



<https://github.com/al-sapsan>

email: [al.sapsan@mail.ru](mailto:al.sapsan@mail.ru)

# ПРАКТИЧЕСКАЯ РАБОТА № 1

курса «Linux для робототехников».

Основание:

изучение модуля «Введение в Linux для робототехники» курса «Linux для робототехников».

Исполнитель:

Соколов Олег Вадимович

# СОДЕРЖАНИЕ:

## Задание 1. Установка Ubuntu Linux.

- 1.1. Объект.
- 1.2. Выполненная работа:
- 1.3. Результат.
- 1.4. Примечание.

## Задание 2. Знакомство с консолью Linux.

- 2.1. Выполненная работа:
- 2.2. Результат.
- 2.3. Примечание.

## Задание 3. Установка программ в Ubuntu Linux на примере ROS.

- 3.1. Выполненная работа:
- 3.2. Результат.
- 3.3. Примечание.

## Задание 4. Консольные текстовые редакторы.

- 4.1. Выполненная работа:
  - 4.1.1. Практика работы с консольным текстовым редактором Nano
  - 4.1.2. Практика работы с консольным текстовым редактором Vim
- 4.2. Результат.
- 4.3. Примечание.

## Приложение 1. Иллюстративный материал.

## **Задание 1. Установка Ubuntu Linux.**

### 1.1. Объект [\[ссылка на содержание\]](#).

Портативный персональный компьютер Apple MacBook Pro модели A1286, оснащенный необходимым системным, прикладным и специализированным лицензионным программным обеспечением, включающим:

- операционную систему macOS High Sierra версия 10.13.6;
- операционную систему Microsoft Windows 7 Профессиональная, установленную с помощью программного обеспечения Apple Boot Camp.

### 1.2. Выполненная работа [\[ссылка на содержание\]](#):

- с официального сайта проекта Ubuntu был загружен ISO-образ дистрибутива Ubuntu Linux 20.04 LTS<sup>1</sup>;
- с официального сайта разработчика Balena было загружено программное обеспечение (далее – ПО) Etcher<sup>2</sup> для операционной системы (далее – ОС) macOS;
- с использованием ПО Etcher с помощью загруженного ISO-образа был получен загрузочный флеш-накопитель дистрибутива Ubuntu Linux 20.04 LTS;
- с помощью стандартных средств ОС macOS High Sierra было проведено выделение свободной области дискового пространства;
- с использованием полученного загрузочного накопителя проведена установка ОС Ubuntu Linux [см. [иллюстрацию 1](#) и [иллюстрацию 2](#)];
- с помощью ПО GPT fdisk<sup>3</sup> произведено восстановление загрузочной области ОС Windows 7<sup>4</sup>.

### 1.3. Результат [\[ссылка на содержание\]](#).

Получен ноутбук с тремя полностью работоспособными операционными системами.

### 1.4. Примечание [\[ссылка на содержание\]](#).

Это не первая моя установка Linux, до этого приходилось работать на дистрибутивах Red Hat, Fedora, Debian, Ubuntu и Lubuntu. Но на MacBook устанавливал Linux впервые. Установка прошла без проблем, за исключением восстановления Windows, но и это решилось через поиск в Интернете [\[ссылка\]](#). Возможно, описание восстановления загрузочной области можно разместить в Ваших учебных материалах для таких маководов как я.

## **Задание 2. Знакомство с консолью Linux.**

### 2.1. Выполненная работа [\[ссылка на содержание\]](#):

---

<sup>1</sup> <https://releases.ubuntu.com/20.04/>

<sup>2</sup> <https://etcher.balena.io/>

<sup>3</sup> <https://sourceforge.net/projects/gptfdisk/>

<sup>4</sup> <https://bullion.ru/forums/topic/4898-kak-vosstanovit-zagruzochnyj-disk-windows-bootcamp/>

- осуществлен просмотр содержимого текущего каталога с помощью команды `ls`, и с использованием опций `l`, `a`, `h` [см. [иллюстрацию 3](#)];
- выполнено перемещение между каталогами с помощью консольной команды `cd`, (в домашний каталог, в каталог на уровень выше, к предыдущему каталогу) [см. [иллюстрацию 4](#)];
- осуществлен просмотр текущего пути с помощью команды `pwd` [см. [иллюстрацию 5](#)];
- выполнено создание каталога для проекта с помощью команды `mkdir` и файла с помощью команды `touch` [см. [иллюстрацию 6](#)];
- выполнен вывод содержимого файла на экран с помощью команды `cat`, осуществлен вывод последних и первых строк файла с помощью команд `tail`, `head`, `more` и поиск текста в файле помощью команды `grep` [см. [иллюстрацию 7](#)];
- выполнено копирование и перемещение файлов и директорий с помощью команд `mv` и `cp` [см. [иллюстрацию 8](#)], а также удаление файла и каталога с помощью команды `rm` и опции `r` [см. [иллюстрацию 9](#)];
- использовались справки и руководства по встроенным и внешним командам оболочки `man`, `help`, и `-help` [см. [иллюстрацию 10](#), [иллюстрацию 11](#) и [иллюстрацию 12](#)];
- были выполнены команды от имени суперпользователя `sudo` [см. [иллюстрацию 13](#)];
- при выполнении вышеуказанных действий использовались сочетания клавиш (`Tab`, стрелки вверх/вниз, `Ctrl + C`, `Ctrl + R`, `Ctrl + A`, `Ctrl + E`, `Ctrl + D`, `Ctrl + L`, `Ctrl + Shift + C`, `Ctrl + Shift + V`).

## 2.2. Результат [\[ссылка на содержание\]](#).

Основные перечисленные команды и сочетания клавиш были изучены, запомнены и применены, по каждому пункту составлен отчет с иллюстрациями.

## 2.3. Примечание [\[ссылка на содержание\]](#).

Консольные команды в Linux практически не отличаются от макетовских, за небольшим исключением. А так для меня лично, что там терминал, что в Ubuntu терминал. Монументальной разницы я не заметил.

# Задание 3. Установка программ в Ubuntu Linux на примере ROS.

## 3.1. Выполненная работа [\[ссылка на содержание\]](#):

- проведено ознакомление с разными способами установки программного обеспечения в Ubuntu Linux (из репозитория, бинарных файлов и исходных кодов);
- проведено ознакомление с инструкцией по установке ПО ROS Noetic для ОС Ubuntu Linux<sup>5</sup>;

---

<sup>5</sup> <https://wiki.ros.org/noetic/Installation/Ubuntu>

- с соблюдением синтаксиса и последовательности команд, указанных в инструкции, произведена установка ПО ROS Noetic, в процессе которой было изучено: добавление репозитория (в sources.list), добавление ключей репозитория ROS (apt-key add -), обновление списка пакетов (apt update), поиск пакетов (apt search), установка пакетов (apt install);

- произведена настройка ROS (настройка переменного окружения .bashrc, доустановка пакетов rosinstall, roscdep, создание и инициализация рабочей области catkin\_ws, выполнение сборки и добавление рабочей области в .bashrc).

### 3.2. Результат [\[ссылка на содержание\]](#).

Установлено и полностью готово к работе окружение, для проверки работоспособности которого были выполнены следующие действия:

- выполнение команды roscd с переходом в каталог catkin\_ws/devel [см. [иллюстрацию 14](#)];

- выполнение команды roscore с получением необходимого результата [см. [иллюстрацию 14](#)];

- выполнение команды rosnode list с получением необходимого результата [см. [иллюстрацию 15](#)];

- выполнение примера с turtlesim\_node из [ROS/Tutorials/UnderstandingTopics](#) [см. [иллюстрации 16, 17](#)] в рамках которого применялись:

- rqt\_graph [см. [иллюстрации 18-20](#)];
- rostopic echo [см. [иллюстрацию 21](#)];
- rostopic list [см. [иллюстрацию 22](#)];
- rostopic type [см. [иллюстрацию 23](#)];
- rostopic pub [см. [иллюстрации 24, 25](#)];
- rostopic hz [см. [иллюстрацию 26](#)];
- rqt\_plot [см. [иллюстрацию 27](#)].

### 3.3. Примечание [\[ссылка на содержание\]](#).

Дополнительно был просмотрен видеотьюториал.

## **Задание 4. Консольные текстовые редакторы.**

### 4.1. Выполненная работа [\[ссылка на содержание\]](#):

*4.1.1. Практика работы с консольным текстовым редактором Nano, в рамках которой были выполнены следующие основные действия:*

- ознакомление с пользовательской документацией<sup>6</sup>;
- открытие файла, навигация по нему [см. [иллюстрацию 28](#)];
- поиск текста [см. [иллюстрацию 29](#)];
- редактирование, копирование и вставка текста [см. [иллюстрации 30, 31](#)]

---

<sup>6</sup> <https://help.ubuntu.ru/wiki/nano>

- замена текста с учётом регистра [см. [иллюстрацию 32, 33](#)];
- сохранение файла, выход [см. [иллюстрацию 34](#)].

*4.1.2. Практика работы с консольным текстовым редактором Vim, в рамках которой были выполнены следующие основные действия:*

- ознакомление с пользовательской документацией и настройкой редактора<sup>7</sup>;
- создание локального конфигурационного файла [см. [иллюстрацию 35](#)];
- сбор и изучение информации о существующих плагинах<sup>8</sup>;
- установка пакета vim-gtk для работы с системным буфером обмена;
- установка плагина NerdTree, который добавляет дерево каталогов;
- установка плагина vim-eunuch, который позволяет выполнять консольные команды прямо в командной строке редактора;
- открытие файла, навигация по нему [см. [иллюстрацию 36](#)];
- поиск текста [см. [иллюстрацию 37](#)];
- редактирование, копирование и вставка текста [см. [иллюстрации 38, 39](#)];
- замена текста с учётом регистра [см. [иллюстрацию 40](#)];
- сохранение файла, выход [см. [иллюстрацию 41](#)].

