

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ГЕОСИСТЕМ И ТЕХНОЛОГИЙ»  
(СГУГИТ)

Е. Ю. Воронкин

## **WEB-ПРОГРАММИРОВАНИЕ**

Утверждено редакционно-издательским советом университета  
в качестве практикума для обучающихся по направлению подготовки  
09.03.02 Информационные системы и технологии (уровень бакалавриата)

Новосибирск  
СГУГИТ  
2020

УДК 004  
В75

Рецензенты: директор ООО «И20» *О. А. Воинова*  
кандидат технических наук, доцент, доцент кафедры карто-  
графии и геоинформатики СГУГиТ *А. А. Колесников*

**Воронкин, Е. Ю.**

**В75** WEB-программирование [Текст] : практикум / Е. Ю. Воронкин. – Новоси-  
бирск : СГУГиТ, 2020. – 68 с.

ISBN

Практикум подготовлен старшим преподавателем Е. Ю. Воронкиным на ка-  
федре прикладной информатики и информационных систем СГУГиТ.

Издание включает лабораторные работы по основным разделам дисциплины  
«Web-программирование»: теоретические основы web-программирования, ос-  
новы html.css, CSS Frameworks, основы работы с CMS и CMF.

Практикум по дисциплине «Web-программирование» предназначен для обу-  
чающихся по направлению подготовки 09.03.02 Информационные системы и тех-  
нологии (уровень бакалавриата).

Рекомендован к изданию кафедрой прикладной информатики и информаци-  
онных систем, Ученым советом Института геодезии и менеджмента СГУГиТ.

Печатается по решению редакционно-издательского совета СГУГиТ

УДК 004

ISBN

© СГУГиТ, 2020

## ОГЛАВЛЕНИЕ

Введение .....	4
Лабораторная работа № 1. Установка операционной системы Ubuntu.....	5
Лабораторная работа № 2. Настройка LAMP.....	20
Лабораторная работа № 3. Распределенная система управления верси- ями.....	30
Лабораторная работа № 4. Установка системы управления содержи- мым .....	44
Лабораторная работа № 5. Модуль views .....	54
Лабораторная работа № 6. Модули slack, trello, addthis, dgis_maps, simple- news, loginza.....	60
Библиографический список рекомендуемой литературы .....	68

## **ВВЕДЕНИЕ**

Данное издание предназначено для обучающихся по направлению подготовки 09.03.02 Информационные системы и технологии (уровень бакалавриата) в качестве практикума по дисциплине «Web-программирование». Оно дополняет изучаемый теоретический материал и позволяет закрепить его на практике.

Практикум содержит описание лабораторных работ: цель, задачи, общие теоретические сведения, примеры выполнения, задания для самостоятельного выполнения, требования к оформлению отчета по выполненной работе.

В пособии приведен список рекомендуемой литературы, позволяющий обучающимся самостоятельно найти и изучить дополнительный материал.

После выполнения каждой работы обучающийся должен предоставить отчет о проделанной работе в письменном виде, в который входят титульный лист, цель работы, задание, решение, полученные результаты и выводы.

# Лабораторная работа № 1

## УСТАНОВКА ОПЕРАЦИОННОЙ СИСТЕМЫ UBUNTU

Время выполнения – 4 часа (аудиторная работа – 2 часа, самостоятельная работа – 2 часа).

**Цель работы:** освоение основных приемов работы с виртуальными машинами и операционной системой *Ubuntu* на ядре *Linux*.

### Задачи работы

1. Получить навыки работы с виртуальной машиной *VMware Workstation*.
2. Изучить основные приемы настройки программного продукта.
3. Научиться использовать виртуальную машину для работы в операционной системе *Ubuntu*.

### Перечень обеспечивающих средств

Задания лабораторной работы выполняются в операционной системе *Ubuntu*.

### Общие теоретические сведения

*VMware Workstation* – это программное обеспечение от компании *VMware*, являющейся лидером в области технологий виртуализации ИТ-инфраструктуры. Благодаря данному программному продукту появилась возможность устанавливать на виртуальном компьютере (виртуальной машине) практически любую операционную систему (гостевую ОС) и использовать ее параллельно с основной. Также на одной машине, персональном компьютере

можно создать несколько виртуальных машин с разными операционными системами и пользоваться ими одновременно, точно также, как это происходило бы на реальном оборудовании. Так, например, на реальном компьютере с операционной системой *Windows*, средства виртуализации *VMware Workstation* позволяют создать виртуальные машины (*VM – Virtual Machine*, *BM*), на которых можно установить операционные системы, например, семейства *Linux*, *Nowell Netware*, или другую *Windows* и пользоваться всеми ОС одновременно.

Все устанавливаемые виртуальные машины имеют свою конфигурацию:

- процессор;
- память;
- диски;
- свой BIOS;
- и т.д.

Каждая виртуальная машина работает в своей, изолированной от других, среде и может быть выключена, перезагружена или сброшена без какого-либо влияния на реальную, или прочие виртуальные, машины.

Любая работа с жестким диском, форматирование, изменение его загрузочных записей, удаление или добавление разделов не несет какое-либо воздействие на реальную или другую виртуальную машину.

Таким образом, виртуальная машина является идеальным средством для опасных экспериментов, отладки системного и прикладного ПО, освоения новых ОС и средств их администрирования, исследования вирусов и средств борьбы с ними. Одной из важных особенностей *VMware Workstation* является возможность создания снимков системы (*snapshot*), т. е. сохранение текущего состояния виртуальной машины. В любой момент времени пользователь может выполнить возврат (откат) состояния виртуальной машины на момент со-

здания снимка. Кроме того, можно настроить взаимодействие между реальной и виртуальными системами по локальной сети, обеспечить обмен данными через сменные носители, общие папки и т. п.

Процесс получения виртуального компьютера можно условно разбить на два этапа:

- создание конфигурации *VMware Workstation*;
- загрузка в виртуальную машину операционной системы с загрузочного носителя.

Рассмотрим порядок создания новой виртуальной машины.

1. Запускаем мастер создания новой виртуальной машины. Выбираем *Файл - Новая виртуальная машина* (рис. 1.1).

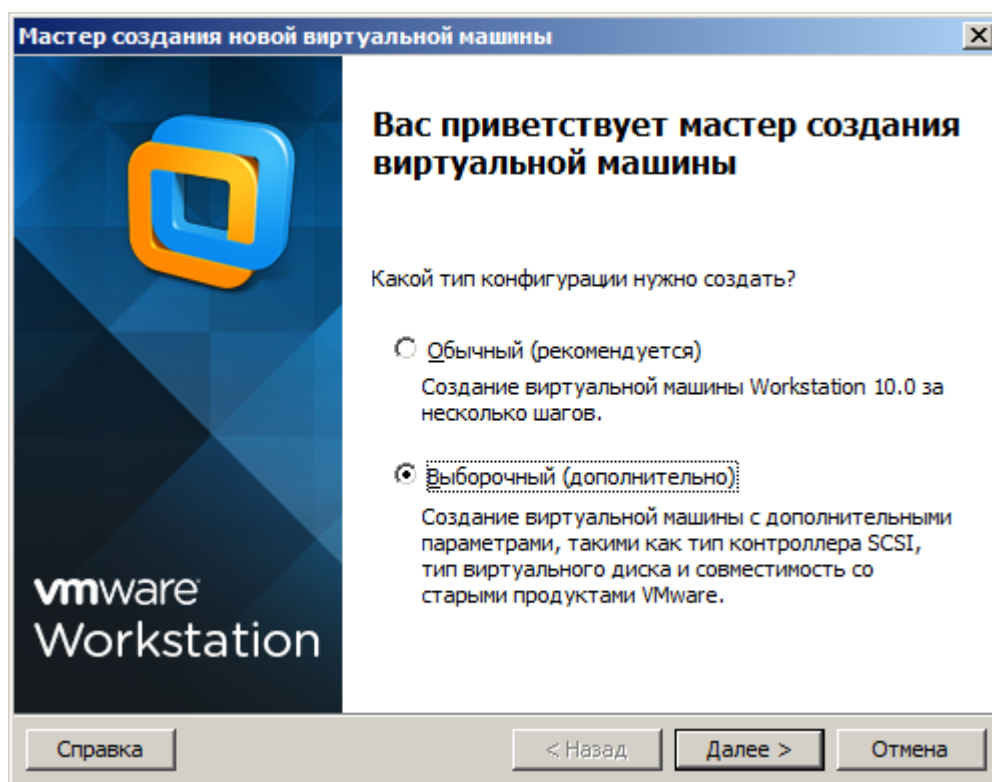


Рис. 1.1. Запуск мастера создания виртуальной машины

Имеется возможность выбора двух режимов работы мастера: *обычного* и *выборочного*, предназначенного для опытных пользователей.

Выборочный режим отличается возможностью настройки совместимости виртуального оборудования с конкретными программными продуктами виртуализации.

При определении параметров установки гостевой ОС нужно выбрать источник ее установки. В нашем случае источником будет носитель с дистрибутивом *Linux* (рис. 1.2).

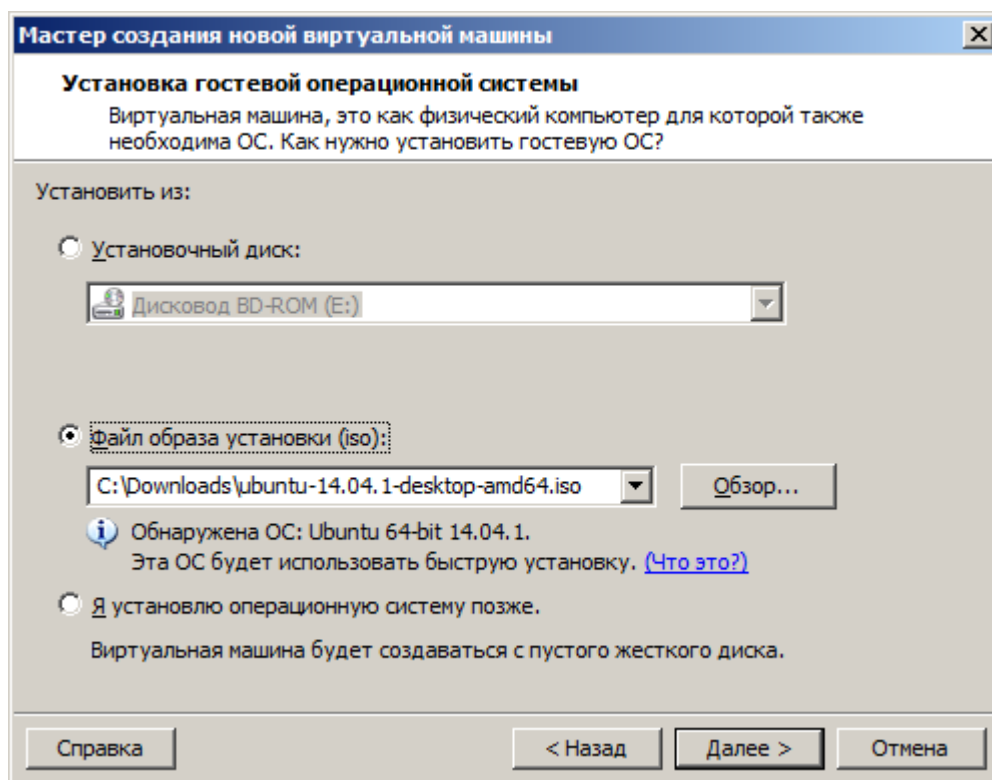


Рис. 1.2. Выбор источника установки

Для загрузки операционной системы можно использовать один из источников: первый – это физический cd/dvd привод с загрузочным носителем; второй – это файл образа загрузочного диска (iso-файл).

Также можно не устанавливать операционную систему и сделать это позже, выбрав соответствующий пункт *Я устанавливаю операционную систему позже*, тогда работа мастера завершится.



Для большинства русскоязычных дистрибутивов *Linux* способ загрузки виртуальной машины с установочного диска или его образа предпочтительнее, поскольку ОС устанавливается абсолютно также, как и на реальном оборудовании, в отличие от режима быстрой установки, который предлагается мастером после определения разновидности операционной системы.

Также при быстрой установке потребуется ввести имя пользователя и пароль для входа в его учетную запись, который будет создан в процессе установки. Во время создания новой виртуальной машины необходимо определить параметры виртуального жесткого диска (рис. 1.3).

