Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированные системы управления

Лабораторная работа №1

по Linux-server

«Средства разработки программного обеспечения в ОС Linux»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент |  |  |  | Кретов И.О. |
| Группа С-АС-22-1 |  |  |  |  |
| Руководитель |  |  |  | Кургасов В.В. |

Липецк 2023 г.

# Цель работы:

 Установить гипервизор, изучить понятия: Виртуализация, гипервизоры, изучить создание виртуальных машин.

- Установить 64 серверную ОС на выбор (Ubuntu, Debian)

- ЛР№3 (из предложенного пособия - 1 задача по варианту + ответы на контрольные вопросы) :

⦁ Изучить компиляцию и компоновку программ на примере инструментального средства разработки программ GCC (http://gcc.gnu.org).

⦁ Изучить сборку проекта с использованием make (⦁ http://www.gnu.org/software/make/).

⦁ Изучить возможности отладки программ с использованием GDB (⦁ http://www.gnu.org/s/gdb/).

⦁ Ознакомиться с интегрированными средами разработки и используемыми редакторами, о которых идет упоминание в работе. Выполнить сравнительную характеристику по выбранным критериям (не менее 8). Критерии сравнения выбираются самостоятельно, однако рекомендуется включить условие использования (тип лицензии), поддерживаемые ОС, официальный сайт разработчика, перечень поддерживаемых языков разработки, номер и дату выпуска последней версии.

⦁ Установить (если отсутствует в системе) редактор Vim. Провести его настройку, включая установку не менее 5 плагинов.

Напишите программу в соответствии с назначенным вариантом, используя редактор Vim.

⦁ Продемонстрируйте работу отладчика на примере написанной вами программы.

⦁ Подготовьте отчет о выполненной работе, содержащий скриншоты с описанием всех основных этапов

Задача вариант № 4:

1. Задать произвольное число k в интервале 2… N-2. Уменьшить все элементы, стоящего после k-го, на единицу.
2. **Установка виртуальной машины Virtual Box.**

Сначала загрузите установочный файл VirtualBox с официального сайта Oracle VirtualBox: ссылка на скачивание.

<https://www.virtualbox.org/wiki/Downloads>



После загрузки запустите установочный файл и следуйте инструкциям мастера установки. Выберите все параметры, подходящие для вашей операционной системы.

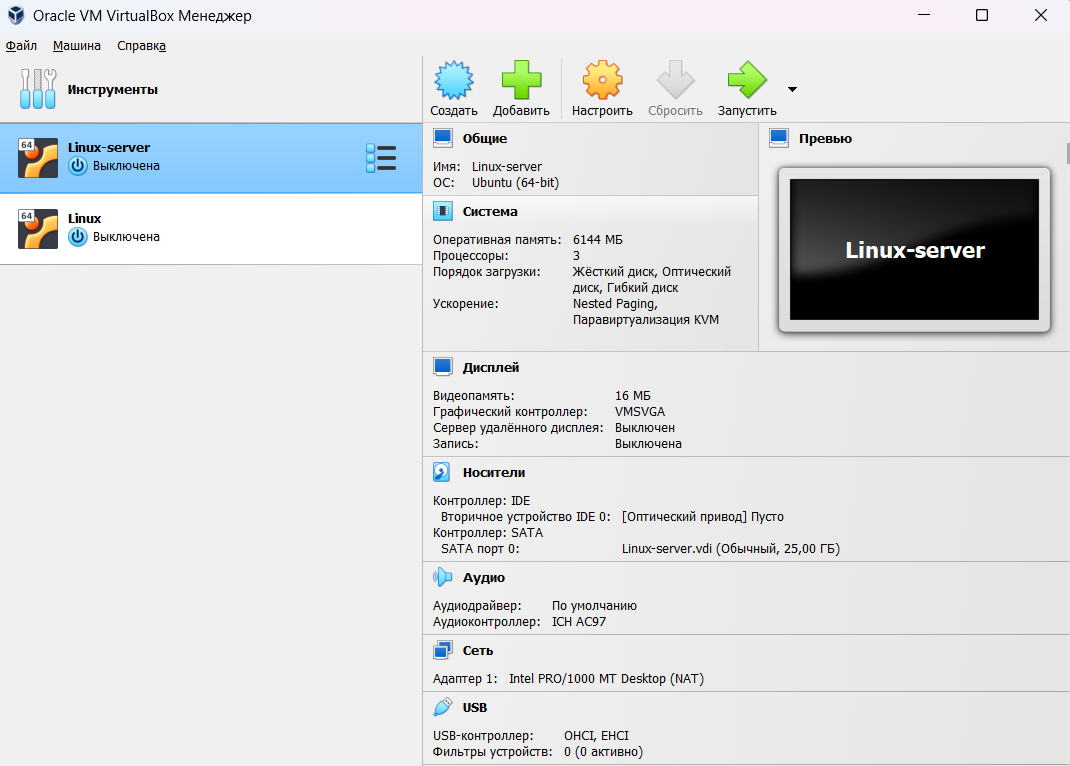
При работе с виртуальными машинами нельзя не затронуть такое понятие, как виртуализация.

Итак, виртуализация – это сокрытие конкретной реализации за универсальным стандартизованным методом обращения к ресурсам. Иными словами, это создание абстракции над аппаратным обеспечением.

