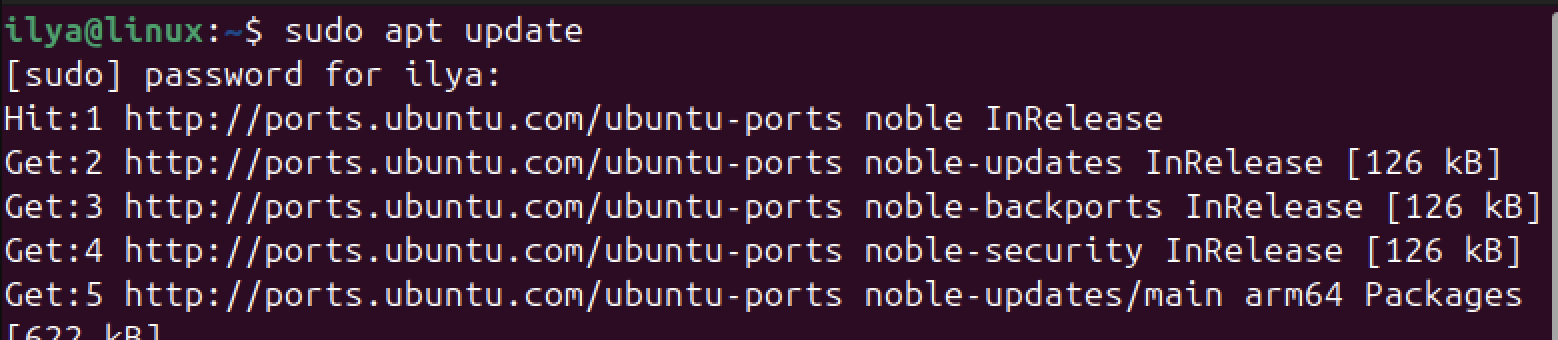
**Lx4.1. Управление пакетами.**

Наша главная цель семинара - Научиться устанавливать новое программное обеспечение, как при помощи специальных программ инсталляторов, так и путем компилирования из исходных файлов.

**Основное задание.**

1.Откройте окно менеджера программ и познакомьтесь с набором программного обеспечения, входящего в стандартный репозиторий вашей операционной системы.

Для этого сначала обновим пакеты с помощью команды “sudo apt update”, чтобы в дальнейшем получить актуальную информацию о доступных наборах программного обеспечения.



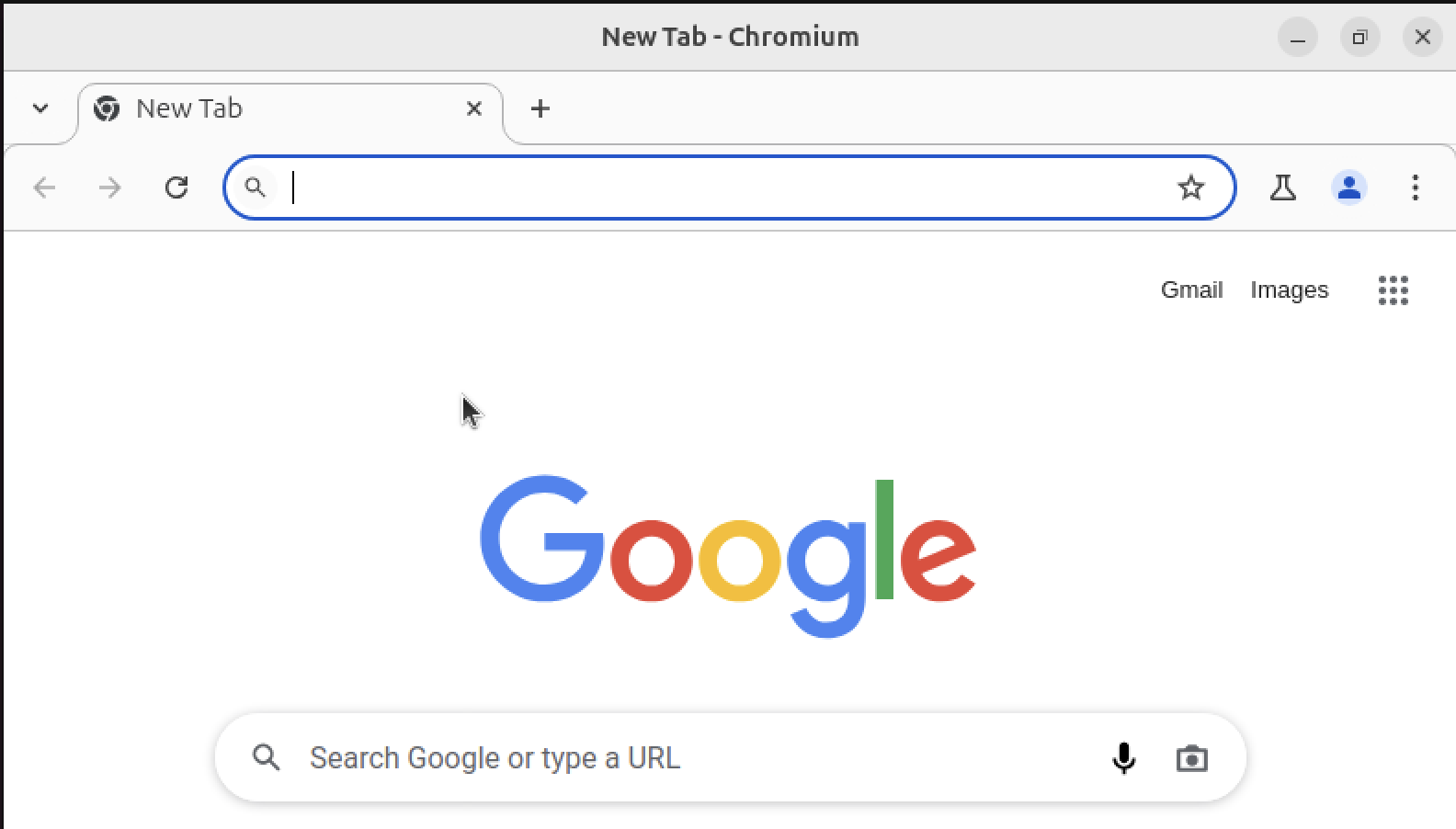
Теперь воспользуемся командой “apt-cache pkgnames” для просмотра доступных пакетов. Данная команда выведет имена всех пакетов, доступных в кэше APT (Advanced Package Tool).



Получим огромный список пакетов, доступных для пользователя.

2. Найдите и установите программу chromium-browser. Проверьте ее работоспособность.

