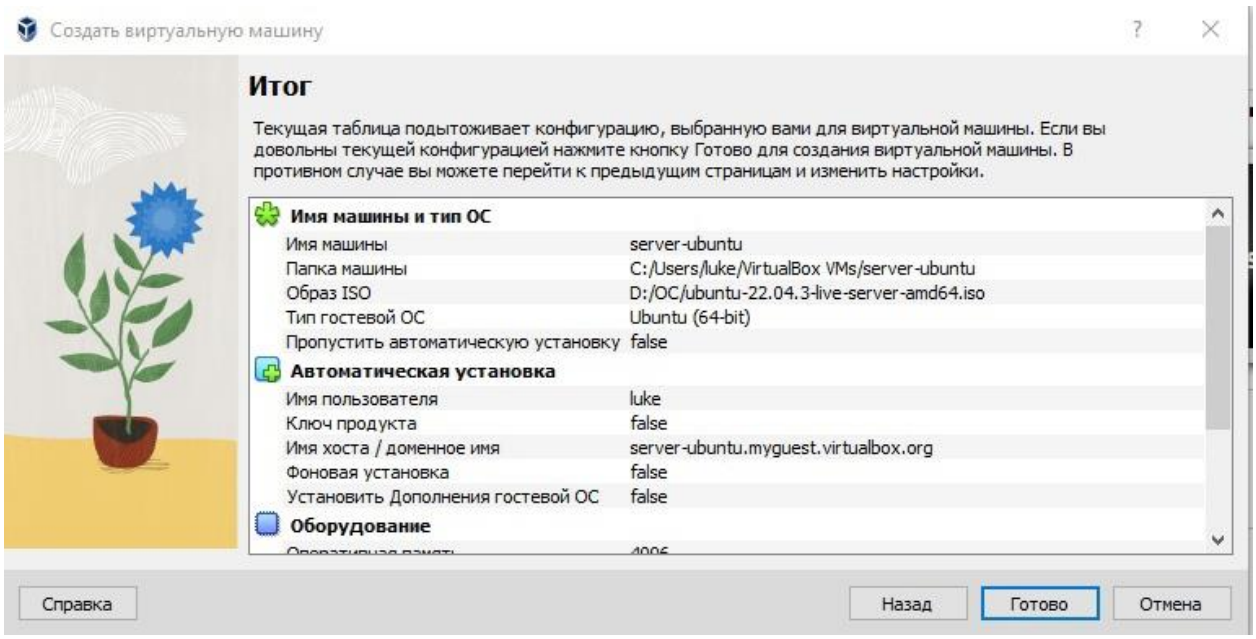


Рис. 5 – Итог

3) Установка и настройка ОС

Запустим виртуальную машину и проведем установку ОС

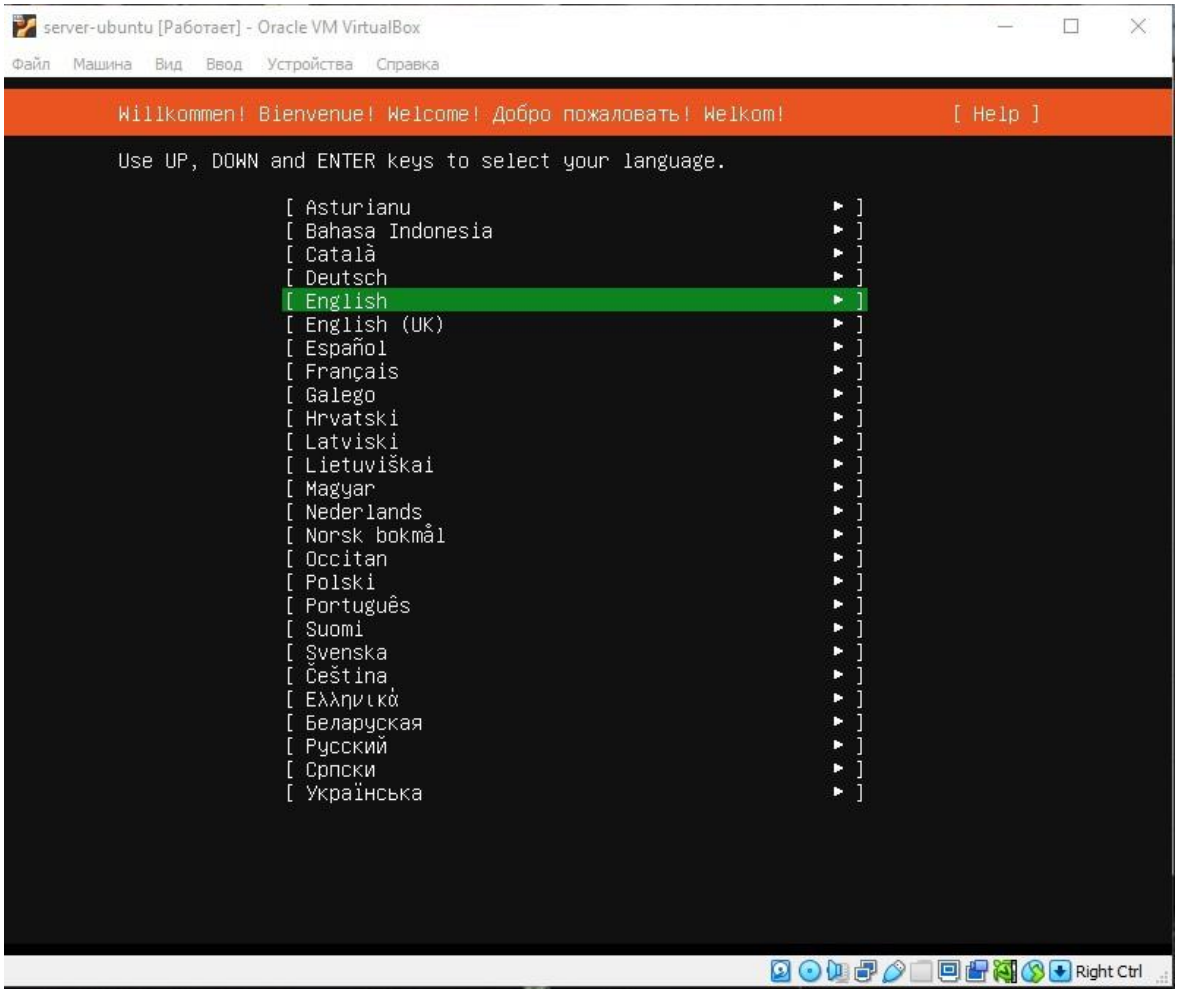