Существует много видов виртуализации, однако можно выделить три основных:

* Аппаратная виртуализация.  
  Позволяет создавать независимые и изолированные друг от друга виртуальные компьютеры с помощью программной имитации ресурсов (процессора, памяти, сети, диска и др.) физического сервера. Физический сервер называют хостовой машиной (хостом), виртуальные компьютеры – виртуальными машинами, ВМ (иногда их также называют гостями). Программное обеспечение, которое создает виртуальные машины и управляет ими, называют гипервизором (а также виртуальным монитором или контрольной программой – вспомните CP-40 из начала статьи). На практике на виртуальных машинах могут использоваться разные ОС для разных целей – например, Windows Server под контроллер домена Active Directory и Debian под веб-сервер NGINX.
* Виртуализация рабочих столов.  
  Позволяет отделить логический рабочий стол (набор пользовательских программ, работающий под ОС) от физической инфраструктуры (например, персональных компьютеров). Одной из наиболее распространенных форм виртуализации рабочих столов является VDI (Virtual Desktop Infrastructure) – инфраструктура виртуальных рабочих столов. Каждый пользователь VDI имеет программную имитацию ОС с необходимым набором программ на физическом сервере под управлением гипервизора и может подключаться к ней по сети. На практике VDI может использоваться для работы большого количества сотрудников на «удаленке» для того, чтобы не закупать им отдельные рабочии станции и управлять инфраструктурой централизованно.
* Виртуализация на уровне ОС (контейнеризация).  
  Позволяет запускать программное обеспечение в изолированных на уровне операционной системы пространствах. Наиболее распространенной формой виртуализации на уровне ОС являются контейнеры (например, Docker). Контейнеры более легковесны, чем виртуальные машины, так как они опираются на функционал ядра ОС и им не требуется взаимодействовать с аппаратным обеспечением. На практике контейнеры представляют из себя изолированную среду для запуска любого приложения со всеми его зависимостями и настройками.

После успешной установки VirtualBox запустите программу. В меню выберите "Файл" -> "Создать виртуальную машину". Задайте имя, тип ОС (Linux) и выберите версию Linux (64-битную) при создании новой виртуальной машины.



Экран установленного образа linux-server на Virtual Box.

 В разделе “Сеть” выбираю тип подключения “Сетевой мост” для дальнейшей работы с ssh.

1. **Установка Linux – сервера.**

Скачайте 64-битный образ ISO для выбранной вами версии Linux-сервера с официального сайта дистрибутива. Обратимся к официальному сайту Linuc Ubuntu ссылка:

<https://ubuntu.com/download/server>

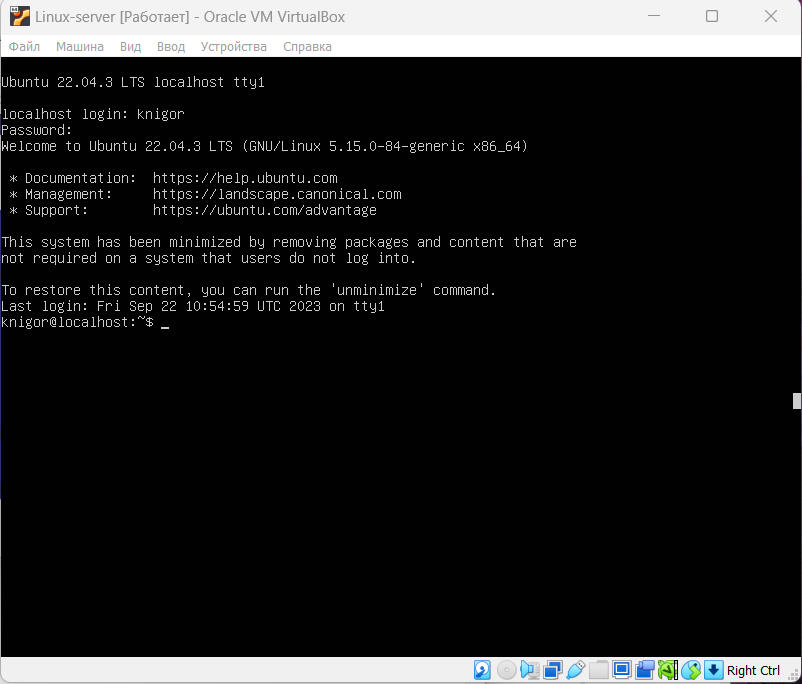
Перед запуском виртуальной машины укажите ей параметры, такие как количество выделенной оперативной памяти, количество процессоров и создайте виртуальный жесткий диск, если необходимо.

Запустите виртуальную машину и начните процесс установки Linux. Загрузите образ ISO, который вы скачали ранее, в виртуальный CD/DVD-привод виртуальной машины. Затем перезагрузите виртуальную машину, чтобы начать установку.

Следуйте инструкциям на экране, выбирая язык, раскладку клавиатуры, разметку диска и другие необходимые параметры. Создайте учетную запись пользователя и установите пароль.

После завершения установки Linux-сервера дождитесь, пока система завершит настройку.

После успешной установки войдите в систему, используя созданную учетную запись пользователя и пароль. Теперь ваша виртуальная машина готова для дальнейшей настройки и использования.



