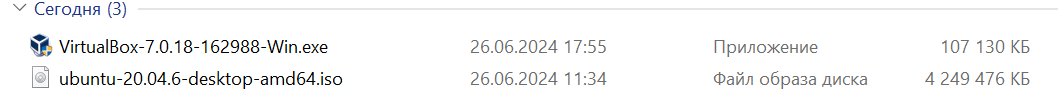
**ЛАБОРАТОРНАЯ РАБОТА 1**

**Часть 1**

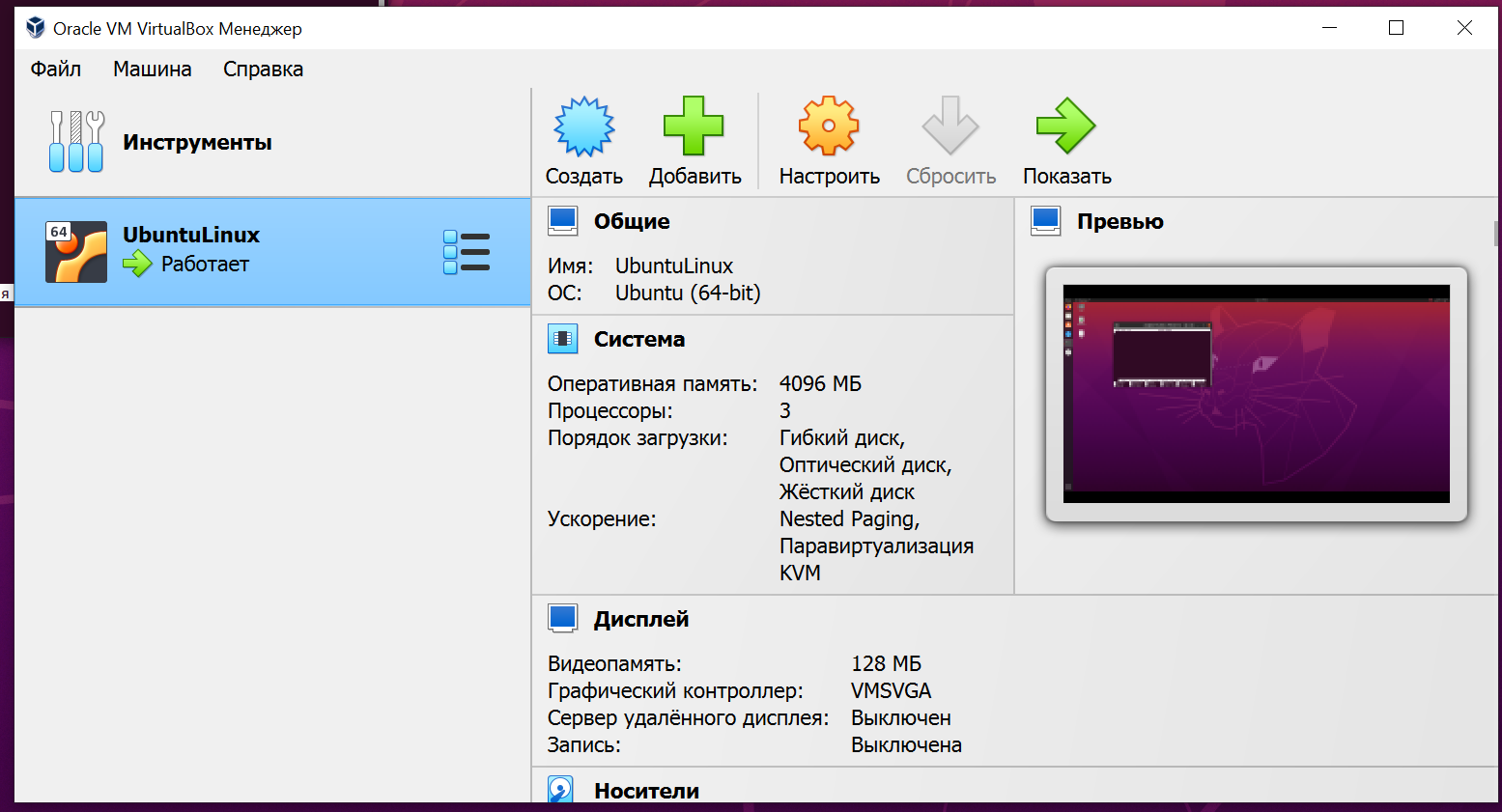
**УСТАНОВКА VIRTUAL BOX, ДИСТРИБУТИВ UBUNTU**

Загрузили следующие файлы:

****

Установили первую, а второе дистрибутив. Дистрибутив — это форма распространения программного обеспечения.

Создали виртуальную машину и настроили ее:



**УСТАНОВКА NANO И РАБОТА С ЭТИМ ТЕКСТОВЫМ РЕДАКТОРОМ**

Nano — это консольный текстовый редактор для UNIX и Unix-подобных операционных систем. Редактор относится к немодальным: вводить и редактировать текст можно сразу после открытия файла.

Устанавливаем через терминал:

Шаг 1. Откройте консоль и обновите репозитории apt — введите команду:

sudo apt update

Шаг 2. Введите в терминале команду для установки Nano:

sudo apt install nano

Готово, редактор установлен на вашем сервере.

****

Для создания файла вводим следующую команду nano.



В процессе работы с файлом вы можете в любой момент выходить из него, используя сочетание клавиш Ctrl+X. Ваши данные не потеряются: после нажатия комбинации редактор спросит, хотите ли вы сохранить изменения. Также вы можете сразу дать название своему файлу.

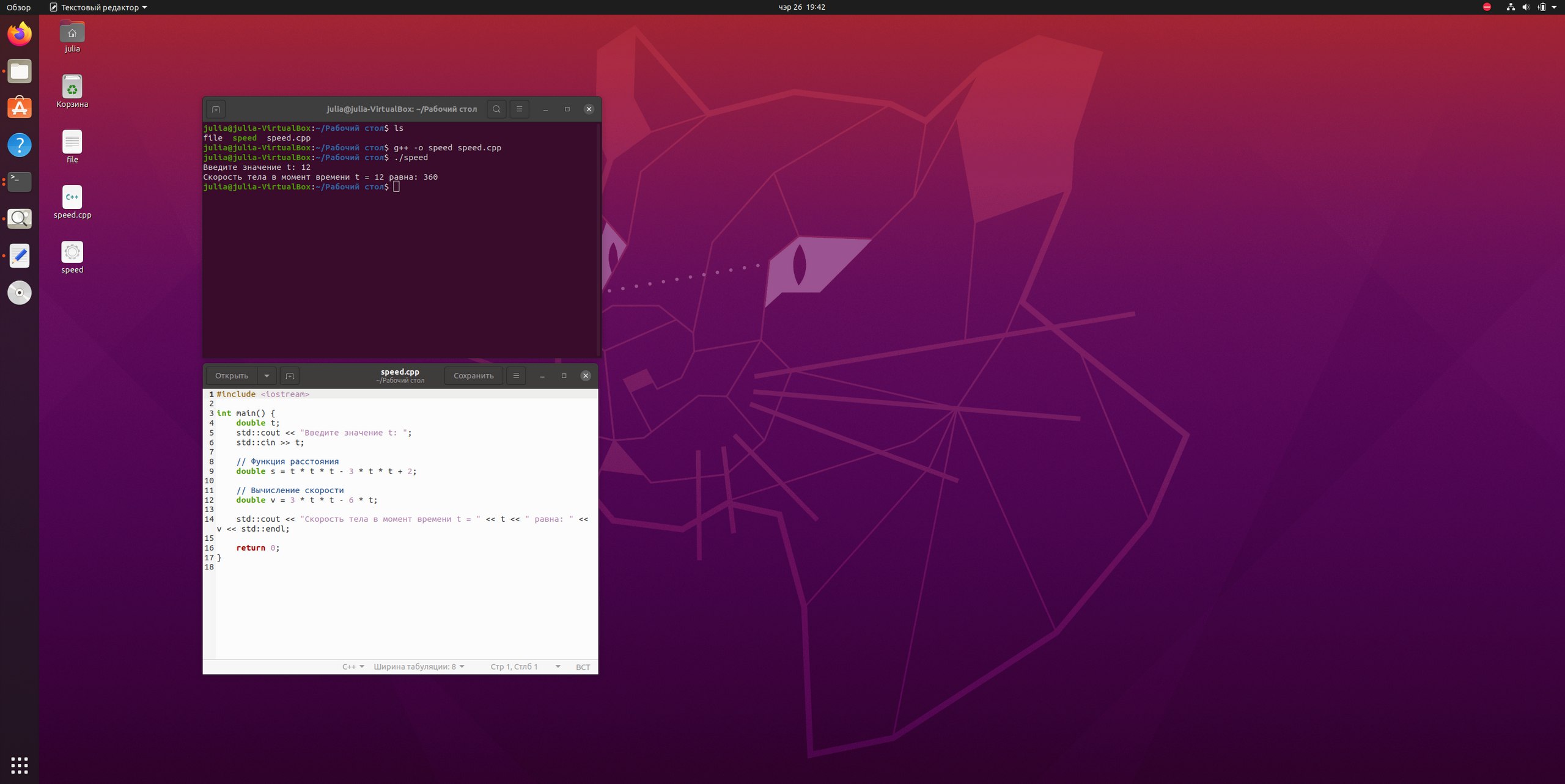
**ЗАДАНИЕ 1**

Тело движется по закону S =t^3 – 3t^2 + 2. Вычислить скорость тела в момент времени t. Значение t ввести с клавиатуры (Функция скорости есть производная от функции расстояния по времени).