Рис. 6 – Выбор языка

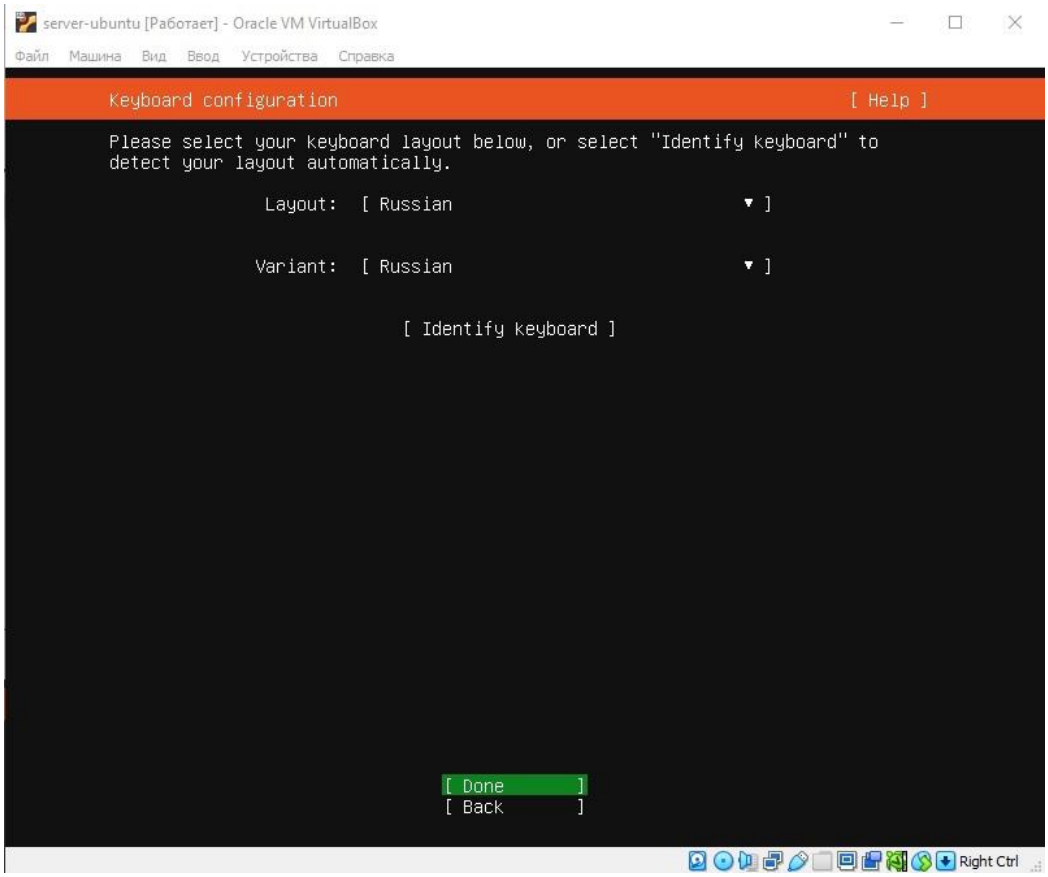


Рис. 7 – Выбор конфигурации клавиатуры

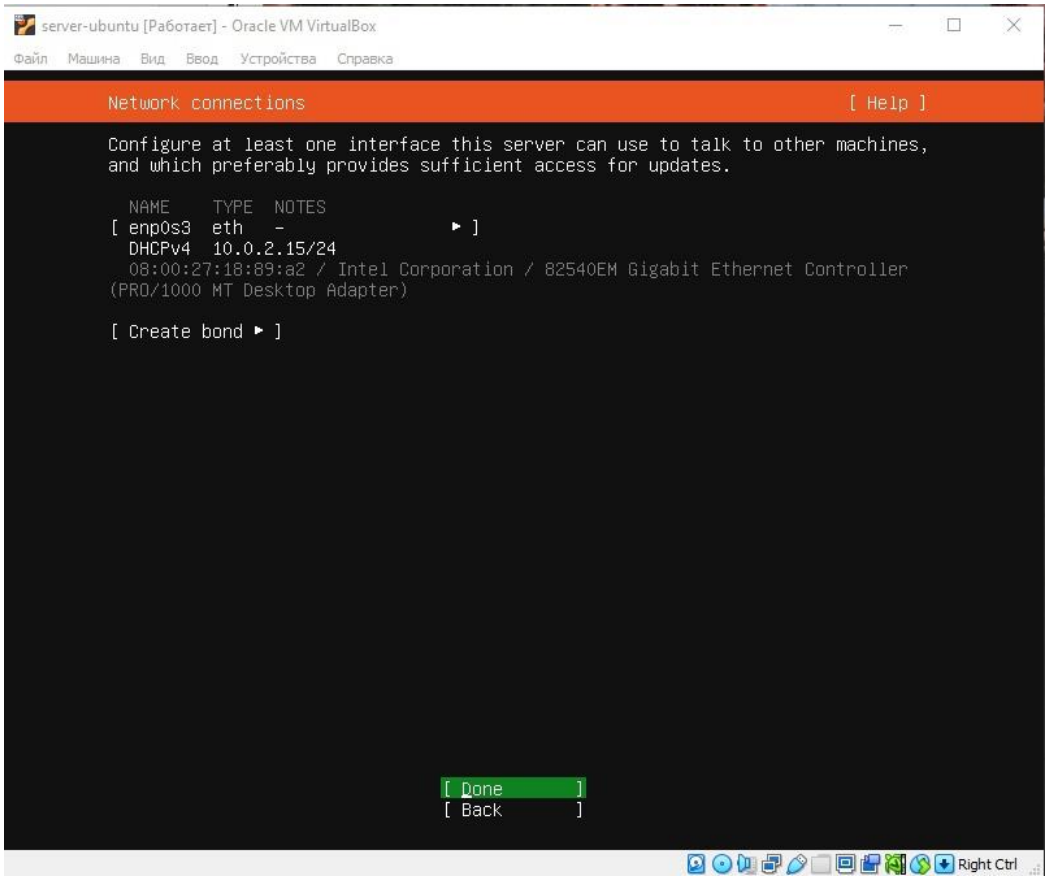