Экран успешного входа

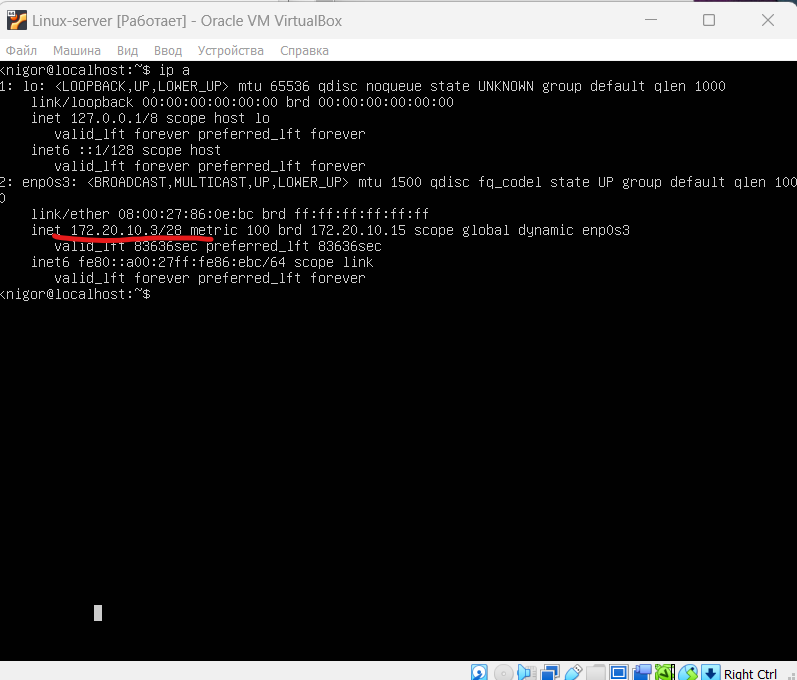
1. **Подключение к ssh-server.**

OpenSSH - это свободная и открытая реализация протокола Secure Shell (SSH), который обеспечивает защищенное удаленное управление компьютерами и передачу файлов по сети. OpenSSH обеспечивает шифрование данных, аутентификацию пользователей и безопасное выполнение команд на удаленных машинах. Это одно из наиболее популярных программных решений для безопасного удаленного доступа и используется во многих операционных системах.

Для того чтобы подключиться к локальной машине по ssh нам нужно установить утилиту mobaXterm на основную ОС ссылка:

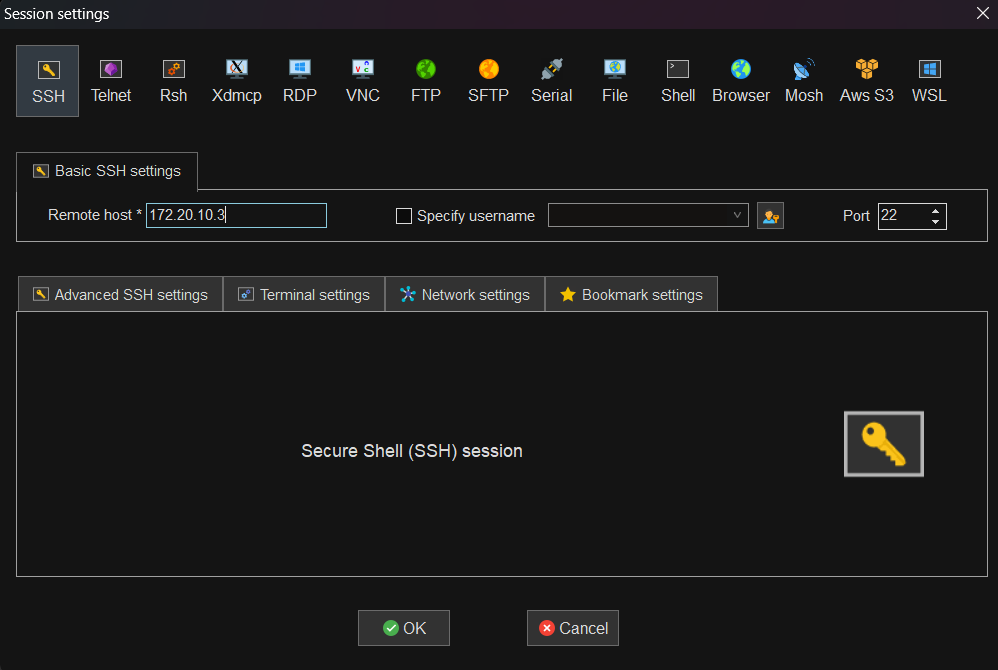
<https://mobaxterm.mobatek.net/>

Дальше нужно настроить подключение, на сервере прописать команда ip a , а затем узнать свой ip.



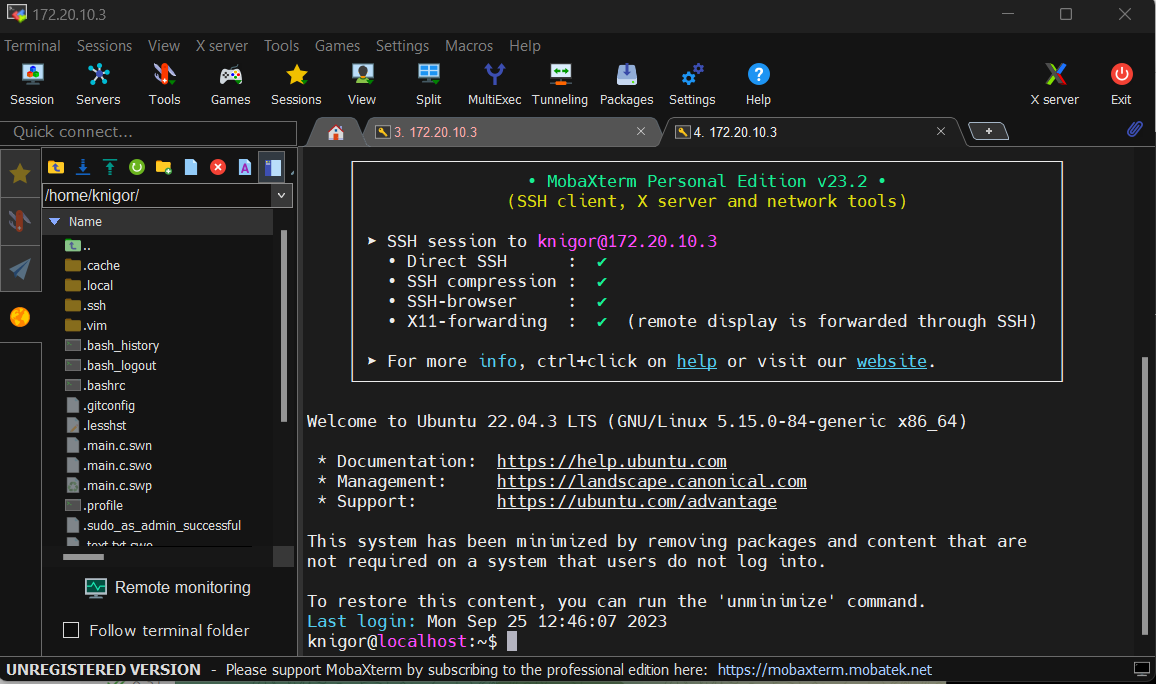
Ip – адрес.