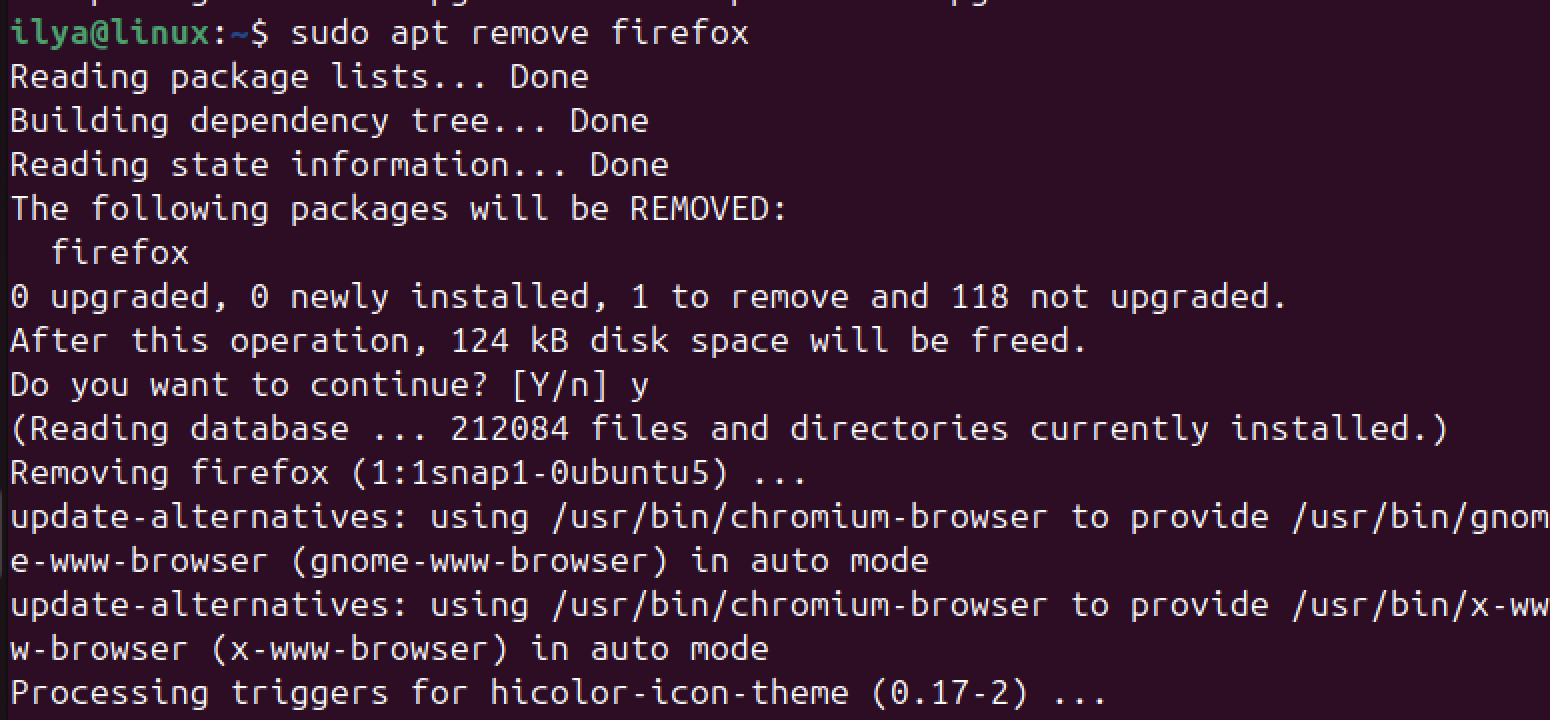
Для установки chromium-browser воспользуемся командой “sudo apt install chromium-browser ”. После успешной установки, проверим на работоспособность программу, для этого запустим её с помощью команды chromium-browser в строке терминала.



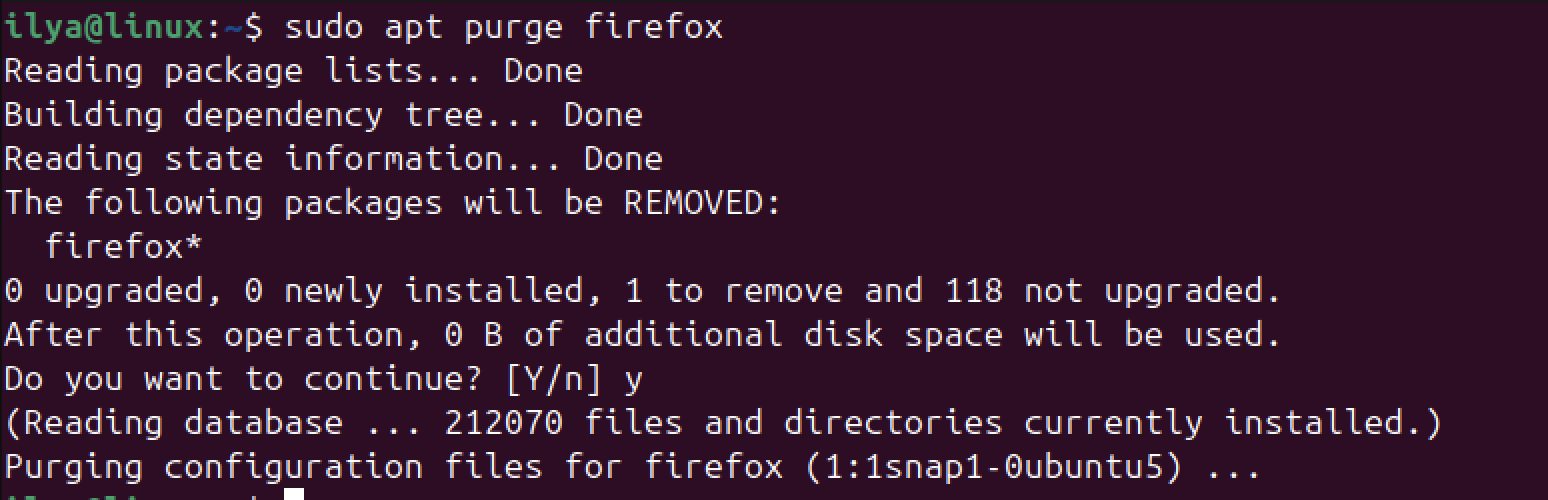
После запуска команды, открывается вкладка на рабочем столе с иконкой google браузера и соответственно запускается сам браузер для дальнейшей работы.

3. Удалите установленную программу firefox.

Для того, чтобы удалить программу firefox, следует прописать команду “sudo apt remove firefox”, предварительно произведя обновление пакетов, для корректной работы.