Решение:

Предварительно мы создали файл speed.cpp

nano speed.cpp

****

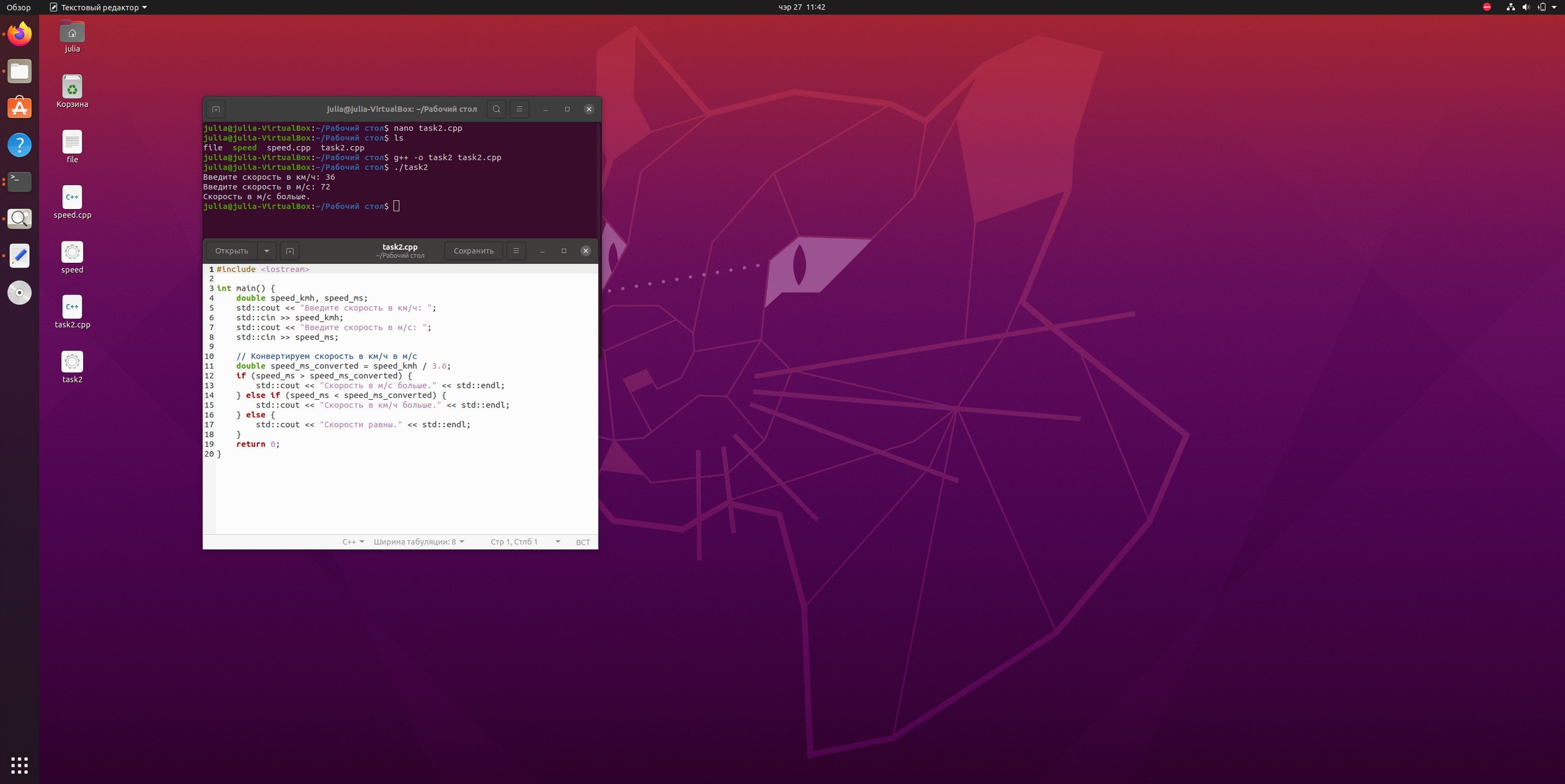
g++ -o speed speed.cpp - это команда для компиляции исходного кода speed.cpp в исполняемый файл speed. Компилятор g++ преобразует ваш С++ код в машинный код, который может быть выполнен компьютером.

./speed - это команда для запуска скомпилированной программы speed. Когда

**ЗАДАНИЕ 2**

Известны две скорости: одна в километрах в час, другая — в метрах в секунду. Какая из скоростей больше?

Решение:



**Часть 2**

**GIT ЭТО?**

Git — распределенная система контроля версий. Git представляет собой серверную часть, которая отвечает за версионное хранение документов и обработку запросов, клиентскую часть, позволяющую формировать запросы к серверу Git и набора конфигурационных файлов, хранящих настройки пользователя и позволяющих адаптировать Git для самых разных задач.

По умолчанию конфигурация репозитория хранится в каталоге “.git”, корневого каталога проекта. Таким образом достаточно легко превратить любой каталог в репозиторий или импортировать существующий репозиторий вызвав соответствующую утилиту, которая создаст подкаталог с необходимыми файлами или исправит существующие файлы в случае импорта.

В классическом обычном сценарии в репозитории git есть три типа объектов — файл, дерево и коммит. Файл есть какая-то версия какого-то пользовательского файла, дерево — совокупность файлов из разных поддиректорий, коммит — дерево + некая дополнительная информация (например, родительский(е) коммит(ы), а также комментарий).

**ОСНОВНЫЕ КОМАНДЫ**

∙who – вывод информации об активных пользователях;

∙whoami, who am i – вывод информации о пользователе терминала;

∙uname – вывод информации о версии операционной системы;

∙echo – вывод сообщений на терминал;

∙banner – вывод сообщений на терминал прописными буквами;

∙man – вызов оперативной справочной системы;

∙date – вывод текущей даты;

∙cal – календарь;

∙write – передача сообщений на терминал другого пользователя;

∙wall – передача сообщений на все терминалы;

∙mesg – разрешение/запрет вывода сообщений от других пользователей;

∙mail – отправка/получение почты;

∙news – знакомство с новостями системы;

∙ps – печать списка запущенных программ;

∙clear – очистка экрана;

∙login – регистрация в системе;

∙logout – выход из системы;

∙ksh – запуск Korn shell;

∙bash – запуск Bourne-Again Shell;

∙exit – завершение текущего shell.

**КОМАНДЫ РАБОТЫ С ФАЙЛАМИ И КАТАЛОГАМИ**

• pwd – вывод абсолютного маршрутного имени текущего рабочего

каталога;

• cd – изменение рабочего каталога;

• ls – вывод информации о содержимом каталога;

• mkdir – создание каталога;

• rmdir – удаление каталога;

• touch – обновление временной метки файла;

• cp – копирование файлов;

• mv – перемещение или переименование файла;

• cat – объединение и вывод на экран содержимого файлов;

• more – постраничный просмотр содержимого файла;

• rm – удаление файла;

• alias – создание псевдонима.

Для выполнения ряда заданий потребуется знание шаблонов генерации имен

файлов:

• ? (вопросительный знак) соответствует любому одному символу, кроме

первой точки;

• [ ] (квадратные скобки) определяют группу символов (выбирается один

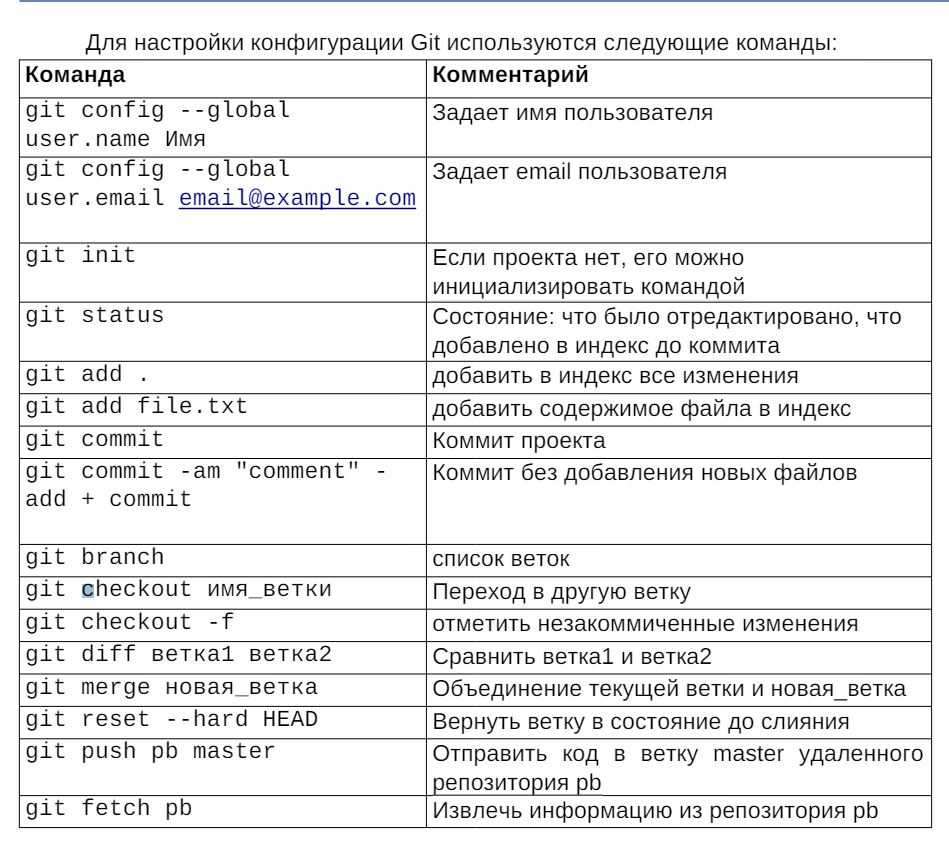
символ из группы);

• - (знак «минус») определяет диапазон допустимых символов;

• ! (восклицательный знак) отвергает следующую за ним группу символов;

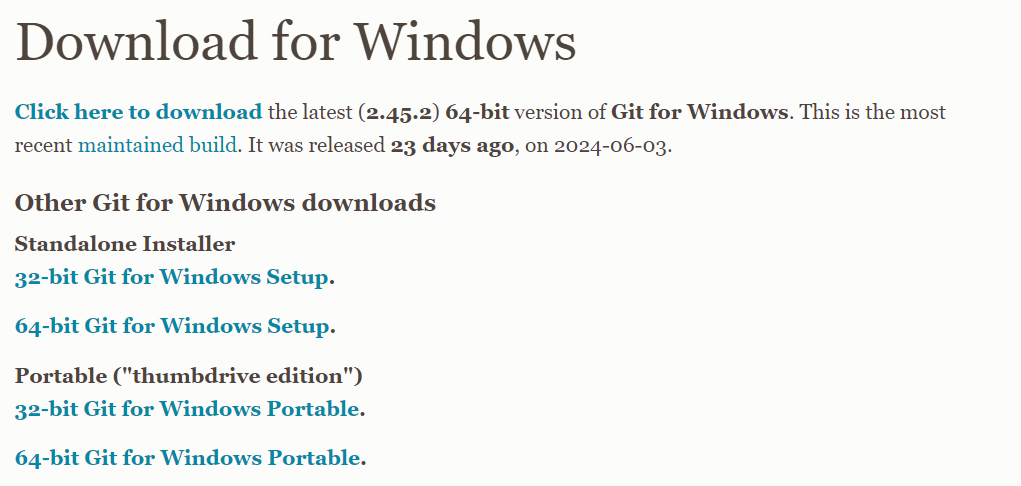
• \* (символ «звёздочка») соответствует любому количеству символов,

кроме первой точки.