В программе mobaXterm открыть вкладку session и перейти в SSH, а затем в открывшемся окне установить подключение к серверу.(Рисунок 4)



Подключение.

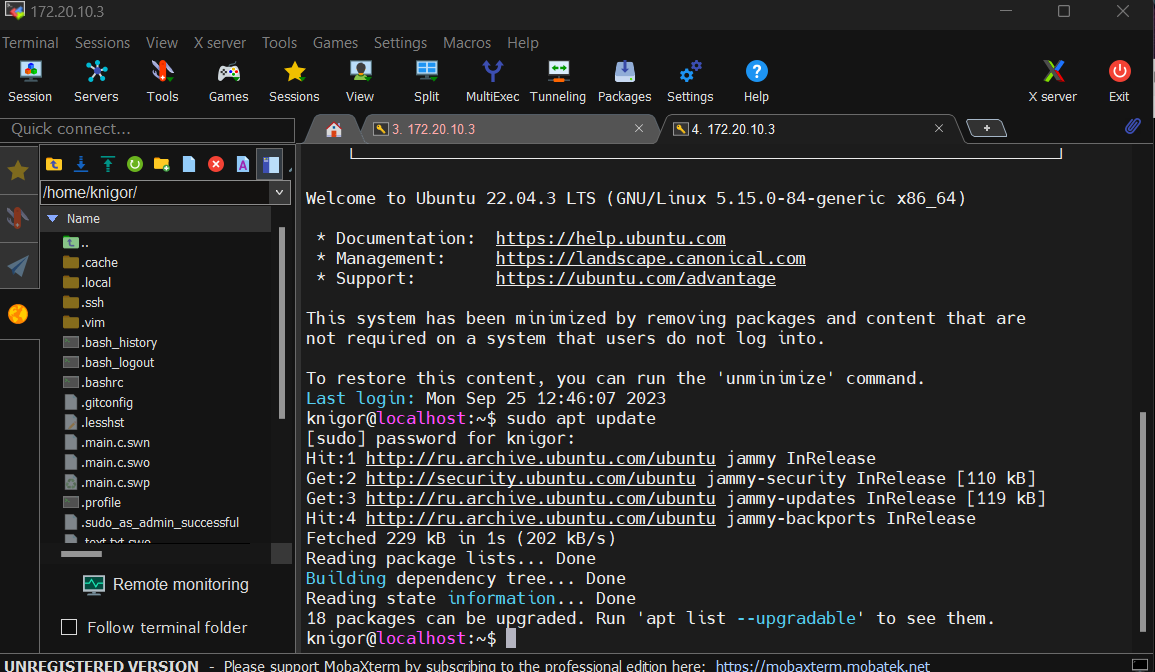
После успешного подключения установить соединение с клиента на адрес серевера, выполнить login(Рисунок 5)



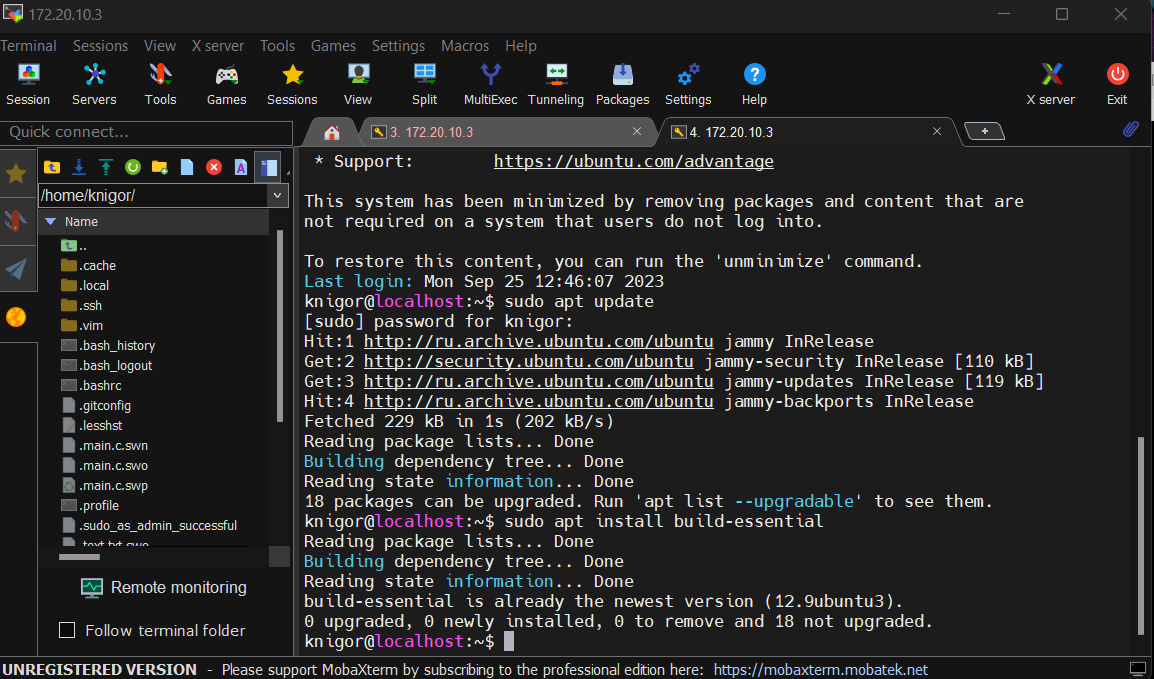
Консоль разработчика(успешное подключение)