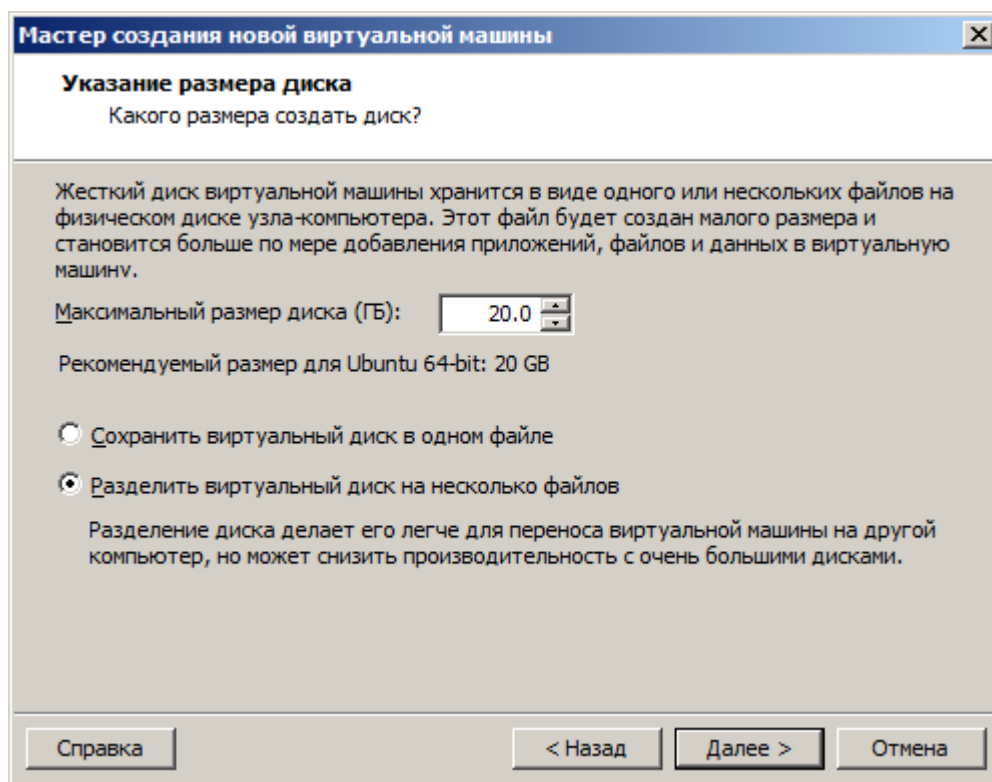


Рис. 1.3. Настройка размера диска

В среде реальной машины виртуальный жесткий диск представляет собой один или несколько файлов, размещенных в каталоге, используемом *VMware Workstation* для хранения данных виртуальных машин.

Рекомендуемый размер виртуального диска, как правило, соответствует стандартному требованию выбранной операционной системы. Если говорить

об ОС *Linux*, то 64-разрядная версия *Ubuntu* при необходимости позволяет изменить размер жесткого диска. Режим сохранения диска в одном файле более предпочтителен для максимальной производительности виртуальной машины. Также для повышения быстродействия данные виртуальных машин можно разместить на *SSD*. При создании новой виртуальной машины требуется задать ее имя. В соответствии с этим именем будет создан каталог с файлами виртуальной машины (рис. 1.4).

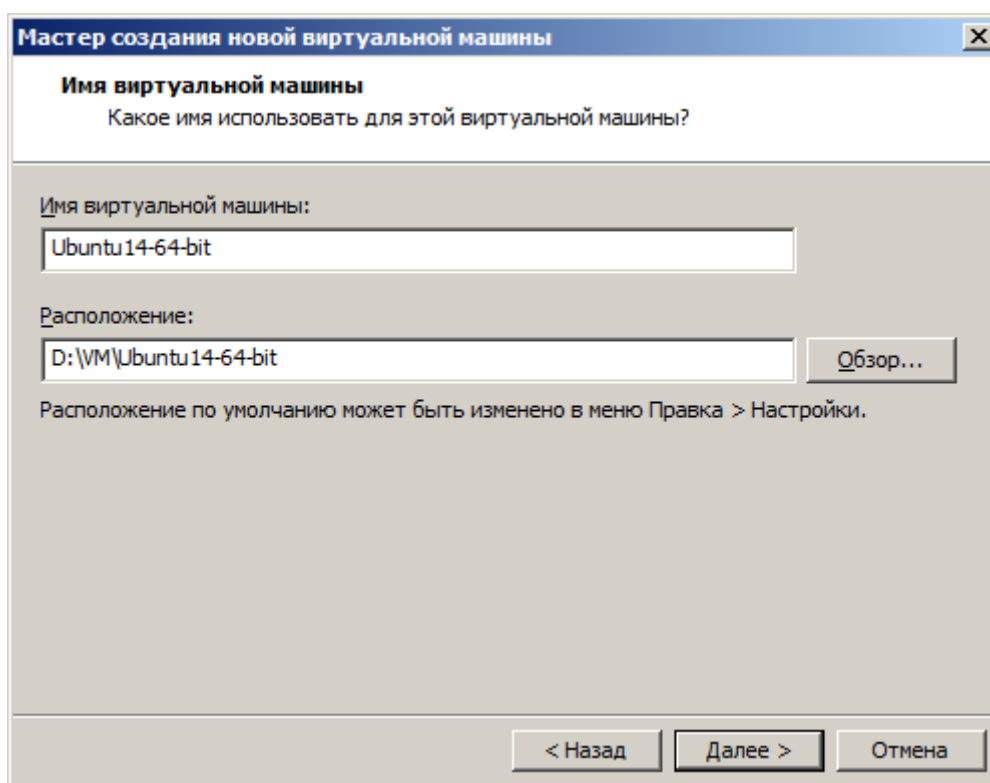


Рис. 1.4. Имя виртуальной машины

После определения всех необходимых параметров выводится итоговое сообщение о конфигурации виртуальной машины. Можно сразу приступить к работе с этой конфигурацией, выбрав галочку *Включить виртуальную машину после ее создания*. После нажатия кнопки *Готово* выполняется ее запуск и процесс установки операционной системы *Ubuntu* (рис. 1.5).

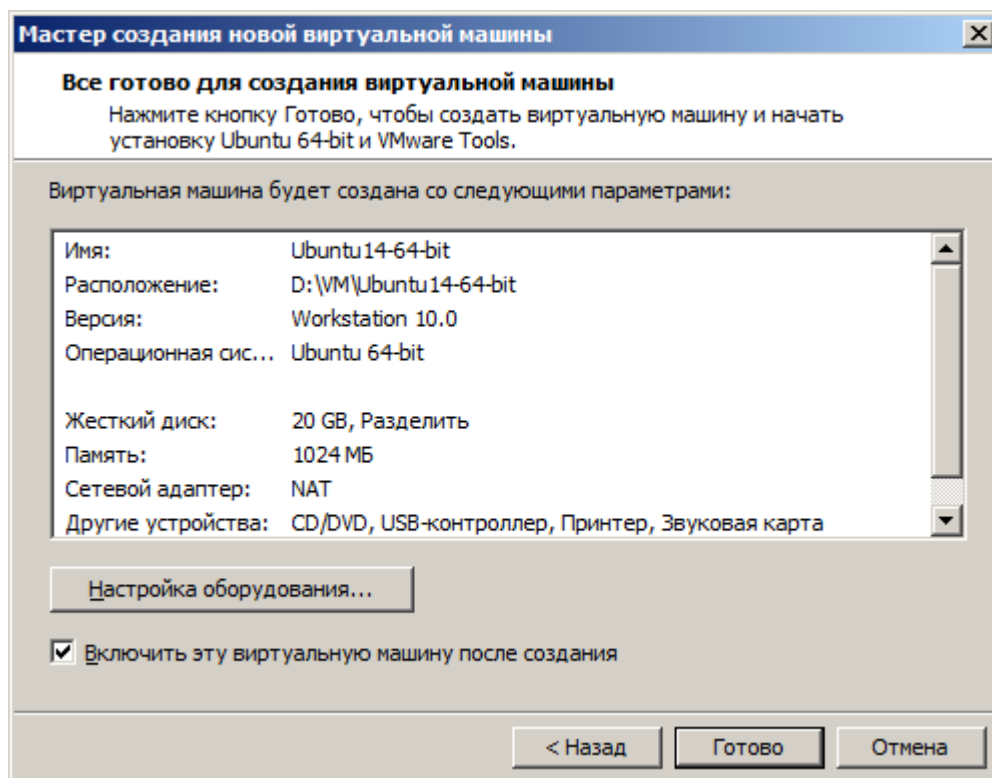


Рис. 1.5. Результат установки виртуальной машины

После того как операционная система установится с виртуального жесткого диска конфигурации, необходимо будет произвести регистрацию пользователя. Для изменения языка системы нужно выбрать пункт меню *System settings* (внизу слева значок в виде шестеренки) подпункт *Language settings* (рис. 1.6).

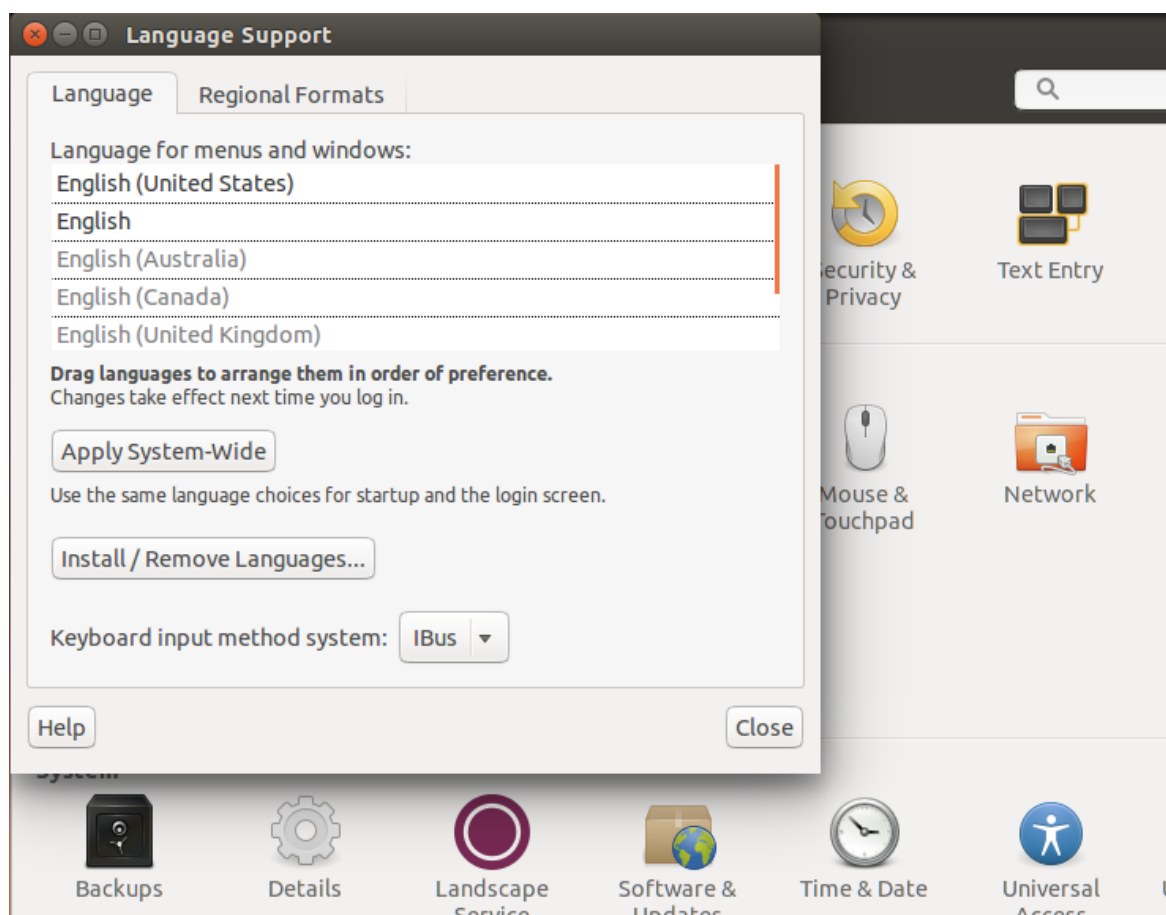


Рис. 1.6. Пункт меню *System settings*

Кнопка *Install/Remove Languages* служит для добавления или удаления языков системы. Добавленный язык отображается последним в списке предпочитаемых для меню и окон (*Languages for menus and windows*). Его нужно перетащить с помощью мыши на первую позицию вместо *English (United States)*. Изменение языка системы вступит в силу после перезагрузки системы и будет предложено обновить в соответствии с русским языком имена стандартных папок системы (рис. 1.7).