**УСТАНОВКА GIT**

Официальная сборка доступна для скачивания на официальном сайте Git. Просто переходим на страницу <https://git-scm.com/download/win>, выбираем нужное и загрузка запустится автоматически.

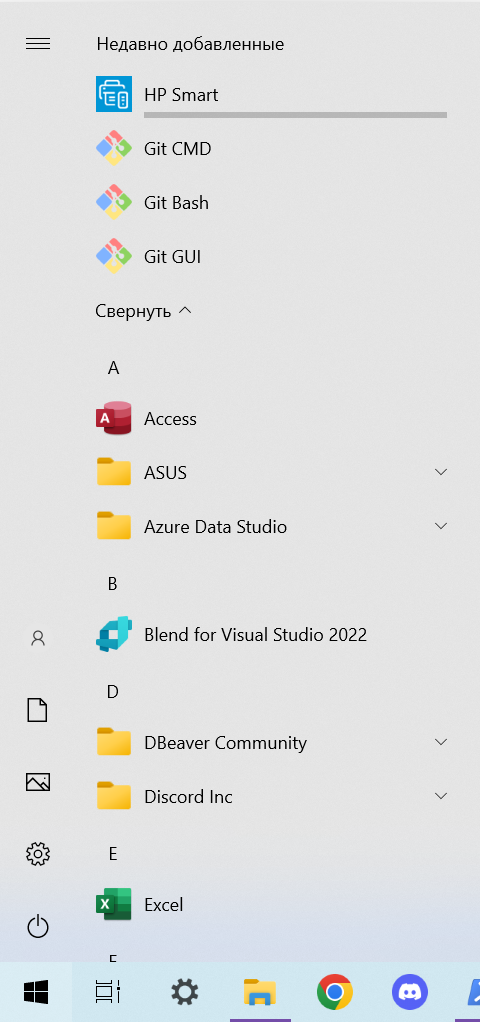


Скачивается следующее:

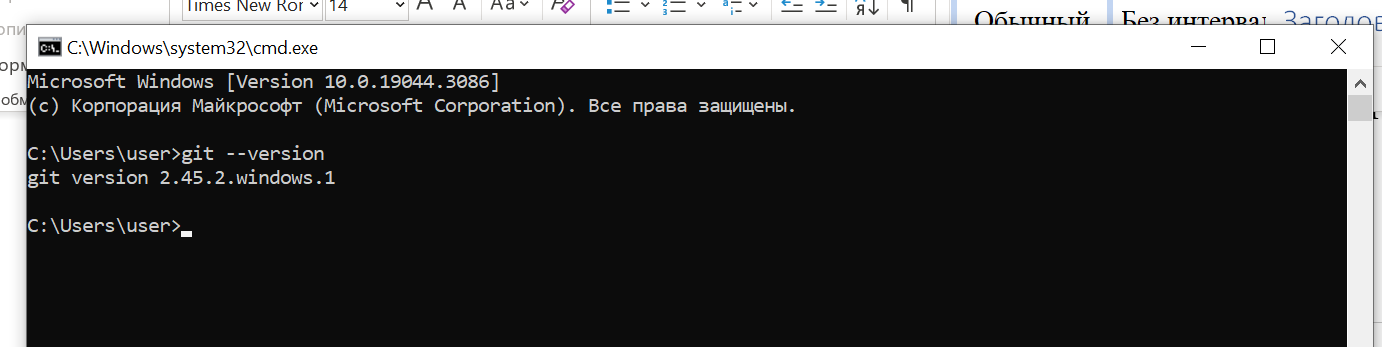


Далее по шагам устанавливаем, выбирая нужные пункты. Следовала я им здесь. <https://timeweb.cloud/tutorials/git/ustanovka-git-na-windows>.

В Пуск появляется следующее, что свидетельствует о том, что все установлено верно.



Проверили версию через командную строку на Windows:

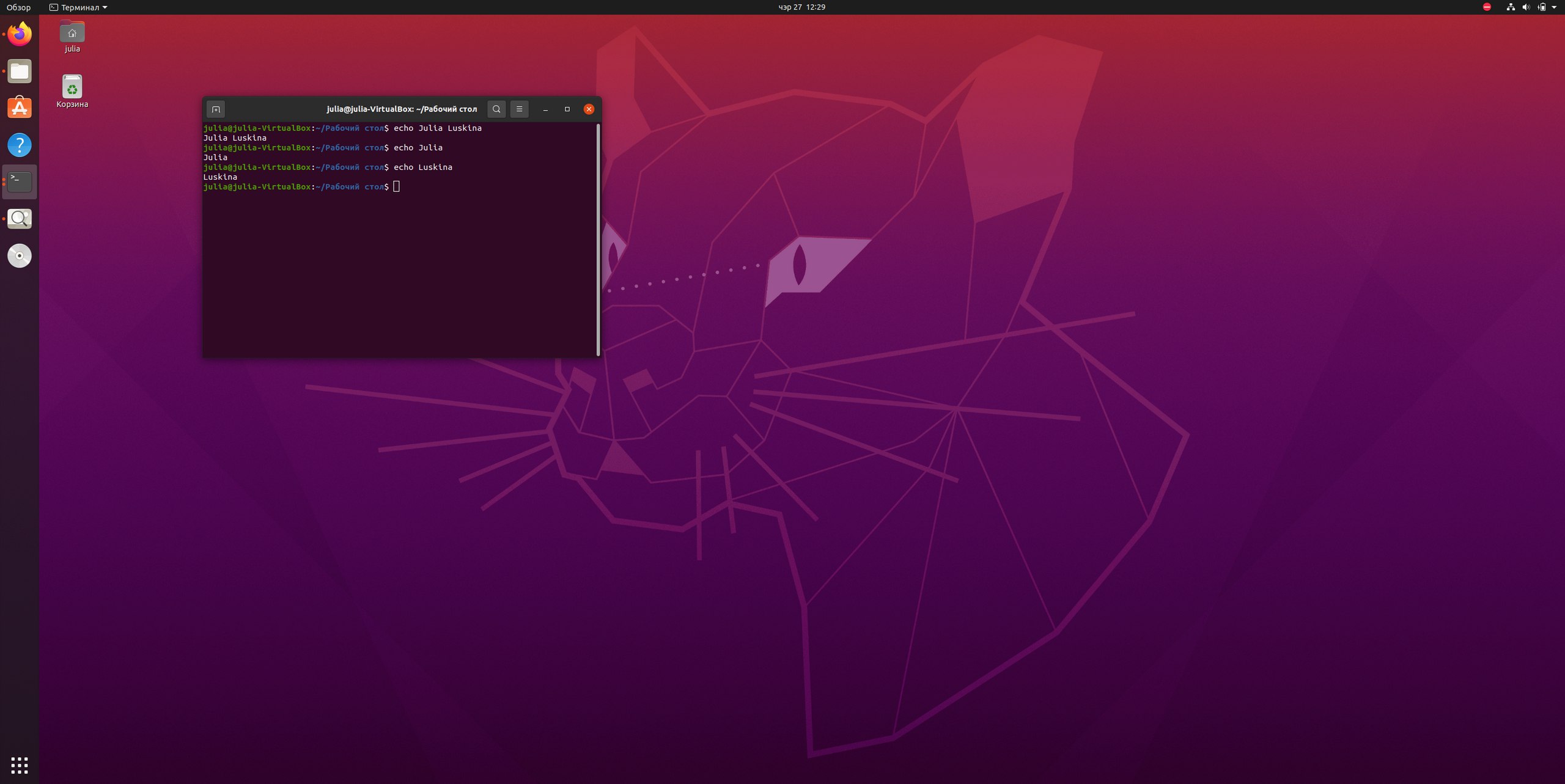


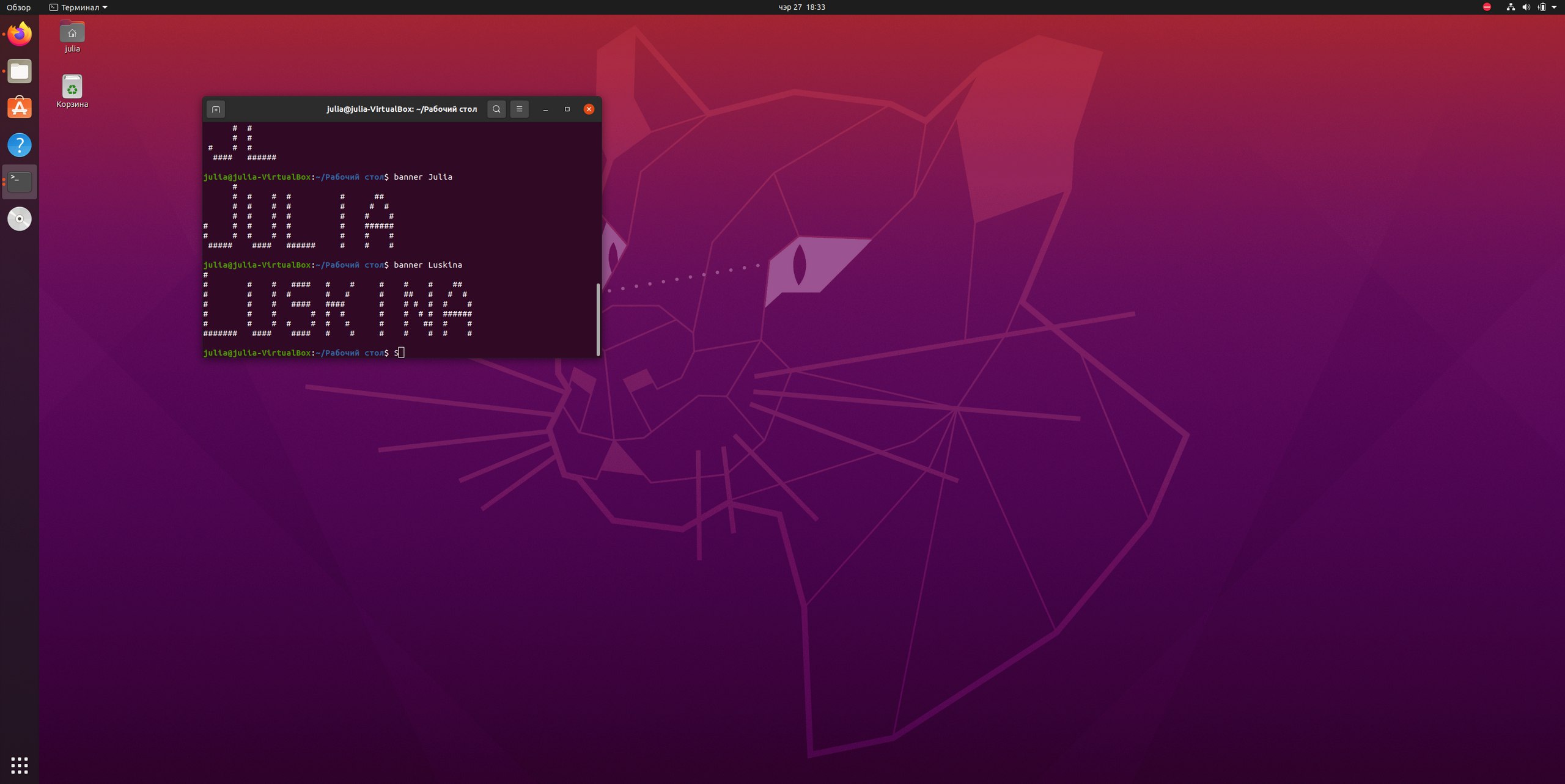
**РАЗДЕЛ 1. РАБОТА В КОМАНДНОЙ СТРОКЕ**

**ЗАДАНИЕ 1**

Выведите сообщение «Имя Фамилия», указав Ваше имя и фамилию, в виде

нескольких строк с помощью команд echo.

