Министерство науки и высшего образования Российской Федерации

Калужский филиал

федерального государственного бюджетного образовательного

учреждения высшего образования

**«Московский государственный технический университет**

**имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

(КФ МГТУ им. Н.Э. Баумана)

Создание FLASK-приложения под управлением OS Ubuntu Linux

Методические указания к выполнению лабораторной работы

по курсу «Операционные системы»

Калуга – 2019

УДК 004.62

ББК 32.972.1

Б435

Методические указания составлены в соответствии с учебным планом КФ МГТУ им. Н.Э. Баумана по направления подготовки 09.03.01 «Информационные системы и сети» кафедра «Программного обеспечения ЭВМ, информационных технологий»

Методические указания рассмотрены и одобрены:

- Кафедрой «Программного обеспечения ЭВМ, информационных технологий» (ИУ4-КФ) протокол №\_\_ от «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

Зав. Кафедрой ИУ4-КФ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., доцент Ю.Е. Гагарин

- Методической комиссией факультета ИУ-КФ протокол №\_\_ от «\_\_» \_\_\_\_\_\_\_\_ 20\_\_ г.

Председатель методической

Комиссии факультета ИУ-КФ \_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., доцент М.Ю. Адкин

-Методической комиссией

КФ МГТУ им Н.Э. Баумана протокол №\_\_ от «\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

Председатель методической комиссии

КФ МГТУ им Н.Э. Баумана \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ д.э.н. профессор О.Л.Перерва

Рецензент:

к.т.н. доцент кафедры ИУ3-КФ \_\_\_\_\_\_\_\_\_\_\_\_ А.В. Финошин

Авторы

Аннотация

Методические указания к выполнению лабораторной работы по курсу «Операционные системы» содержат общие сведения о системных каталогах в ОС семейства Linux, а также синтаксис основных команд для монтирования устройств, архивации и сжатия файлов.

Предназначены для студентов 3-ого курса бакалавриата КФ МГТУ им Н.Э. Баумана, обучающихся по направлению подготовка 09.03.01 «Информационные системы и сети»

Калужский филиал МГТУ им Н.Э. Баумана, 2019 г.

Оглавление

ВВЕДЕНИЕ 5

ЦЕЛЬ И ЗАДАЧИ РАБОТА, ТРЕБОВАНИЯ 7

К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ 7

КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ 8

FLASK 10

УСТАНОВКА И НАСТРОЙКА PYTHON 11

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ 20

КОНТРОЛЬНЫЕ ВОПРОСЫ 21

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ 22

ОСНОВНАЯ ЛИТЕРАТУРА 23

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА 23

# ВВЕДЕНИЕ

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Программное обеспечение ЭВМ, информационные технологии» факультета «Информатика и Управление»на кафедре «Программное обеспечение ЭВМ, информационные технологии» Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 3-го курса направления подготовки 09.03.01 «Информационные системы и сети», содержат краткое описание команд для установки и настройки сетевого интерфейса с сетью в ОС Linux Ubutu.

Методические указания составлены для ознакомления студентов с работой с сетевыми интерфейсами, файлами конфигурации, командами проверки сетевого интерфейса.

Для выполнения лабораторных работ студенту необходимы минимальные навыки программирования и знания в ОС Linux Ubutu

Требования к программному обеспечению:

* Microsoft Windows 7/8/10
* Программный продукт виртуализации для операционной системы Windows (Virtual Box).

# ЦЕЛЬ И ЗАДАЧИ РАБОТА, ТРЕБОВАНИЯ

# К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является приобретение практических навыков по работе с устройствами и файловой системой ОС Linux.

Основными задачами выполнения лабораторной работы являются:

1. Ознакомиться с устройством файловой системы ОС Linux.
2. Получить навыки работы с устройствами ОС Linux.
3. Изучить основные команды монтирования файловых систем ОС Linux.
4. Получить навыки работы с утилитами для архивации и сжатия файлов в ОС Linux.
5. Изучить основные опции команд для архивации и сжатия файлов в ОС Linux.

Результатами работы являются:

1. Демонстрация выполнения команд для работы с файловой системой, архивации и сжатия файлов.
2. Подготовленный отчет.

# КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ

**Что такое Python**

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

Python — активно развивающийся язык программирования, новые версии с добавлением/изменением языковых свойств выходят примерно раз в два с половиной года. Язык не подвергался официальной стандартизации, роль стандарта выполняет CPython, разрабатываемый под контролем автора языка.