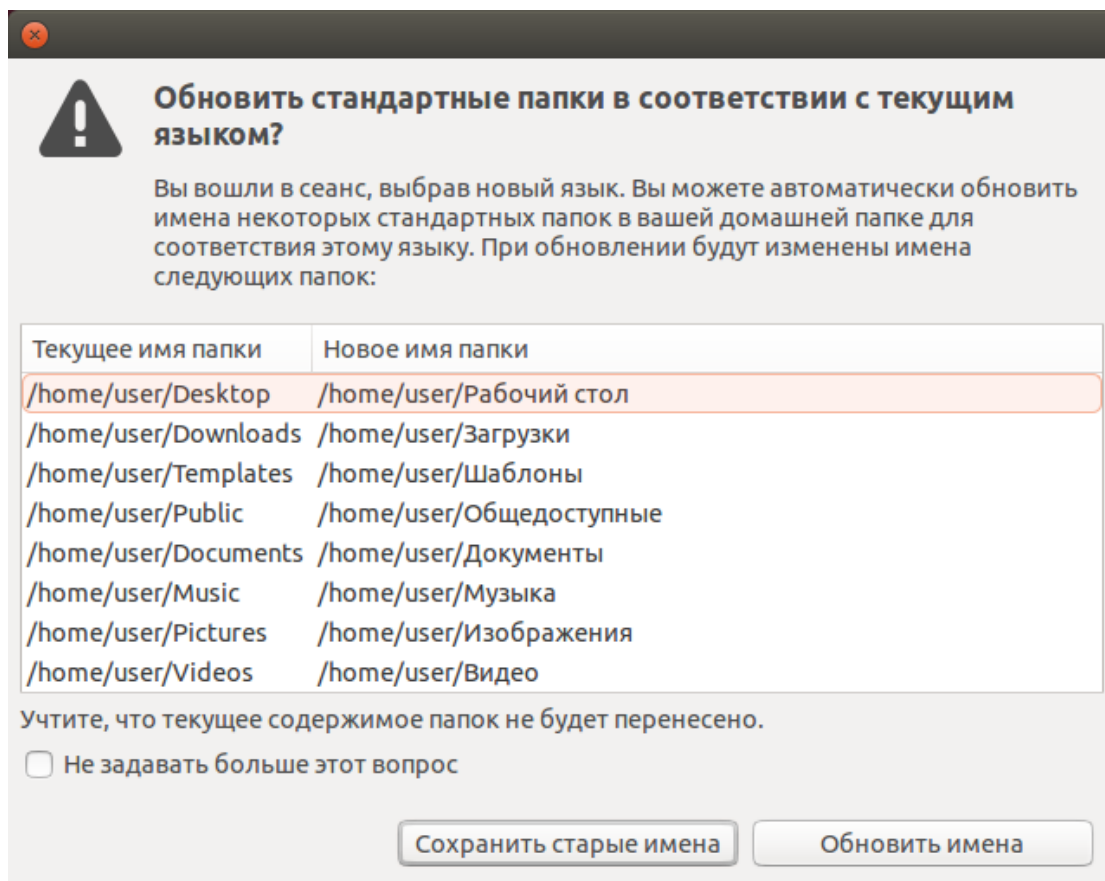


Рис. 1.7. Обновление стандартов языков

Необходимо выбрать значение по умолчанию *Сохранить старые имена*, так как это предпочтительнее для работы в командной оболочке. Кроме того, русские символы в именах папок могут стать проблемой при работе некоторых приложений.

#### *Изменение настроек виртуальной машины VMware*

Конфигурацию оборудования виртуальной машины можно изменить с использованием меню *Виртуальная машина – Параметры*. При этом изменение некоторых параметров возможно только в выключенном состоянии (рис. 1.8).

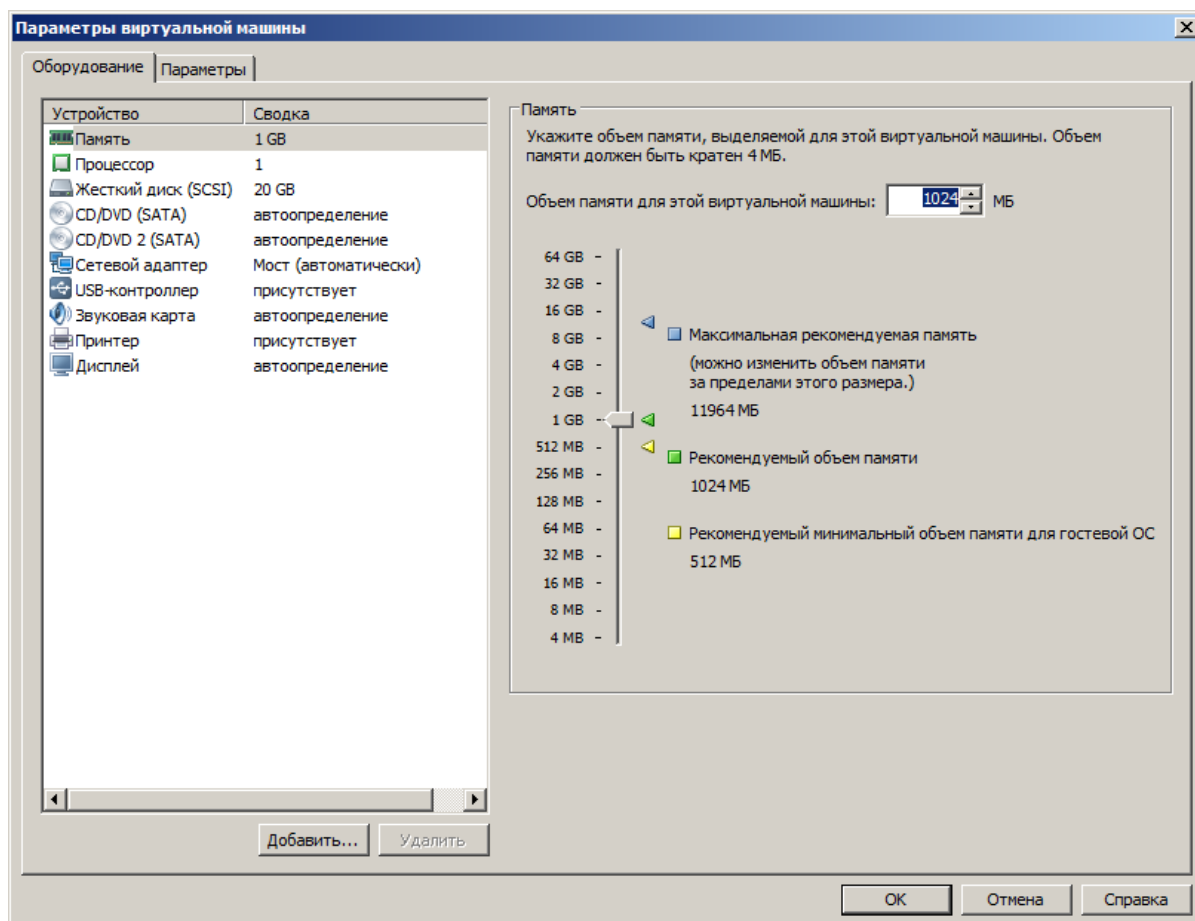


Рис. 1.8. Параметры виртуальной машины

Пользователь может добавлять новые устройства в существующую конфигурацию, а также удалять отдельные устройства или изменять их параметры. Так, например, можно изменить количество ядер, процессоров, объем оперативной памяти, добавить новые диски или удалить существующие. Имеется возможность изменения отдельных параметров устройств, входящих в состав оборудования виртуальной машины.

Изменение настроек сетевых адаптеров потребуется, если возникнет необходимость подключения виртуальной машины к существующей локальной сети в качестве обычного реального компьютера. При создании виртуальной машины с параметрами по умолчанию сетевой адаптер настраивается на использование в режиме трансляции адресов *NAT* (*Network Address*

*Translation*), когда настройки сетевого интерфейса выполняются автоматически собственной службой *DHCP*, имеющейся в составе установленного ПО VMware Workstation. Такой подход позволяет избежать конфликтов *IP*-адресов виртуальных машин с *IP*-адресами реальной локальной сети и позволяет наиболее простым способом подключиться как к сети интернет, так и к ресурсам реального компьютера. Однако режим *NAT* не позволяет компьютерам реальной локальной сети использовать ресурсы виртуальной машины. Для них она невидима, или, как иногда выражаются специалисты, «находится за *NAT*». Чтобы обеспечить к ней доступ потребуется переключить виртуальную сетевую карту в режим *сетевого моста (network bridge)*. *Сетевой мост* или просто *мост* – это режим, при котором обмен данными между виртуальным и реальным сетевыми адаптерами выполняется напрямую без какого-либо их изменения в среде гостевой ОС. Этот режим позволяет получить доступ к сетевым службам гостевой ОС таким же образом, как и для обычных реальных узлов локальной сети. В режиме моста виртуальный адаптер получает адрес из диапазона существующей локальной сети, а для передачи и приема пакетов используется реальный сетевой адаптер, что фактически аналогично добавлению *IP*-адреса виртуального адаптера в настройки *TCP/IP* реального. Таким образом, виртуальная машина становится обычным узлом локальной сети, обмениваясь данными с прочими узлами через сетевой адаптер реальной машины.

В качестве примера рассмотрим случай, когда в локальную сеть с диапазоном адресов 192.168.0.0 – 192.168.0.255 (с учетом маски – 192.168.0.0/255.255.255.0 или 192.168.0.0/24) нужно добавить виртуальную машину, которой выделен (вручную или автоматически по *DHCP*) свободный адрес из этого диапазона. Речь идет, конечно же, о *DHCP*-сервере локальной сети, а не *DHCP VMware*. Если такой сервер имеется в наличии, то достаточно

переключить виртуальный сетевой адаптер в режим моста и средствами *Linux* настроить его на получение сетевых параметров (адрес, маска, шлюз, *DNS* и т. п.) от *DHCP*-сервера. Другой вариант – настройка параметров вручную, например, с выделением виртуальной машине статического *IP*-адреса. В любом случае, первым шагом необходимо перевести виртуальный сетевой адаптер в режим моста (рис. 1.9).

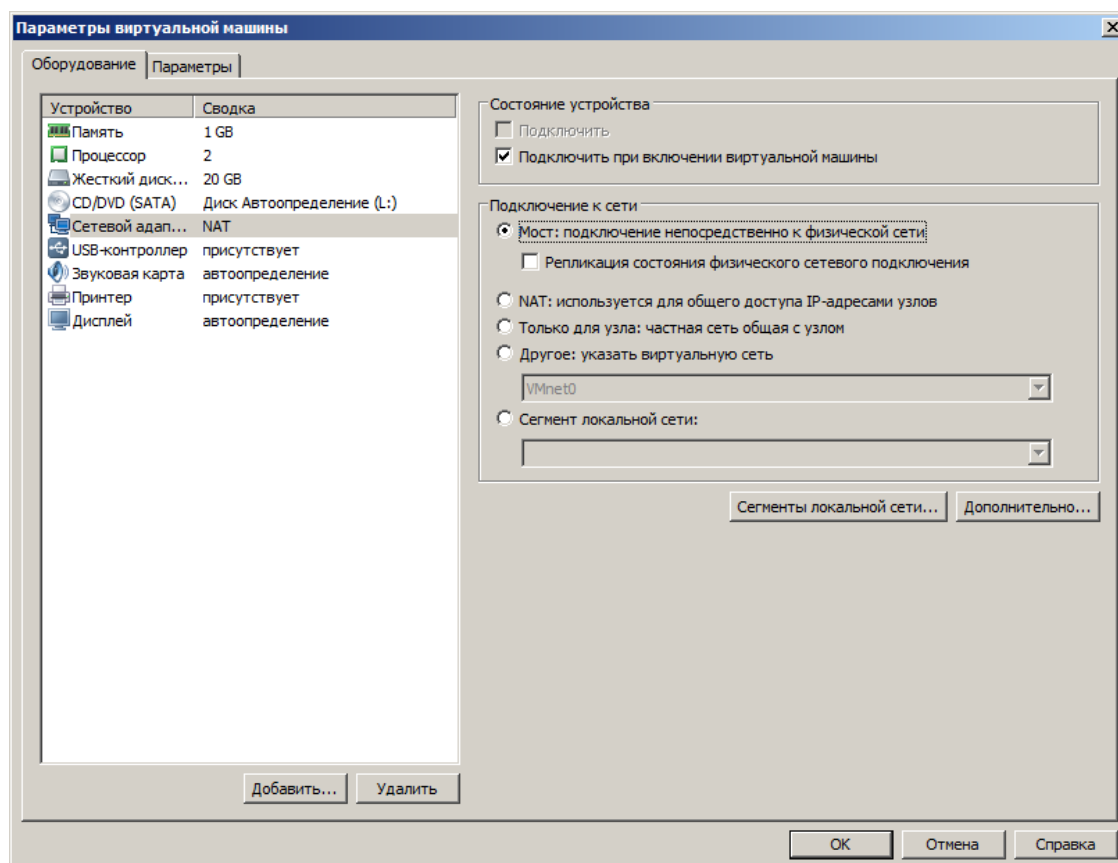


Рис. 1.9. Ручная настройка сетевого адаптера

Затем средствами гостевой ОС (в данном случае – средствами *Linux Ubuntu*) нужно настроить виртуальный сетевой адаптер для использования нового значения *IP*-адреса. В графической среде эту настройку можно сделать с помощью команды *Параметры системы - Сеть - Выбрать соединение* и нажать кнопку *Параметры* (рис. 1.10).