Рис. 8 – Настройка сетевого подключения

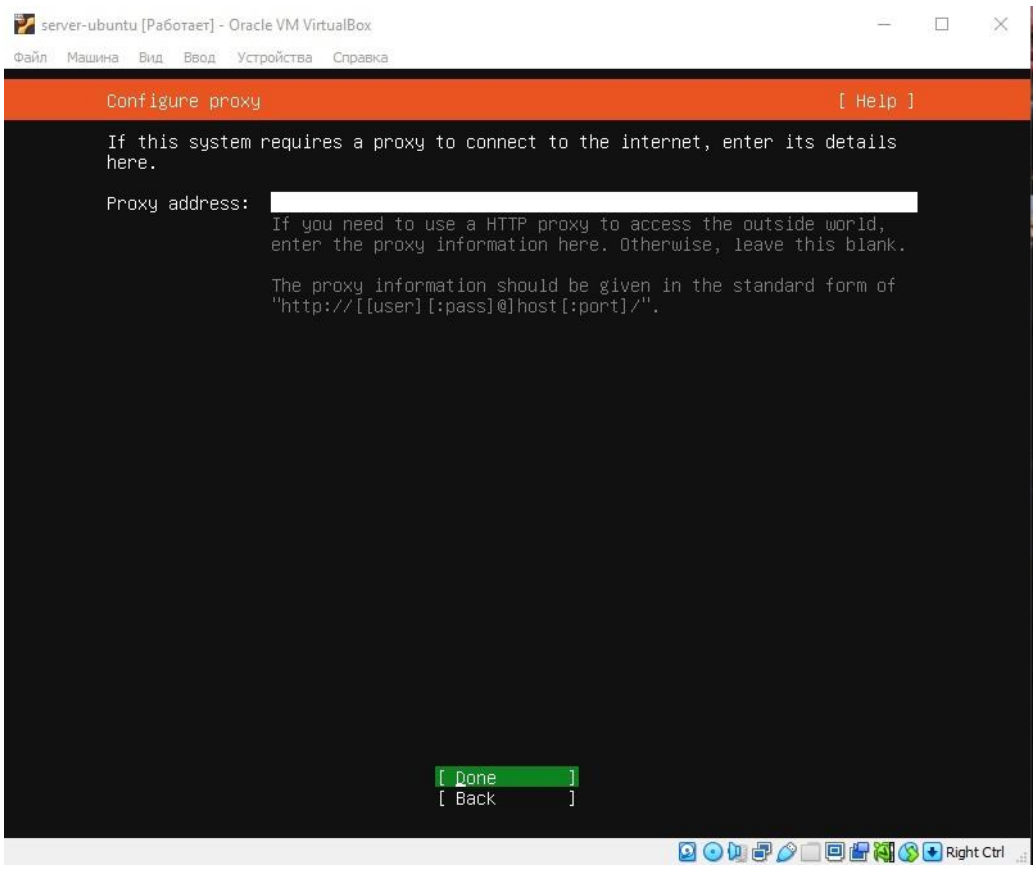


Рис. 9 – Выбор Proxy адреса

Ничего не указываем, потому что Гипервизор автоматически их укажет.