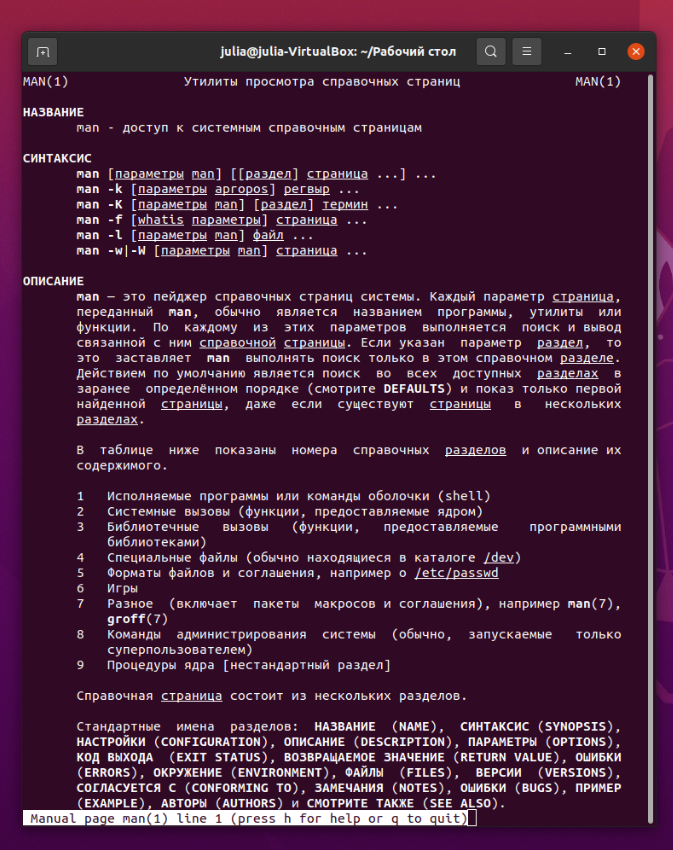




**ЗАДАНИЕ 2**

Прочитайте статью справочной системы man о пользовании справочной системой.





**ЗАДАНИЕ 3**

Прочитайте статью справочной системы о команде uname. Из какого раздела справочника Вы прочитали статью?

Здесь мы видим, что в заголовке страницы указано UNAME(1), что означает, что эта страница руководства относится к разделу 1 справочника Linux, который содержит описание пользовательских команд.

Другие возможные разделы справочника:

Раздел 2: Системные вызовы

Раздел 3: Библиотечные функции

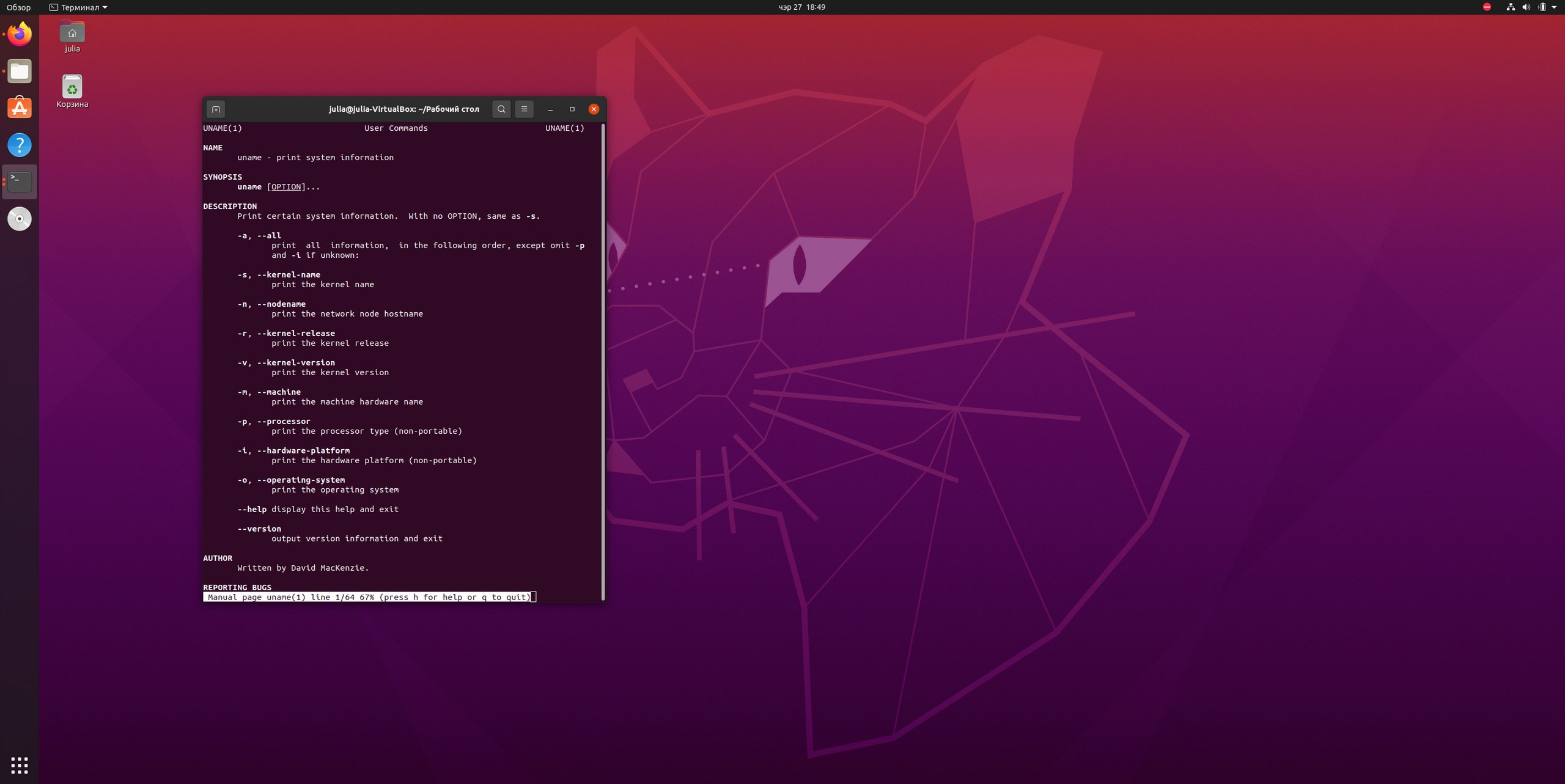
Раздел 4: Специальные файлы (устройства)