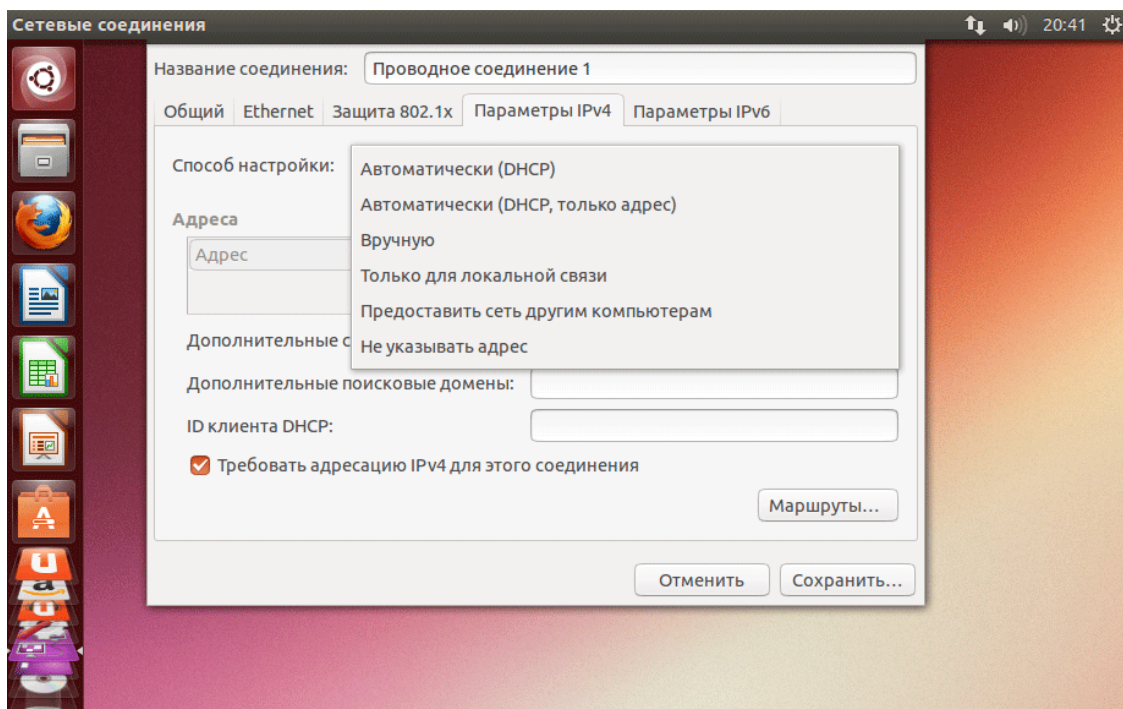


Рис. 1.10. Настройка виртуального сетевого адаптера

Для случая использования статического *IP*-адреса нужно выбрать режим *Вручную* и ввести в требуемые поля *IP*-адрес и маску подсети, а также адрес шлюза и *DNS*-серверов, если они используются в реальной локальной сети. При использовании виртуальной машины в качестве узла реальной локальной сети необходимо учитывать тот факт, что могут потребоваться дополнительные настройки брандмауэров для разрешения необходимого сетевого трафика.

### *Настройка и установка VMware Tools*

*VMware Tools* – дополнительное программное обеспечение для повышения производительности и улучшения управляемости виртуальных машин. Установка данного ПО выполняется в среде гостевой операционной системы. Установку (переустановку) *VMware Tools* можно выполнить при первой загрузке гостевой ОС, либо в любой момент времени с использованием меню

*Виртуальная машина – Установить VMware Tools.* Для установки используются файлы с образами в формате *ISO*, которые временно монтируются в среде гостевой ОС в качестве виртуальных *CD/DVD*-приводов. Естественно, для разных типов операционных систем используются свои инсталляционные образы (свои файлы *.iso* для *Windows*, *Linux*, *Solaris* и т. п., которые хранятся в каталоге, куда устанавливалась *VMware Workstation*). Процесс установки *VMware Tools* зависит от типа гостевой ОС. Виртуальная машина с *Linux* должна быть запущена. Поскольку инсталлятор *VMware Tools* написан на *Perl*, он должен быть установлен на гостевой ОС. Установка должна выполняться под учетной записью суперпользователя *root*.

На виртуальном приводе инсталляционный пакет представляет собой сжатый файл с именем *VMwareTools-x.x.x-yyyy.tar.gz*, где *x* и *y* определяются версией программы. Необходимо распаковать его, например, во временный каталог */tmp* (или созданный в нем подкаталог, например *vmtools*).

Для этого переходим в каталог */tmp*

```
cd /tmp
```

Создаем каталог для распакованных файлов

```
mkdir vmtools
```

Переходим в созданный каталог

```
cd vmtools
```

Распаковываем инсталляционный пакет. Стандартно после монтирования виртуального привода в *Ubuntu* он находится по адресу */media/пользователь/VMware Tools*, например */media/user/VMware Tools* : *tar xvf /media/user/VMware Tools/VMwareTools-x.x.x-yyyy.tar.gz*

После завершения распаковки, переходим в каталог инсталлятора

```
cd vmware-tools-distrib
```

Запускаем процесс инсталляции под учетной записью *root*

```
sudo ./vmware-install.pl
```

В ходе установки *VMware Tools* на экран выдаются запросы о размещении файлов в различных каталогах файловой системы, на которые можно отвечать нажатием *Enter*, если используется стандартное размещение. После завершения установки необходимо перезагрузить систему.

### **Задание**

Установить операционную систему *Ubuntu*, полученную от преподавателя на виртуальную машину *VMWare WorkStation*.

### **Содержание отчета**

1. Титульный лист
2. Цель работы.
3. Формулировка задания.
4. Подробное описание выполненных действий в процессе установки и настройки операционной системы *Ubuntu*.
5. Общий вывод о проделанной работе.

## Лабораторная работа № 2

### НАСТРОЙКА LAMP

Время выполнения – 8 часов (аудиторная работа – 4 часа, самостоятельная работа – 4 часа).

**Цель работы:** освоение основных приемов работы с *Linux*, *Apache*, *MySQL*, *PHP*.

#### Задачи работы

1. Получить навыки работы с *LAMP*.
2. Изучить основные приемы настройки программного продукта.
3. Научиться использовать *Apache*, *MySQL*, *PHP* для работы с web-ресурсами.

#### Перечень обеспечивающих средств

Задания лабораторной работы выполняются в операционной системе *Ubuntu*.

#### Общие теоретические сведения

*LAMP* – это сочетание технологий *Linux*, *Apache*, *MySQL*, *PHP* с открытым исходным кодом, которые используются для веб-разработки.

Стек *LAMP* устанавливается через утилиту *tasksel*. Для этого необходимо в командную строку *Linux* ввести следующие команды (рис. 2.1):

```
$ sudo apt install -y curl tasksel
```

```
$ sudo tasksel install lamp-server
```

```
kas@DESKTOP-RA654U1: ~
Processing triggers for initramfs-tools (0.130ubuntu3.9) ...
kas@DESKTOP-RA654U1:~$ sudo apt install tasksel
[sudo] password for kas:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libdumbnet1 libfreetype6
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  laptop-detect tasksel-data
The following NEW packages will be installed:
  laptop-detect tasksel tasksel-data
0 upgraded, 3 newly installed, 0 to remove and 2 not upgraded.
Need to get 40.2 kB of archives.
After this operation, 312 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 tasksel-data all 3.34ubuntu11 [5476 B]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 tasksel all 3.34ubuntu11 [28.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 laptop-detect all 0.16 [6016 B]
Fetched 40.2 kB in 1s (60.8 kB/s)
Preconfiguring packages ...
Selecting previously unselected package tasksel-data.
(Reading database ... 28678 files and directories currently installed.)
Preparing to unpack .../tasksel-data_3.34ubuntu11_all.deb ...
Unpacking tasksel-data (3.34ubuntu11) ...
Selecting previously unselected package tasksel.
Preparing to unpack .../tasksel_3.34ubuntu11_all.deb ...
Unpacking tasksel (3.34ubuntu11) ...
Selecting previously unselected package laptop-detect.
```

Рис. 2.1. Процесс скачивания пакетов

После того как процесс установки и распаковки файлов завершиться, необходимо выполнить проверку.

1. Перезагружаем apache командой – *sudo service apache2 restart*.

Переходим по ссылке: *http://localhost/* (рис. 2.2).

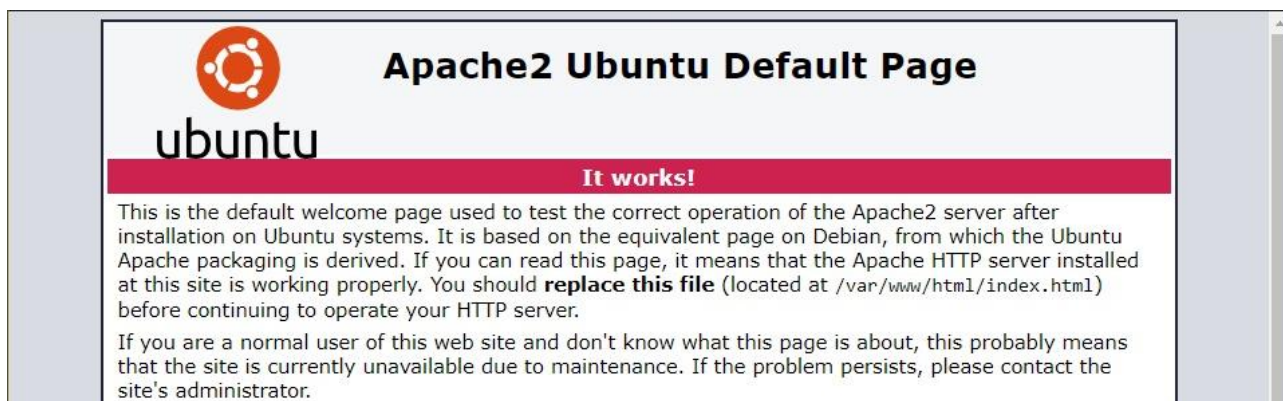
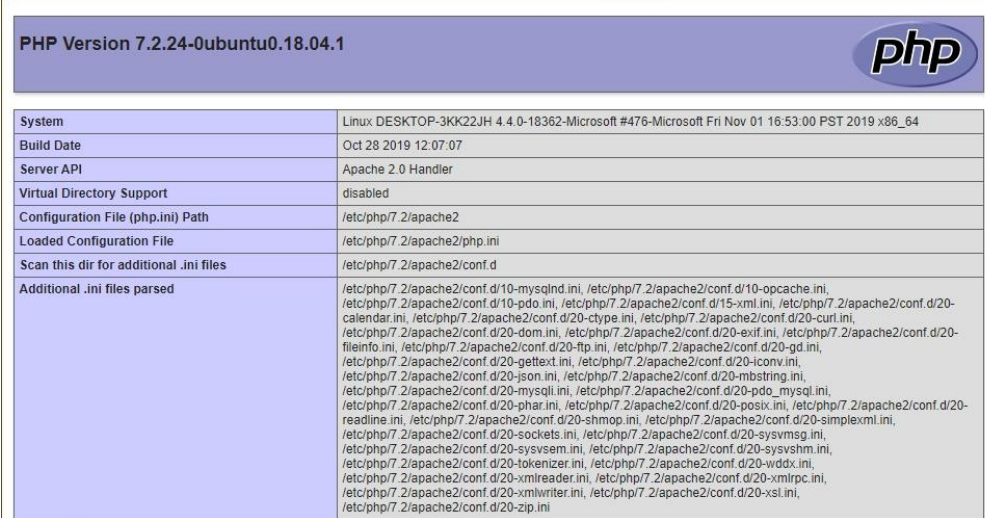


Рис. 2.2. Результат перезагрузки Apache2

2. Для проверки *PHP* в терминал вводим:

```
sudo nano /var/www/html/phpinfo.php
```

В открывшемся окне прописываем - `<?php phpinfo(); ?>`, после для выхода из редактора используем сочетание «*ctrl+s*», «*ctrl+x*» и переходим по ссылке: <http://localhost/phpinfo.php> (рис. 2.3).



PHP Version 7.2.24-0ubuntu0.18.04.1	
System	Linux DESKTOP-3KK22JH 4.4.0-18362-Microsoft #476-Microsoft Fri Nov 01 16:53:00 PST 2019 x86_64
Build Date	Oct 28 2019 12:07:07
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqld.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/15-xml.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-curl.ini, /etc/php/7.2/apache2/conf.d/20-dom.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gd.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mbstring.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-simplexml.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini, /etc/php/7.2/apache2/conf.d/20-wddx.ini, /etc/php/7.2/apache2/conf.d/20-xmlreader.ini, /etc/php/7.2/apache2/conf.d/20-xmlrpc.ini, /etc/php/7.2/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.2/apache2/conf.d/20-xsl.ini, /etc/php/7.2/apache2/conf.d/20-zip.ini

Рис. 2.3. Результат проверки PHP

Следующим шагом после проверки корректной установки *Apache2* и *PHP* будет запуск базы данных *MySQL* и ее первичная настройка. Для этого необходимо выполнить следующие действия:

1. Запустить *mysql* – `$ sudo service mysql start`.

Запускаем настройку безопасности: `$ sudo mysql_secure_installation`.