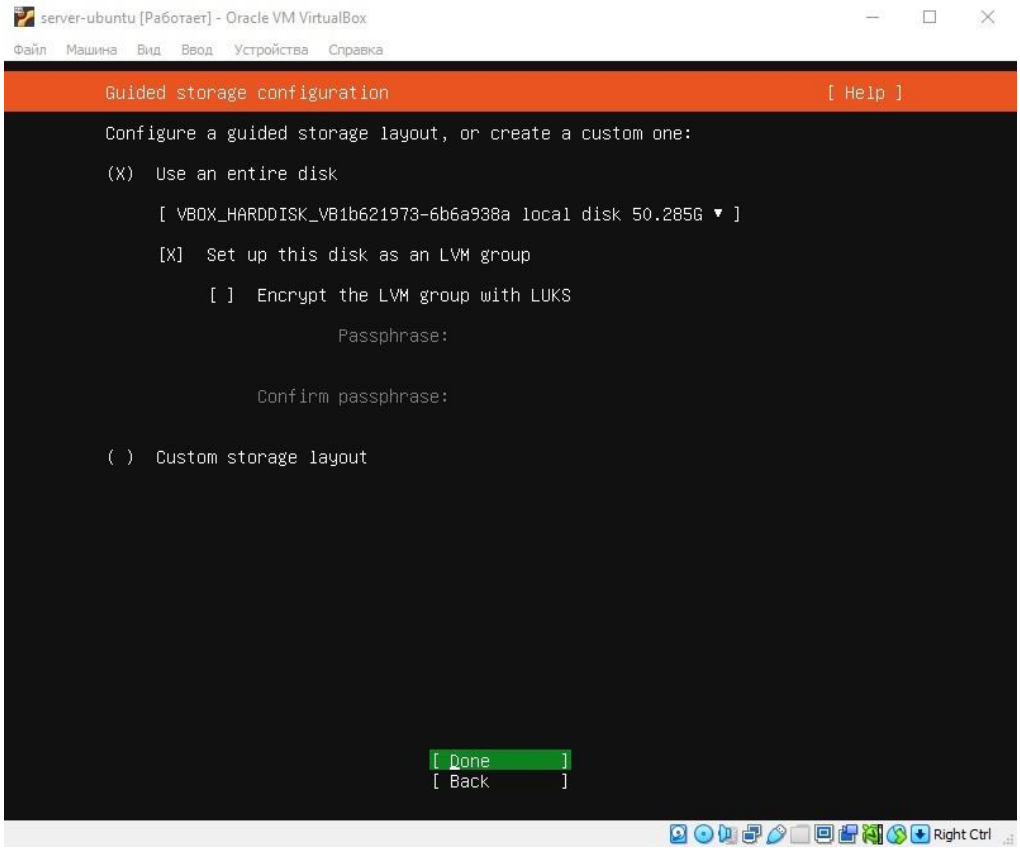


Рис. 10 – Сборка конфигурации памяти

Используем диск созданный VirtualBox

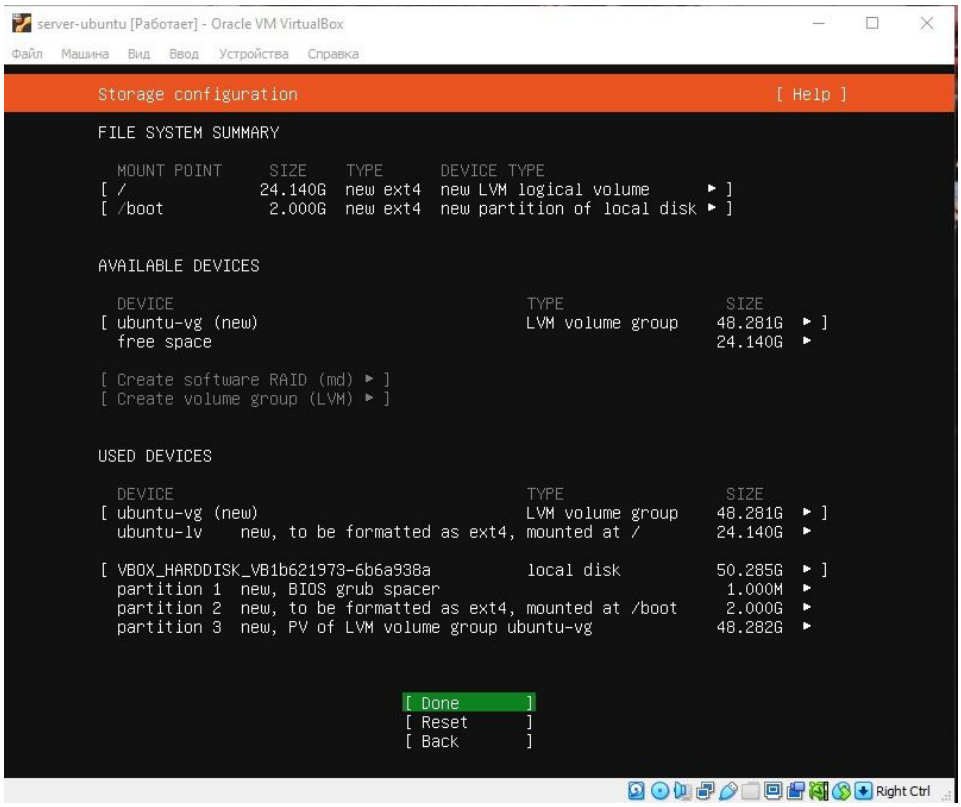


Рис. 11 – Настройка конфигурации памяти

Используем автоматические настройки.