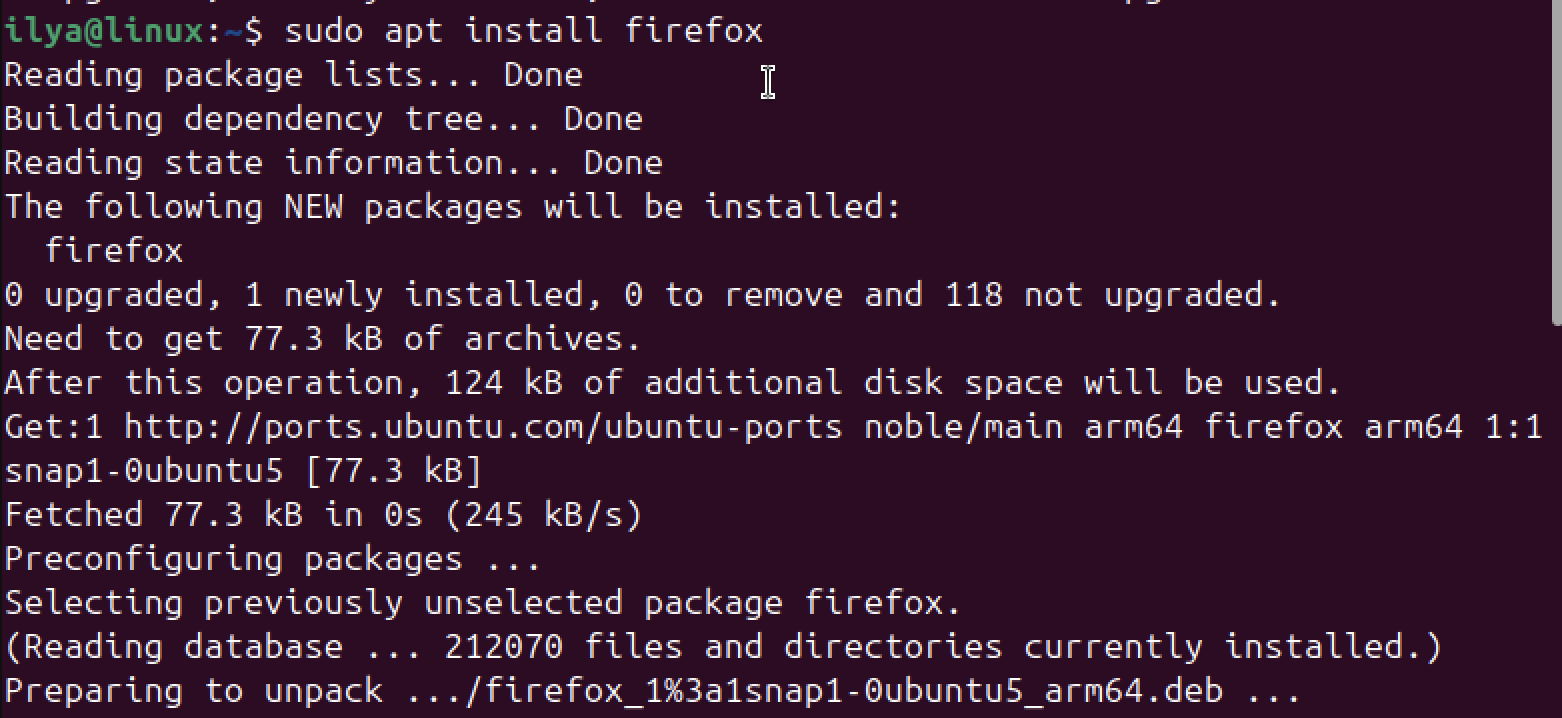


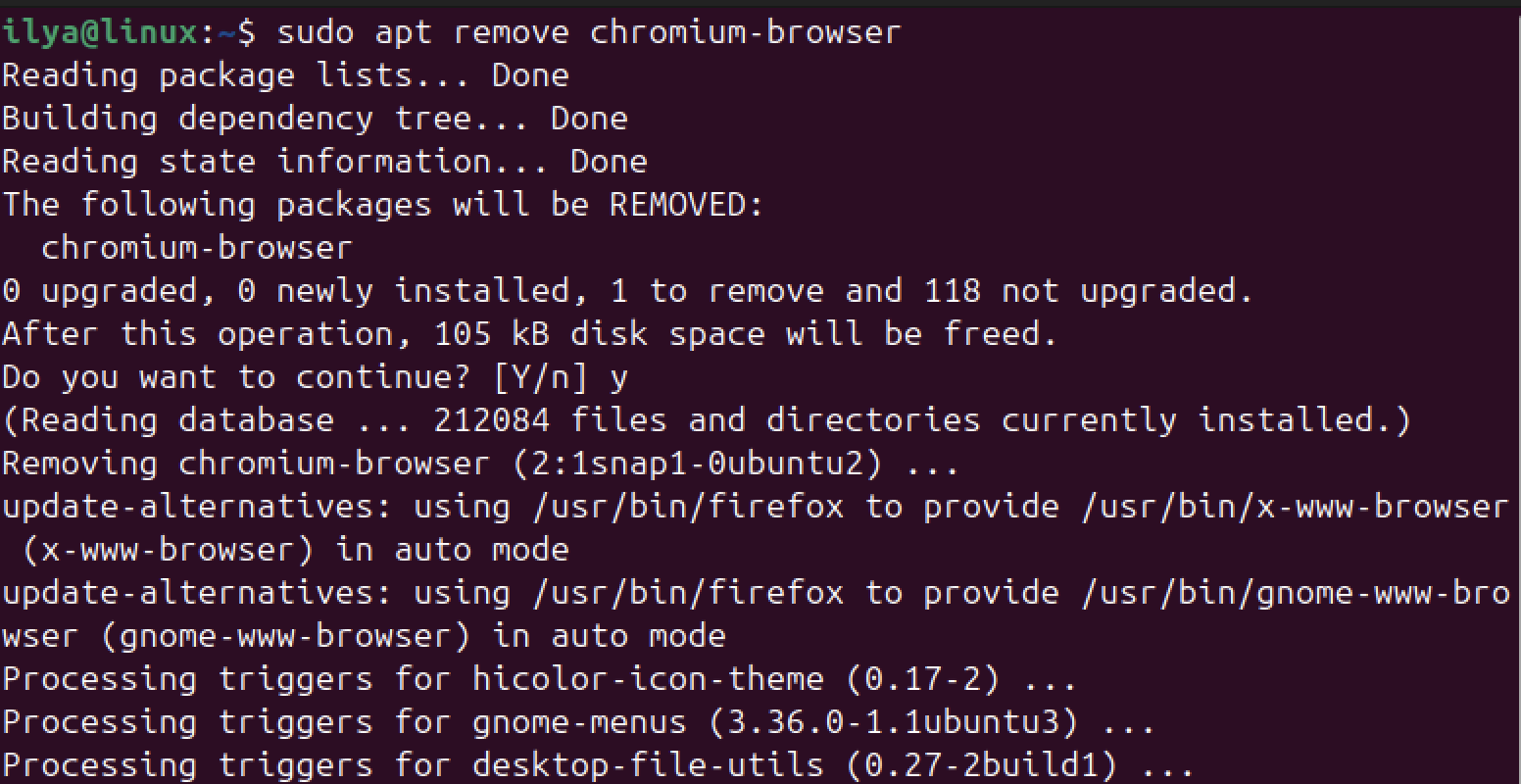
Для полноценного удаления программы firefox с устройства, нам требуется удалить и её конфигурационные файлы. Для этого пропишем команду “sudo apt purge firefox”, в которой команда “purge” будет отвечать за удаление конфигурационных файлов программы.



4. Произведите обратные действия (установку firefox и удаление chromium) командами в терминале.

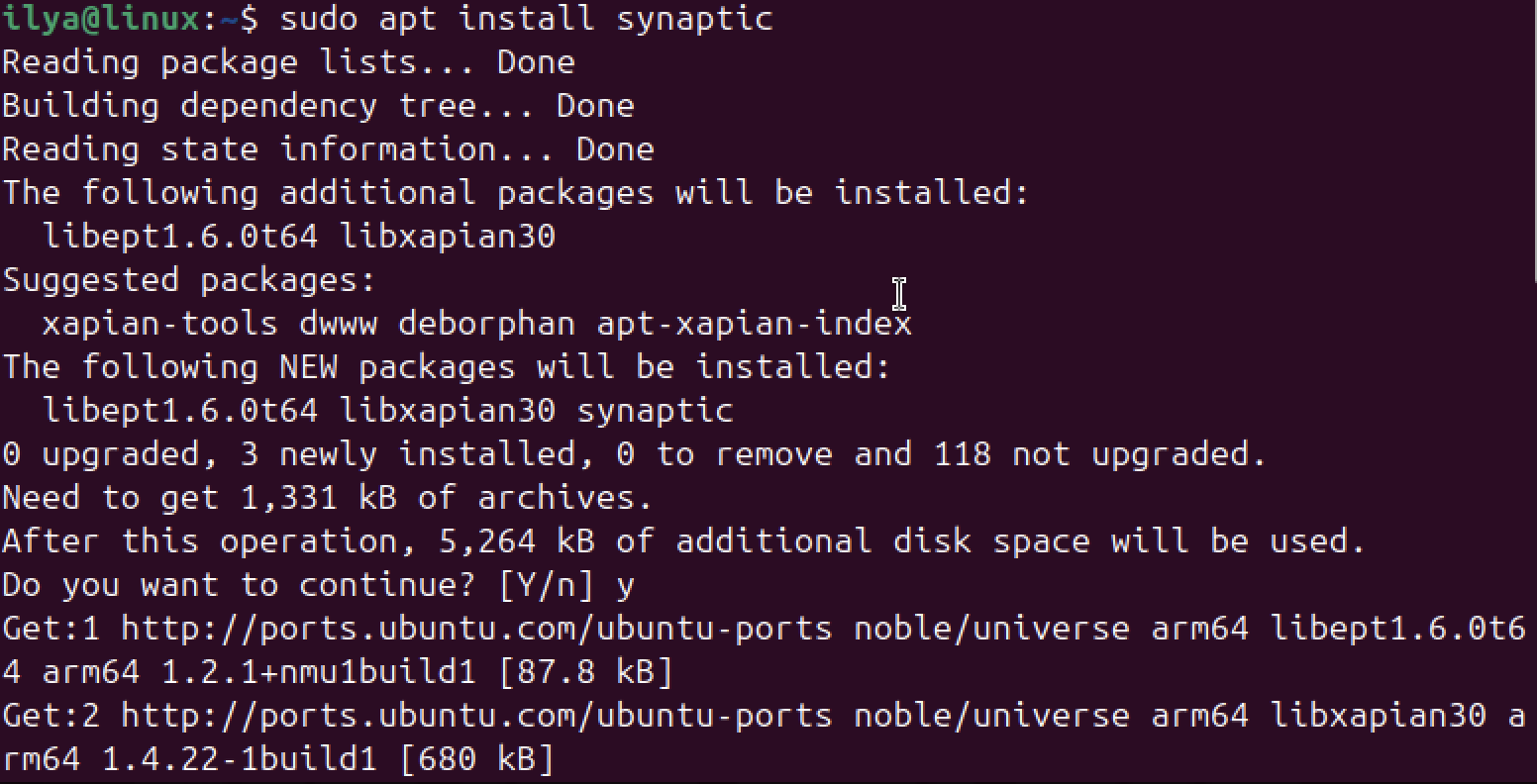
Чтобы произвести обратные действия, нам нужно прописать 2 команды: 1 - “sudo apt install firefox” - для установки программы firefox и “sudo apt remove chromium-browser” - для удаления браузера Google Chrome.