Соглашаемся на установку пароля, выбираем сложность пароля. С оставшимися запросами соглашаемся (рис. 2.4).

```
kas@DESKTOP-RA654U1: ~  
No directory, logging in with HOME=/  
[ OK ]  
kas@DESKTOP-RA654U1:~$ sudo mysql_secure_installation  
Securing the MySQL server deployment.  
Connecting to MySQL using a blank password.  
VALIDATE PASSWORD PLUGIN can be used to test passwords  
and improve security. It checks the strength of password  
and allows the users to set only those passwords which are  
secure enough. Would you like to setup VALIDATE PASSWORD plugin?  
Press y|Y for Yes, any other key for No: y  
There are three levels of password validation policy:  
LOW Length >= 8  
MEDIUM Length >= 8, numeric, mixed case, and special characters  
STRONG Length >= 8, numeric, mixed case, special characters and dictionary file  
Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0  
Please set the password for root here.  
New password:  
Re-enter new password:  
Estimated strength of the password: 25  
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y
```

Рис. 2.4. Настройка безопасности *MySQL*

Для комфортной работы с *MySQL* следует установить *PhpMyAdmin*:

```
$ sudo apt update
```

```
$ sudo apt install phpmyadmin php-mbstring php-gettext
```

Далее будет предложен выбор веб-сервера для автоматической настройки *phpMyAdmin*. Для этого выделяем *apache2*, нажимаем пробел, *Tab*, *Enter* (рис. 2.5).

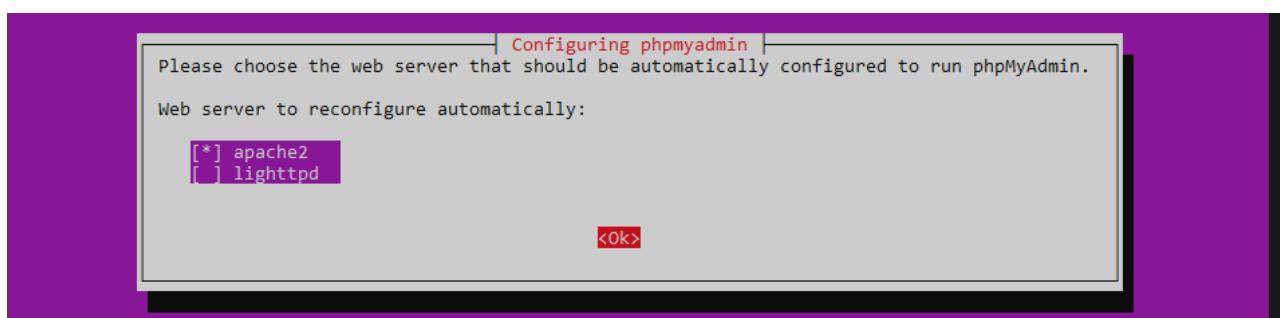


Рис. 2.5. Выбор web-сервера

Соглашаемся с настройкой базы данных для *phpMyAdmin* с помощью *dbconfig-common* (рис. 2.6) и на последнем шаге создаем пароль.

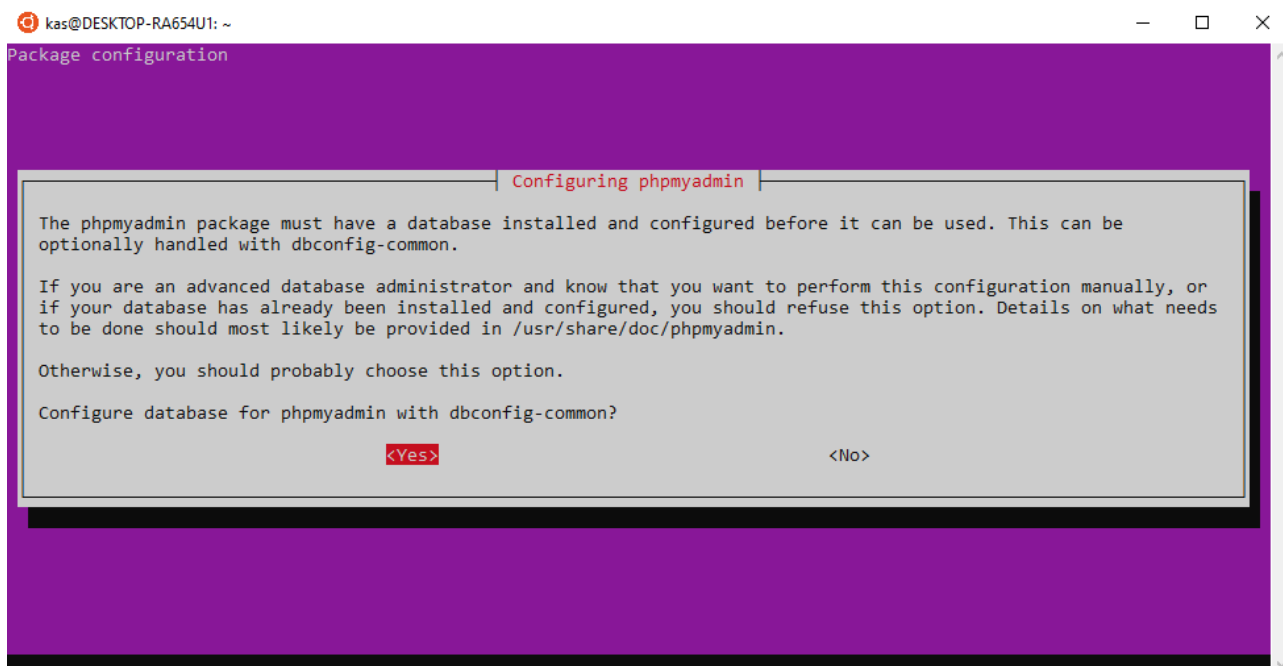


Рис. 2.6. Подтверждение настроек

После окончания работы с установкой и первичной настройкой *phpMyAdmin* выполняем перезагрузку *Apache2*: `$ sudo apache2ctl restart` и переходим по ссылке: `http://localhost/phpmyadmin/` для проверки работоспособности. Должна открыться страница авторизации *phpMyAdmin* (рис. 2.7).

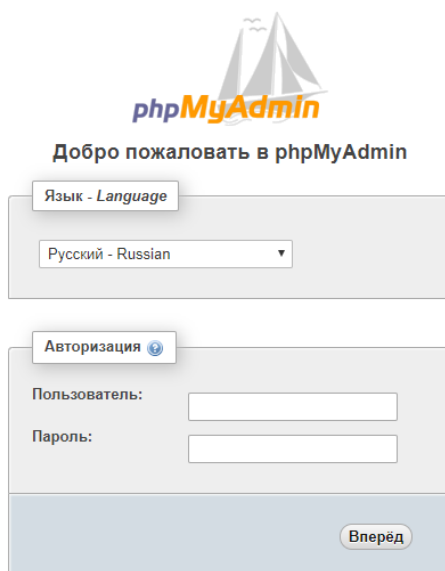


Рис. 2.7. Страница авторизации *phpMyAdmin*



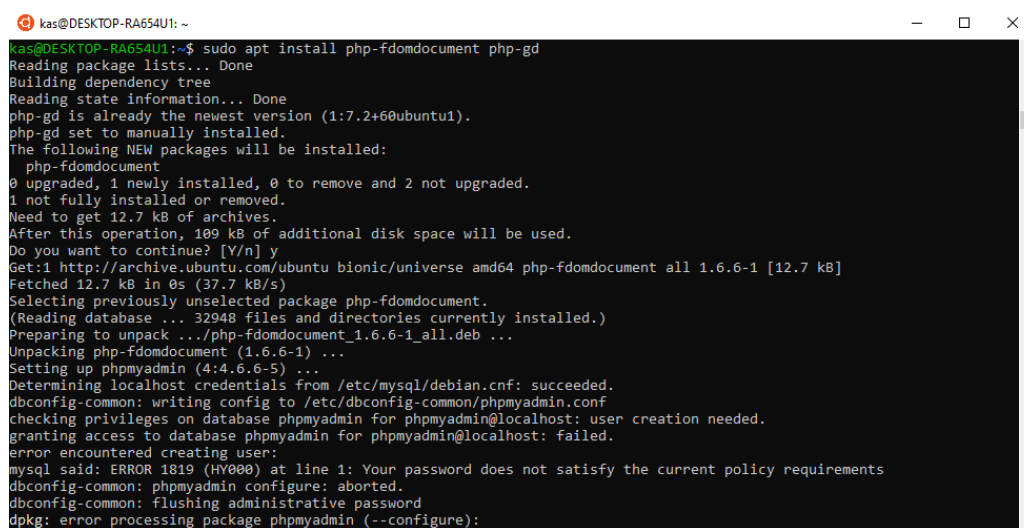
После того как мы убедились в работоспособности выполненных действий, включаем возможность редактирования файлов и устанавливаем дополнение к *php*, для этого вводим следующие команды:

`$ sudo a2enmod rewrite` – редактирование

`$ sudo service apache2 restart` – перезагрузка

`$ sudo apt install php-fpm php-gd` – дополнения

(рис. 2.8).



```
kas@DESKTOP-RA654U1: ~  
kas@DESKTOP-RA654U1:~$ sudo apt install php-fpm php-gd  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
php-gd is already the newest version (1:7.2+60ubuntu1).  
php-gd set to manually installed.  
The following NEW packages will be installed:  
  php-fpm  
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.  
1 not fully installed or removed.  
Need to get 12.7 kB of archives.  
After this operation, 109 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://archive.ubuntu.com/ubuntu bionic/universe amd64 php-fpm all 1.6.6-1 [12.7 kB]  
Fetched 12.7 kB in 0s (37.7 kB/s)  
Selecting previously unselected package php-fpm.  
(Reading database ... 32948 files and directories currently installed.)  
Preparing to unpack .../php-fpm_1.6.6-1_all.deb ...  
Unpacking php-fpm (1.6.6-1) ...  
Setting up phpmyadmin (4:4.6.6-5) ...  
Determining localhost credentials from /etc/mysql/debian.cnf: succeeded.  
dbconfig-common: writing config to /etc/dbconfig-common/phpmyadmin.conf  
checking privileges on database phpmyadmin for phpmyadmin@localhost: user creation needed.  
granting access to database phpmyadmin for phpmyadmin@localhost: failed.  
error encountered creating user:  
mysql said: ERROR 1819 (HY000) at line 1: Your password does not satisfy the current policy requirements  
dbconfig-common: phpmyadmin configure: aborted.  
dbconfig-common: flushing administrative password  
dpkg: error processing package phpmyadmin (--configure):
```

Рис. 2.8. Установка дополнений

*Настройка apache и создание директории для сайта.*

Следующим шагом необходимо создать каталог для сайта:

`cd /var/www/html/` - переход в расположение сайта

В нем создаем каталог:

`$ sudo mkdir abracadabra,`

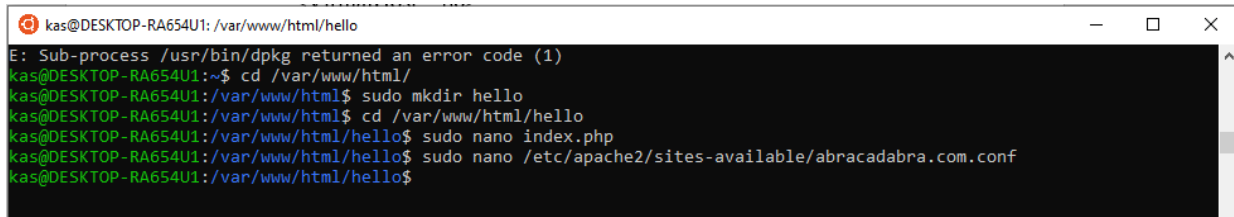
где *abracadabra* – название каталога.

Для проверки создания вводим: `ls`.

Перейдем в только что созданный каталог и создадим там файл *index.php* (рис. 2.9).

`cd /var/www/html/abracadabra`

```
$ sudo nano index.php
```

A screenshot of a terminal window with a black background and green text. The window title is 'kas@DESKTOP-RA654U1: /var/www/html/hello'. The terminal shows the following commands and output: 'E: Sub-process /usr/bin/dpkg returned an error code (1)', 'kas@DESKTOP-RA654U1:~\$ cd /var/www/html/', 'kas@DESKTOP-RA654U1:/var/www/html\$ sudo mkdir hello', 'kas@DESKTOP-RA654U1:/var/www/html\$ cd /var/www/html/hello', 'kas@DESKTOP-RA654U1:/var/www/html/hello\$ sudo nano index.php', and 'kas@DESKTOP-RA654U1:/var/www/html/hello\$ sudo nano /etc/apache2/sites-available/abracadabra.com.conf'.

```
kas@DESKTOP-RA654U1: /var/www/html/hello
E: Sub-process /usr/bin/dpkg returned an error code (1)
kas@DESKTOP-RA654U1:~$ cd /var/www/html/
kas@DESKTOP-RA654U1:/var/www/html$ sudo mkdir hello
kas@DESKTOP-RA654U1:/var/www/html$ cd /var/www/html/hello
kas@DESKTOP-RA654U1:/var/www/html/hello$ sudo nano index.php
kas@DESKTOP-RA654U1:/var/www/html/hello$ sudo nano /etc/apache2/sites-available/abracadabra.com.conf
kas@DESKTOP-RA654U1:/var/www/html/hello$
```

Рис. 2.9. Создание файла *index*

В созданный файл прописываем *php*-скрипт:

```
<?php
    echo "Это мой сайт ABRACADABRA!!!!";
?>
```

Для выхода и сохранения изменений нужно нажать: «*ctrl+s*», «*ctrl+x*».