#### 4.2. Результат [\[ссылка на содержание\]](#).

Проведена практика работы с двумя консольными текстовыми редакторами: Nano и Vim. Изготовлены иллюстрации к соответствующим пунктам и вынесены в Приложение № 1.

#### 4.3. Примечание [\[ссылка на содержание\]](#).

Для начала, следует заметить, что у меня ноутбук Apple MacBook Pro и, вследствие этого, кнопка Alt просто отсутствует. Начал я свою практическую работу с изучения редактора Nano, как наиболее простого и понятного, даже не удостоив вниманием документацию на него. Но сразу столкнулся с проблемой, что у меня ничего не получается. Это был локальный шок! Ведь в редакторе нет ничего сложного. Обратился к официальной документации [\[ссылка\]](#), но там моя проблема не описана. И только после этого набрел на русскоязычную документацию [\[ссылка\]](#), где моя «проблема» решалась простым использованием клавиши Esc.

Но на этом мои приключения не окончились. Так как я по своей опрометчивости сделал файл с русским текстом, то мне приходилось постоянно менять язык клавиатуры. И если в Vim ты видишь, какая команда печатается, то в Nano при нажатии комбинации клавиш просто ничего не происходит. И пока до тебя дойдет, что ты забыл переключить клавиатуру, ты уже тихо ненавидишь данный редактор. Даже в «Лексиконе» под DOS работать было легче!

<sup>7</sup> <https://losst.pro/nastrojka-vim>

<sup>8</sup> <https://neg-serg.github.io/2016/01/vim-plugin-overview/#valloricyoucomplete>

Исходя из моего многострадального опыта общения с Nano, к работе в Vim я подошел системно, сначала полностью изучив документацию по его настройке и работе. И вот тут я обнаружил, что Vim, оказывается, очень удобный, функциональный и настраиваемый редактор, который по функционалу превосходит Nano практически по всем параметрам. Это была любовь с первого раза! ☺

Да, Vim требует определенной тренировки и практики работы в нем, но здесь мне вспомнилось мое обучение набору текста на клавиатуре «слепым способом» (не смотря на клавиатуру). По началу, я печатал один абзац текста в час, потом быстрее и к концу месяца, я уже не смотрел на клавиатуру. Так и с Vim, если немного потрудиться и напрячь свои извилины, память и волю, то на выходе получится очень эффективный, быстрый и функциональный результат. Мой выбор, однозначно, Vim!

Приложение: иллюстративный материал на 14 л., в 1 экз.

**О.В. Соколов**

## ИЛЛЮСТРАТИВНЫЙ МАТЕРИАЛ

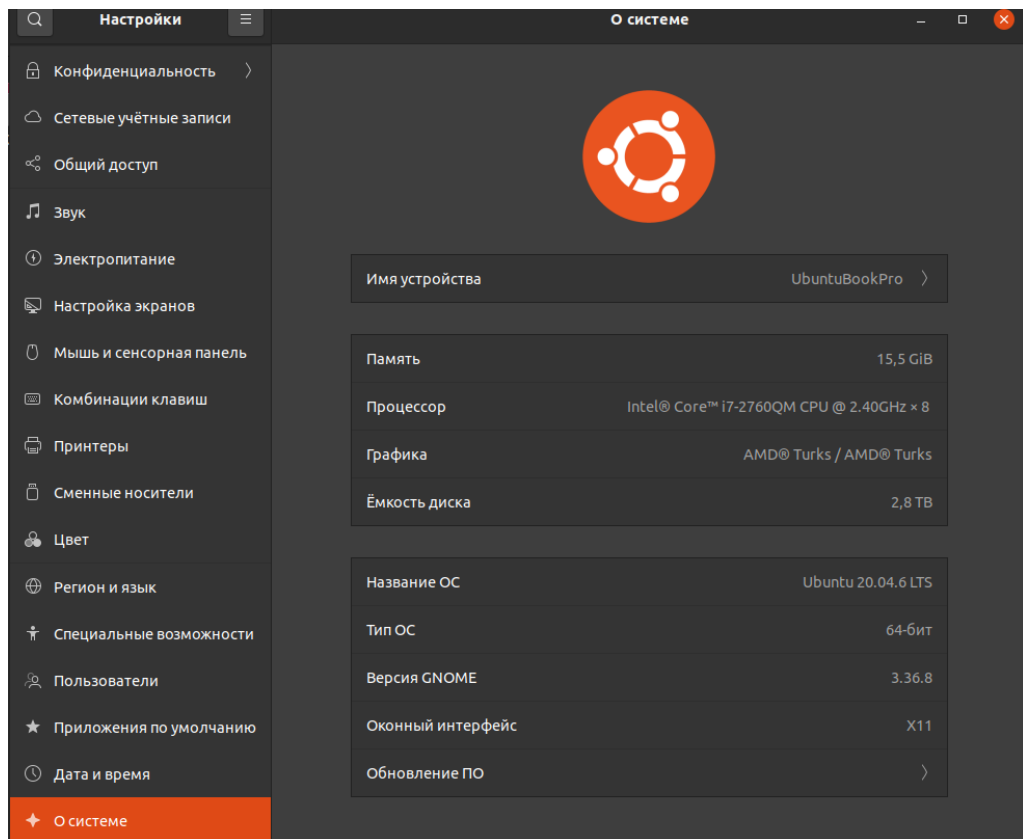


Иллюстрация 1. Вкладка «О системе» приложения «Настройки» [[обратная ссылка](#)].

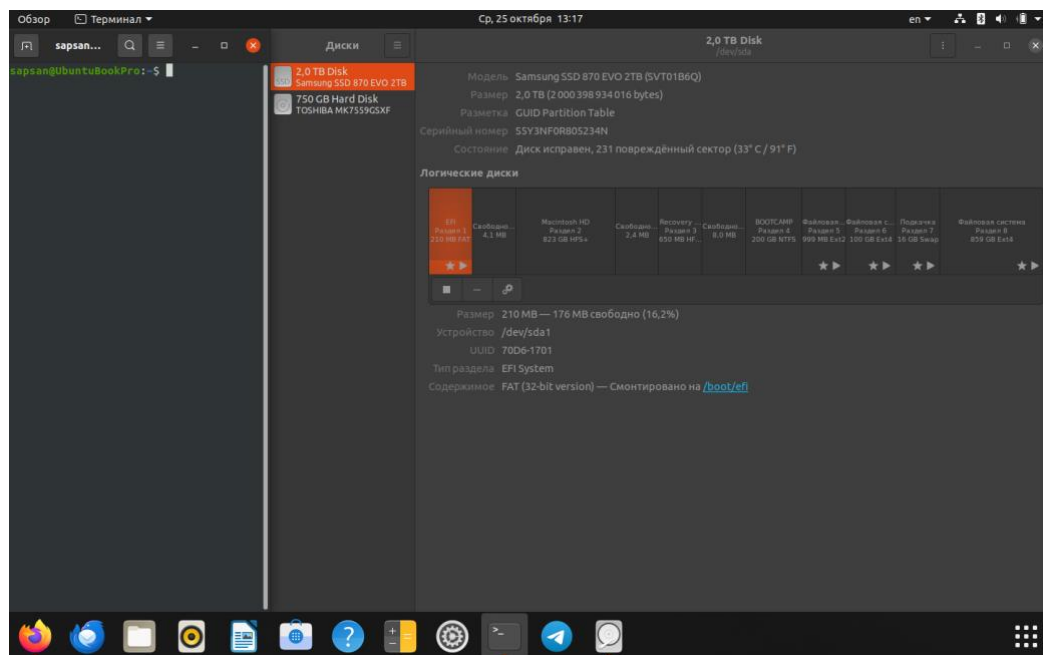


Иллюстрация 2. Окно запущенного терминала и окно ПО «Диски» [[обратная ссылка](#)].



```
sapsan@UbuntuBookPro:~$ ls
catkin_ws  snap  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
sapsan@UbuntuBookPro:~$ ls -l
итого 40
drwxrwxr-x 5 sapsan sapsan 4096 окт 22 01:10 catkin_ws
drwx----- 4 sapsan sapsan 4096 окт 19 23:57 snap
drwxr-xr-x 2 sapsan sapsan 4096 окт 19 21:15 Видео
drwxr-xr-x 3 sapsan sapsan 4096 окт 22 18:00 Документы
drwxr-xr-x 2 sapsan sapsan 4096 окт 22 14:31 Загрузки
drwxr-xr-x 2 sapsan sapsan 4096 окт 25 14:17 Изображения
drwxr-xr-x 2 sapsan sapsan 4096 окт 19 21:15 Музыка
drwxr-xr-x 2 sapsan sapsan 4096 окт 19 21:15 Общедоступные
drwxr-xr-x 3 sapsan sapsan 4096 окт 21 12:40 'Рабочий стол'
drwxr-xr-x 2 sapsan sapsan 4096 окт 19 21:15 Шаблоны
sapsan@UbuntuBookPro:~$ ls -h
catkin_ws  snap  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
sapsan@UbuntuBookPro:~$ ls -lh
итого 40K
drwxrwxr-x 5 sapsan sapsan 4,0K окт 22 01:10 catkin_ws
drwx----- 4 sapsan sapsan 4,0K окт 19 23:57 snap
drwxr-xr-x 2 sapsan sapsan 4,0K окт 19 21:15 Видео
drwxr-xr-x 3 sapsan sapsan 4,0K окт 22 18:00 Документы
drwxr-xr-x 2 sapsan sapsan 4,0K окт 22 14:31 Загрузки
drwxr-xr-x 2 sapsan sapsan 4,0K окт 25 14:17 Изображения
drwxr-xr-x 2 sapsan sapsan 4,0K окт 19 21:15 Музыка
drwxr-xr-x 2 sapsan sapsan 4,0K окт 19 21:15 Общедоступные
drwxr-xr-x 3 sapsan sapsan 4,0K окт 21 12:40 'Рабочий стол'
drwxr-xr-x 2 sapsan sapsan 4,0K окт 19 21:15 Шаблоны
sapsan@UbuntuBookPro:~$ ls -ah
.      .bash_logout  catkin_ws  .local      .ros        .sudo_as_admin_successful  Видео  Изображения  'Рабочий стол'
..     .bashrc      .config   .mozilla    snap        .thunderbird  Документы  Музыка  Шаблоны
.bash_history  .cache     .gnupg    .profile    .ssh        .viminfo     Загрузки  Общедоступные
```