1. **Выполнение ЛР №3.**
2. **Установка компилятора gcc и настройка его.**

Первым делом нам нужно обновить информацию о доступных пакетах в репозиториях. Прописать команду sudo apt update



Для установки компилятора GCC на Linux-сервер, установим пакет build-essential, который содержит GCC, выполнив команду: sudo apt install build-essential

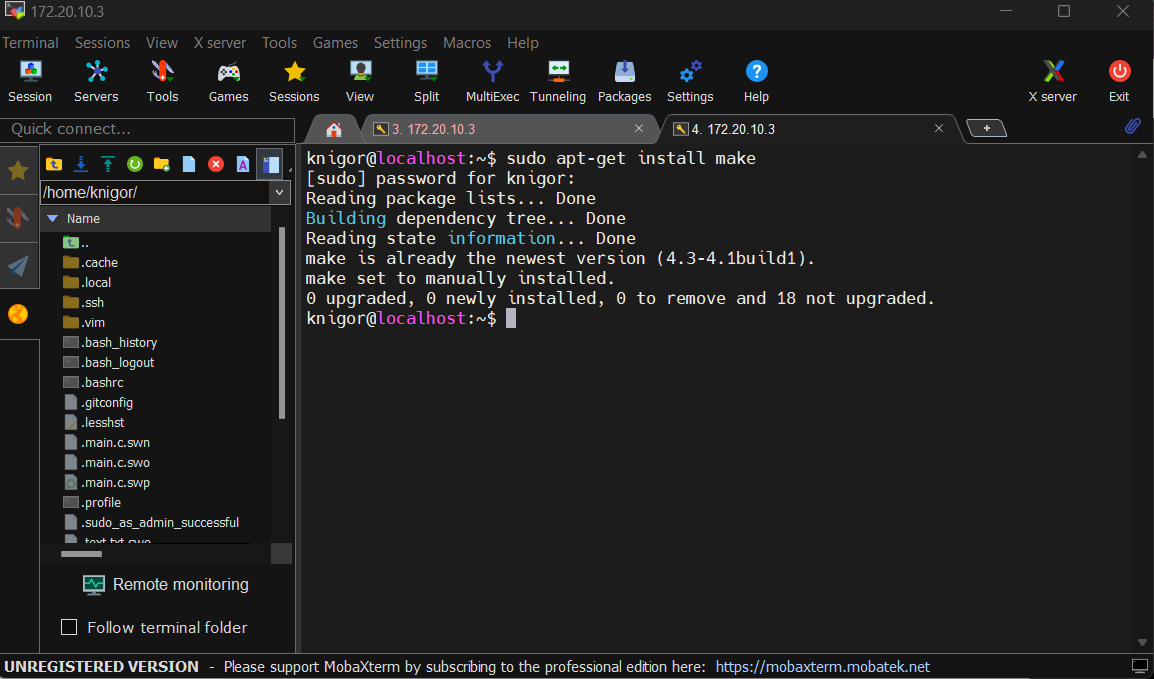


1. **Сборка проекта с помощью make**

Make - это инструмент для автоматизации сборки проектов на языке программирования Cи. Он используется для создания исполняемых файлов из исходных кодов. Make читает специальный файл, называемый Makefile, который содержит информацию о том, какие файлы нужно скомпилировать и как их скомпилировать. Затем Make выполняет команды, указанные в Makefile, чтобы собрать проект.

Для установки make необходимо выполнить следующую команду:

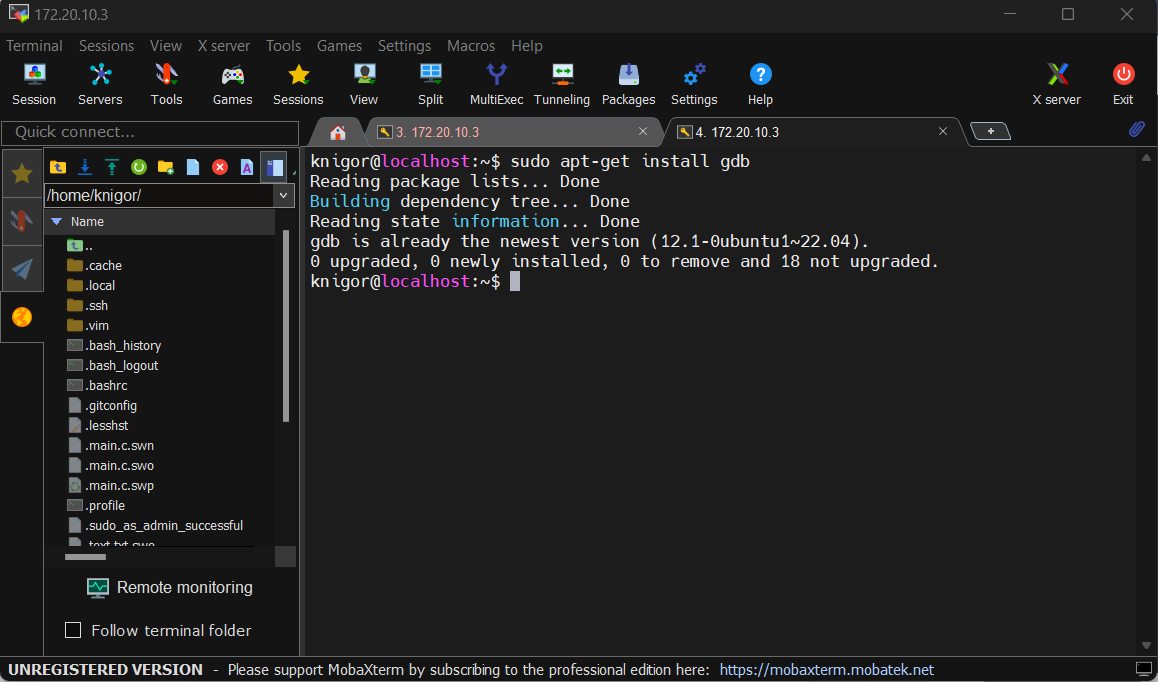
sudo apt-get install make



1. **Отладка программ с помощью GDB.**

GDB (GNU Debugger) - это отладчик программ, написанных на языке Си. Он позволяет отлаживать программы, то есть находить и исправлять ошибки в коде. Для установки GDB необходимо выполнить следующую команду:

sudo apt-get install gdb

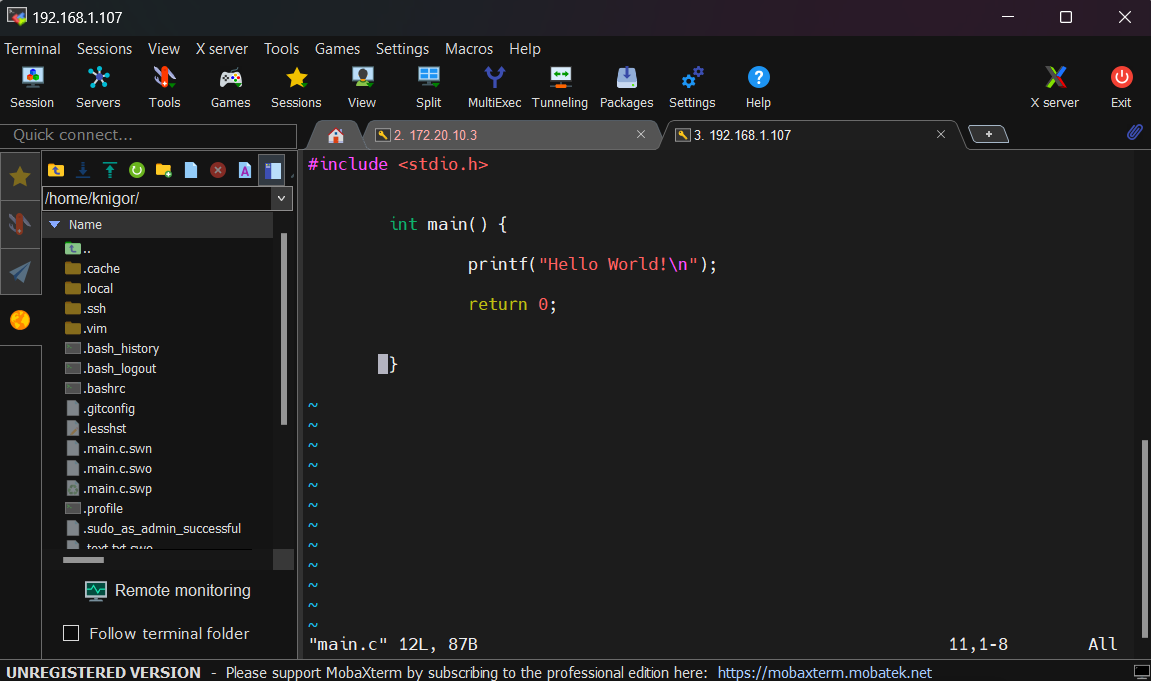


1. **Сравнительная характеристика IDE.**

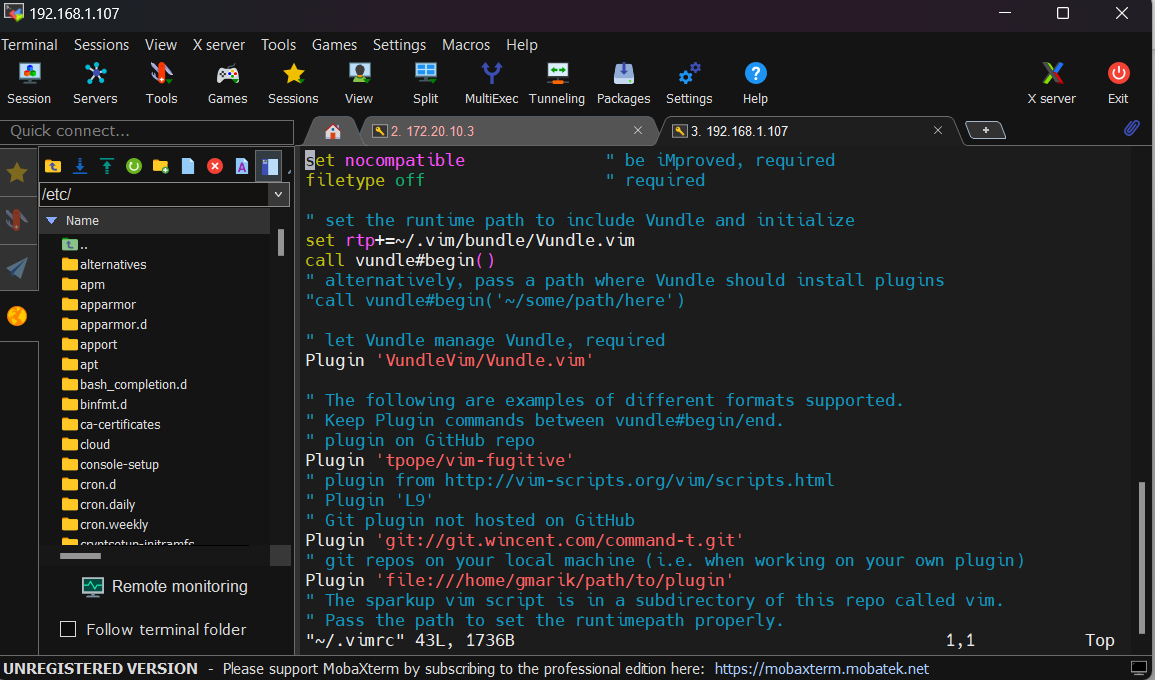
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IDE** | **Перечень поддерживаемых языков** | **Дата выпуска последней версии** | **Официальный сайт** | **Цена** | **Интерфейс** | **Интеграция с системами контроля версий** |
| CodeBlocks | C++, PHP, Python, Vala, Rascal | 2019 | [https://www.codeblocks.org](https://www.codeblocks.org/) | Бесплатно | Устаревший | Да |
| Visual Studio Code | JavaScript, TypeScript, C#, Java и другие | 2020 | [https://code.visualstudio.com](https://code.visualstudio.com/) | Бесплатно | Современный | Да |
| Atom | JavaScript, CoffeeScript, Python и другие | 2021 | [https://atom.io](https://atom.io/) | Бесплатно | Современный | Да |
| Sublime Text | Python, JavaScript, HTML, CSS и другие | 2022 | [https://www.sublimehq.com](https://www.sublimehq.com/) | Платно, но есть пробная версия | Интуитивный, с возможностью кастомизации | Да |
| NetBeans | Java, JavaScript, PHP и другие | 2023 | [https://netbeans](https://netbeans/). | Бесплатно | Минималистичный дизайн | Да |