Раздел 5: Форматы файлов и протоколы

Раздел 6: Игры и развлечения

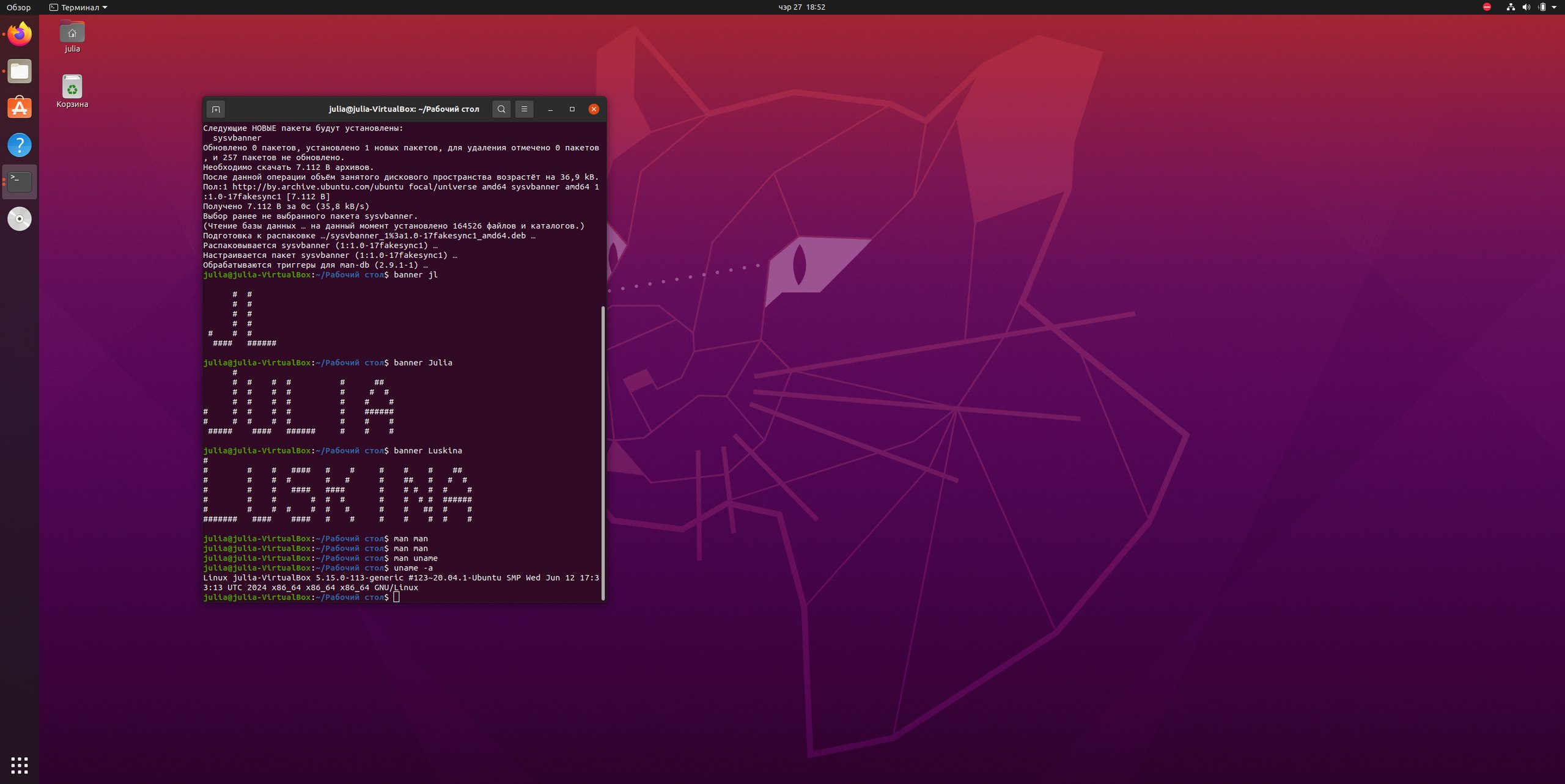
Раздел 7: Разное

Раздел 8: Коман



**ЗАДАНИЕ 4**

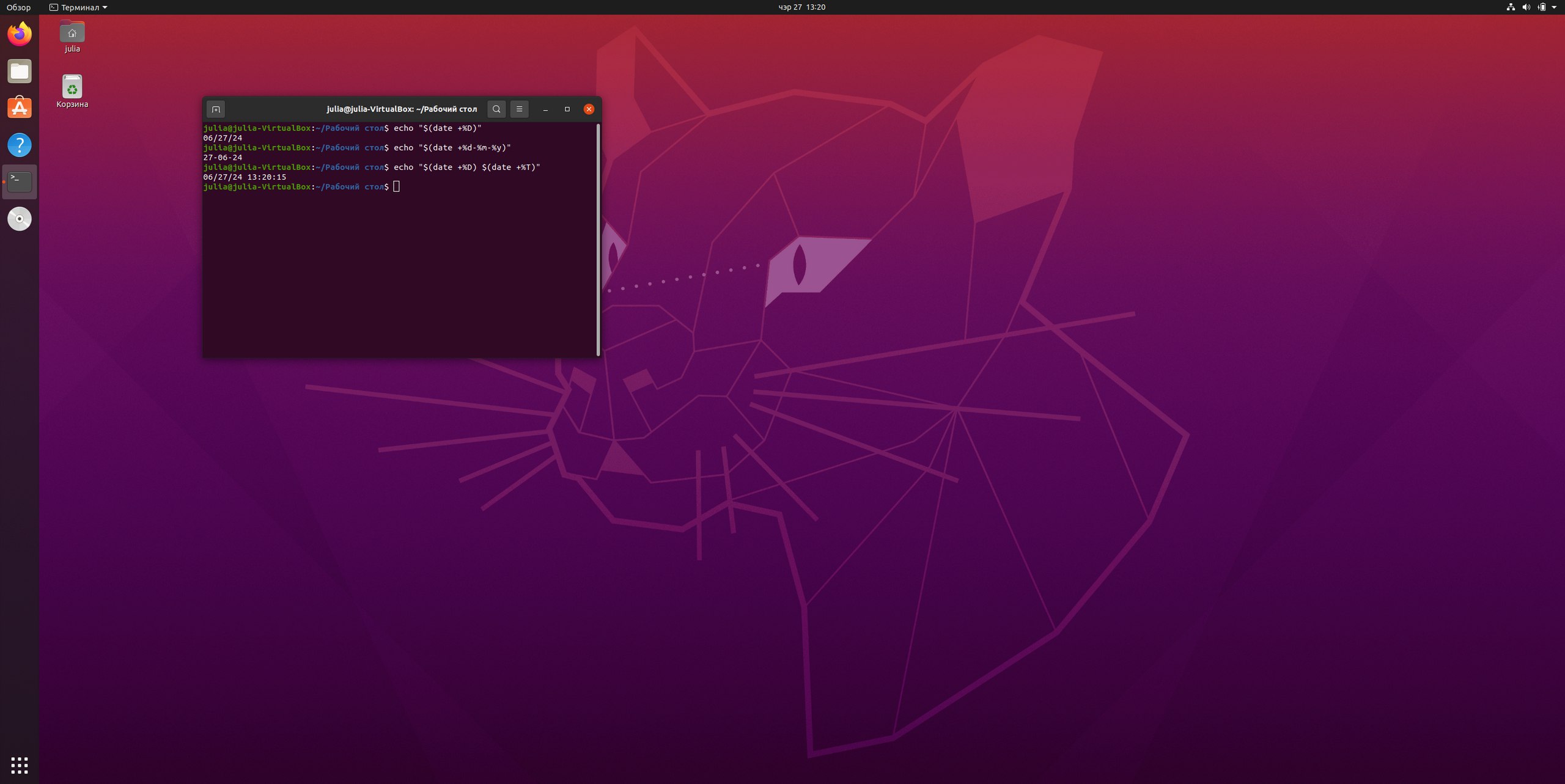
Определите имя машины, название и версию операционной системы, с которой Вы работаете. Каков аппаратный тип системы?



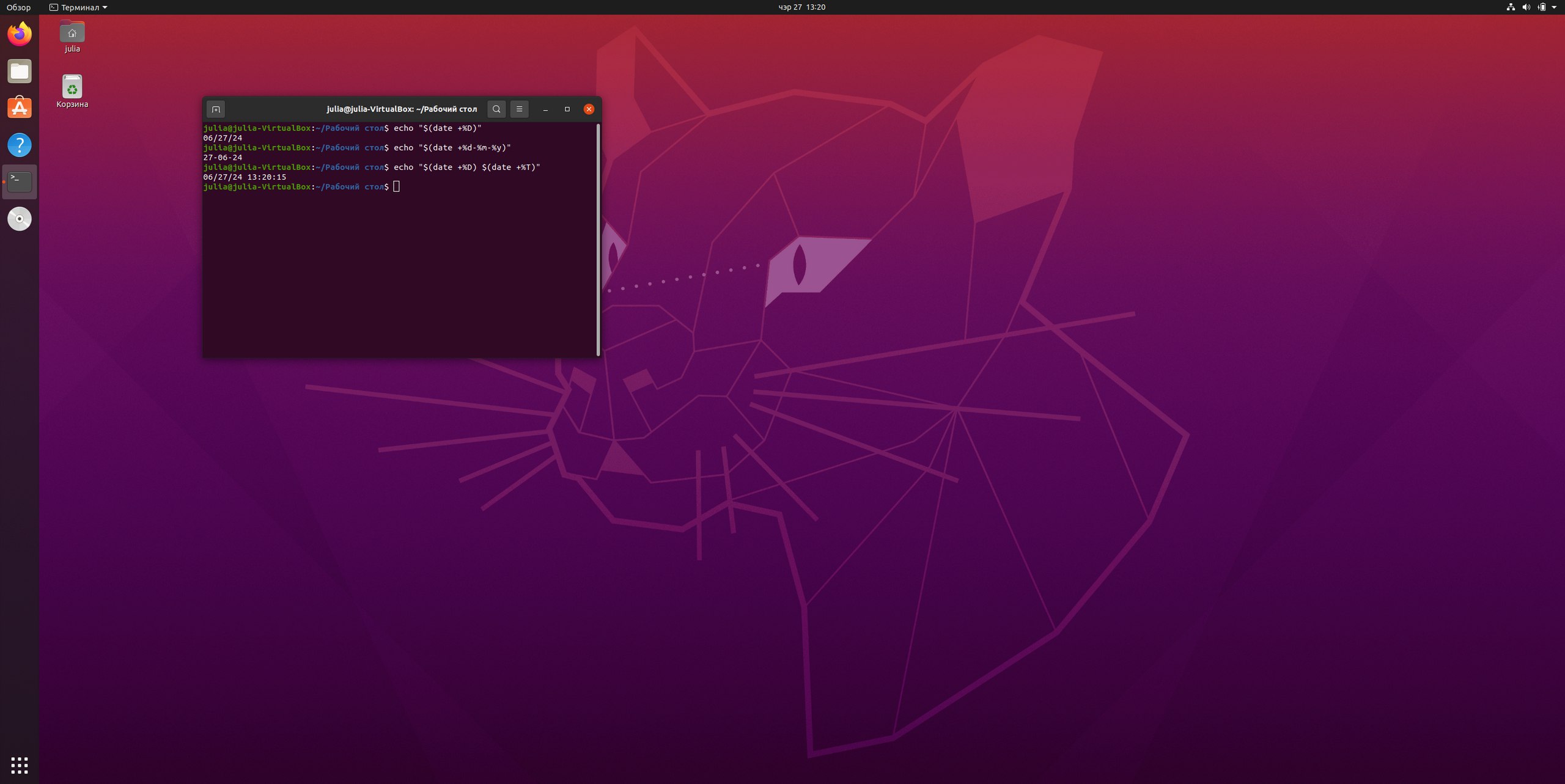
**ЗАДАНИЕ 5**

Выведите дату в форматах dd-mm-yy, mm-dd-yy hh:mm:ss.