#### *Отладка конфигурации apache*

Для редактирования файла *conf* вводим команду:

```
$ sudo nano /etc/apache2/sites-available/abracadabra.com.conf
```

В открывшемся окне прописываем:

```
<VirtualHost *:80>

    ServerAdmin webmaster@abracadabra.com
    ServerName abracadabra.com

    DocumentRoot /var/www/html/abracadabra
    <Directory />
        Options FollowSymLinks
        AllowOverride All
    </Directory>
    <Directory /var/www/html/abracadabra/>
        Options Indexes FollowSymLinks MultiViews
```

```

        AllowOverride All
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride All
        Options +ExecCGI -MultiViews +Sym-
LinksIfOwnerMatch

        Order allow,deny
        Allow from all
    </Directory>

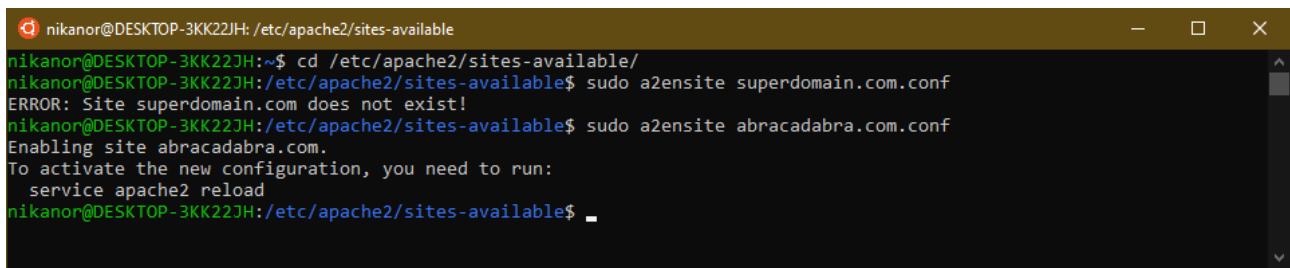
    ErrorLog ${APACHE_LOG_DIR}/error.log

    # Possible values include: debug, info, notice,
warn, error, crit,
    # alert, emerg.
    LogLevel warn

    CustomLog ${APACHE_LOG_DIR}/access.log com-
bined
</VirtualHost>

```

Выходим из редактора *nano*, переходим в каталог *apache* командой: *cd /etc/apache2/sites-available/* и вводим команду *\$ sudo a2ensite abracadabra.com.conf* (рис. 2.10) и перезагружаем *apache*: *service apache2 reload*.



```
nikanor@DESKTOP-3KK22JH: /etc/apache2/sites-available
nikanor@DESKTOP-3KK22JH:~$ cd /etc/apache2/sites-available/
nikanor@DESKTOP-3KK22JH:/etc/apache2/sites-available$ sudo a2ensite superdomain.com.conf
ERROR: Site superdomain.com does not exist!
nikanor@DESKTOP-3KK22JH:/etc/apache2/sites-available$ sudo a2ensite abracadabra.com.conf
Enabling site abracadabra.com.
To activate the new configuration, you need to run:
  service apache2 reload
nikanor@DESKTOP-3KK22JH:/etc/apache2/sites-available$ _
```

Рис. 2.10. Применение изменений в файле конфигурации

### *Редактирование hosts*

Для редактирования *hosts* на *Ubuntu*, вводим команду:  
`$ sudo nano /etc/hosts` в открывшемся окне в конце прописываем:  
`127.0.0.1 abracadabra.com` и снова перезагружаем *Apache2* для принятия изменений.

Чтобы найти файл *host* нужно перейти в следующий каталог: *C:\Windows\System32\drivers\etc\hosts*.

Для проверки переходим по ссылке: <http://abracadabra.com/>.

В результате на экране отобразится результат выполнения файла *index.php*.

## **Задание**

Установить и настроить *Apache*, *MySQL*, *PHP* для запуска интернет-ресурсов, как личной разработки, так и проектов, функционирующих на известных *CMS*.

## **Содержание отчета**

1. Титульный лист
2. Цель работы.
3. Формулировка задания.

4. Подробное описание выполненных действий в процессе установки и настройки *LAMP*.

5. Общий вывод о проделанной работе

## Лабораторная работа № 3

### РАСПРЕДЕЛЕННАЯ СИСТЕМА УПРАВЛЕНИЯ ВЕРСИЯМИ (GIT)

Время выполнения – 8 часов (аудиторная работа – 4 часа, самостоятельная работа – 4 часа).

**Цель работы:** освоение основных приемов работы в распределенной системе управления версиями.

#### Задачи работы

1. Получить навыки работы с распределенной системой управления версиями.
2. Изучить основные приемы настройки программного продукта.
3. Научиться использовать распределенные системы управления версиями для работы с *web*-ресурсами.

#### Перечень обеспечивающих средств

Задания лабораторной работы выполняются в операционной системе *Ubuntu*.

#### Общие теоретические сведения

*Git* (произносится «гит») – распределенная система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра *Linux*, первая версия выпущена 7 апреля 2005 года. На сегодняшний день его поддерживает Джунио Хамано.

Система спроектирована как набор программ, специально разработанных с учетом их использования в сценариях. Это позволяет удобно создавать специализированные системы контроля версий на базе *Git* или пользовательские интерфейсы. Например, *Cogito* является именно таким примером оболочки к репозиториям *Git*, а *StGit* использует *Git* для управления коллекцией исправлений (патчей).

*Git* поддерживает быстрое разделение и слияние версий, включает инструменты для визуализации и навигации по нелинейной истории разработки. Как и *Darcs*, *BitKeeper*, *Mercurial*, *Bazaar* и *Monotone*, *Git* предоставляет каждому разработчику локальную копию всей истории разработки, изменения копируются из одного репозитория в другой.

Удаленный доступ к репозиториям *Git* обеспечивается *git*-демоном, *SSH*- или *HTTP*-сервером. *TCP*-сервис *git-daemon* входит в дистрибутив *Git* и является наряду с *SSH* наиболее распространённым и надежным методом доступа. Метод доступа по *HTTP*, несмотря на ряд ограничений, очень популярен в контролируемых сетях, потому что позволяет использовать существующие конфигурации сетевых фильтров

### *Настройка Git для операционных систем семейства Linux.*

Для установки *Git* понадобятся библиотеки, от которых он зависит: *curl*, *zlib*, *openssl*, *expat* и *libiconv*. Например, если в используемой системе менеджер пакетов - *yum* (*Fedora*), или *apt-get* (*Debian*, *Ubuntu*), можно воспользоваться следующими командами, чтобы разрешить все зависимости:

```
$ yum install curl-devel expat-devel gettext-devel \
openssl-devel zlib-devel
```

данная команда для тех, у кого менеджер пакетов – *yum* (*Fedora*)

```
$ apt-get install libcurl4-gnutls-dev libexpat1-dev
gettext \
libz-dev libssl-dev
```

данная команда для тех, у кого менеджер пакетов – *eum* (*Debian*, *Ubuntu*)

Установив все необходимые библиотеки, можно идти дальше и скачать последнюю версию с сайта *Git*: <http://git-scm.com/download>.

Теперь скомпилируем и установим:

```
$ tar -zxf git-1.7.2.2.tar.gz
$ cd git-1.7.2.2
$ make prefix=/usr/local all
$ sudo make prefix=/usr/local install
```

После этого можно скачать *Git* с помощью самого *Git*, чтобы получить обновления:

```
$ git clone git://git.kernel.org/pub/scm/git/git.git
```

Если же у нас дистрибутив, основанный на *Debian*, например, *Ubuntu*, попробуем *apt-get*:

```
$ apt-get install git
```

### Настройка *Git* для Windows

#### 1. Загрузка и установка

Загрузим установщик с официального сайта. Загрузка начнется автоматически (рис. 3.1).



Рис. 3.1. Загрузка дистрибутива *Git* с официального сайта



Перейдем в папку *Downloads* и запустим на исполнение загруженный файл и укажем путь до каталога, в который будет установлен *Git* (рис. 3.2).

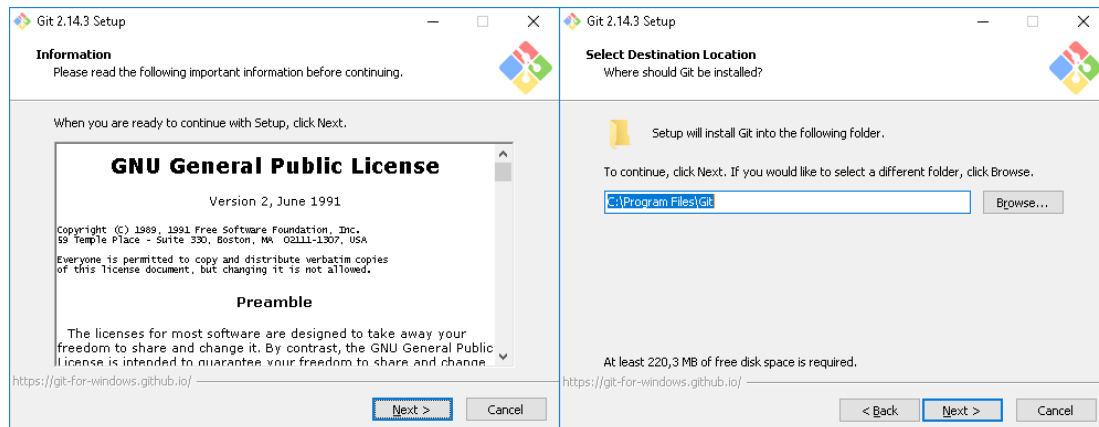


Рис. 3.2. Запуск дистрибутива

Чтобы на рабочем столе была иконка *Git*, на следующем шаге отметим галочкой *On the Desktop* (рис. 3.3).

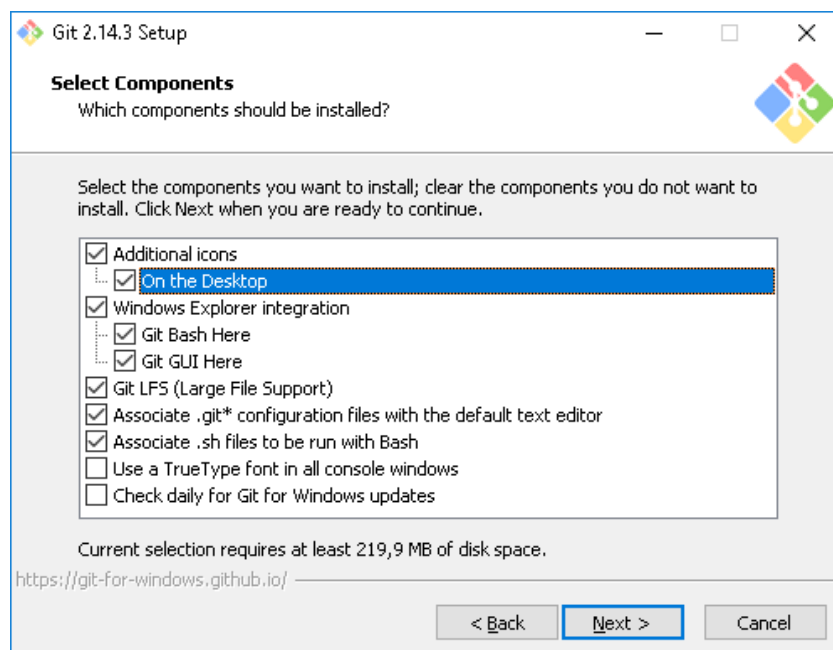


Рис. 3.3. *On the Desktop*

Введем имя директории, которая будет создана в *Start Menu*. При необходимости можно изменить путь с помощью кнопки *Browse* (рис. 3.4).

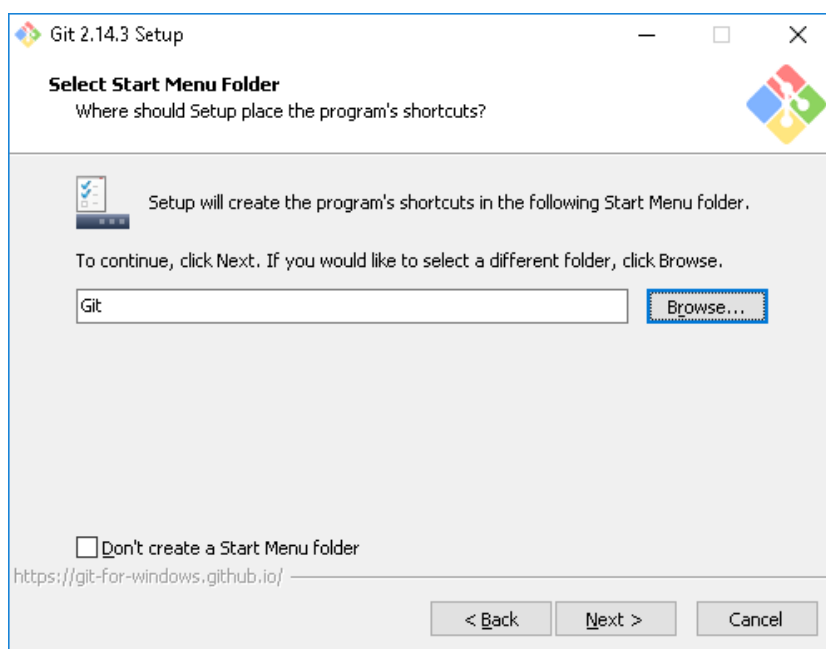


Рис. 3.4. Имя директории

Следующим шагом будет выбор одного из способов использования *git* из командной строки. На данный момент существует 3 способа использования:

1) *Use Git from Git Bash only* - использование только из командной строки *Bash*;

2) *Use Git from the Windows Command Prompt* - использование командной строки *Bash*, а также минимальный набор команд *Git* из консоли *Windows*;

3) *Use Git and optional Unix tools from the Windows Command Prompt* - использование *Git* и утилит *Unix* из командной строки *Windows*, в этом случае будут перезаписаны некоторые утилиты *Windows*, например, *find* и *sort* (рис. 3.5).