Иллюстрация 3. Использование консольной команды ls [\[обратная ссылка\]](#).

```
sapsan@UbuntuBookPro:~$ cd Изображения/
sapsan@UbuntuBookPro:~/Изображения$ cd ..
sapsan@UbuntuBookPro:~$ cd Документы/
sapsan@UbuntuBookPro:~/Документы$ cd -
sapsan@UbuntuBookPro:~/Изображения$ cd -
/home/sapsan
sapsan@UbuntuBookPro:~$ cd -
/home/sapsan/Изображения
sapsan@UbuntuBookPro:~/Изображения$
```

Иллюстрация 4. Использование консольной команды cd [\[обратная ссылка\]](#).

```
sapsan@UbuntuBookPro:~$ pwd
/home/sapsan
sapsan@UbuntuBookPro:~$ cd ..
sapsan@UbuntuBookPro:/home$ cd -
/home/sapsan
```

Иллюстрация 5. Использование консольной команды pwd [\[обратная ссылка\]](#).

```
sapsan@UbuntuBookPro:~$ mkdir my_test_directory
sapsan@UbuntuBookPro:~$ ls
catkin_ws  my_test_directory  snap  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
sapsan@UbuntuBookPro:~$ cd my_test_directory/
sapsan@UbuntuBookPro:~/my_test_directory$ ls
sapsan@UbuntuBookPro:~/my_test_directory$ touch first_test_file.yaml
sapsan@UbuntuBookPro:~/my_test_directory$ ls
first_test_file.yaml
sapsan@UbuntuBookPro:~/my_test_directory$
```

Иллюстрация 6. Использование консольных команд mkdir и touch [\[обратная ссылка\]](#).

```
sapsan@UbuntuBookPro:~$ cd Документы/
sapsan@UbuntuBookPro:~/Документы$ cat text_test_file.txt
Привет!
Я тестовый текстовый файл :)
sapsan@UbuntuBookPro:~/Документы$ more text_test_file.txt
Привет!
Я тестовый текстовый файл :)
sapsan@UbuntuBookPro:~/Документы$ less text_test_file.txt
sapsan@UbuntuBookPro:~/Документы$ tail -n1 text_test_file.txt
Я тестовый текстовый файл :)
sapsan@UbuntuBookPro:~/Документы$ head -n1 text_test_file.txt
Привет!
sapsan@UbuntuBookPro:~/Документы$ grep "er" text_test_file.txt
Привет!
sapsan@UbuntuBookPro:~/Документы$ grep "re" text_test_file.txt
Я тестовый текстовый файл :)
sapsan@UbuntuBookPro:~/Документы$ grep -A1 "re" text_test_file.txt
Я тестовый текстовый файл :)
sapsan@UbuntuBookPro:~/Документы$ grep -A1 "er" text_test_file.txt
Привет!
Я тестовый текстовый файл :)
sapsan@UbuntuBookPro:~/Документы$
```

Иллюстрация 7. Использование консольных команд cat, tail, head, more и grep [\[обратная ссылка\]](#).

```
sapsan@UbuntuBookPro:~$ mkdir my_test_directory
sapsan@UbuntuBookPro:~$ ls
catkin_ws  my_test_directory  snap  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
sapsan@UbuntuBookPro:~$ cd my_test_directory/
sapsan@UbuntuBookPro:~/my_test_directory$ ls
sapsan@UbuntuBookPro:~/my_test_directory$ touch first_test_file.yaml
sapsan@UbuntuBookPro:~/my_test_directory$ ls
first_test_file.yaml
sapsan@UbuntuBookPro:~/my_test_directory$ mv first_test_file.yaml second_test_file.yaml
sapsan@UbuntuBookPro:~/my_test_directory$ ls
second_test_file.yaml
sapsan@UbuntuBookPro:~/my_test_directory$ mv second_test_file.yaml ..
sapsan@UbuntuBookPro:~/my_test_directory$ cd
sapsan@UbuntuBookPro:~$ ls
catkin_ws  my_test_directory  second_test_file.yaml  snap  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'
sapsan@UbuntuBookPro:~$ ls
catkin_ws  my_test_directory  snap  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
sapsan@UbuntuBookPro:~$ cd my_test_directory/
sapsan@UbuntuBookPro:~/my_test_directory$ ls
second_test_file.yaml
sapsan@UbuntuBookPro:~/my_test_directory$ cp second_test_file.yaml copy_test_file.yaml
sapsan@UbuntuBookPro:~/my_test_directory$ ls
copy_test_file.yaml  second_test_file.yaml
sapsan@UbuntuBookPro:~/my_test_directory$ cp second_test_file.yaml ../copy_test_file.yaml
sapsan@UbuntuBookPro:~/my_test_directory$ ls -
catkin_ws  copy_test_file.yaml  my_test_directory  snap  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'
sapsan@UbuntuBookPro:~/my_test_directory$
```

Иллюстрация 8. Использование консольных команд mv и cp [[обратная ссылка](#)].

```
sapsan@UbuntuBookPro:~/my_test_directory$ rm copy_test_file.yaml
sapsan@UbuntuBookPro:~/my_test_directory$ ls
second_test_file.yaml
sapsan@UbuntuBookPro:~/my_test_directory$ rm ~/copy_test_file.yaml
sapsan@UbuntuBookPro:~/my_test_directory$ ls -
catkin_ws  my_test_directory  snap  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
sapsan@UbuntuBookPro:~/my_test_directory$ rm -r ~/my_test_directory/
sapsan@UbuntuBookPro:~/my_test_directory$ ls -
catkin_ws  snap  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
sapsan@UbuntuBookPro:~/my_test_directory$
```

Иллюстрация 9. Использование консольной команды rm [[обратная ссылка](#)].

```
GREP(1) User Commands GREP(1)
NAME
    grep, egrep, fgrep, rgrep - print lines that match patterns
SYNOPSIS
    grep [OPTION...] PATTERNS [FILE...]
    grep [OPTION...] -e PATTERNS ... [FILE...]
    grep [OPTION...] -f PATTERN_FILE ... [FILE...]
DESCRIPTION
    grep searches for PATTERNS in each FILE. PATTERNS is one or more patterns separated by newline characters, and grep prints each line that matches a pattern. Typically PATTERNS should be quoted when grep is used in a shell command.
    A FILE of "-" stands for standard input. If no FILE is given, recursive searches examine the working directory, and nonrecursive searches read standard input.
    In addition, the variant programs egrep, fgrep and rgrep are the same as grep -E, grep -F, and grep -r, respectively. These variants are deprecated, but are provided for backward compatibility.
OPTIONS
    Generic Program Information
    --help Output a usage message and exit.
    -V, --version Output the version number of grep and exit.
    Pattern Syntax
    -E, --extended-regexp Interpret PATTERNS as extended regular expressions (EREs, see below).
    -F, --fixed-strings Interpret PATTERNS as fixed strings, not regular expressions.
    -G, --basic-regexp Interpret PATTERNS as basic regular expressions (BREs, see below). This is the default.
    -P, --perl-regexp Interpret PATTERNS as Perl-compatible regular expressions (PCREs). This option is experimental when combined with the -z (--null-data) option, and grep -P may warn of unimplemented features.
```

Иллюстрация 10. Использование консольной команды man [[обратная ссылка](#)].

```
sapsan@UbuntuBookPro:~$ help cd
cd: cd [-L|[-P [-e]] [-@]] [каталог]
Change the shell working directory.

Change the current directory to DIR. The default DIR is the value of the
HOME shell variable.

The variable CDPATH defines the search path for the directory containing
DIR. Alternative directory names in CDPATH are separated by a colon (:).
A null directory name is the same as the current directory. If DIR begins
with a slash (/), then CDPATH is not used.

If the directory is not found, and the shell option 'cdable_vars' is set,
the word is assumed to be a variable name. If that variable has a value,
its value is used for DIR.

Options:
-L      force symbolic links to be followed: resolve symbolic
-P      links in DIR after processing instances of '..'
        use the physical directory structure without following
        symbolic links: resolve symbolic links in DIR before
        processing instances of '..'
-e      if the -P option is supplied, and the current working
        directory cannot be determined successfully, exit with
        a non-zero status
-@      on systems that support it, present a file with extended
        attributes as a directory containing the file attributes

The default is to follow symbolic links, as if '-L' were specified.
'..' is processed by removing the immediately previous pathname component
back to a slash or the beginning of DIR.

Exit Status:
Returns 0 if the directory is changed, and if $PWD is set successfully when
-P is used; non-zero otherwise.
sapsan@UbuntuBookPro:~$
```