У Python много готовых библиотек для решения задач

Библиотеками в программировании называют инструменты для решения конкретных типов задач. Вот несколько примеров популярных библиотек для Python:

* Pygame. Библиотека для создания небольших игр и мультимедийных приложений.
* NumPy. Библиотека для работы с искусственным интеллектом и машинным обучением. Используется для сложных математических вычислений.
* Pandas. Библиотека для работы с большими данными.
* SQLAlchemy. Библиотека для работы с базами данных.
* Django, Flask. Библиотеки для разработки серверной части приложений.

Наличие библиотек значит, что под каждую задачу есть свой инструмент.

Универсальный. Python не предназначен для чего-то конкретного, это язык общего назначения: подходит для web-проектов, для desktop- и серверных приложений.

* Мультипарадигменный. Питон поддерживает любой стиль программирования.
* Скриптовый. До 1960-х годов было только две группы языков: системные и командные. На системных языках писали программисты, а на командных – системные администраторы. Со временем появились скриптовые языки – нечто среднее между системными и командными языками.

Одним из главных недостатков является его относительно низкая скорость выполнения. Python является языком с полной динамической типизацией, автоматическим управлением памятью. Если на первый взгляд это может казаться преимуществом, то при разработке программ с повышенным требованием к эффективности, Python может значительно проигрывать по скорости своим статическим братьям (C/C++, Java, Go). Что касается динамических собратьев (PHP, Ruby, JavaScript), то здесь дела обстоят намного лучше, Python в большинстве случаев выполняет код быстрее за счет предварительной компиляции в байт-код и значительной части стандартной библиотеки, написанной на Си.

# FLASK

Flask — фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2. Относится к категории так называемых микрофреймворков — минималистичных каркасов веб-приложений, сознательно предоставляющих лишь самые базовые возможности.

Flask является микрофреймворком Python, который имеет модульный дизайн. Данный фреймворк предназначен для создания веб-приложений. У Flask нет определенной системы базы данных или системы ORM. Если вы хотите использовать базу данных, то вам потребуется отдельный модуль. Flask часто комбинируют с SQLAlchemy для использования баз данных.

Платформа поддерживает маршрутизацию URL-адресов, шаблоны (с Jinja2), управление сеансами и имеет некоторые фишки в области безопасности.

Особенности:

* Маршрутизация URL-адресов и их представления
* Механизм шаблонов
* Управление сеансами

# 

# УСТАНОВКА И НАСТРОЙКА PYTHON

Если ваша операционная система не имеет предустановленный пакет Python, вы можете скачать программу установки с официального сайта Python. Если вы используете Microsoft Windows вместе с WSL или Cygwin, обратите внимание, что вы не будете использовать родную версию Python для Windows, а версию, совместимую с Unix, которую вам нужно получить от Ubuntu (если вы используете WSL) или от Cygwin (Рисунок 1).

Чтобы убедиться, что ваша установка Python является функциональной, вы можете открыть окно терминала и ввести python3, или если это не работает, просто python. Вот что вам следует ожидать:

$ python3

Python 3.5.2 (default, Nov 17 2016, 17:05:23)

[GCC 5.4.0 20160609] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> \_

или так в cmd (окно командной строки Microsoft Windows) :

c:\cp>c:\python33\python

Python 3.3.5 (v3.3.5:62cf4e77f785, Mar 9 2014, 10:37:12) [MSC v.1600 32 bit (In

tel)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>>

Интерпретатор Python теперь находится в ожидании пользовательского ввода. Но пока Вы подтвердили, что Python установлен в вашей системе. Чтобы выйти из интерактивного приглашения, вы можете ввести exit() и нажать Enter.

В версиях Python для Linux и Mac OS X вы также можете выйти из интерпретатора, нажав Ctrl-D.

В Windows, ярлык выхода — Ctrl-Z, затем Enter.

Рисунок 1 - Установка Python



В Python пакеты, такие как Flask, доступны в общем репозитории, откуда их можно загрузить и установить. Официальный репозиторий пакетов Python называется PyPI, что означает Python Package Index (также известный, как «cheese shop»). Установка пакета из PyPI очень проста, потому что у Python есть инструмент под названием pip, который выполняет эту работу (в Python 2.7 pip не входит в комплект с Python и его нужно устанавливать отдельно).