1. **Установка редактора VIM.**

Vim (Vi Improved) - это текстовый редактор, который часто используется системными администраторами и разработчиками программного обеспечения. Это консольное приложение, работающее в текстовом режиме, без использования графического интерфейса. Чтобы установить VIM, нужно ввести команду sudo apt install vim. Затем откроем файл main.c с помощью VIM и убедимся что VIM установлен и все работает.



Для того чтобы нормально установить плагины, установим Vundle, а зачем через этот менеджер в файле ~/.vimrc зададим наши плагины, я выбрал первые мне понравившиеся конфигурации, в дальнейшем по мере наращивания опыта можно будет поставить больше плагинов. Как выглядит файл vimrc



Вот список плагинов которые я установил

1. Vundle - это менеджер плагинов для текстового редактора Vim. Он позволяет легко устанавливать, обновлять и удалять плагины Vim. Bundle упрощает процесс настройки Vim для разработчиков.
2. Плагин tpope для Vim это набор плагинов, которые расширяют функциональность Vim. Этот плагин позволяет пользователям использовать более продвинутые функции, такие как автоматическое дополнение кода, подсветка синтаксиса и другие.
3. плагин command-t для Vim автоматически дополняет команды и код в редакторе. Он также предоставляет функции поиска и замены, подсветку синтаксиса и другие возможности для удобства работы с кодом.
4. плагин sparkup предназначен для создания веб-приложений на языке программирования Scala. Он содержит набор инструментов, которые упрощают процесс разработки и позволяют быстро создавать приложения. Sparkup поддерживает различные версии Scala и Java, а также позволяет использовать различные базы данных и системы управления версиями.

**6.Программа для задания по варианту**

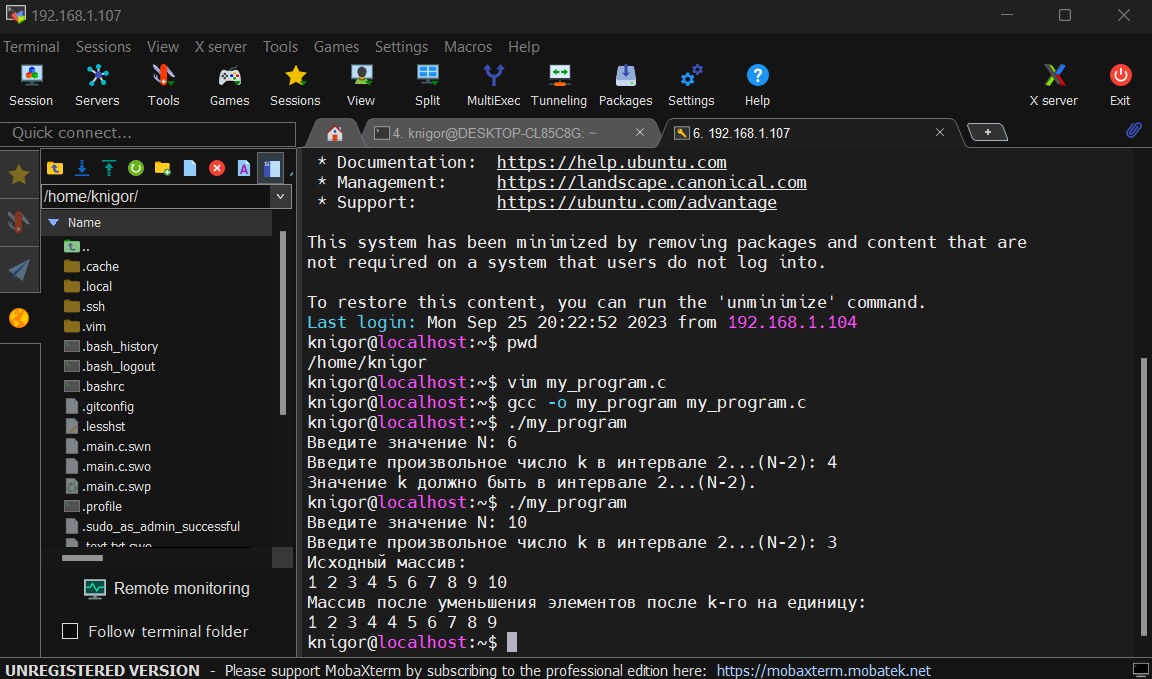
Задача вариант № 4:

Задать произвольное число k в интервале 2… N-2. Уменьшить все элементы, стоящего после k-го, на единицу.