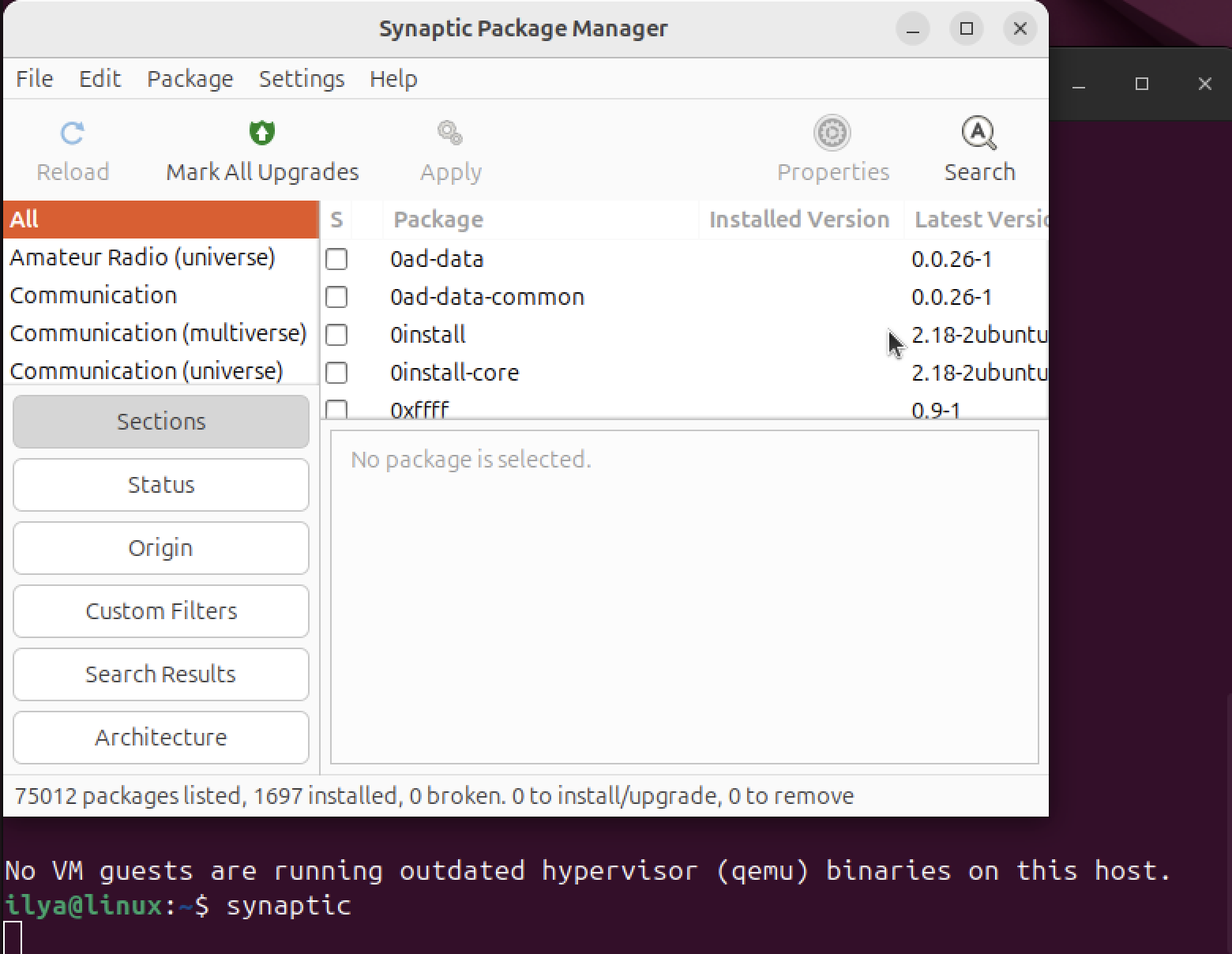




5. Откройте менеджер пакетов Synaptic в графическом режиме.

Чтобы открыть менеджр пакетов Synaptic в графическом режиме, пропишем команду synaptic в строку терминала, предварительно установив данную команду на наше устройство командой “sudo apt install synaptic”.





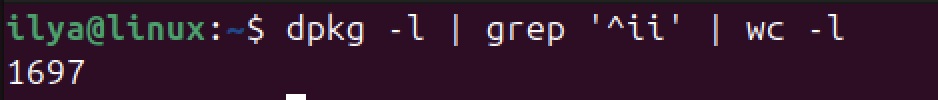
Перед нами откроется менеджер файлов (в графическом режиме) для работы.

6. Выведите полный список пакетов, установленных в системе. Подсчитайте общее количество пакетов, установленных в системе.

В первую очередь, выведем все установленные пакеты на устройстве с помощью команды “dpkg -l”, где dpkg — это низкоуровневый менеджер пакетов, отвечающий за управление пакетами (установка и удаление) в системах Linux, которые основаны на Debian и имеют расширение .dep, а соответственно -l отвечает за отображение всех установленных пакетов. Далее, подсчитаем пакеты, установленные на устройстве. Пропишем команду “dpkg -l | grep ‘^ii’ | wc -l”.

Grep ‘^ii’ - задаем регулярное выражение, которое захватывает только те файлы, которые на данный момент установлены на устройстве. ^ - отвечает за начало строки, ii - код статуса установки, первая I отвечает за установку файла, вторая I отвечает за правильную настройку пакеты и отсутствие в нем ошибок.

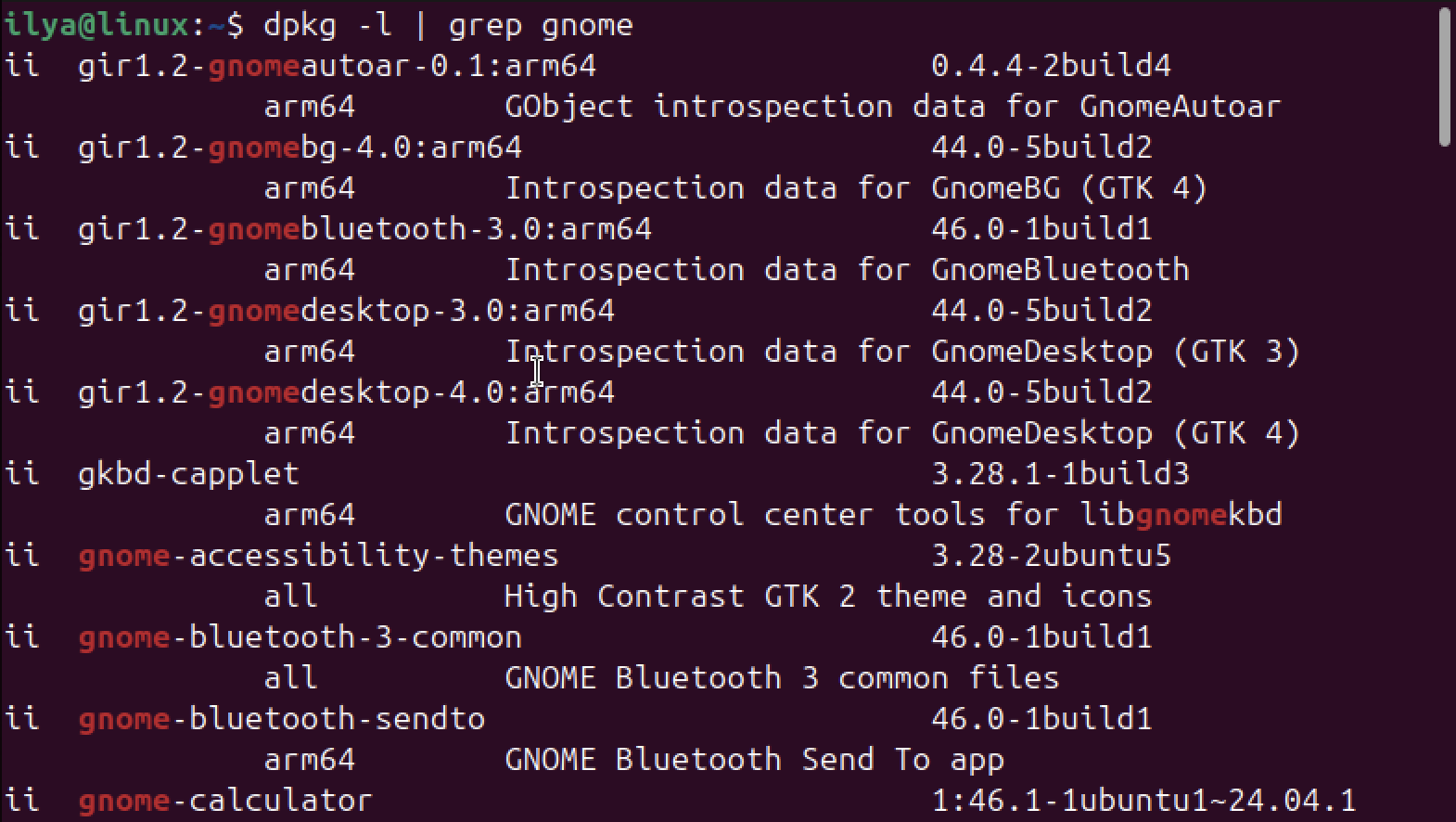
И wc -l отвечает за окончательный подсчет строк установленных пакетов. Wc - word count, -l указывает для wc, что следует считать только количество строк.