Иллюстрация 11. Использование консольной команды help [[обратная ссылка](#)].

```
sapsan@UbuntuBookPro:~$ man grep
sapsan@UbuntuBookPro:~$ grep --help
Использование: grep [ПАРАМЕТР]... ШАБЛОНЫ [ФАЙЛ]...
Поиск ШАБЛОНОВ в каждом ФАЙЛЕ.
Пример: grep -l 'hello world' menu.h main.c
ШАБЛОНЫ можно указать несколько, разделяя их символом новой строки.

Шаблон выбора и его интерпретация:
-E, --extended-regexp  ШАБЛОНЫ — расширенные регулярные выражения
-F, --fixed-regexp     ШАБЛОНЫ — строки
-G, --basic-regexp     ШАБЛОНЫ — простые регулярные выражения
-P, --perl-regexp      ШАБЛОНЫ — регулярные выражения языка Perl
-e, --regexp=ШАБЛОНЫ  использовать ШАБЛОНЫ для поиска
-f, --file=ФАЙЛ       брать ШАБЛОНЫ из ФАЙЛА
-i, --ignore-case       игнорировать различие регистра
-no-ignore-case        учитывать регистр (по умолчанию)
-m, --word-regexp      совпадение обязательно с целым словом
-x, --line-regexp      совпадение обязательно с целой строкой
-z, --null-data        строки разделяются байтом с нулевым значением, а не
                        символом конца строки

Разное:
-s, --no-messages      не показывать сообщения об ошибках
-v, --invert-match      выбирать не подходящие строки
-V, --version           показать информацию о версии и закончить работу
--help                 показать эту справку и закончить работу
```

Иллюстрация 12. Использование консольной справки -help [[обратная ссылка](#)].

```
sapsan@UbuntuBookPro:~$ ls
catkin_ws  snap  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
sapsan@UbuntuBookPro:~$ cd Документы/
sapsan@UbuntuBookPro:~/Документы$ cd
sapsan@UbuntuBookPro:~$ sudo su
[sudo] пароль для sapsan:
root@UbuntuBookPro:/home/sapsan# exit
sapsan@UbuntuBookPro:~$ pwd
/home/sapsan
sapsan@UbuntuBookPro:~$ cd ..
sapsan@UbuntuBookPro:/home$ cd -
/home/sapsan
```

Иллюстрация 13. Использование консольной команды суперпользователя sudo [[обратная ссылка](#)].

```
sapsan@UbuntuBookPro:~$
sapsan@UbuntuBookPro:~$ roscd
sapsan@UbuntuBookPro:~/catkin_ws/devel$ rosnodetool list
ERROR: Unable to communicate with master!
sapsan@UbuntuBookPro:~/catkin_ws/devel$ roscore
... logging to /home/sapsan/.ros/log/64841d90-736a-11ee-a86b-4930ecb75a95/roslaunch-UbuntuBookPro-10963.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://UbuntuBookPro:43341/
ros_comm version 1.16.0

SUMMARY
=====
PARAMETERS
* /roscd: noetic
* /rosversion: 1.16.0

NODES
auto-starting new master
process[master]: started with pid [10972]
ROS_MASTER_URI=http://UbuntuBookPro:11311/

setting /run_id to 64841d90-736a-11ee-a86b-4930ecb75a95
process[rosout-1]: started with pid [10982]
started core service [/rosout]
```

Иллюстрация 14. Использование внешних консольных команд `roscd` и `roscore` [\[обратная ссылка\]](#).

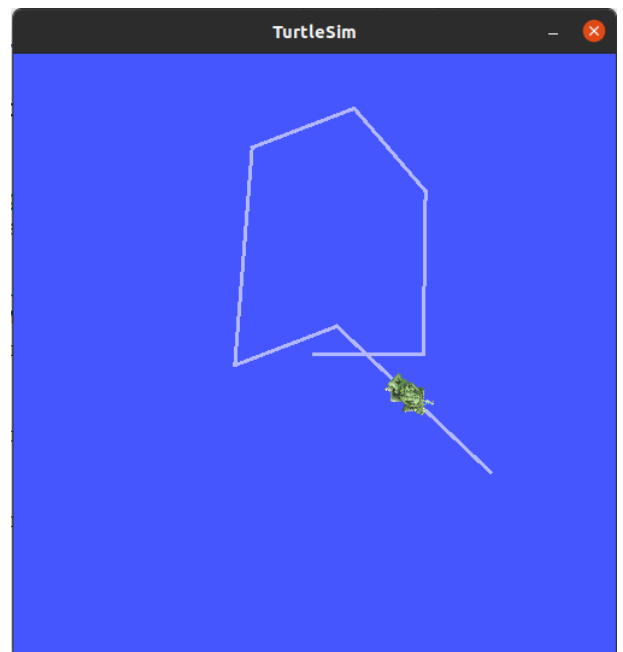
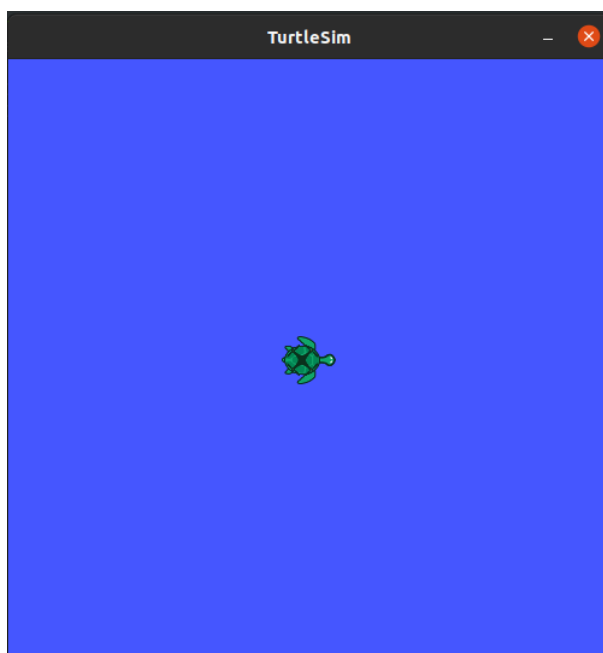
```
sapsan@UbuntuBookPro:~$ rosnodetool list
/roscd
sapsan@UbuntuBookPro:~$ rosnodetool info /roscd
-----
Node [/roscd]
Publications:
* /roscd_agg [roscd_msgs/Log]

Subscriptions:
* /roscd [unknown type]

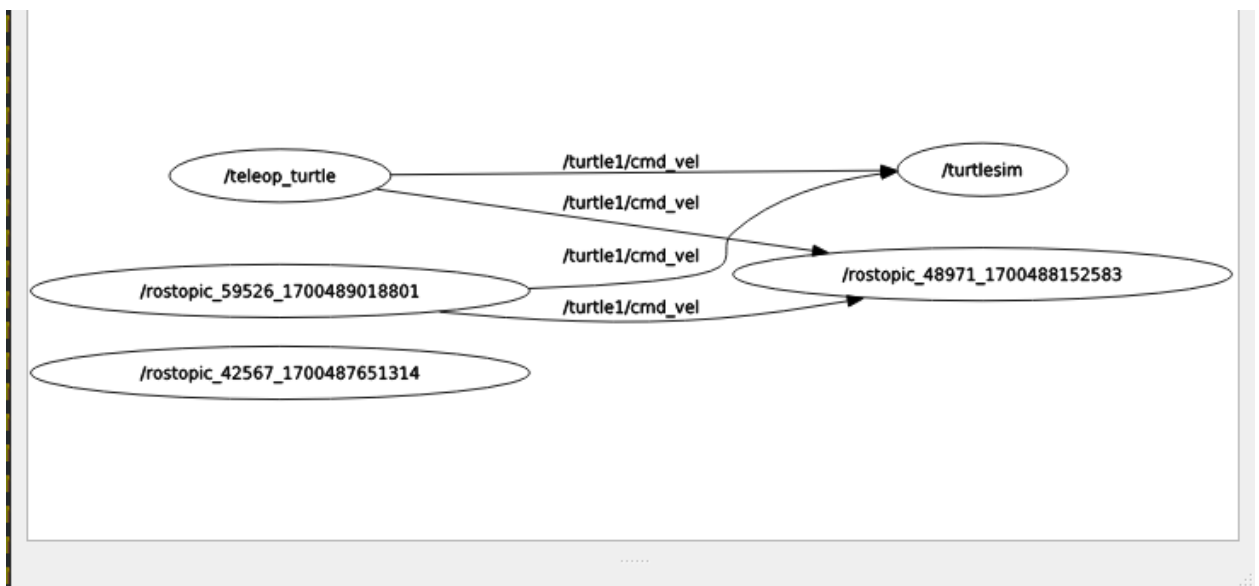
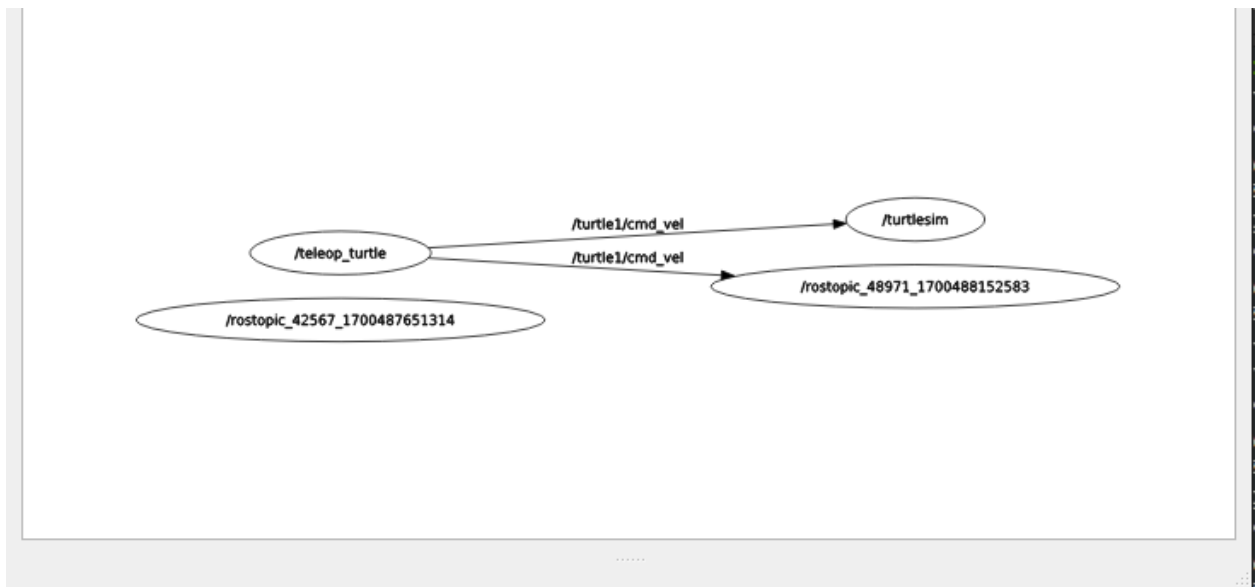
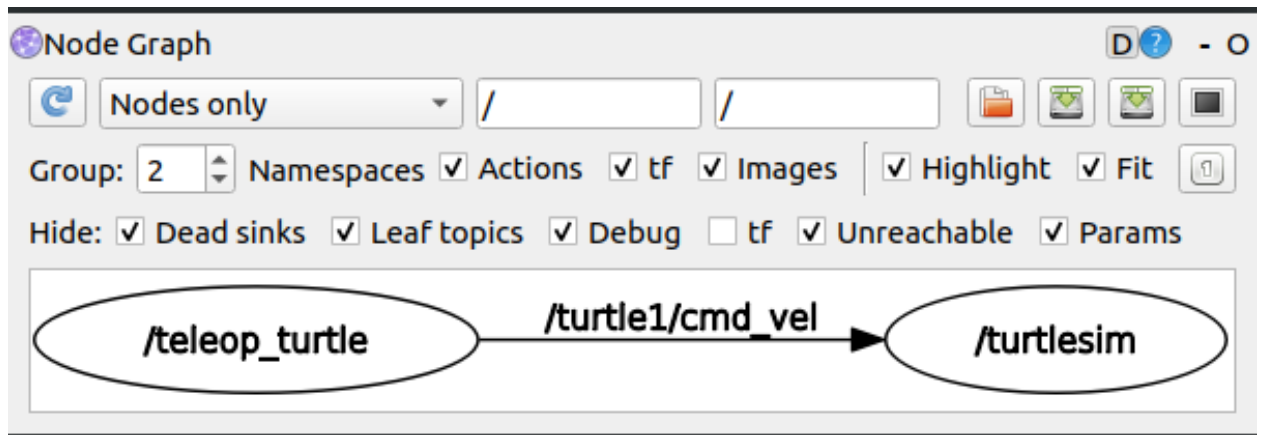
Services:
* /roscd/get_loggers
* /roscd/set_logger_level

contacting node http://UbuntuBookPro:46369/ ...
Pid: 10982
sapsan@UbuntuBookPro:~$
```