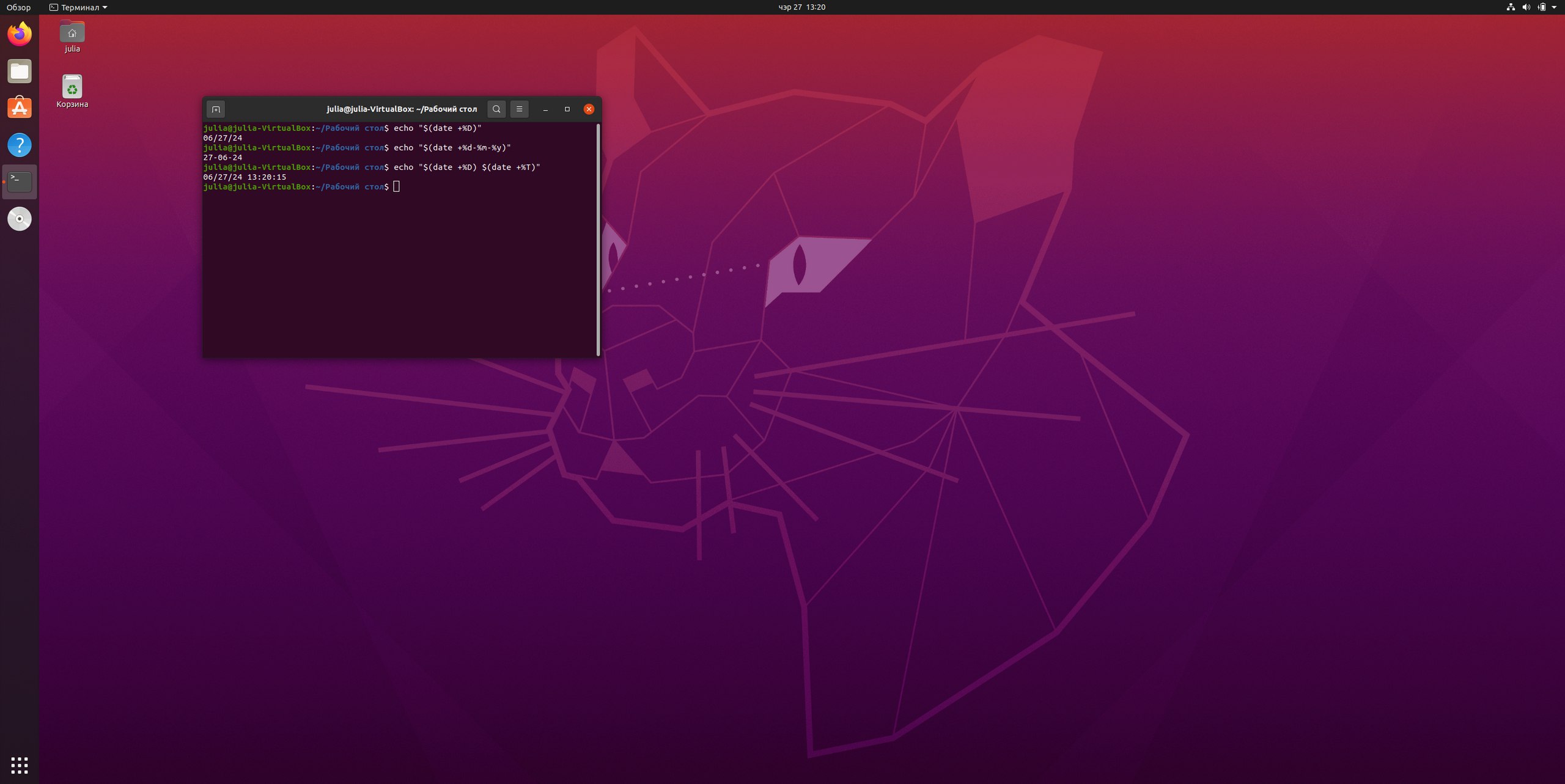
Дата в формате mm-dd-yy



Дата в формате dd-mm-yy

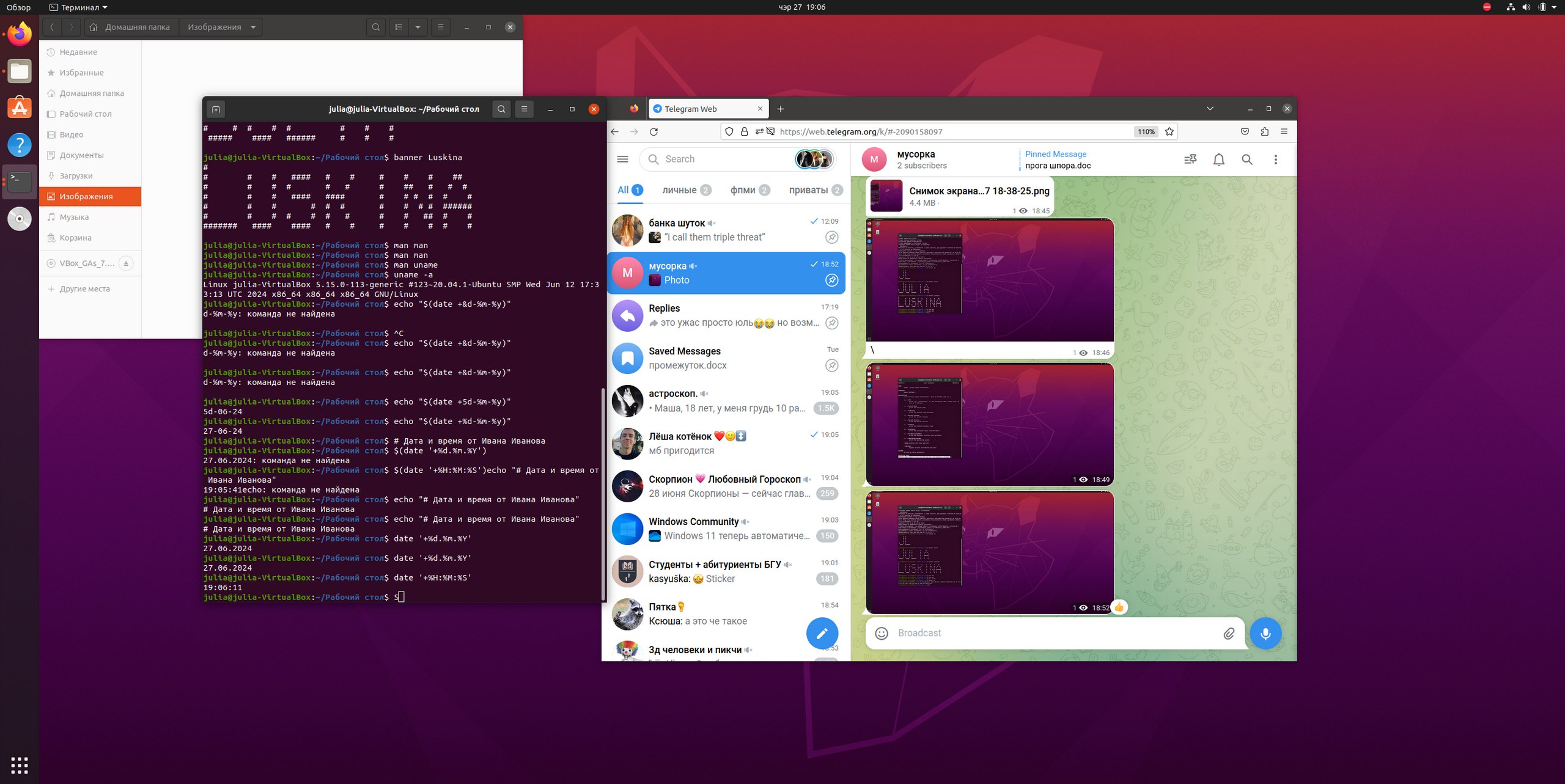


Дата и время в формате mm-dd-yy hh:mm:ss



$ - это стандартный визуальный индикатор командной строки, который помогает отличить ввод пользователя от вывода команды.

Еще вариант:



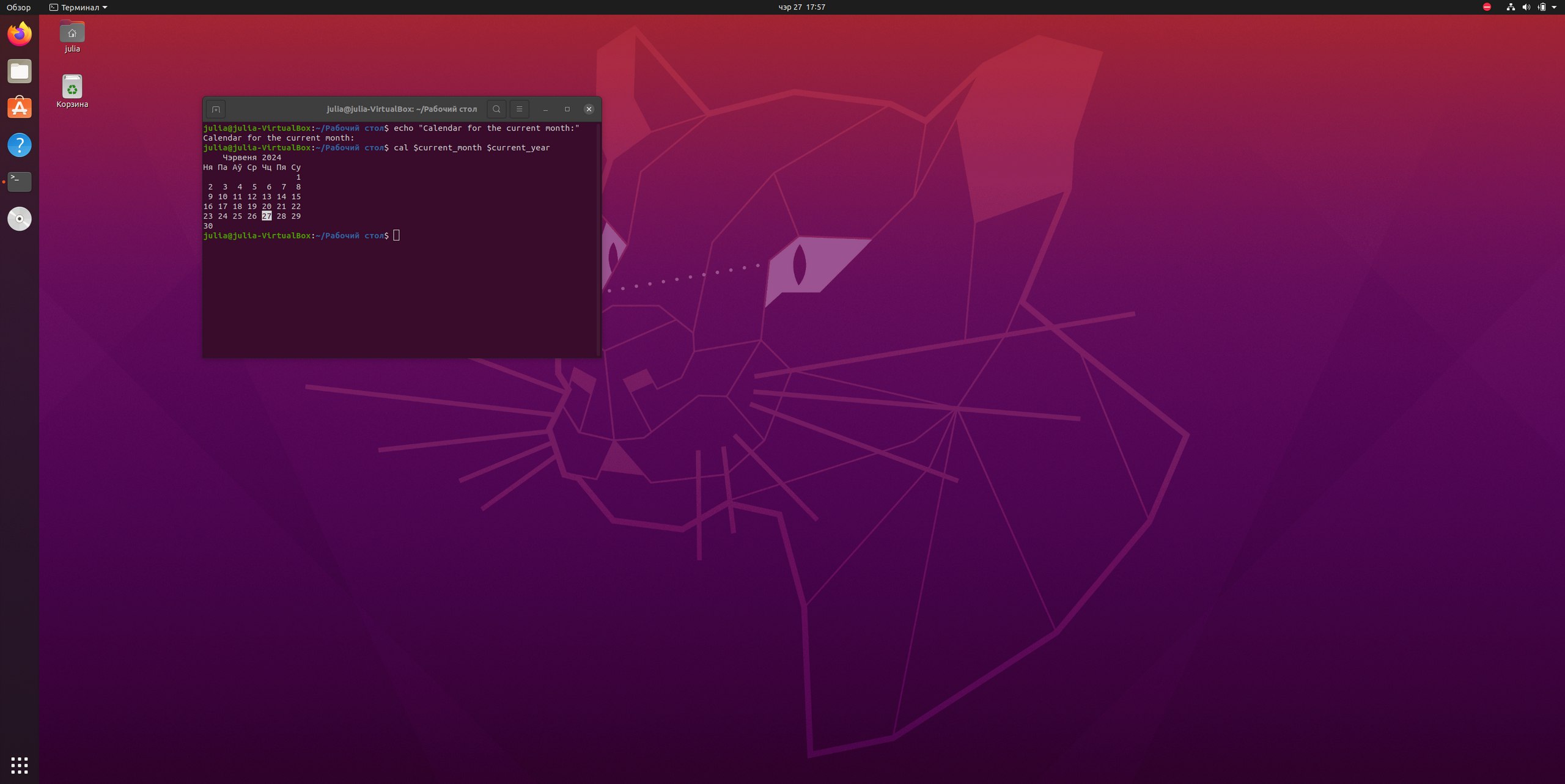
**ЗАДАНИЕ 6**

Выведите дату в две строки: на первой – день, месяц, год, на второй – текущее время, снабдив вывод комментарием с Вашим именем и фамилией.