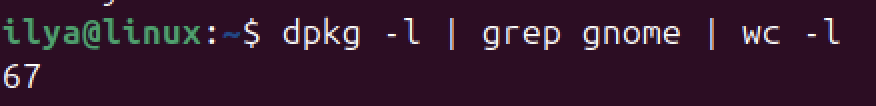


Прописав команду, мы получили значение 1697.

7. Выберите из общего списка все пакеты, имеющие в названии слово gnome (т.е. те пакеты, которые принадлежат интегрированной среде GNOME) и подсчитайте их количество.

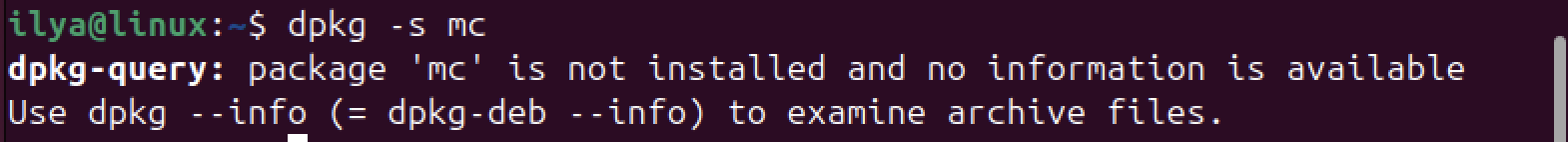
Аналогично прошлому пункту, мы сможем вывести все пакеты, имеющие в названии gnome и затем узнать их количество с помощью команды “dpkg -l | grep gnome” и “dpkg -l | grep gnome | wc -l”.



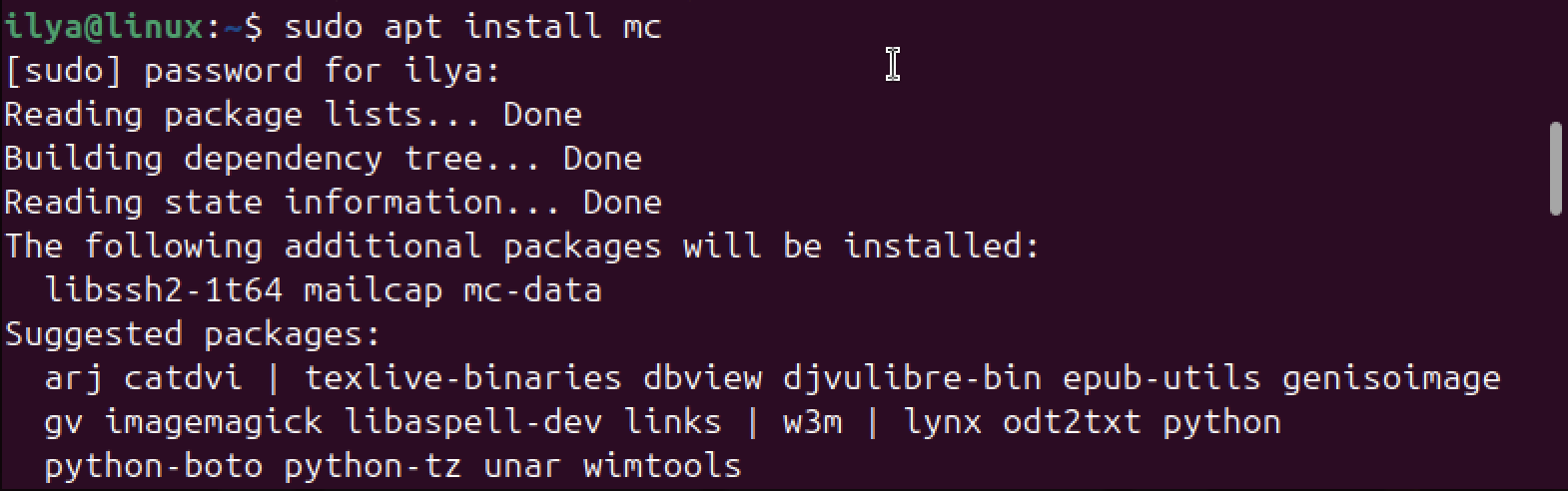


Прописав команду, мы получили значение 67.

8. Проверьте, установлен ли в системе пакет mc и получите расширенную информацию об этом пакете.

Для того, чтобы проверить, существует ли на устройстве пакет mc, воспользуемся командой “dpkg -s mc” для получения расширенной информации о пакете. Параметр -s отвечает за отображение информации о состоянии пакета. После этого параметра в команде, записывается имя пакета, информацию о котором мы хотим вывести.

Получили сообщение, что данные пакет не установлен. Займемся установкой, прописав команду “sudo apt install mc”.