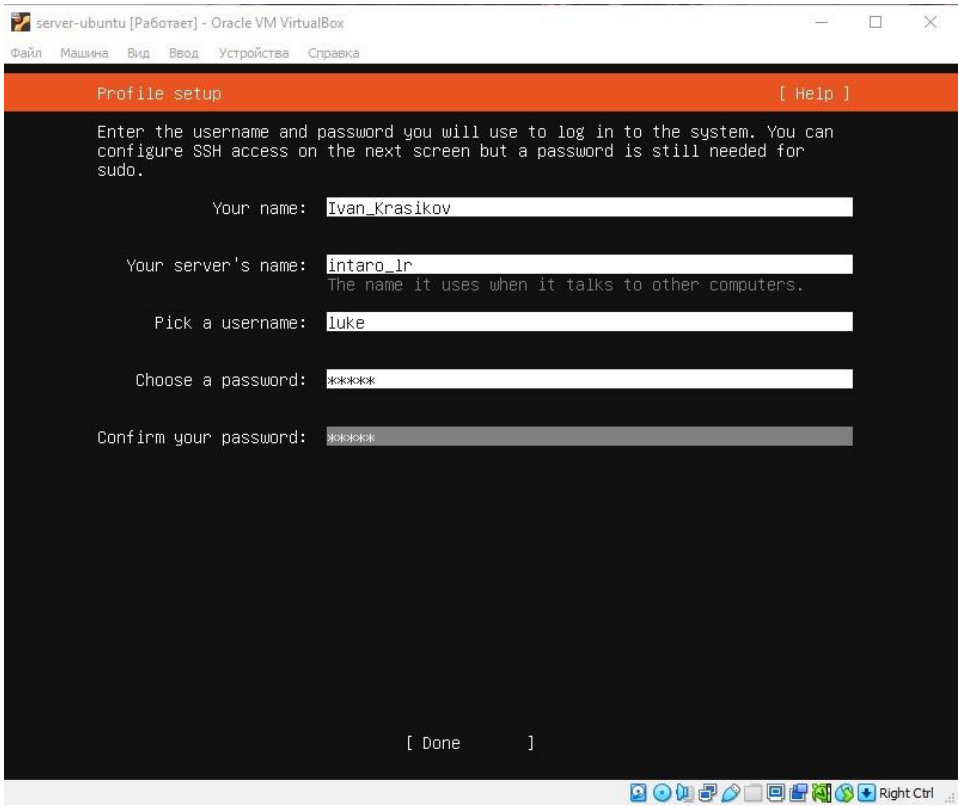


Рис. 12 – Задаем данные пользователя, название сервера, пароль

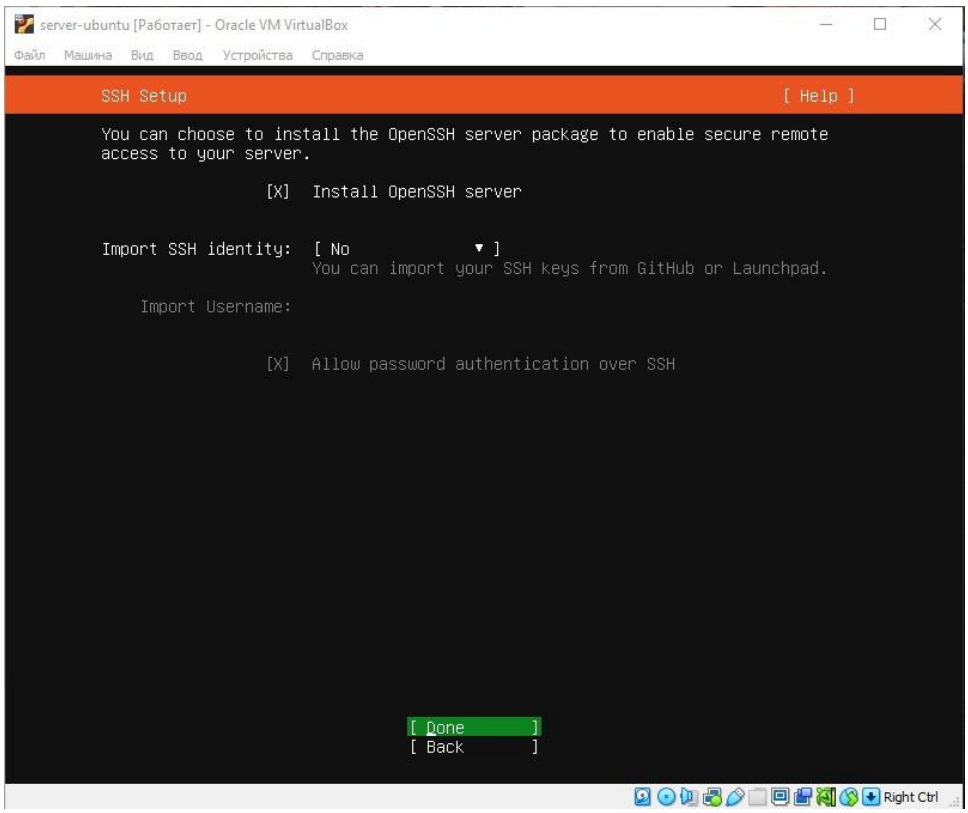


Рис. 13 – Настройка SSH

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

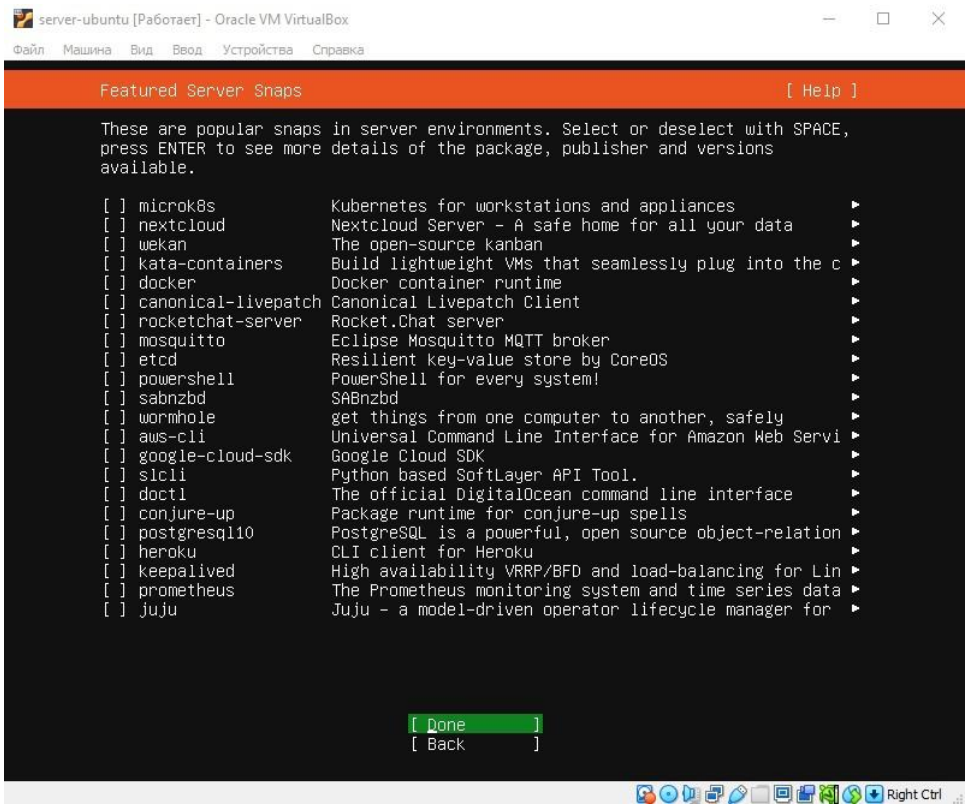