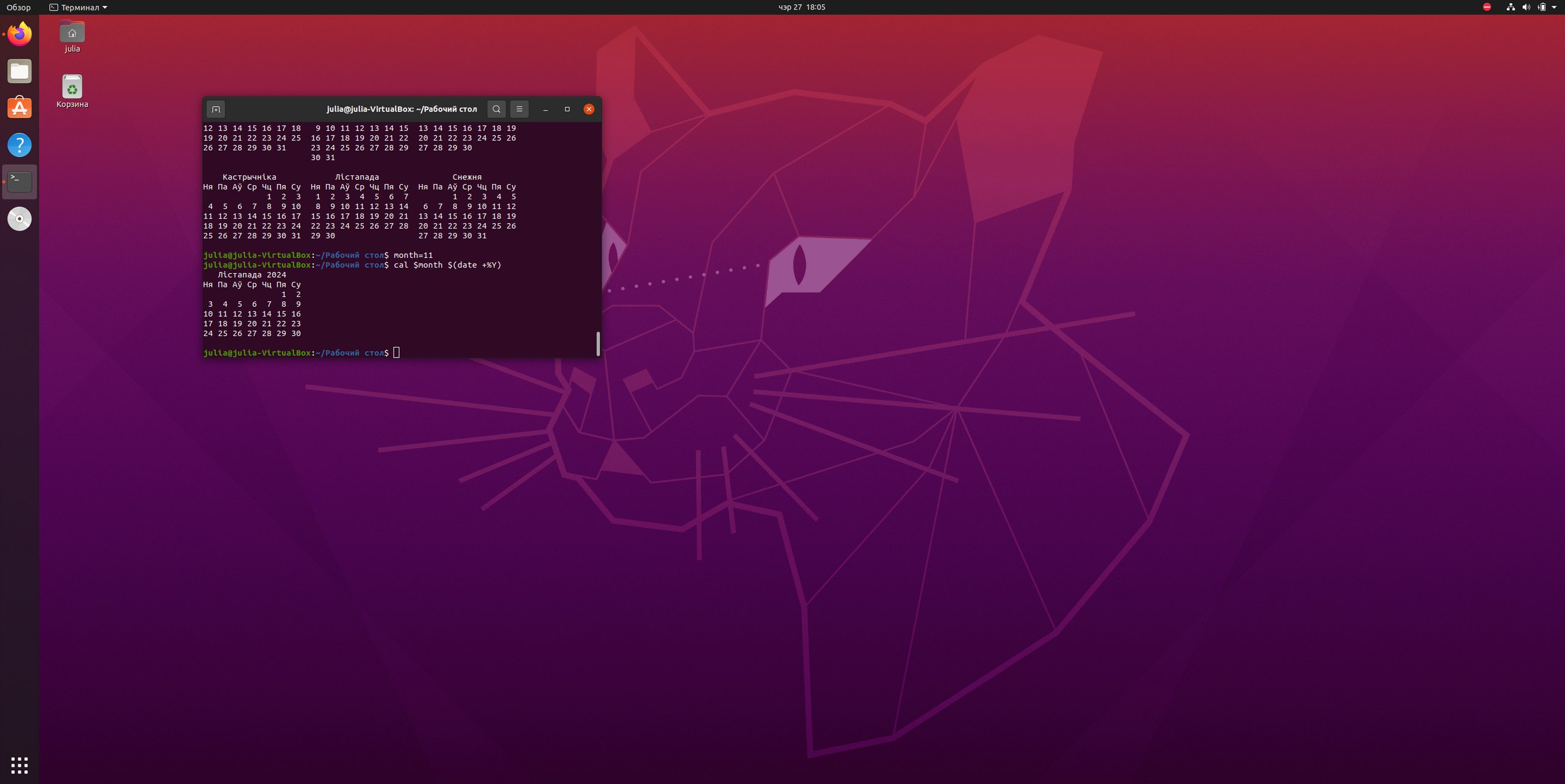


**ЗАДАНИЕ 7**

Выведите календарь на текущий месяц.

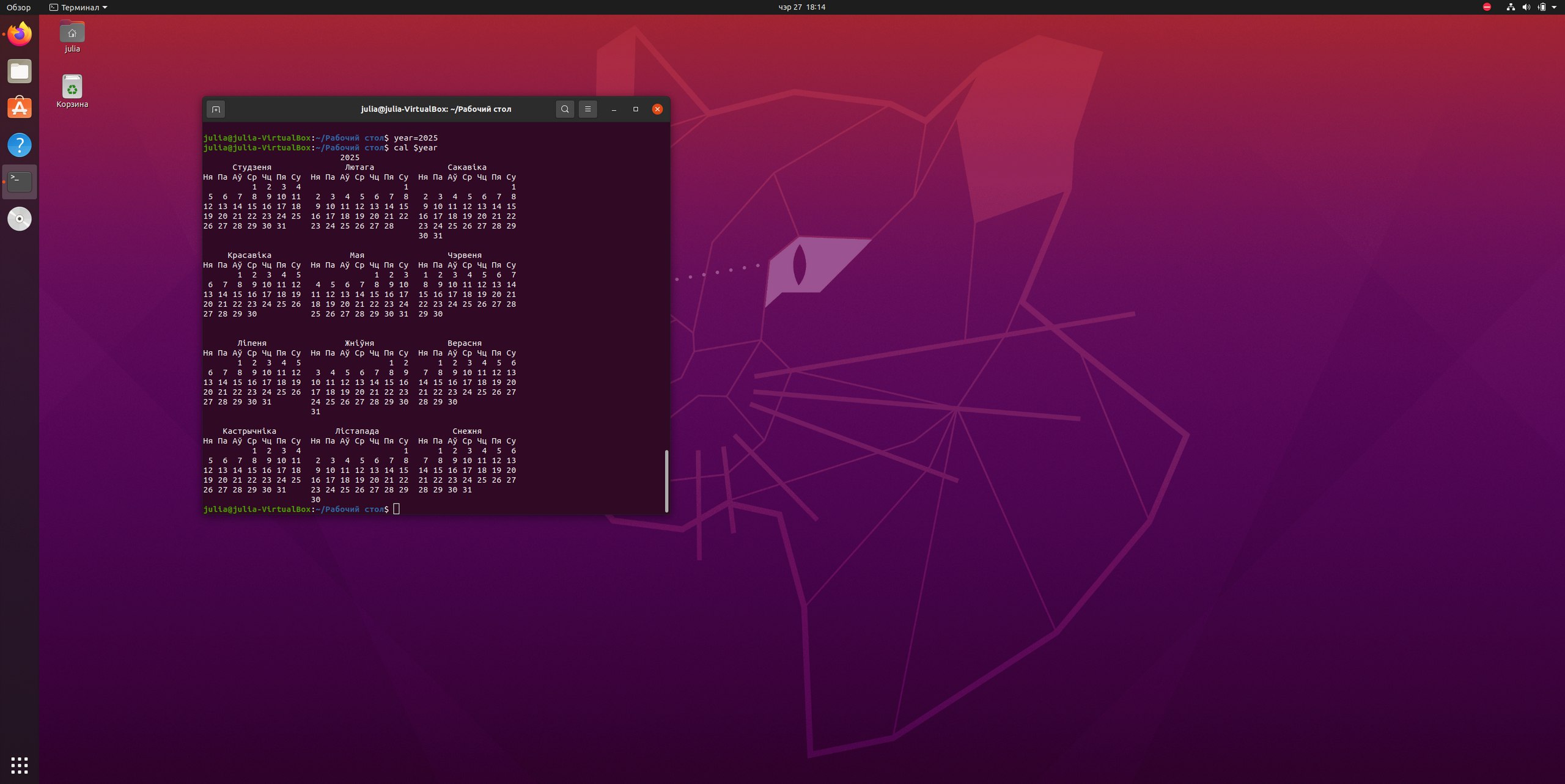


Выведите календарь на месяц Вашего рождения текущего года.



**ЗАДАНИЕ 8**

Выведите календарь на будущий год.