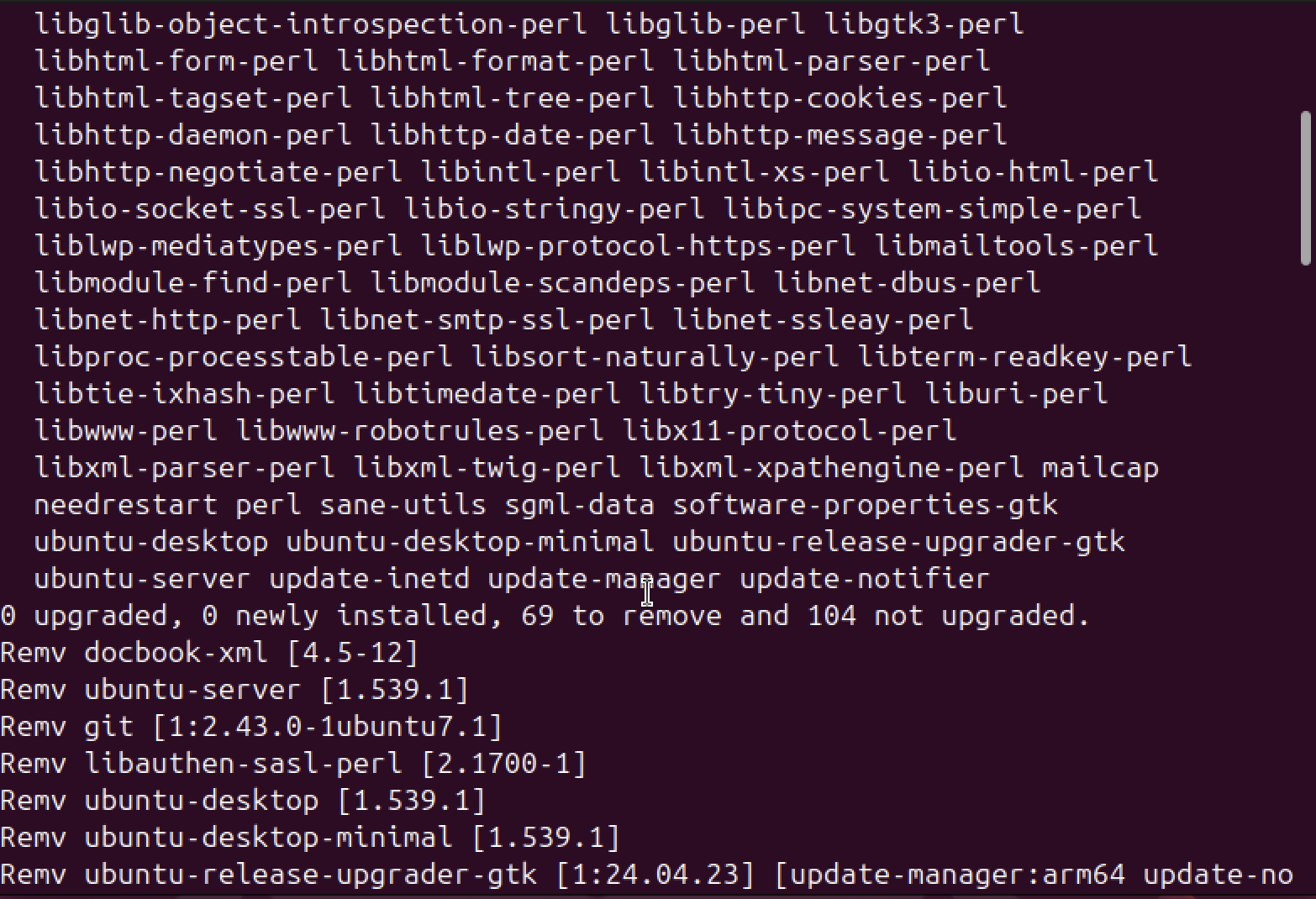


Заново пропишем команду “dpkg -s mc”, для получения информации о пакете. Получим следующее:



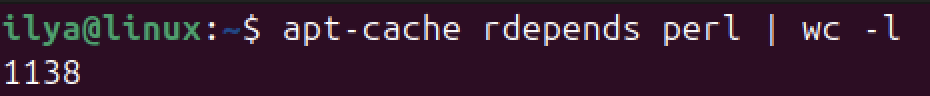
9. Выполните имитацию удаления (НЕ УДАЛЕНИЕ!) пакета perl. Подсчитайте количество пакетов, которые от него зависят.

Сначала выполним имитацию удаления пакета. Команда практически идентичная, как и для удаления, но мы дополнительно используем параметр --simualte, чтобы показать, что удаление не настоящие, а лишь симуляция. “sudo apt remove --simulate perl” покажет, что произойдет при удалении пакета.



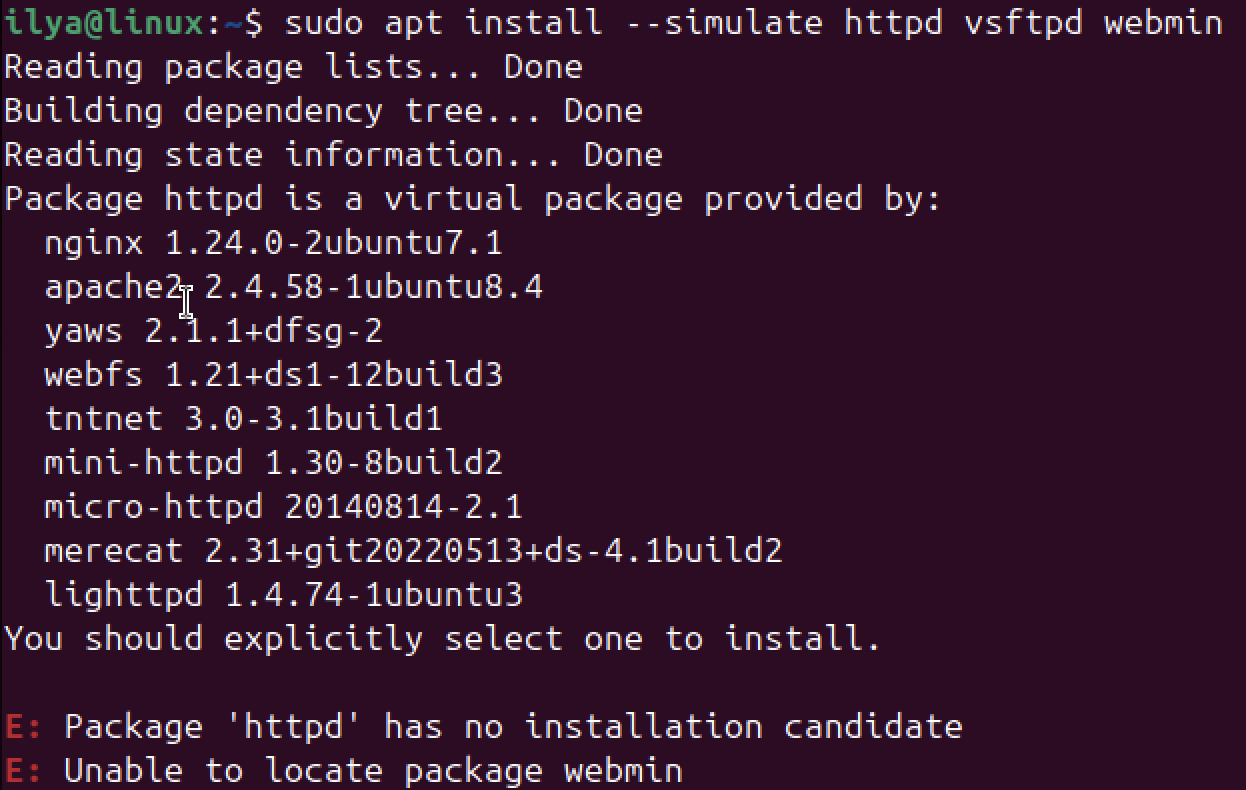
На экране, будет отображаться информация о тех пакетах и файлах, которые будут затронуты при удалении perl.

Теперь подсчитаем количество пакетов, который зависят от пакета perl. Для удобства будем использовать команду “apt-cache rdepends perl | wc -l”, в которой apt-cache будет работать с кэшем устройства, для получения информации об установленных пакетах. Параметр rdepends отвечает за отображение всех пакетов, которые зависят от указанного нами в строке пакета.



Получим следующее значение.

10. Выполните имитацию установки пакетов httpd, vsftpd и webmin, проанализировать результаты (убедиться, что все зависимости удовлетворены).

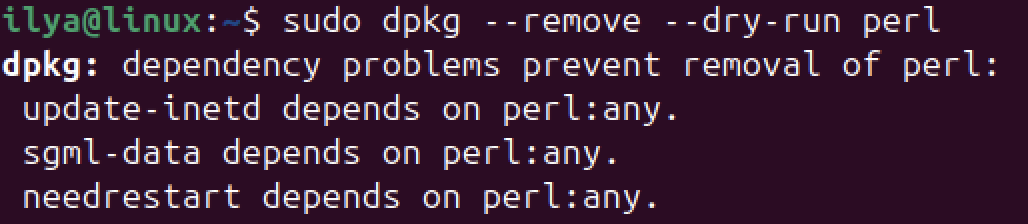
Так как у нас снова нужно произвести имитацию, то воспользуемся командой. “sudo apt install --simulate httpd vsftpd webmin” и пропишем названия пакетов через пробел. Получим следующее сообщение: 

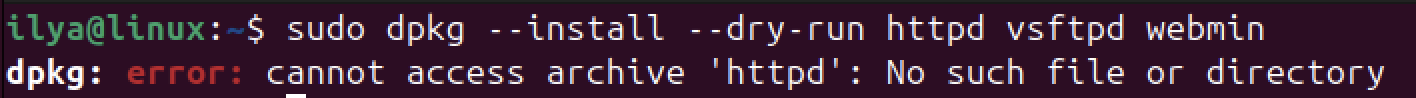
Как мы можем заметить, для пакета “httpd” не найден кандидат для установки и нет возможности найти пакет webmin, следовательно некоторые зависимости не могут быть удовлетворены, значит установка будет проблематичной.