Иллюстрация 15. Использование внешней консольной команды `roscd` [\[обратная ссылка\]](#).



Иллюстрации 16, 17. Выполнение примера с `turtlesim_node` [\[обратная ссылка\]](#).



Иллюстрации 18, 19, 20. Использование rqt\_graph [\[обратная ссылка\]](#).

```
sapsan@UbuntuBookPro: ~  
sapsan@UbuntuBookPro:~$ rostopic echo /turtle1/cmd_vel  
linear:  
  x: 2.0  
  y: 0.0  
  z: 0.0  
angular:  
  x: 0.0  
  y: 0.0  
  z: 0.0  
---  
linear:  
  x: 0.0  
  y: 0.0  
  z: 0.0  
angular:  
  x: 0.0  
  y: 0.0  
  z: 2.0  
---  
linear:  
  x: 2.0  
  y: 0.0  
  z: 0.0  
angular:
```

Иллюстрации 21. Использование rostopic echo [[обратная ссылка](#)].

```
sapsan@UbuntuBookPro: ~  
sapsan@UbuntuBookPro:~$ rostopic list -v  
Published topics:  
* /rosout_agg [rosgraph_msgs/Log] 1 publisher  
* /rosout [rosgraph_msgs/Log] 6 publishers  
* /turtle1/pose [turtlesim/Pose] 1 publisher  
* /turtle1/color_sensor [turtlesim/Color] 1 publisher  
* /turtle1/cmd_vel [geometry_msgs/Twist] 1 publisher  
  
Subscribed topics:  
* /rosout [rosgraph_msgs/Log] 1 subscriber  
* /turtle1/cmd_vel [geometry_msgs/Twist] 3 subscribers  
* /statistics [rosgraph_msgs/TopicStatistics] 1 subscriber  
  
sapsan@UbuntuBookPro:~$ rostopic type /turtle1/cmd_vel  
geometry_msgs/Twist  
sapsan@UbuntuBookPro:~$ rosmmsg show geometry_msgs/Twist  
geometry_msgs/Vector3 linear  
  float64 x  
  float64 y  
  float64 z  
geometry_msgs/Vector3 angular  
  float64 x  
  float64 y  
  float64 z  
  
sapsan@UbuntuBookPro:~$
```

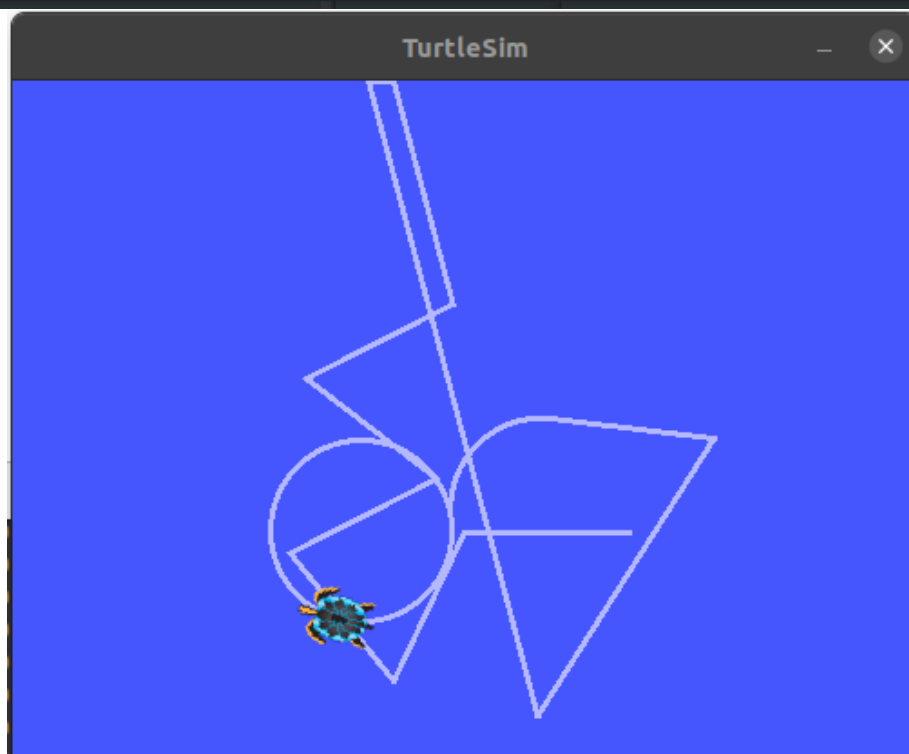
Иллюстрации 22. Использование rostopic list и rostopic type [[обратная ссылка](#)].



```
sapsan@UbuntuBookPro: ~  
sapsan@UbuntuBookPro:~$ rostopic type /turtle1/cmd_vel | rosmmsg show  
geometry_msgs/Vector3 linear  
  float64 x  
  float64 y  
  float64 z  
geometry_msgs/Vector3 angular  
  float64 x  
  float64 y  
  float64 z  
sapsan@UbuntuBookPro:~$
```

Иллюстрации 23. Использование rostopic type [\[обратная ссылка\]](#).

```
sapsan@UbuntuBookPro:~$ rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist -- '[2.0,  
0.0, 0.0]' '[0.0, 0.0, 1.8]'  
publishing and latching message for 3.0 seconds  
sapsan@UbuntuBookPro:~$
```



Иллюстрации 24, 25. Использование rostopic pub [\[обратная ссылка\]](#).

```
sapsan@UbuntuBookPro: ~  
sapsan@UbuntuBookPro:~$ rostopic hz /turtle1/pose  
subscribed to [/turtle1/pose]  
average rate: 62.460  
  min: 0.015s max: 0.017s std dev: 0.00043s window: 63  
average rate: 62.471  
  min: 0.015s max: 0.017s std dev: 0.00042s window: 125  
average rate: 62.492  
  min: 0.015s max: 0.017s std dev: 0.00043s window: 188  
average rate: 62.498  
  min: 0.015s max: 0.017s std dev: 0.00045s window: 250  
average rate: 62.489  
  min: 0.015s max: 0.017s std dev: 0.00045s window: 313  
average rate: 62.497  
  min: 0.015s max: 0.017s std dev: 0.00045s window: 375  
average rate: 62.488  
  min: 0.015s max: 0.017s std dev: 0.00045s window: 438  
average rate: 62.493  
  min: 0.015s max: 0.017s std dev: 0.00045s window: 501  
average rate: 62.493  
  min: 0.015s max: 0.017s std dev: 0.00045s window: 563  
average rate: 62.495
```

Иллюстрация 26. Использование rostopic hz [\[обратная ссылка\]](#).