Рис. 14 – Выбор дополнительных пакетов для установки

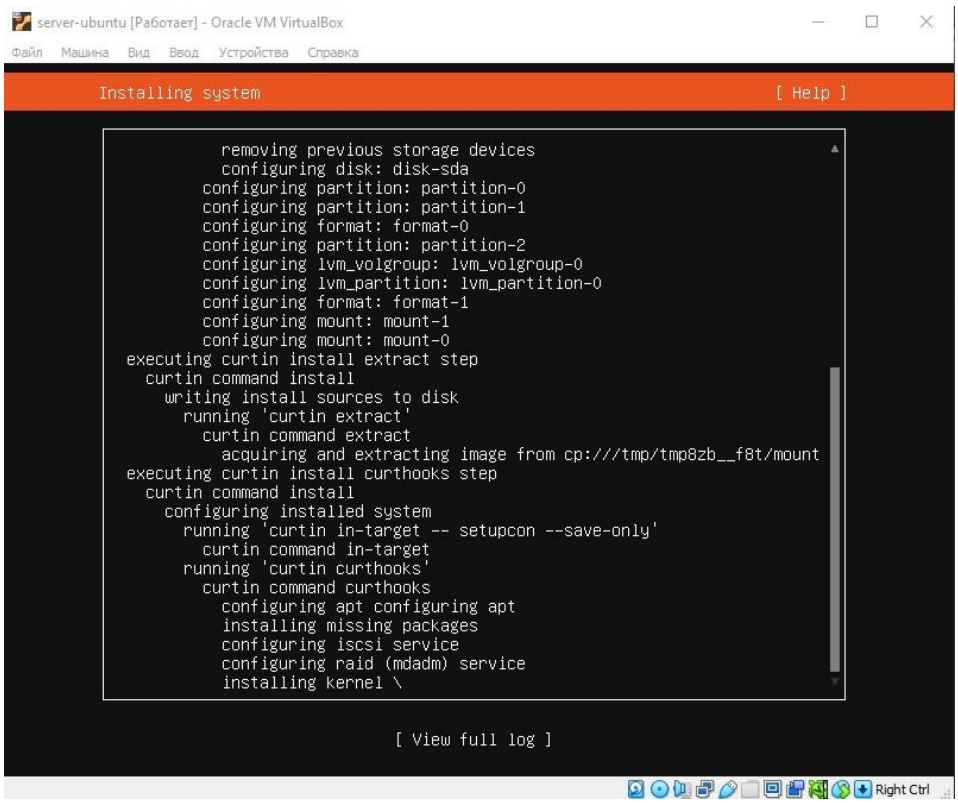


Рис. 15 – Установка ОС

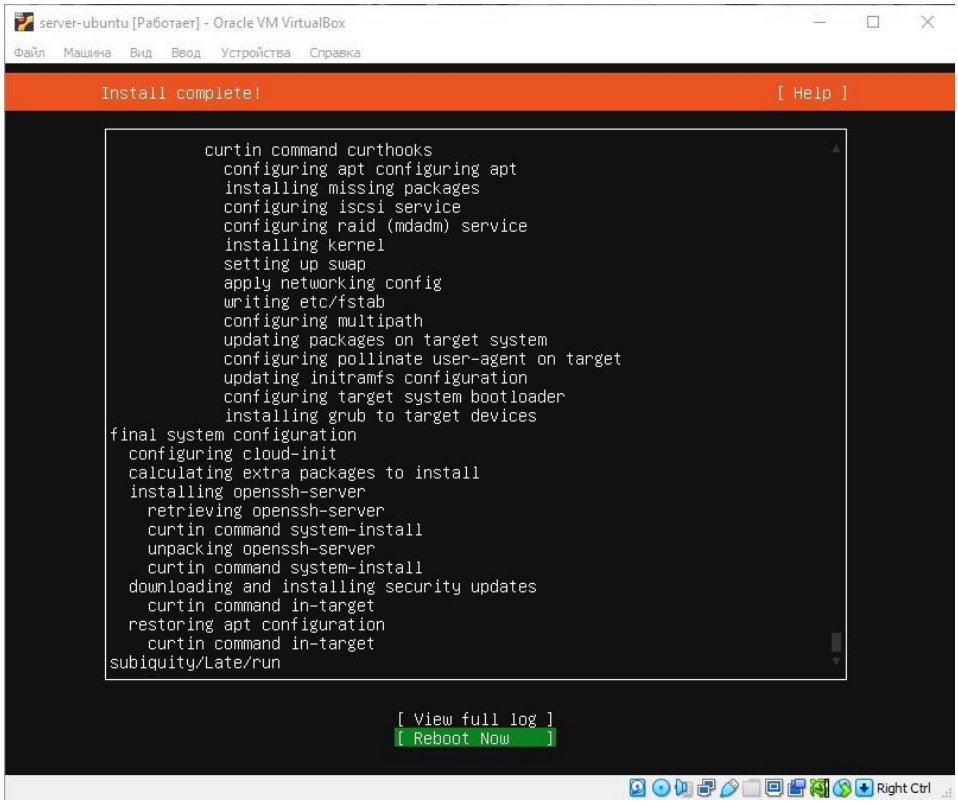


Рис. 16 – Конец установки

Установка закончена, после чего выбираем перезагрузку.