11. Повторите задания 6-10 в консольном режиме с использованием программы dpkg.

Так как я выполнил задания 6 - 8 с использованием программой dpkg, то теперь выполним задания 9 - 10 с командой dpkg.

Чтобы выполнить имитацию удаления или установки пакета, с командой dpkg нужно использовать параметр --dry-run.

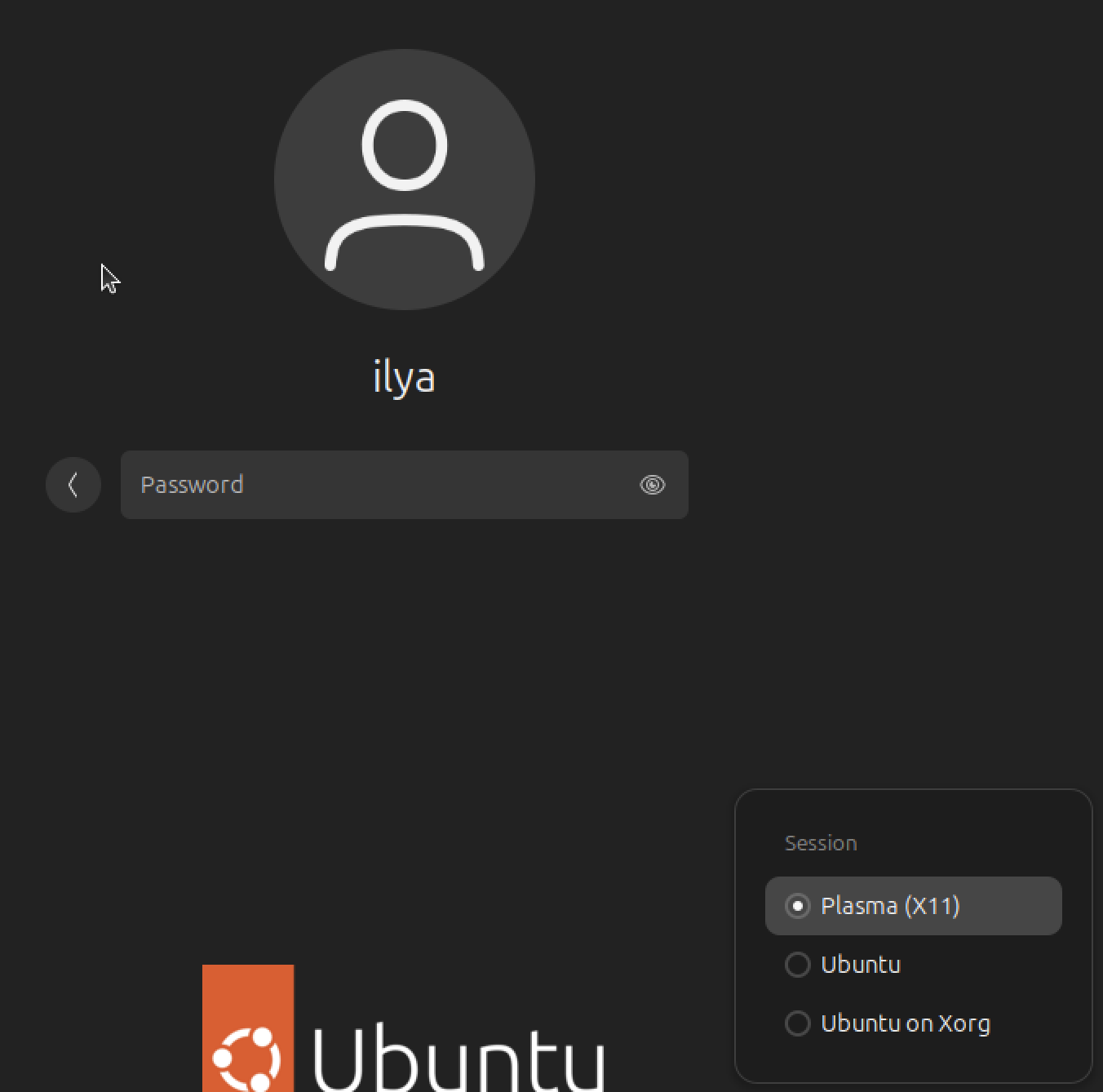




**Задания для самостоятельного выполнения.**

1. Установите другой менеджер рабочего стола (например, KDE или XFCE). Проверьте его работоспособность, залогинившись в него.

Для дальнейшей работы будет использовать менеджер рабочего стола KDE. Для корректной работы обновим пакеты и пропишем команду для установки KDE. “sudo apt install kde-plasma-desktop”. После успешной установки, выйдем из нашей сессии. Заново заходим и в правом углу нажимаем на шестеренку, выбираем сессию plasma, которую мы только что установили. Вводим пароль и видим следующее:

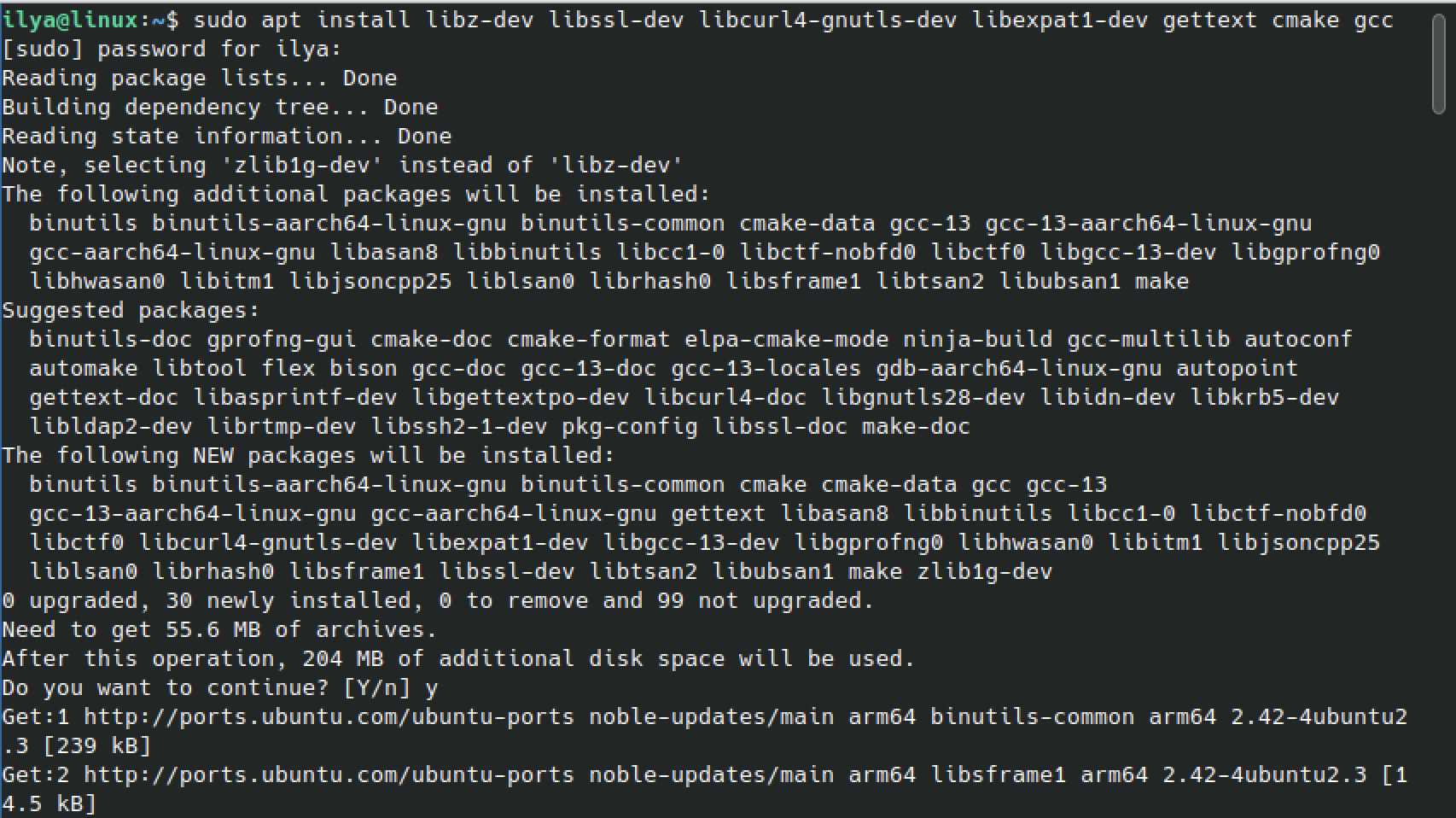




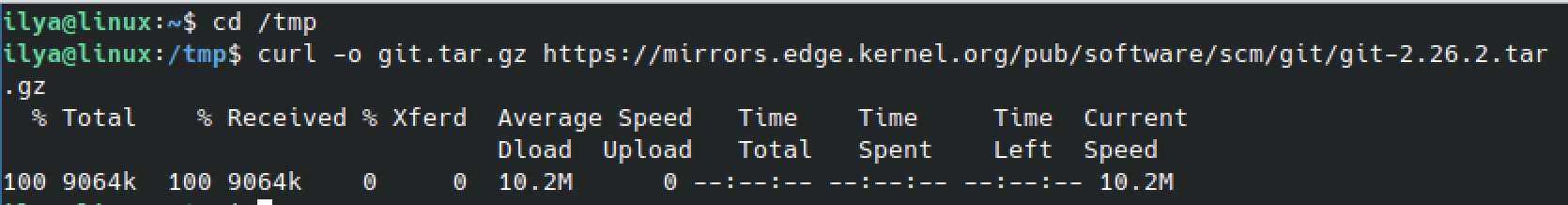
Получили отличный рабочий стол от Ubuntu. Следовательно установка прошла успешно. Продолжим свою работу в данной сессии.

1. Установите какую-либо программу по вашему выбору из исходных кодов.

Будем устанавливать git. Для этого сначала установим зависимости, которые требуются для его компиляции. Пропишем команду sudo apt install libz-dev libssl-dev libcurl4-gnutls-dev libexpat1-dev gettext cmake gcc, где через пробел прописаны необходимые пакеты для установки.

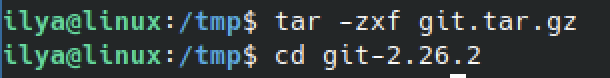


Далее перейдем в директорию tmp и пропишем команду curl -o git.tar.gz <https://mirrors.edge.kernel.org/pub/software/scm/git/git-2.26.2.tar.gz> архив которой, мы в дальнейшем установим. Мы переходим в директорию tmp, чтобы обезопасить процесс загрузки и избежать путаницы с другими файлами и директориями.



Теперь распакуем архив, с помощью команды “tar -zxf git.tar.gz”. -z - указывает, что архив сжат с использованием gzip. X - указывает, что требуется извлечь содержимое. F - опция используется для указания имени файла архива, с которым будет работать команда.

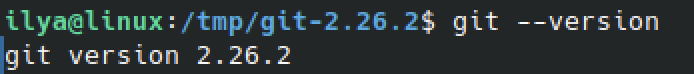
После разархивации перейдем в директорию с распакованным исходным кодом. Команда - “cd git-2.26.2”.



Для компиляции git пропишем команды “make prefix=/usr/local all”,

“sudo make prefix=/usr/local install”. Первая команда указывает на сборку всех компонентов программы с указанием префикса для установки. Вторая команда - установка собранных файлов в системные директории с правами суперпользователя, используя указанный префикс.

После завершения установки проверьте, что Git установлен правильно, выполнив команду: git version 2.26.2

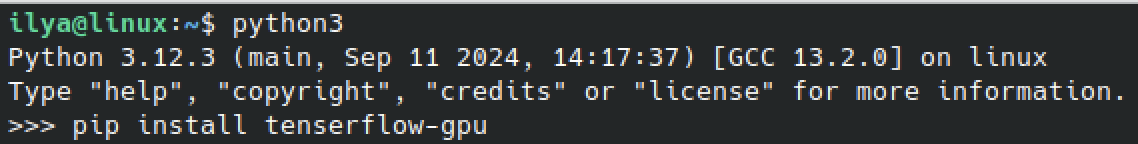


1. Установите на вашу Linux-машину библиотеку tensorflow-gpu. Проверьте правильность установки, запустив в командной строке интерактивную консоль Python командой python и затем введя команду.

Для этого сначала установим python на наше устройство. Команда - “sudo apt install python3 python3-pip python3-venv”.

Далее откроем виртуальное обеспечение с помощью команды “python3 -m venv session”. -m показывает, что модуль venv будет выполняться как скрипт. Venv - модуль для создания виртуальных окружений в питоне. И в конце указываем название виртуального окружения. В нашем случае - ourvenv.

Теперь выполним установку библиотеки tensorflow-gpu. Для этого пропишем команду pip install tensorflow-gpu. Начнется загрузка пакетов для работы с библиотекой.



Теперь проверим правильность установки, запустив в командной строке интерактивную консоль Python командой python3 и затем введя команду “import tensorflow”. После вызова команды, ничего не должно произойти.





**Контрольные вопросы.**

1.Чем отличаются команды apt и apt-get?

apt-get: это более низкоуровневый инструмент, который предоставляет полный доступ к функциям управления пакетами, но может быть менее удобным для обычного пользователя.

apt: это более высокоуровневый и удобный интерфейс, который объединяет функциональность apt-get и apt-cache. Он был разработан с акцентом на удобство использования для конечного пользователя.

2. В чем различие программ для управления пакетами apt и dpkg?

dpkg: это низкоуровневый инструмент, который управляет установкой, удалением и конфигурацией пакетов в формате .deb. Он работает непосредственно с файлами пакетов и не обрабатывает зависимости автоматически. Это означает, что, если пакет требует другие библиотеки или программы, вам придется устанавливать их вручную.

apt: это более высокоуровневый инструмент, который использует dpkg для установки пакетов, но также управляет зависимостями. apt может загружать пакеты из удаленных репозиториев и автоматически устанавливать все необходимые зависимости, что делает его более удобным для пользователей.

3. Какие менеджеры пакетов применяются в других дистрибутивах? В чем их особенности?

Homebrew – используется на macOS. Менеджер пакетов для установки программного обеспечения из исходников или бинарных файлов.

RPM (Red Hat Package Manager) - используется в: Red Hat, CentOS, Fedora и других дистрибутивах на основе RPM.

YUM (Yellowdog Updater Modified) - используется в: Red Hat, CentOS, Fedora (более старая версия).

1. Как устанавливать пакеты из исходных кодов?

1 - установка зависимостей, которые требуются для компиляции программы. 2 - установка архива с исходным кодом программы. 3 - распаковка архива. 4 - конфигурирование для адаптации исходного кода к системе. 5 - компиляция с помощью команды make. 6 - проверка установки.

5. Зачем нужны контейнерные менеджеры пакетов? В чем их преимущества и недостатки?

Контейнерные менеджеры пакетов предоставляют средства для упаковки, распространения и управления контейнерами, которые содержат все необходимые зависимости для работы приложения.

Преимущества:

Контейнеры упаковывают приложение вместе со всеми его зависимостями, что позволяет запускать их на любой системе, поддерживающей контейнерную платформу.

Контейнеры легко масштабируются в зависимости от потребностей приложения.

Каждый контейнер работает в своей изолированной среде, что гарантирует, что изменения в одном контейнере не повлияют на другие.

Недостатки:

Контейнеры по умолчанию работают в памяти, что требует дополнительных инструментов для сохранения данных после завершения работы контейнера.

Большое количество контейнеров может создать нагрузку на локальную сеть, что может привести к проблемам с производительностью.

Хотя контейнеры упрощают управление зависимостями внутри самого контейнера, они могут создать сложности при взаимодействии между различными контейнерами или при обновлении образов.