Чтобы установить пакет на свой компьютер, вы используете pip следующим образом (Рисунок 2).:

$ pip install <package-name>

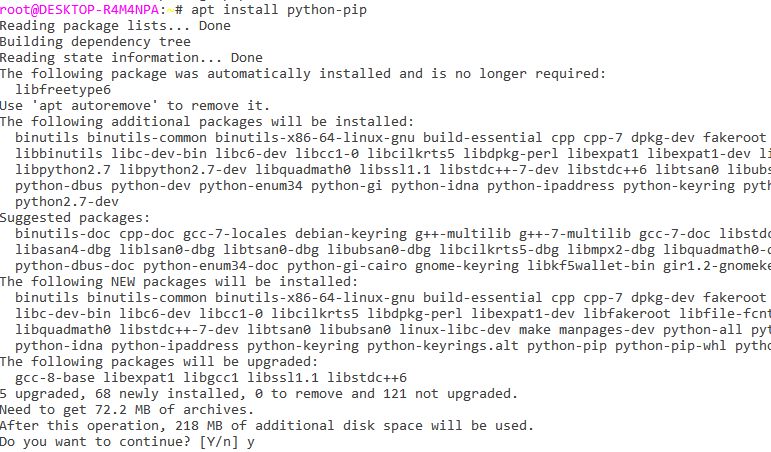


Рисунок 2 - Установка Pip для Python

Интересно, что этот метод установки пакетов не будет работать в большинстве случаев. Если ваш интерпретатор Python был установлен глобально для всех пользователей вашего компьютера, велика вероятность того, что ваша обычная учетная запись пользователя не получит разрешения на внесение в нее изменений, поэтому единственный способ выполнить вышеприведенную команду — запустить ее от имени администратора. Инструмент pip загрузит пакет из PyPI, а затем добавит его в вашу папку Python. С этого момента каждый скрипт Python, который у вас есть в вашей системе, будет иметь доступ к этому пакету.

Начнем с создания рабочего каталога проекта (Рисунок 3).

****Рисунок 3 - Создание папки проекта

Если вы используете версию Python 3, в нее включена поддержка виртуальной среды, поэтому все, что вам нужно сделать для ее создания, это:

$ python3 -m venv venv

С помощью этой команды Python запустит пакет venv, который создает виртуальную среду с именем venv. Первым venv в команде является имя пакета виртуальной среды Python, а второе — имя виртуальной среды, которое будет использоваться для этой конкретной среды (Рисунок 4).



Рисунок 4 - Установка приложения для создания виртуальной среды

По завершении команды вы будете иметь каталог с именем venv, где хранятся файлы виртуальной среды.

Если вы используете любую версию Python старше 3.4 (включая выпуск 2.7), виртуальные среды не поддерживаются изначально. Для этих версий Python вам необходимо загрузить и установить сторонний инструмент virtualenv, прежде чем создавать виртуальные среды. После того, как virtualenv установлен, вы можете создать виртуальную среду со следующей командой:

$ virtualenv venv

Или $ python virtualenv.py venv

С помощью этой команды Python запустит пакет venv, который создает виртуальную среду с именем venv (Рисунок 5).

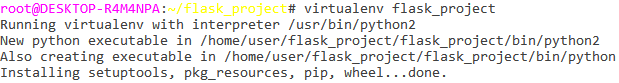


Рисунок 5 - Создание виртуальной среды

Независимо от метода, который вы использовали для его создания, вы создали свою виртуальную среду. Теперь вам нужно сообщить системе, что вы хотите ее использовать, активируя ее. Чтобы активировать новую виртуальную среду, используете следующую команду:

$ source venv/bin/activate

(venv) $ \_



Рисунок 6 - Вход в виртуальную среду

Когда вы активируете виртуальную среду, конфигурация сеанса терминала изменяется так, что интерпретатор Python, хранящийся внутри нее, станет тем, который вызывается при вводе python. Кроме того, в запросе терминала включено имя активированной виртуальной среды. Изменения, внесенные в сеанс терминала, являются временными и частными для этого сеанса, поэтому они не будут сохраняться при закрытии окна терминала. Если вы одновременно работаете с несколькими терминальными окнами, отлично видно, чтобы на каждом из них были задействованы разные виртуальные среды.