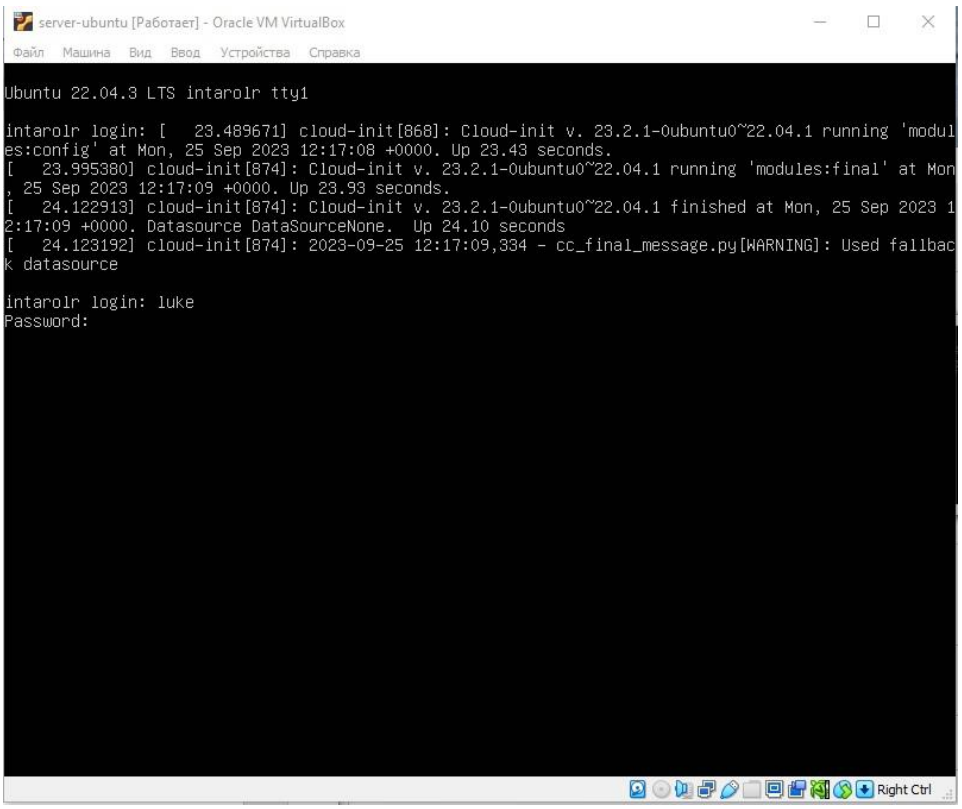


Рис. 17 – Вход в систему

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

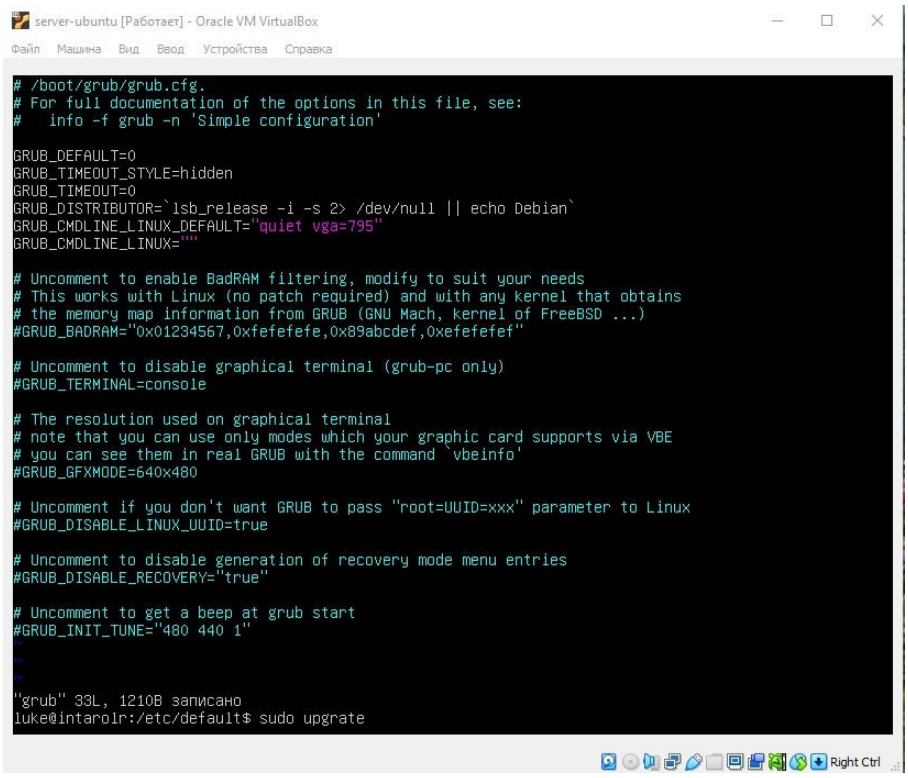


Рис. 18 – Файл grub

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

```
luke@intarolr:/home$ cd luke
luke@intarolr:~$ mkdir lr1 && cd lr1
luke@intarolr:~/lr1$
```

```
luke@intarolr:~/lr1$ touch main.c
luke@intarolr:~/lr1$ ls -ln
total 0
-rw-rw-r-- 1 1000 1000 0 сен 25 18:00 main.c
luke@intarolr:~/lr1$
```

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

```c
#include <stdio.h>
#include <stdlib.h>


void zadacha(int* arr, int N){
    int count = 0;
    for(int i = 0; i < N; i++){
        if(arr[i] % 2 == 0){
           int tmp = arr[count];
           arr[count] = arr[i];
           arr[i] = tmp;
           count++;
        }
    }
}


int main(){
    int N;
    printf("N: ");
    scanf("%d", &N);
    int* arr = malloc(N*sizeof(int));
    for(int in = 0; in < N; in++){
       scanf("%d", &arr[in]);
    }

    zadacha(arr, N);

    for(int out = 0; out < N; out++){
       printf("%d ", arr[out]);
    }
    printf("\n");
    free(arr);
    return 0;
}
~
```

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.

```
luke@intarolr:~/lr1$ touch makefile
luke@intarolr:~/lr1$ ls -ln
total 4
-rw-rw-r-- 1 1000 1000 595 сен 25 18:36 main.c
-rw-rw-r-- 1 1000 1000   0 сен 25 18:40 makefile
luke@intarolr:~/lr1$ _
```

```
zadacha: main.c
	gcc -o zadacha -g main.c
~
~
~
~
```