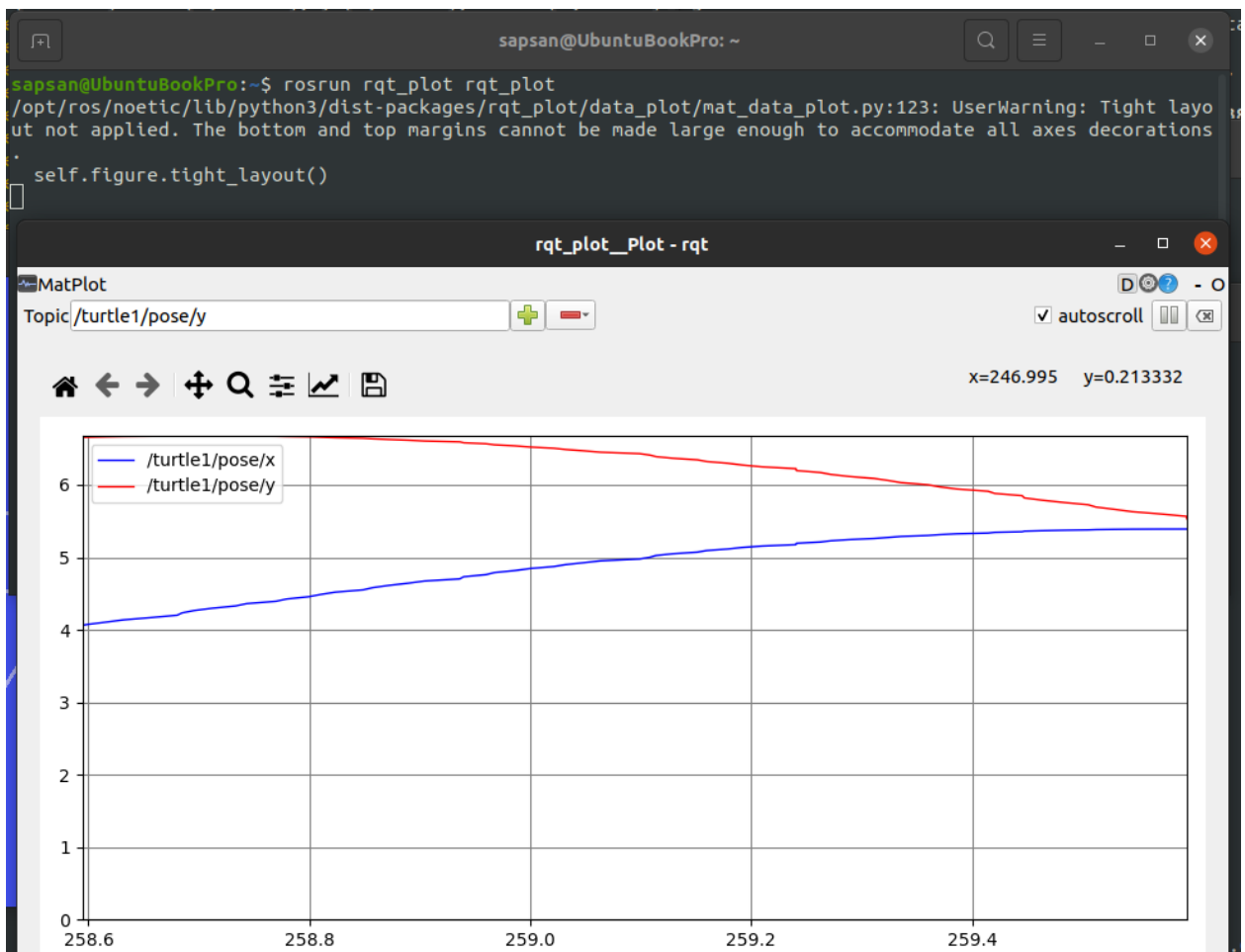


Иллюстрация 27. Использование rqt\_plot [\[обратная ссылка\]](#).



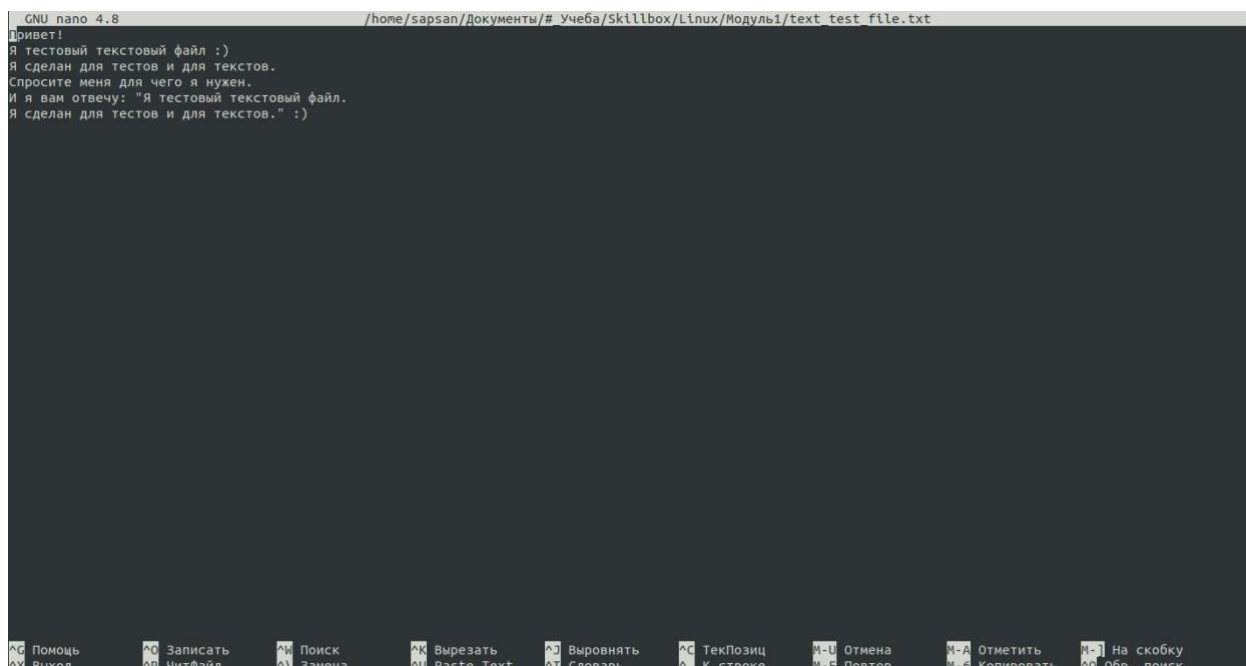


Иллюстрация 28. Открытие файла и навигация по нему в консольном текстовом редакторе Nano [\[обратная ссылка\]](#).