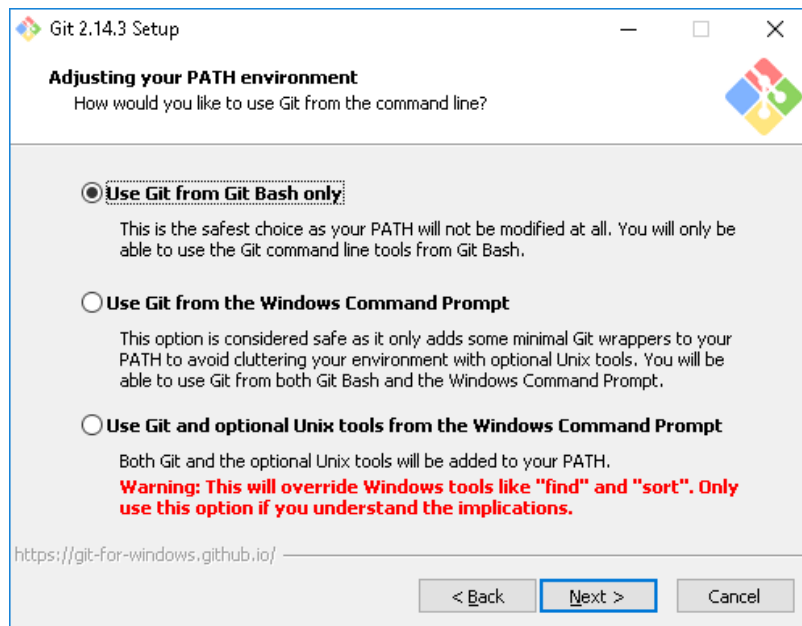


Рис. 3.5. Способы использования *git*

Выберем библиотеку, которая будет использована при подключении по протоколу *HTTPS*:

- *OpenSSL* - сертификаты сервера будут проверяться с использованием *Unix*-файла *ca-bundle.crt*;
- *Windows Secure Channel* - сертификаты сервера будут проверяться с использованием стандартной библиотеки *Windows* (рис. 3.6).

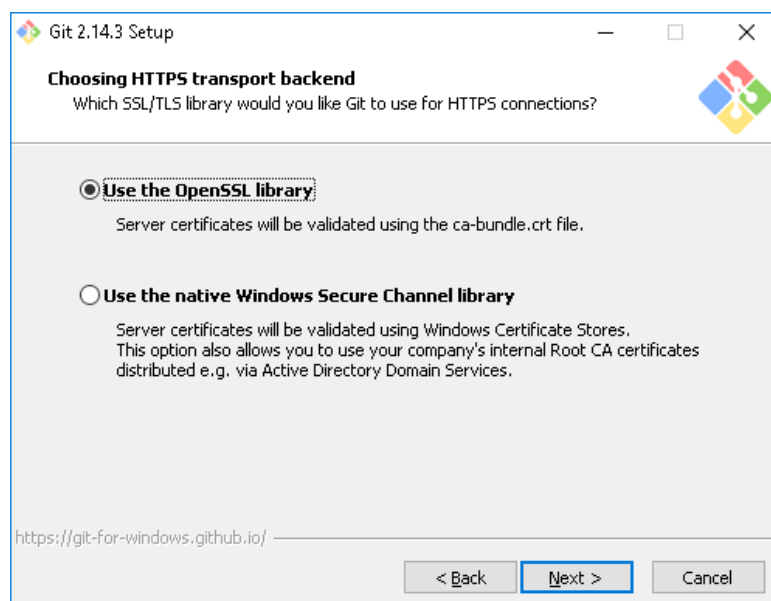


Рис. 3.6. Библиотеки

Убедимся, что мы выбрали способ обработки окончания строк «*Checkout Windows-style, commit Unix-style line endings*». Это значение гарантирует, что *Git* преобразует *LF* в *CRLF* при проверке текстовых файлов. При выполнении текстовых файлов *CRLF* также преобразуется в *LF*. Это мера совместимости для защиты новых строк в текстовых файлах, что позволяет легко работать с текстовыми файлами в *Windows* и на платформах *Unix* (рис. 3.7).

Примечание: *LF* и *CRLF* - управляющий символ для переноса строки в *Unix* и *Windows* соответственно.

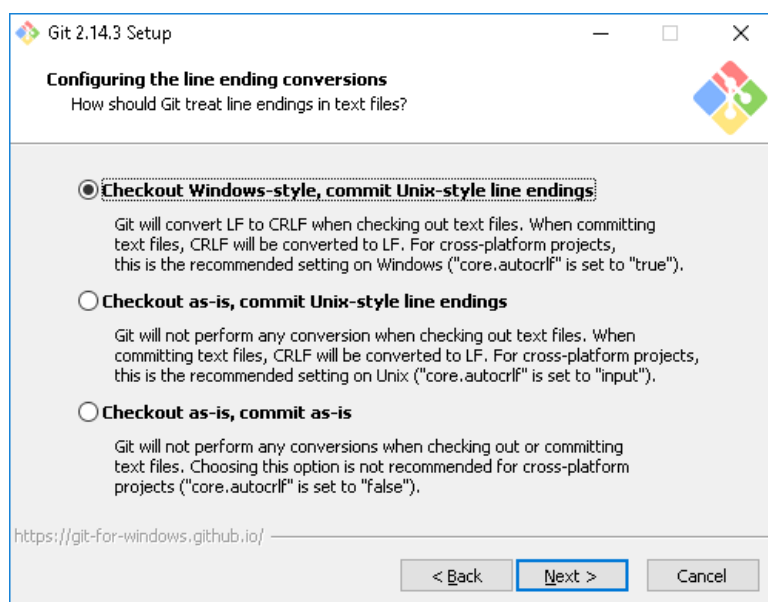


Рис. 3.7. Способы обработки окончания строк

Далее необходимо сконфигурировать используемый терминал (рис. 3.8):

- *MinTTY* - терминал *Unix*;
- *Windows* - стандартный терминал *Windows*.

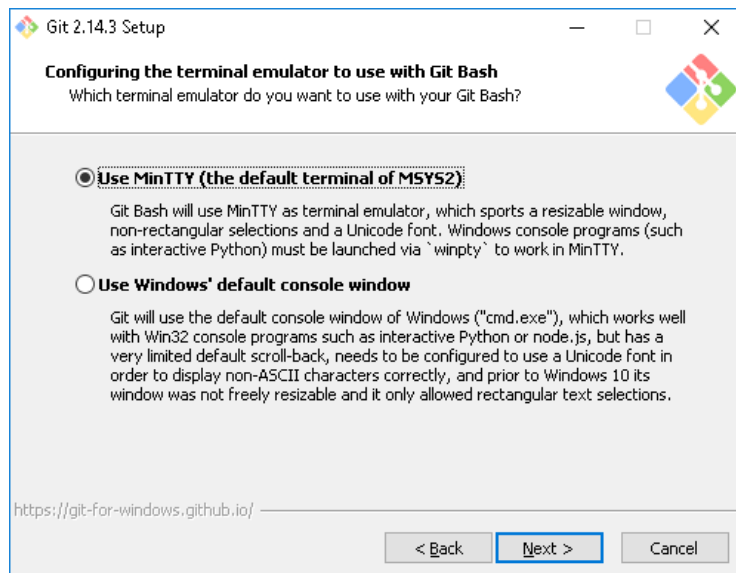


Рис. 3.8. Конфигурация используемого терминала

Отметим галочками нужные нам дополнительные функции:

- *File system caching* - кэширование файловой системы;
- *Git Credential Manager* - включить менеджер учетных данных;
- *Symbolic links* - разрешить символичные ссылки.

Нажмем кнопку *Install*, начнется процесс установки (рис. 3.9).

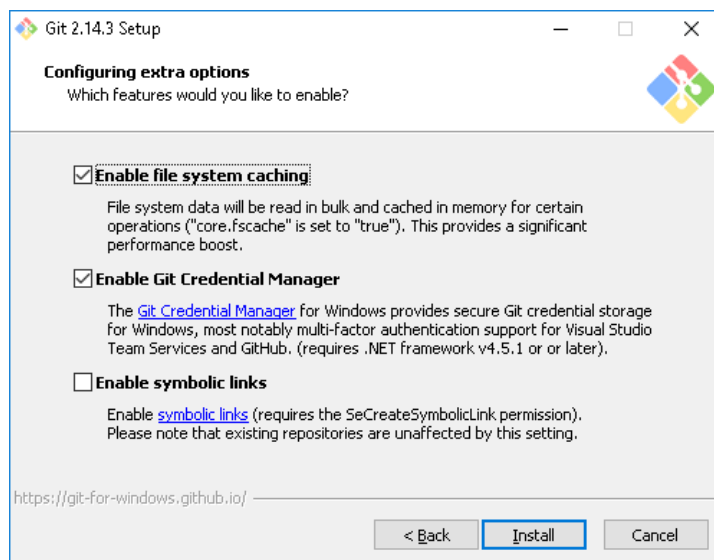


Рис. 3.9. Дополнительные функции

*Подключение к удаленному репозиторию.*

Откроем каталог с файлами, которые необходимо отслеживать в системе контроля версий и выложить на *GitHub*. В пустую часть каталога нажмем правой кнопкой мыши и выберем *Git Bash Here* (рис. 3.10).

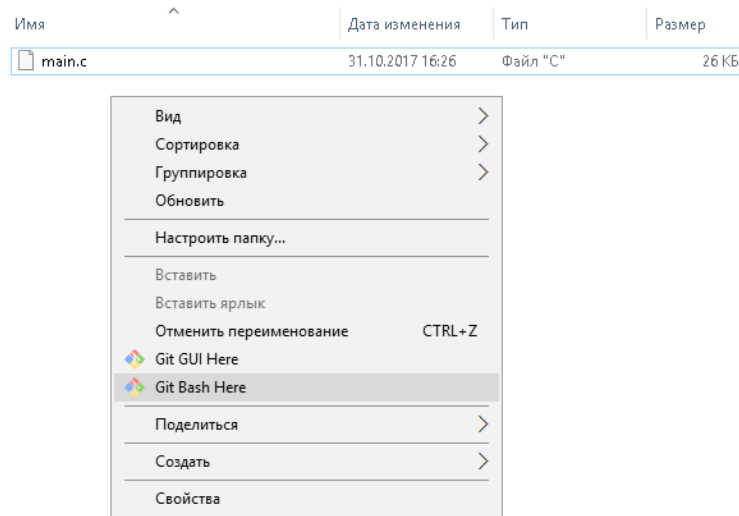


Рис. 3.10. Выборка файлов

Откроется приглашение командной строки в зависимости от настроек (рис. 3.11).