Теперь, когда создана и активирована виртуальная среда, установим в нее Flask :

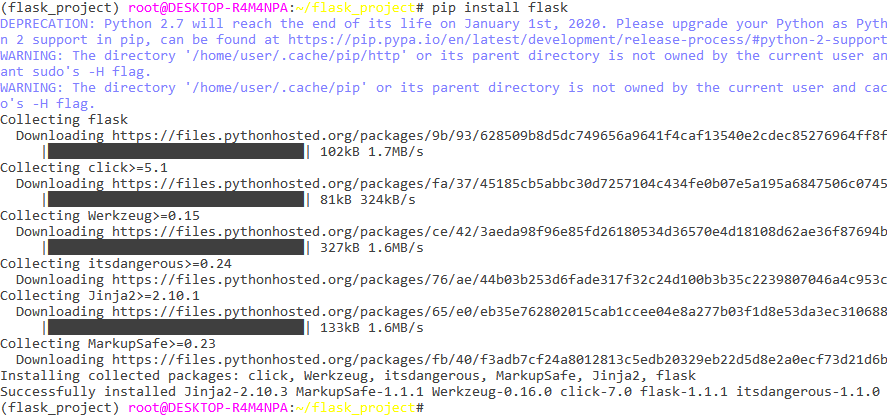


Рисунок 7 - Установка Flask

В Python подкаталог, содержащий файл \_\_init\_\_.py, считается пакетом и может быть импортирован. Когда вы импортируете пакет, \_\_init\_\_.py выполняет и определяет, какие символы предоставляют пакет для внешнего мира (Рисунок 8).

Рисунок 8 - Создание папки приложения и скрипта инициализации

Инициализация \_\_init\_\_.py для пакета приложения будет содержать следующий код (Рисунок 9):



Рисунок 9 - Скрипт инициализации

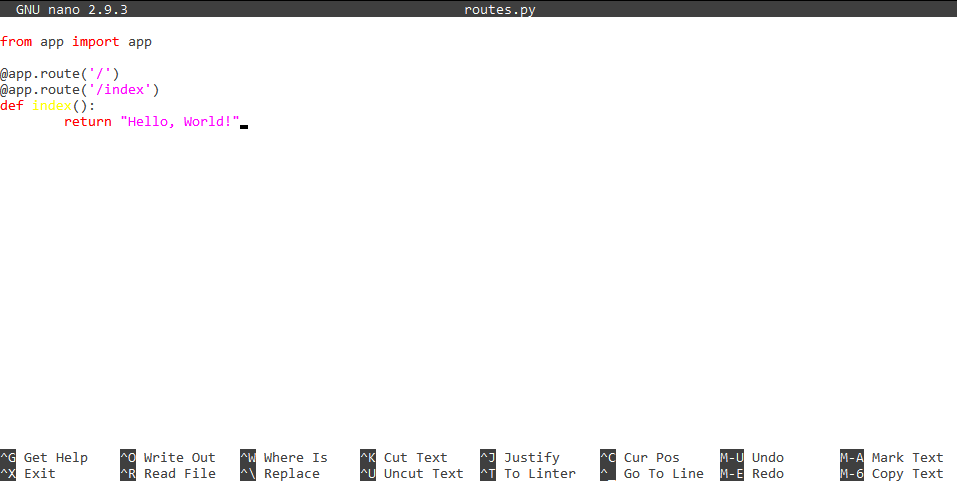
Первая функция просмотра, которую нужно написать в новом модуле с именем app/routes.py (Рисунки 10-11):

Рисунок 10 - Создание скрипта маршрутов

Рисунок 11 - Скрипт маршрутов

Создадим экземпляр приложение Flask (Рисунок 12).

Рисунок 12 - Создание экземпляра приложения Flask

Чтобы завершить приложение, вам нужно создать сценарий Python на верхнем уровне, определяющий экземпляр приложения Flask (Рисунок 13).