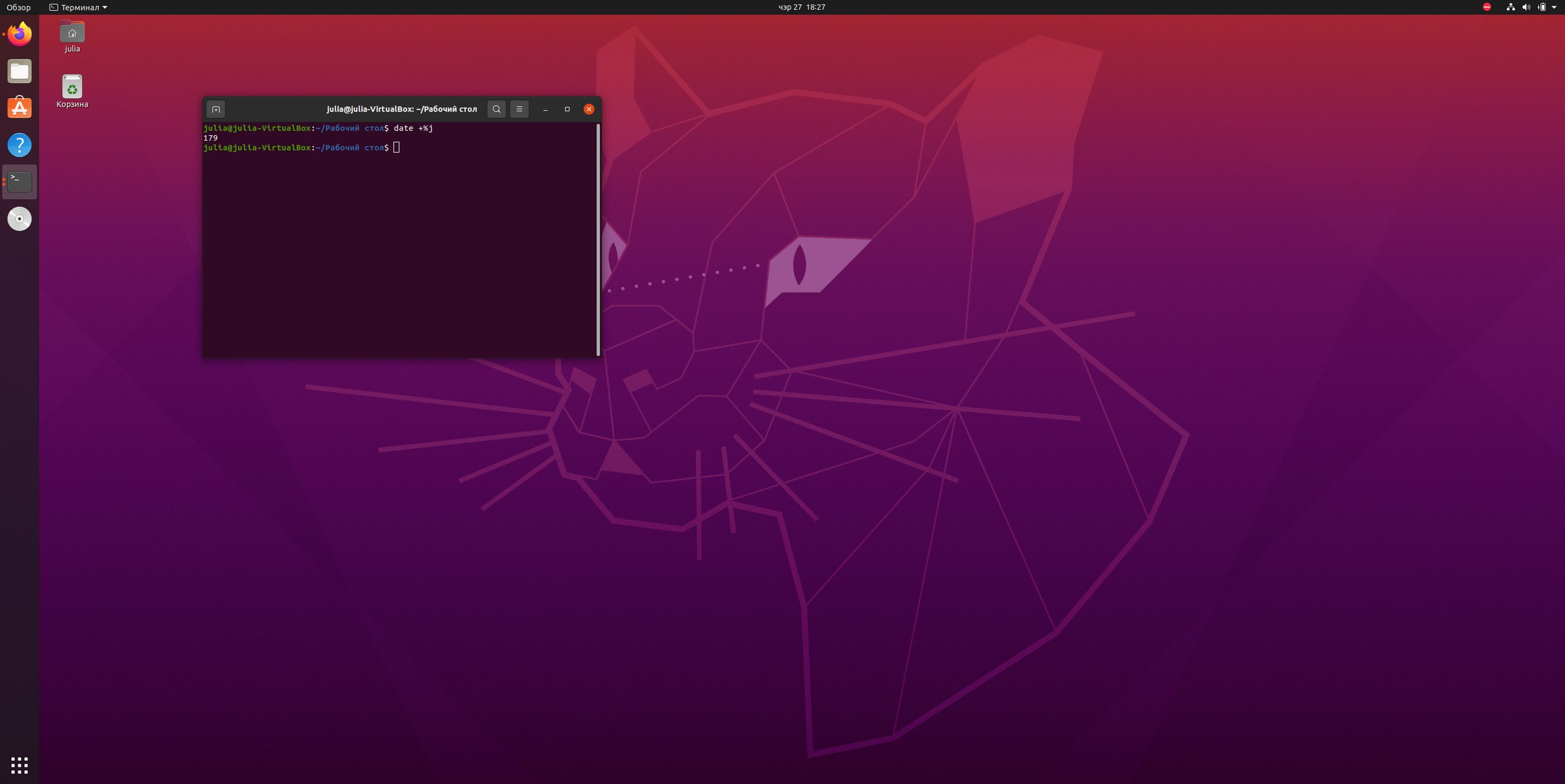


В каком столбце отображаются воскресенья?

По умолчанию в первом

**ЗАДАНИЕ 9**

Определите порядковый номер текущего дня с начала года



**РАЗДЕЛ 2. РАБОТА С ФАЙЛАМИ**

**ЗАДАНИЕ 1**

Создавать файл dirlist.txt, состоящий из 13 строк. Нечётные строки — пустые, чётные — содержат 2) Ваше ФИО, 4) место (город/посёлок) проживания, 6) улица проживания, 8) область проживания, 10) список Ваших хобби и интересов.

ЗАДАНИЕ 2

ЗАДАНИЕ 3

**РАЗДЕЛ 3. Изучение GitHub**

Установила утилиту Git:

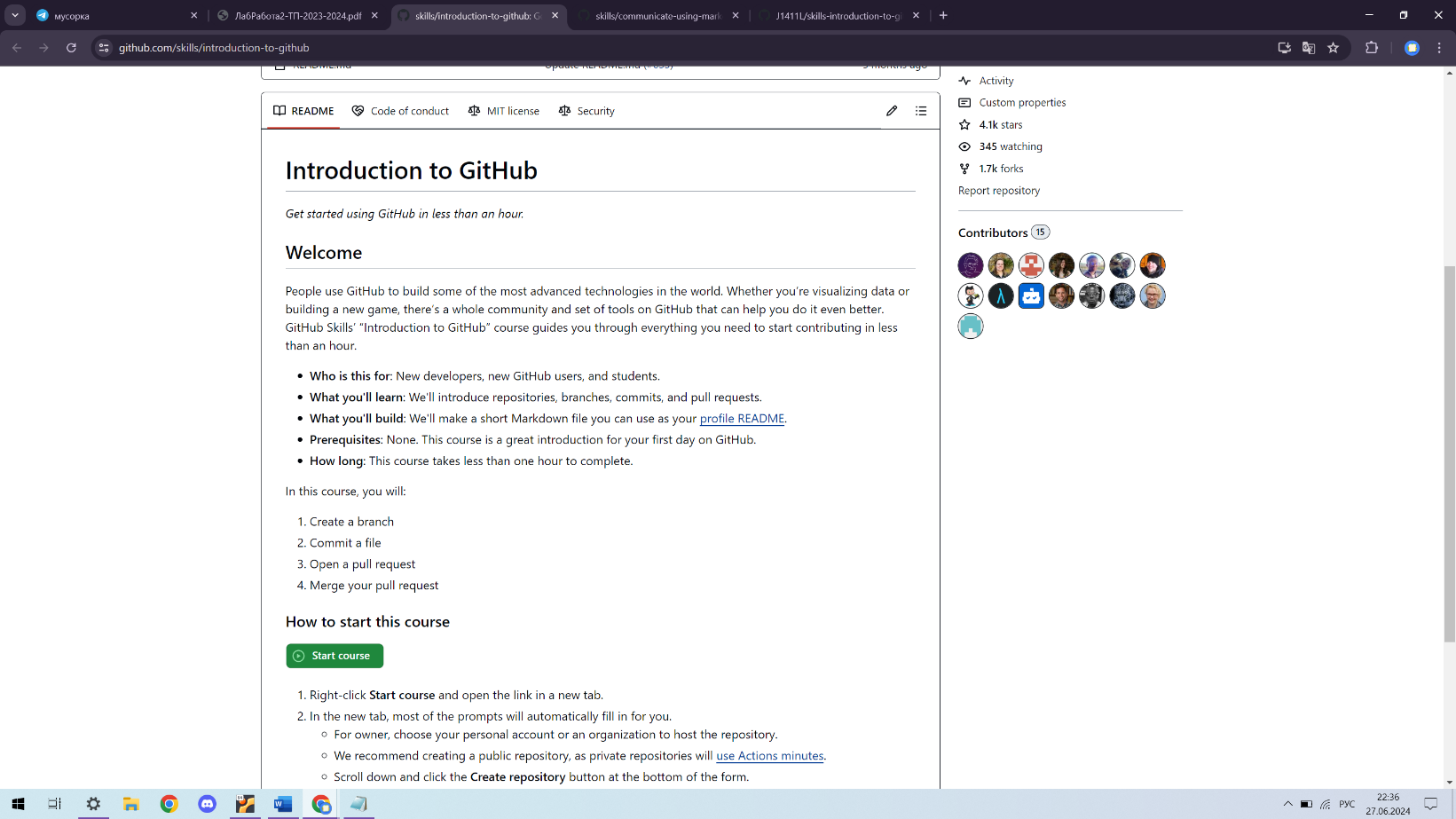
sudo apt install git

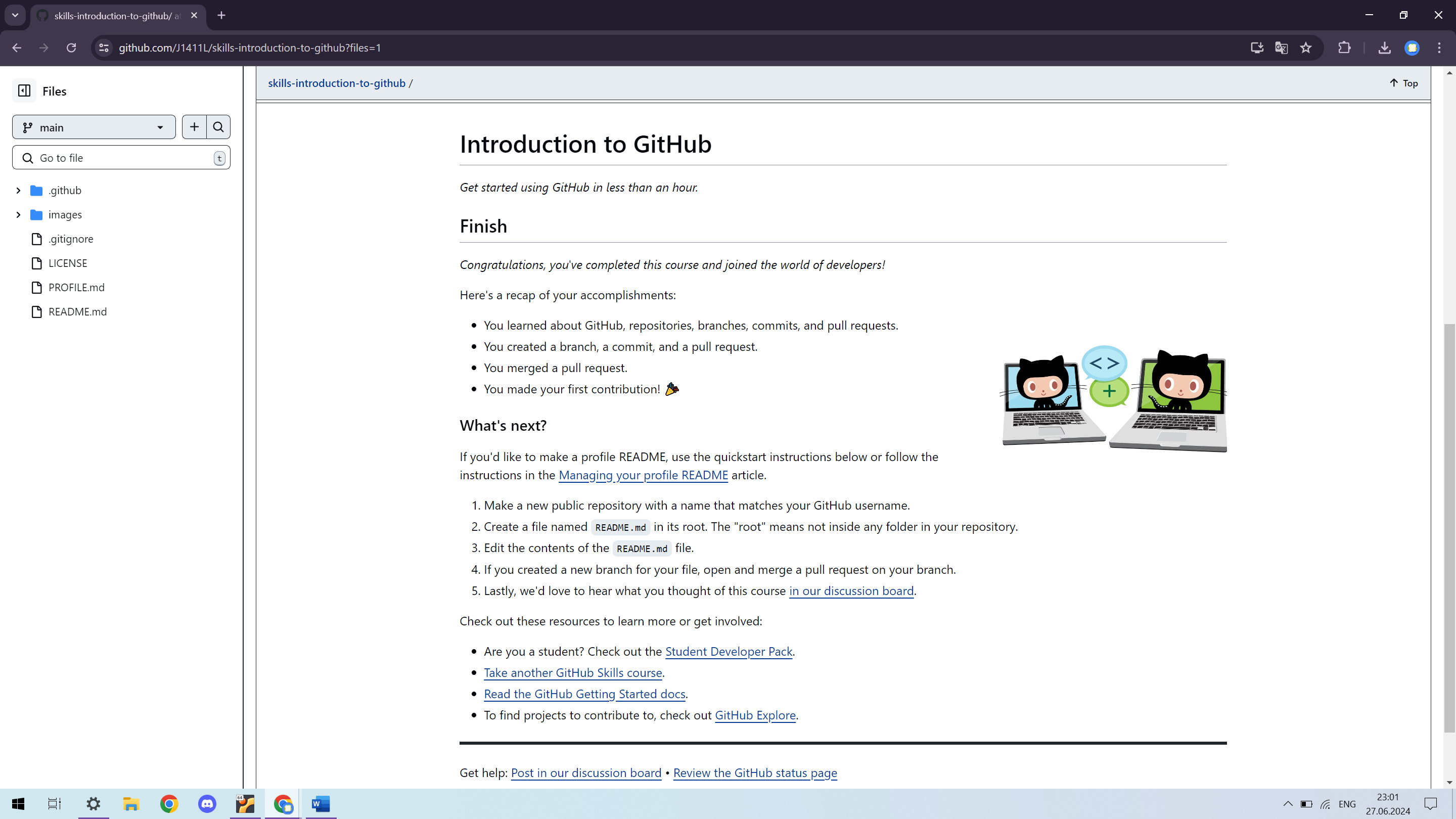
Задала данные пользователя (имя, адрес электронной почты), в консоли с помощью команд:

git config --global user.name "Ваше имя для подписи коммитов"

git config --global user.email "Ваш емейл, с которым зарегистрированы на github"

Прошла курс Introduction to GitHub. <https://github.com/skills/introduction-to-github>





Прошла курс Communicate using Markdown. <https://github.com/skills/communicate-using-markdown>