Теперь мы можем написать команду make и наша программа соберется.

```
luke@intarolr:~/lr1$ make
gcc -o zadacha -g main.c
luke@intarolr:~/lr1$ ls -l
total 28
-rw-rw-r-- 1 luke luke   595 сен 25 18:36 main.c
-rw-rw-r-- 1 luke luke    42 сен 25 18:41 makefile
-rwxrwxr-x 1 luke luke 18168 сен 25 18:42 zadacha
luke@intarolr:~/lr1$
```

Запускаем программу через команду ./zadacha

```
luke@intarolr:~/lr1$ ./zadacha
N: 9
1 3 4 6 8 9 1 3 6
4 6 8 6 1 9 1 3 3
luke@intarolr:~/lr1$ _
```

3) Отладка

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

```
luke@intarolr:~/lr1$ gdb zadacha
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from zadacha...
(gdb)
```

Рис. 20 – Процесс отладки

С помощью команды run можем запустить программу

```
(gdb) run
Starting program: /home/luke/lr1/zadacha
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
N: 6
1 2 3 4 5 6
2 4 6 1 5 3
[Inferior 1 (process 1538) exited normally]
(gdb)
```

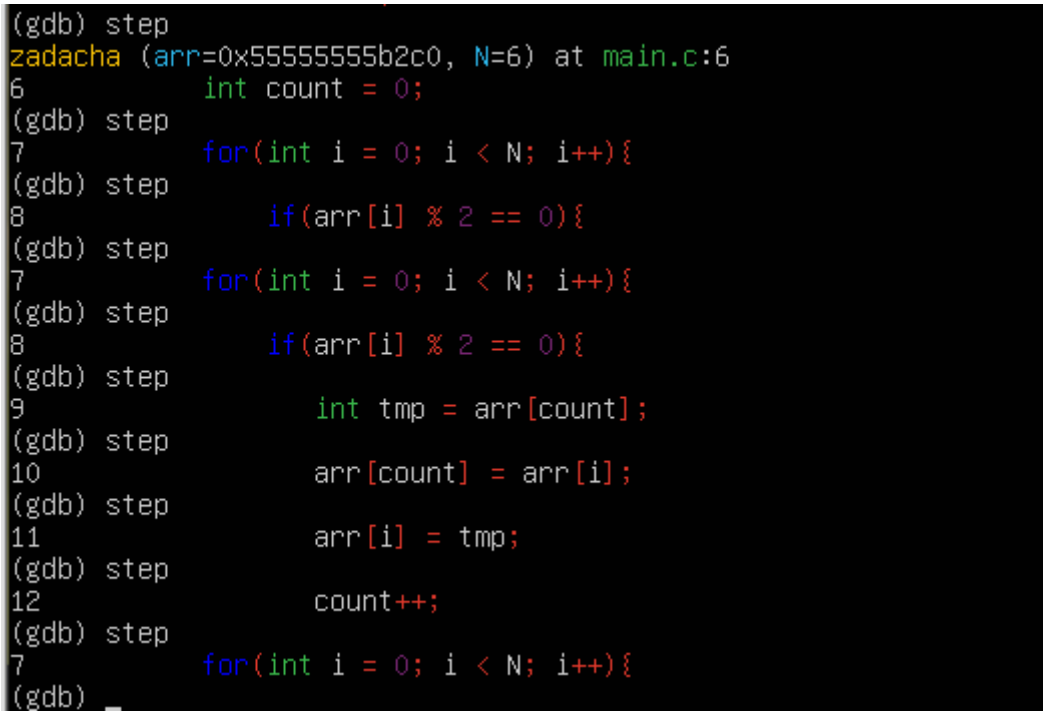
С помощью команды list можно вывести код программы, но по умолчанию выводит только 10 строк, для снятия ограничения нужно вписать команду set list unlimited.

```
(gdb) set listsize unlimited
(gdb) list
Line number 36 out of range; main.c has 35 lines.
(gdb) list 35
1       #include <stdio.h>
2       #include <stdlib.h>
3
4
5       void zadacha(int* arr, int N){
6           int count = 0;
7           for(int i = 0; i < N; i++){
8               if(arr[i] % 2 == 0){
9                   int tmp = arr[count];
10                  arr[count] = arr[i];
11                  arr[i] = tmp;
12                  count++;
13              }
14          }
15      }
16
17
18      int main(){
19          int N;
20          printf("N: ");
21          scanf("%d", &N);
22          int* arr = malloc(N*sizeof(int));
23          for(int in = 0; in < N; in++){
24              scanf("%d", &arr[in]);
25          }
26
27          zadacha(arr, N);
28
29          for(int out = 0; out < N; out++){
30              printf("%d ", arr[out]);
31          }
32          printf("\n");
33          free(arr);
34          return 0;
35      }
(gdb)
```

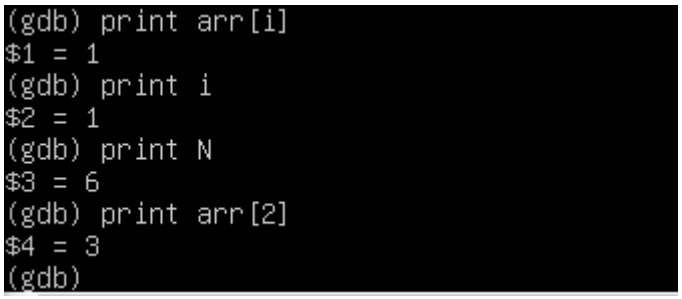
Рис. 21 – Вывод кода программы через gdb

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

С 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 С заключается в том, что программа делиться на несколько файлов и подключается через #include.

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

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.