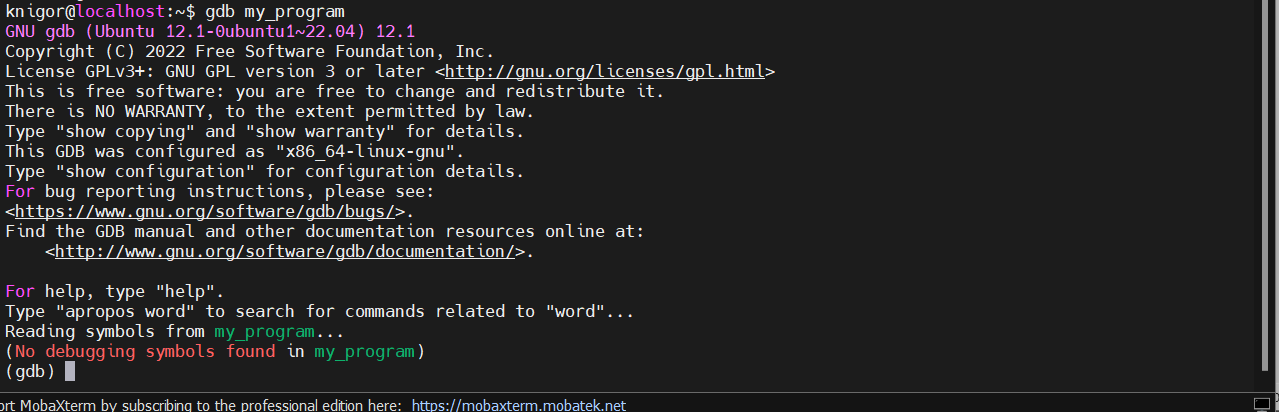


Код программы.

Произведем компиляцию приложения через набор компиляторов GCC



Результат отладки программы через GDB



Командой break main мы найдем breakpoint, то есть точку, после которой

Командой “next” мы можем перемащаться шагами по нашему коду. Таким образом, мы можем отлаживать и напрямую видеть, что происходит в коде на каждом участке кода.

Ответы на контрольные вопросы:

⦁ Что такое API? -  API (Application Programming Interface) - это набор правил и соглашений, которые позволяют различным программам взаимодействовать друг с другом. Он определяет, каким образом две программы могут обмениваться данными и использовать функции друг друга.

⦁ Что такое библиотека в программировании? - Библиотека в программировании - это файл или набор файлов, содержащих функции и переменные, которые могут быть использованы в программе. Они могут содержать различные алгоритмы, структуры данных, математические функции и многое другое.

⦁ Понятия Статической и Динамической библиотек. -  Статическая библиотека - это библиотека, которая компилируется вместе с вашей программой и становится ее частью. Ее код вставляется в вашу программу на этапе компиляции. Динамическая библиотека (также известная как динамическая ссылка или shared library) - это библиотека, которую операционная система может загружать и выгружать по мере необходимости. Это позволяет программам использовать код и ресурсы совместно, уменьшая размер каждой отдельной программы и снижая затраты на обслуживание. Динамические библиотеки часто используются для системных утилит, таких как графические и аудио драйверы.

⦁ Что такое плагин? - Плагин - это модуль, который расширяет функциональность программы, добавляя новые функции или изменяя существующие. Плагины обычно используются для добавления новых возможностей без необходимости полной перекомпиляции программы.

⦁ Назовите несколько консольных текстовых редакторов для Linux. Vim, NanoVim, Emacs, Midnight Commander, Gedit

⦁ Что делает команда gcc? - Команда gcc (GNU Compiler Collection) собирает исходный код программы на языке Си или C++, переводит его в машинный код и создает исполняемый файл.

⦁ Что делает команда make? - Команда make - это утилита, которая автоматизирует процесс компиляции и сборки программ. Она анализирует зависимости между файлами исходного кода и автоматически определяет, какие файлы нужно перекомпилировать.

⦁ Что делает команда gdb? - Команда gdb (GNU Debugger) используется для отладки программ. Она позволяет ставить точки останова, просматривать значения переменных, выполнять код пошагово и т.д.

⦁ Дайте определение заголовочного файла и файла реализации. Приведите пример. -  Заголовочный файл - это текстовый файл, содержащий определения функций, структур данных и макросы, которые используются в программе. Файл реализации - это файл, который содержит реализацию функций, определенных в заголовочном файле.

Заголовочный файл:

#ifndef MYLIB\_H

#define MYLIB\_H

int add(int a, int b);

#endif /\* MYLIB\_H \*/

Файл реализации:

#include "mylib.h"

int add(int a, int b) {

    return a + b;

}

⦁ Что означает единица трансляции? В чем особенность разработки программ из нескольких единиц трансляции? - Единица трансляции - это часть исходного кода программы, которая обрабатывается компилятором за один раз. Особенность разработки программ из нескольких единиц трансляции заключается в том, что необходимо следить за зависимостями между этими единицами, чтобы избежать ошибок и повысить производительность компиляции.

⦁ Дайте краткую характеристику каждому этапу трансляции программ, написанных на Си. - Процесс трансляции программ на языке Си обычно включает следующие этапы:

– Лексический анализ - разбиение исходного кода на лексемы (слова, числа, операторы и т.д.).

– Синтаксический анализ - проверка правильности написания программы с точки зрения грамматики языка Си.

– Семантический анализ - проверка соответствия типов данных, корректности использования переменных и т.п.

– Генерация промежуточного кода - преобразование исходного кода в более компактный и удобный для дальнейшей обработки вид.

– Оптимизация промежуточного кода - удаление избыточного кода, упрощение выражений и т.д., чтобы повысить эффективность программы.

– Генерация объектного кода - перевод промежуточного кода в набор инструкций для конкретного процессора.

– Сборка объектных файлов - объединение объектных файлов в исполняемый файл с учетом зависимостей между ними.

– Линковка (связывание) - подключение необходимых библиотек и разрешение внешних ссылок.