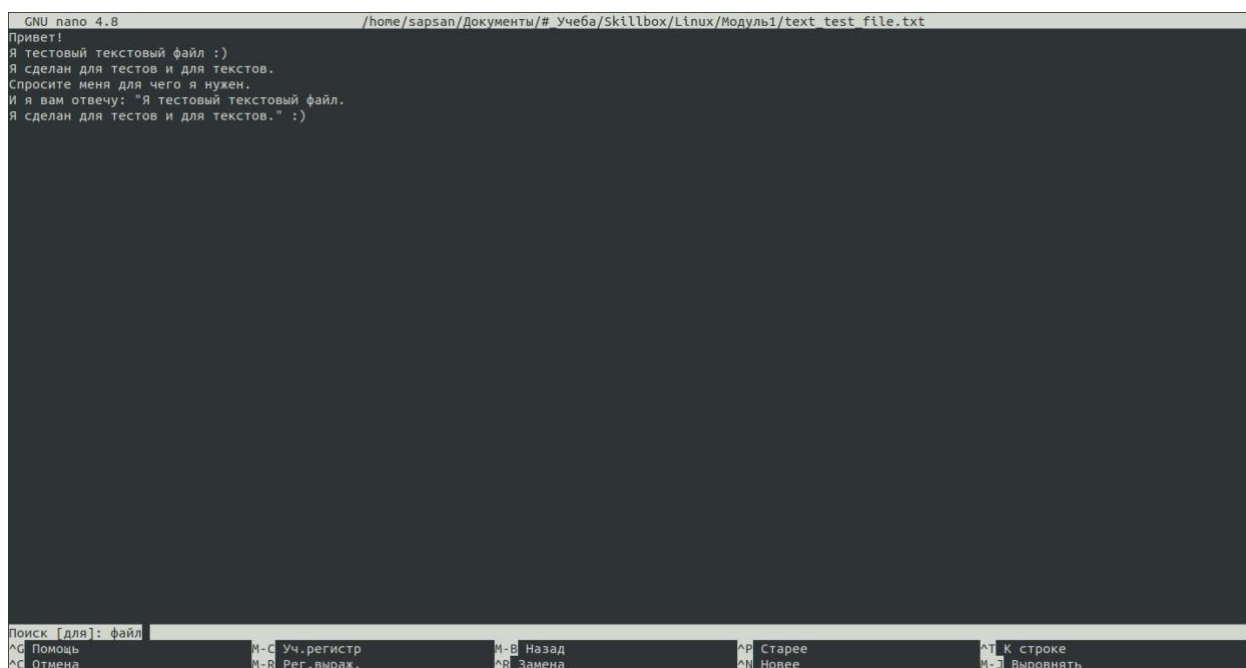
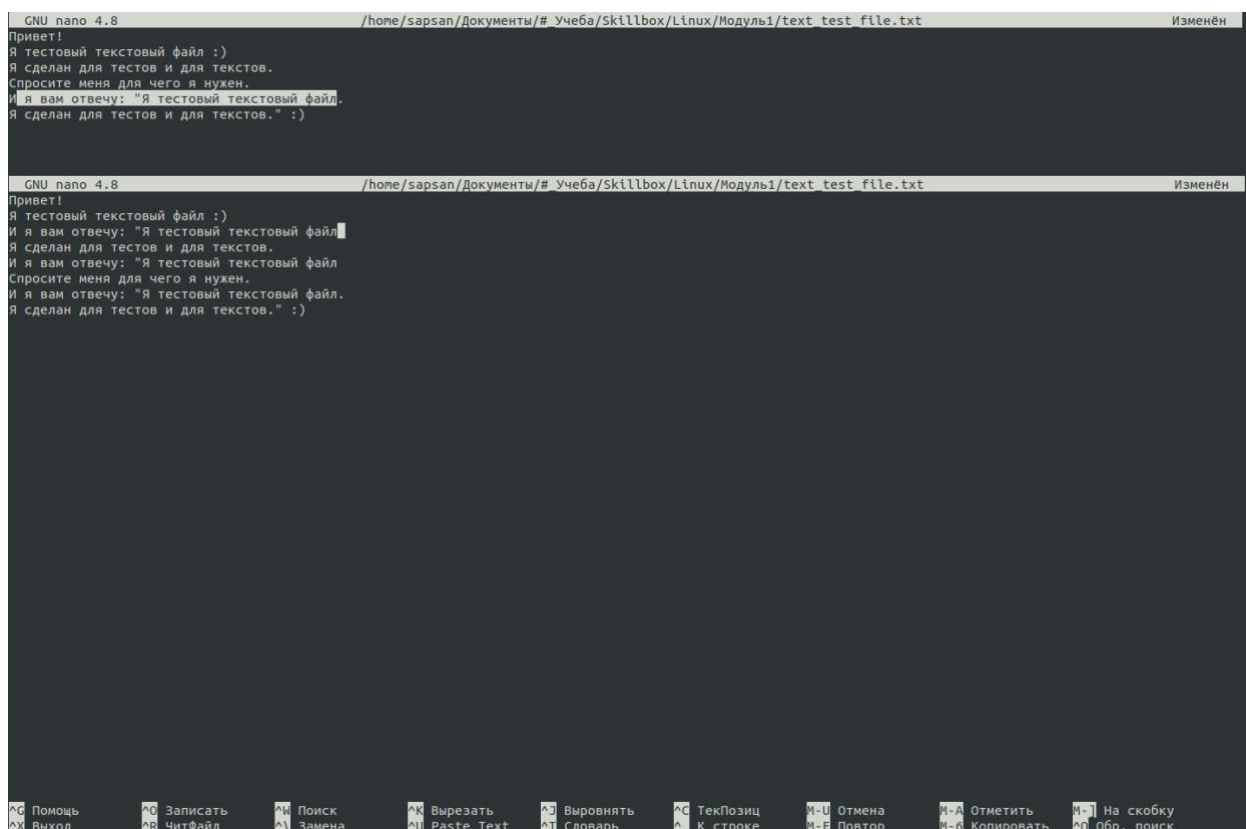


Иллюстрация 29. Поиск текста в консольном текстовом редакторе Nano [\[обратная ссылка\]](#).



Иллюстрации 30, 31. Редактирование, копирование и вставка текста в консольном текстовом редакторе Nano (в верхнем окне первоначальное состояние текста, в нижнем – результирующее) [[обратная ссылка](#)].

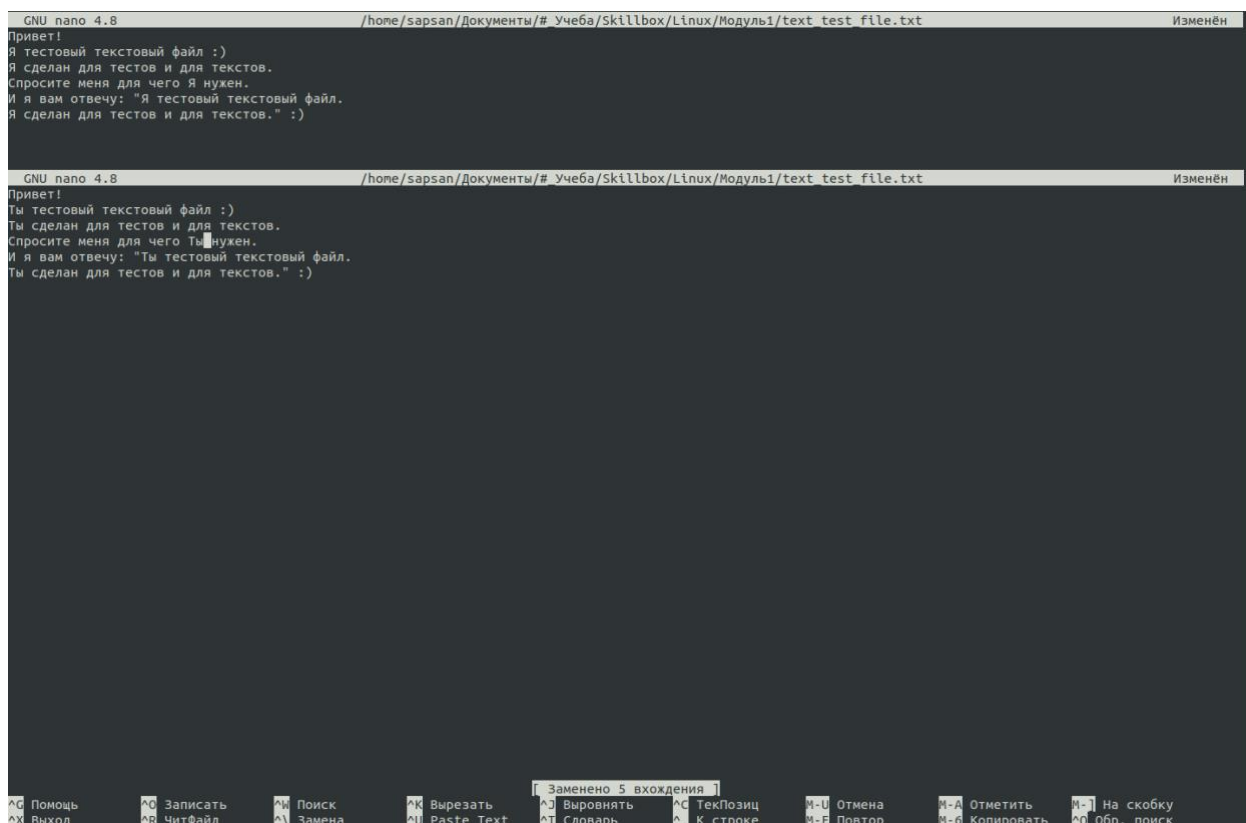


Иллюстрация 32, 33. Замена текста с учётом регистра в консольном текстовом редакторе Nano (в верхнем окне первоначальное состояние текста, в нижнем – результирующее) [[обратная ссылка](#)].

```
GNU nano 4.8 /home/sapsan/документы/# Учеба/Skillbox/Linux/Модуль1/text_test file.txt Изменён
Привет!
Ты тестовый текстовый файл :)
Ты сделан для тестов и для текстов.
Спросите меня для чего Ты нужен.
И я вам отвечу: "Ты тестовый текстовый файл.
Ты сделан для тестов и для текстов." :)

Имя файла для записи: /home/sapsan/документы/# Учеба/Skillbox/Linux/Модуль1/text_test file.txt
^C Помощь ^H-D Формат DOS ^M-A Доп. в начало ^M-B Резерв. копия
^O Отмена ^H-M Формат Mac ^M-E Доп. в конец ^M-T К файлам
```