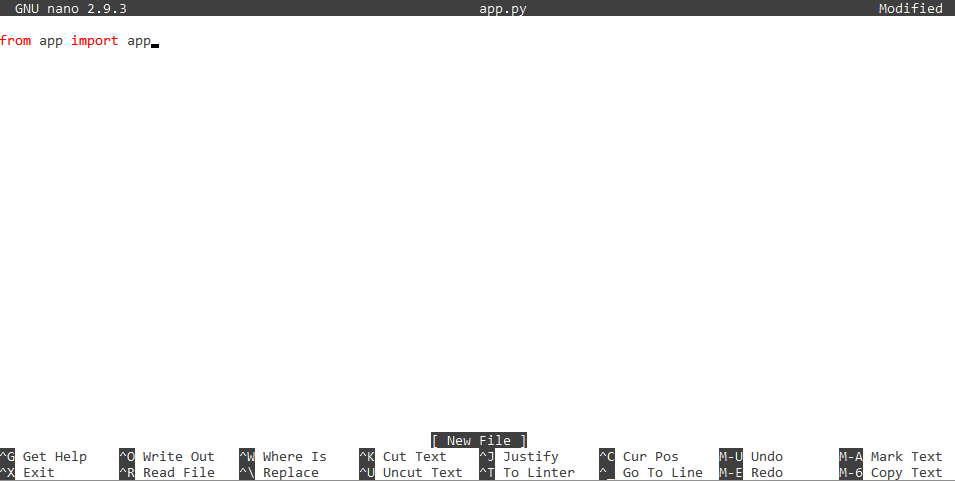


Рисунок 13 - Содержимое экземпляра приложения

Для продолжения работы нам потребуется создать переменную среду FLASK\_APP (Рисунок 14).

Рисунок 14 - Установка переменной среды FLASK\_APP

Команда для запуска готового проекта представлена на рисунке 15.

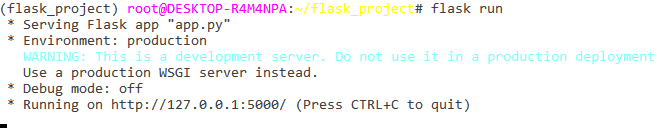


Рисунок 15 - Запуск проекта

Работа приложения демонстрируется на рисунке 16.

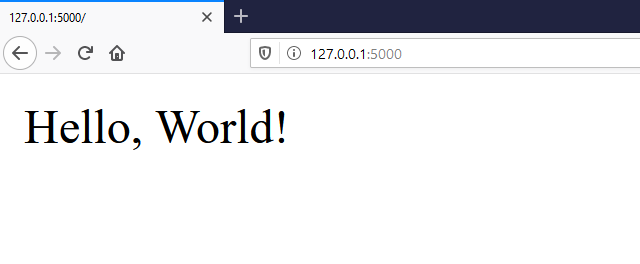


Рисунок 16 - Работа проекта

# ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Научиться использовать команды для установки пакетов, создания виртуальных сред в ОС Linux. Изучить особенности языка python и микрофреймворка Flask. Продемонстрировать выполнение команд:

* 1. Установки приложений через встроенную утилиту apt
  2. Установки дополнительных библиотек для языка Python
  3. Создания виртуальных сред
  4. Создания директорий
  5. Управления виртуальной средой
  6. Создания скриптов языка Python
  7. Установки переменной среды FLASK\_APP
  8. Запуска экземпляра приложения

# КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Опишите организацию Flask приложения в ОС Linux.
2. Раскройте понятие виртуальной среды Python.
3. Опишите минимальный синтаксис языка Python.
4. Перечислите каталоги и файлы, образующие структуру проекта.

# ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 2 занятия (4 академических часа: 3 часа на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания, ответы на контрольные вопросы, описание процесса выполнения лабораторной работы, выводы.

# ОСНОВНАЯ ЛИТЕРАТУРА

1. Гринберг, М. «Разработка веб-приложений с использованием Flask на языке Python» [Электронный ресурс] / М. Гринберг. — Электрон. дан. — Москва : ДМК Пресс, 2014. — 272 с. — Режим доступа: https://e.lanbook.com/book/90103

# ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

1. Крищенко, В.А. Сервисы Windows [Электронный ресурс] : учебное пособие / В.А. Крищенко, Н.Ю. Рязанова. — Электрон. дан. — Москва : МГТУ им. Н.Э. Баумана, 2011. — 47 с. — Режим доступа: https://e.lanbook.com/book/52416..
2. Войтов, Н.М. Администрирование ОС Red Hat Enterprise Linux. Учебный курс [Электронный ресурс] : учебное пособие / Н.М. Войтов.

— Электрон. дан. — Москва : ДМК Пресс, 2011. — 192 с. — Режим доступа: https://e.lanbook.com/book/1081

1. Стащук, П.В. Администрирование и безопасность рабочих станций под управлением Mandriva Linux: лабораторный практикум [Электронный ресурс] : учебно-методическое пособие / П.В. Стащук.

— Электрон. дан. — Москва : ФЛИНТА, 2015. — 182 с. — Режим доступа: <https://e.lanbook.com/book/70397>

**Электронные ресурсы:**

1. Научная электронная библиотека [http://eLIBRARY.RU](http://eLIBRARY.RU/)
2. Электронно-библиотечная система [http://e.lanbook.com](http://e.lanbook.com/)
3. The Flask Mega-Tutorial https://blog.miguelgrinberg.com/