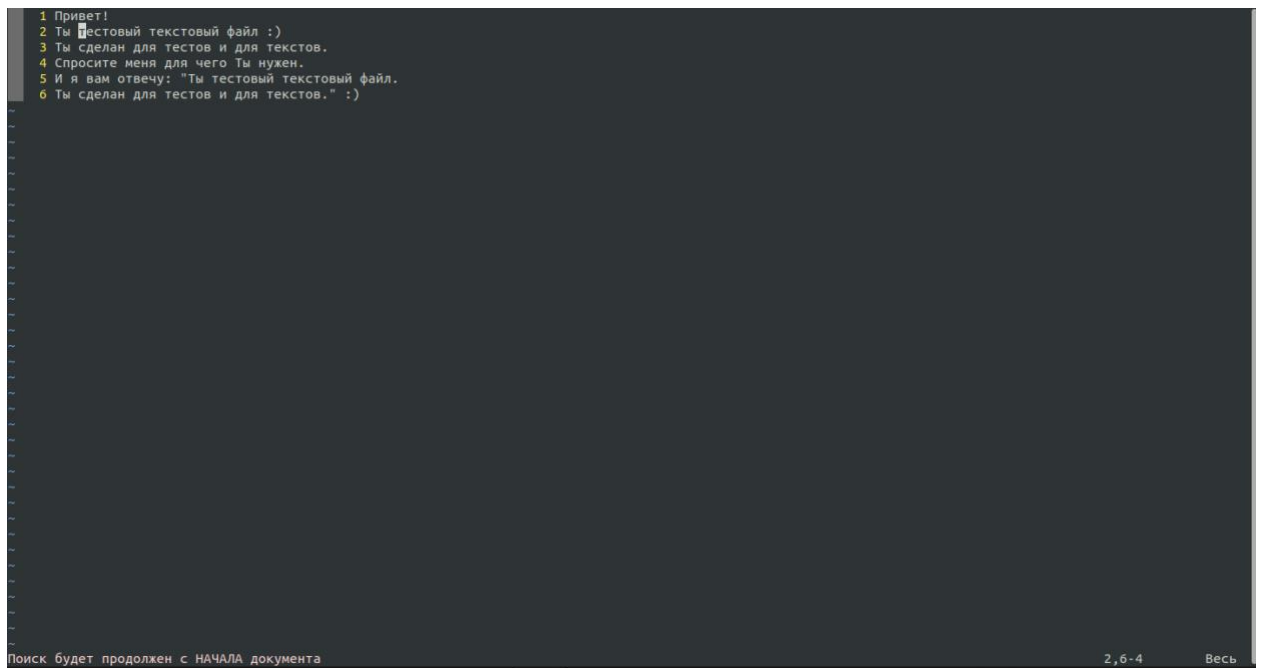
Иллюстрация 34. Сохранение изменений и выход из консольного текстового редактора Nano [[обратная ссылка](#)].

```
1 nmap <C-t> :NERDTreeToggle<CR> "запускает дерево каталогов через Ctrl+t
2 set number "добавляет строки
3 "добавляет отступы и включает замену табов на пробелы
4 set smarttab "при нажатии таба в начале строки добавляет количество
5 "пробелов равное shiftwidth
6 set tabstop=4 "количество пробелов в одном обычном табе
7 set softtabstop=4 "количество пробелов в табе при удалении
8 set shiftwidth=4
9 set foldcolumn=2 "отступ между левой частью окна
10 syntax on "подсветка синтаксиса языков программирования
11 set mouse=a "включить мышь во всех режимах
12 "горячие клавиши
13 inoremap <C-v> <ESC>"+ra
14 vnoremap <C-c> "+y
15 vnoremap <C-d> "+d
```

Иллюстрация 35. Текстовое содержимое локального конфигурационного файла Vim с комментариями [[обратная ссылка](#)].

```
" Press ? for help
.. (up a dir)
/home/sapsan/
├─ apt/
├─ catkin_ws/
├─ install/
├─ msoffice/
├─ playonlinux/
├─ PlayonLinux's virtual drives/
├─ snap/
├─ wine-8.0.2/
├─ Видео/
├─ Документы/
├─ # Учеба/
│   ├─ Python/
│   └─ Skillbox/
│       └─ Linux/
│           └─ Книжки/
│               └─ Модуль1/
│                   1.1.робототехников.
│                   1.2.Введение в Linux
│                   1.3.Демо установка
│                   1.3.Установка и нас
│                   1.4.Демо консоль Li
│                   1.4.Консоль Linux и
│                   1.5.Демо установка
│                   1.5.Установка прогр
│                   1.6.Демо консольные
│                   1.6.Консольные текс
│                   text_test file.txt
├─ Программирование мобиль
├─ Заключение.docx
├─ Иллюстрации.docx
├─ PPLfR.docx
├─ Загрузки/
├─ Изображения/
├─ Музыка/
├─ Общедоступные/
├─ Рабочий стол/
/home/sapsan
```

Иллюстрация 36. Открытие файла и навигация по нему в консольном текстовом редакторе Vim [[обратная ссылка](#)].

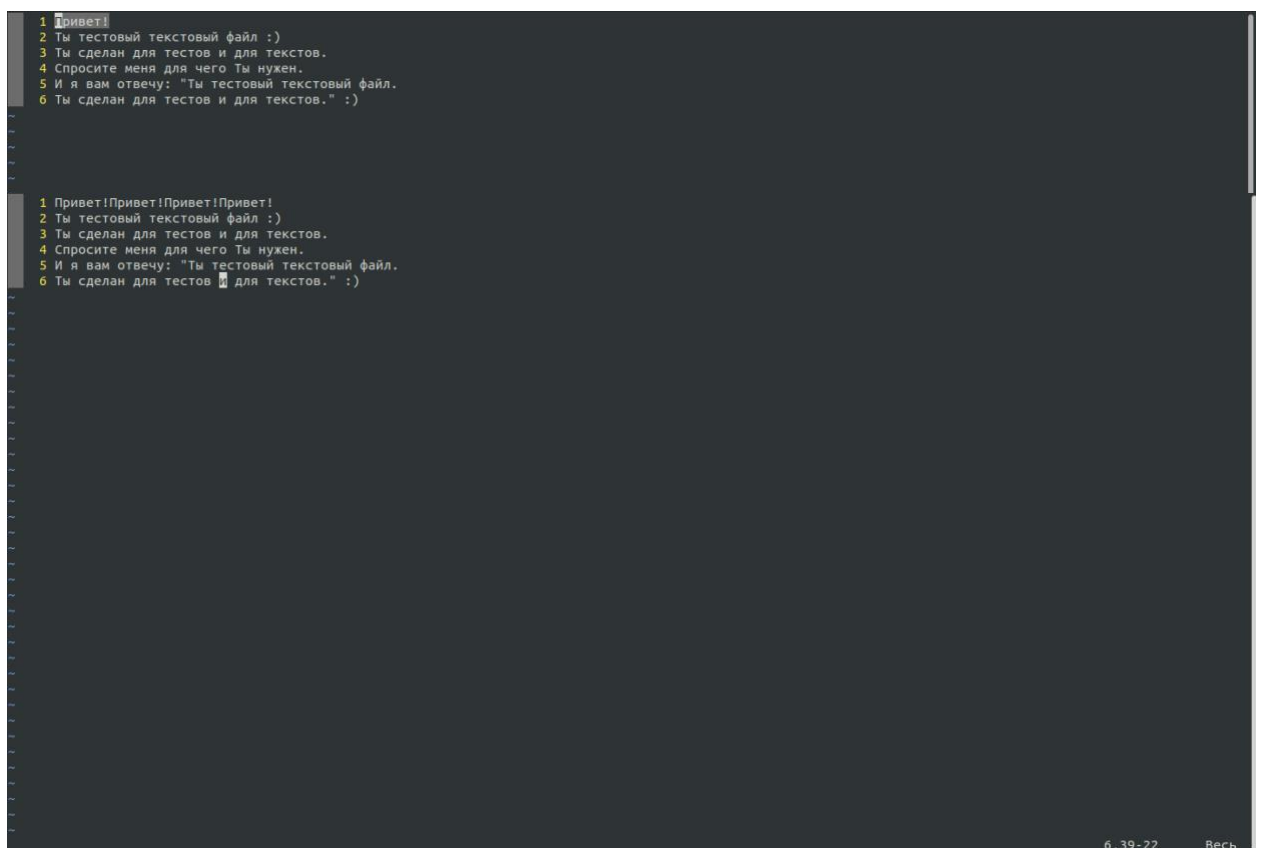


```
1 Привет!  
2 Ты тестовый текстовый файл :)  
3 Ты сделан для тестов и для текстов.  
4 Спросите меня для чего Ты нужен.  
5 И я вам отвечу: "Ты тестовый текстовый файл.  
6 Ты сделан для тестов и для текстов." :)
```

Поиск будет продолжен с НАЧАЛА документа

2,6-4    Весь

Иллюстрация 37. Поиск текста в консольном текстовом редакторе Vim [[обратная ссылка](#)].



```
1 Привет!  
2 Ты тестовый текстовый файл :)  
3 Ты сделан для тестов и для текстов.  
4 Спросите меня для чего Ты нужен.  
5 И я вам отвечу: "Ты тестовый текстовый файл.  
6 Ты сделан для тестов и для текстов." :)
```

1 Привет!Привет!Привет!Привет!  
2 Ты тестовый текстовый файл :)  
3 Ты сделан для тестов и для текстов.  
4 Спросите меня для чего Ты нужен.  
5 И я вам отвечу: "Ты тестовый текстовый файл.  
6 Ты сделан для тестов и для текстов." :)

6,39-22    Весь

Иллюстрации 38, 39. Редактирование, копирование и вставка текста в консольном текстовом редакторе Vim (в верхнем окне первоначальное состояние текста, в нижнем – результирующее) [[обратная ссылка](#)].

```
1 Привет!
2 Ты тестовый текстовый файл :)
3 Ты сделан для тестов и для текстов.
4 Спросите меня для чего Ты нужен.
5 И я вам отвечу: "Ты тестовый текстовый файл.
6 Ты сделан для тестов и для текстов." :)

:xs/Ты/Я/с
```

Иллюстрация 40. Замена текста с учётом регистра в консольном текстовом редакторе Vim [\[обратная ссылка\]](#).

```
1 Привет!Привет!
2 Ты тестовый текстовый файл :)
3 Ты сделан для тестов и для текстов.
4 Спросите меня для чего Ты нужен.
5 И я вам отвечу: "Ты тестовый текстовый файл.
6 Ты сделан для тестов и для текстов." :)
7 Привет!Привет!

:wq
```

Иллюстрация 41. Сохранение изменений и выход из консольного текстового редактора Vim [\[обратная ссылка\]](#).