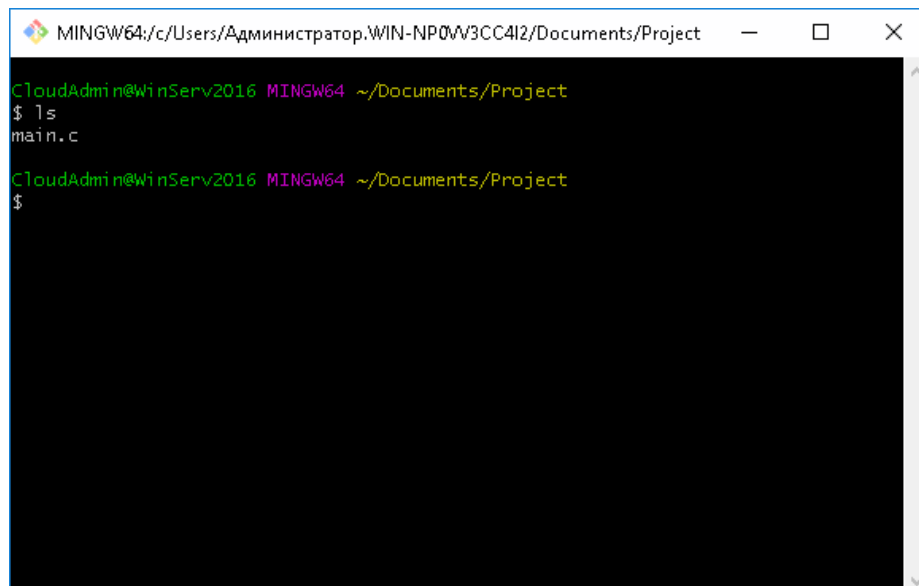


Рис. 3.11. Командная строка под *Windows*, с приглашением

Для настройки необходимо указать имя и электронную почту:

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Ваше имя"
```

Для того чтобы начать отслеживать содержимое папки в системе, выполним команды:

```
git init
```

```
git add
```

Выполним первый коммит:

```
git commit -m "Init"
```

Чтобы добавить изменения, например, на *github* выполним действие:

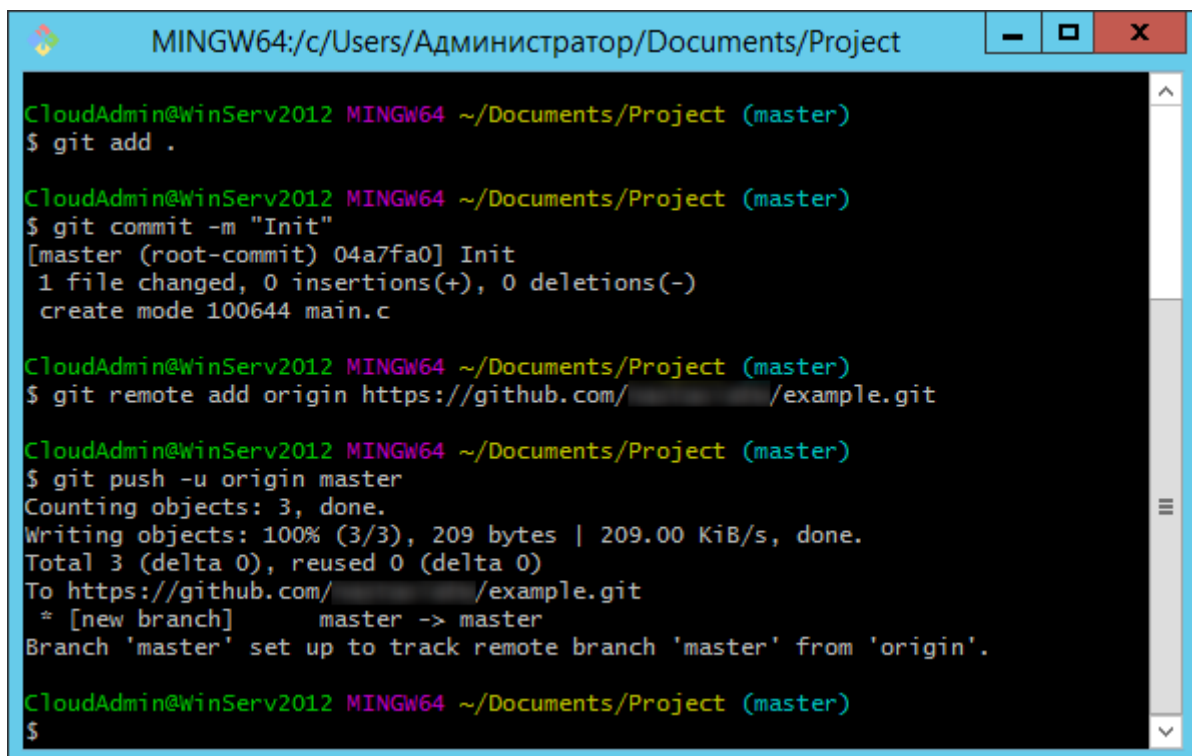
```
git remote add origin https://github.com/пользователь/репозиторий.git
```

```
git push -u origin master
```

На GitLab:

```
git remote add gitlab https://server/namespace/project.git
```

```
git push -u gitlab master (рис. 3.12)
```



```
MINGW64/c/Users/Администратор/Documents/Project

CloudAdmin@WinServ2012 MINGW64 ~/Documents/Project (master)
$ git add .

CloudAdmin@WinServ2012 MINGW64 ~/Documents/Project (master)
$ git commit -m "Init"
[master (root-commit) 04a7fa0] Init
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 main.c

CloudAdmin@WinServ2012 MINGW64 ~/Documents/Project (master)
$ git remote add origin https://github.com/_____/example.git

CloudAdmin@WinServ2012 MINGW64 ~/Documents/Project (master)
$ git push -u origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 209 bytes | 209.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/_____/example.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

CloudAdmin@WinServ2012 MINGW64 ~/Documents/Project (master)
$
```

Рис. 3.12. Ввод команд для подключения к репозиторию

Откроется окно входа (консольное или стандартное окно *Windows*). В качестве пользователя укажем логин на *GitHub*, репозиторий - название существующего репозитория (рис. 3.13).

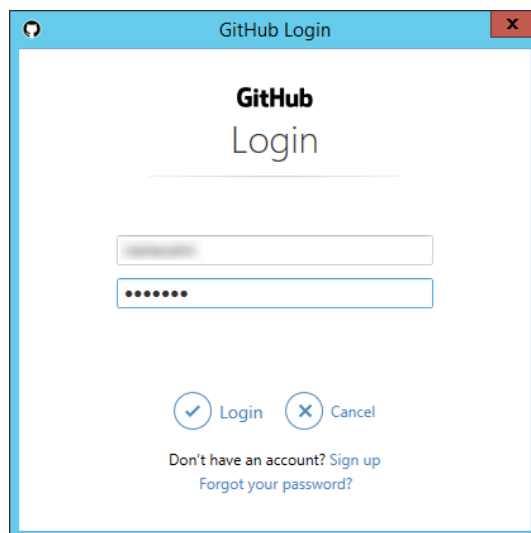


Рис. 3.13. Вход в *GitHub*

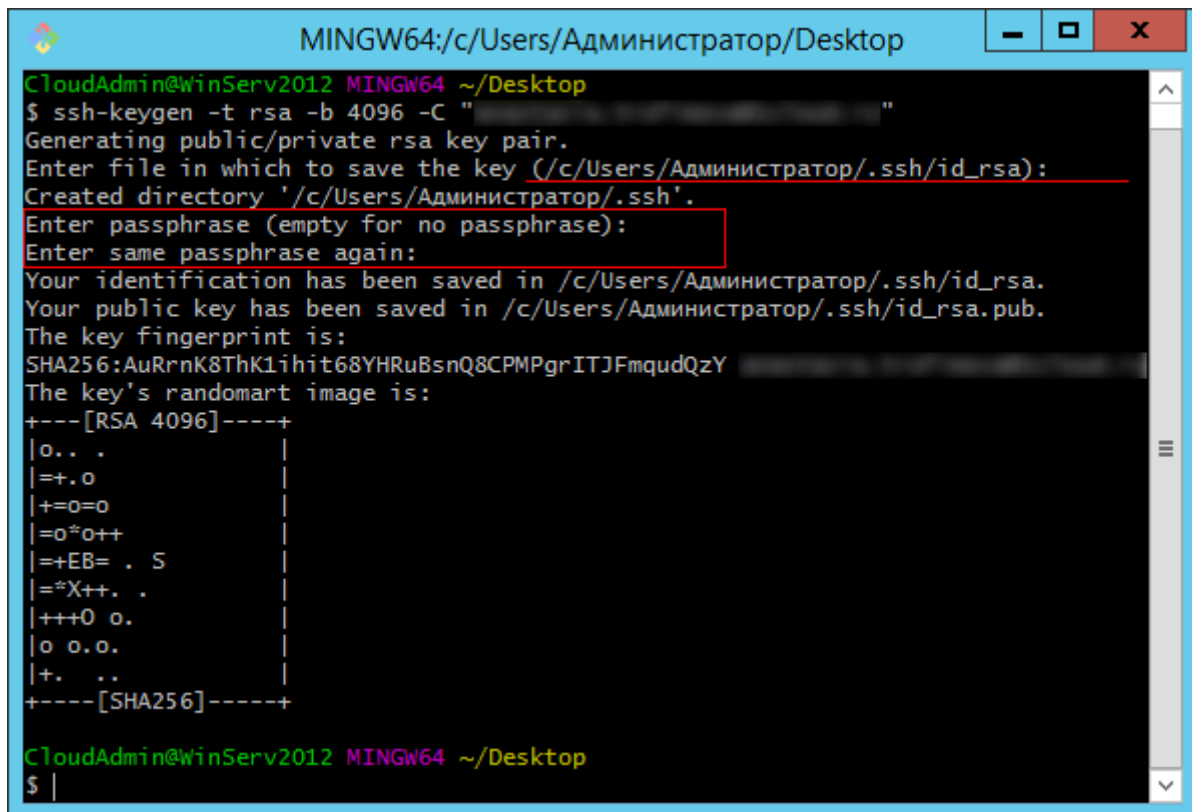
### *Аутентификация по SSH-ключам*

Выполнить подключение к удаленному репозиторию можно по протоколу *SSH* с помощью приватного и публичного ключа. Чтобы сгенерировать пару, выполним команду в консоли *GitBash*, указав реальный почтовый адрес:

```
ssh-keygen -t rsa -b 4096 -C email@example.com
```

В процессе генерации будет предложено указать директорию и имя файла для ключа, примем значение по умолчанию, нажав *Enter*. Далее, при желании можно указать ключевую фразу в качестве дополнительной защиты, при ее отсутствии нажмем *Enter*. В результате будут сгенерированы ключи (рис. 3.14).





```
CloudAdmin@WinServ2012 MINGW64 ~/Desktop
$ ssh-keygen -t rsa -b 4096 -C ""
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Администратор/.ssh/id_rsa):
Created directory '/c/Users/Администратор/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Администратор/.ssh/id_rsa.
Your public key has been saved in /c/Users/Администратор/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:AuRrnK8ThK1ihit68YHRuBsnQ8CPMPgrITJFmqudQzY
The key's randomart image is:
+---[RSA 4096]-----+
|o.. .|
|+=+.o|
|+=o=o|
|=o*o++|
|=+EB= . S|
|=*X++ . .|
|+++0 o.|
|o o.o.|
|+ . .|
+-----[SHA256]-----+
CloudAdmin@WinServ2012 MINGW64 ~/Desktop
$ |
```

Рис. 3.14. Генерация *ssh*-ключей

После генерации ключей необходимо добавить их к самому агенту.

Для этого необходимо проделать следующие действия:

Запустим *ssh*-агент:

```
eval $(ssh-agent -s)
```

Мы увидим идентификатор процесса.

Добавим ключи к *ssh*-агенту:

```
ssh-add ~/.ssh/id_rsa
```

Теперь нужно добавить ключ к удаленному серверу *Git*. Для этого заходим на сервер *Git* и в настройках добавляем *SSH*-ключ.

Например, на сервере GitHub это можно сделать следующим образом: откроем *Settings - SSH and GPG keys*, нажмем *New SSH key*. В поле *Title* введем понятное название, в поле *Key* вставим публичный ключ (содержимое

файла `~/.ssh/id_rsa.pub`). Нажмем *Add SSH key* (рис. 3.15). В результате появится **НОВЫЙ** ключ.

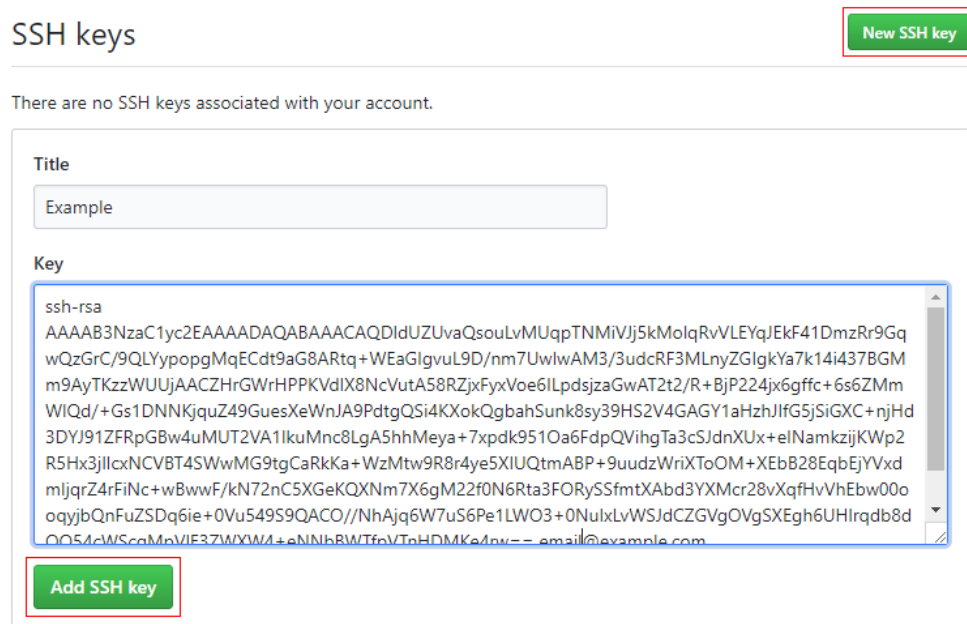


Рис. 3.15. Добавление *ssh*-ключ в *Git*

Для проверки настроек выполним команду:

```
ssh -T git@<домен или ip-адрес сервера git>
```

Например:

```
ssh -T git@github.com
```

Если все настроено верно, то мы увидим следующее сообщение (рис. 3.16).

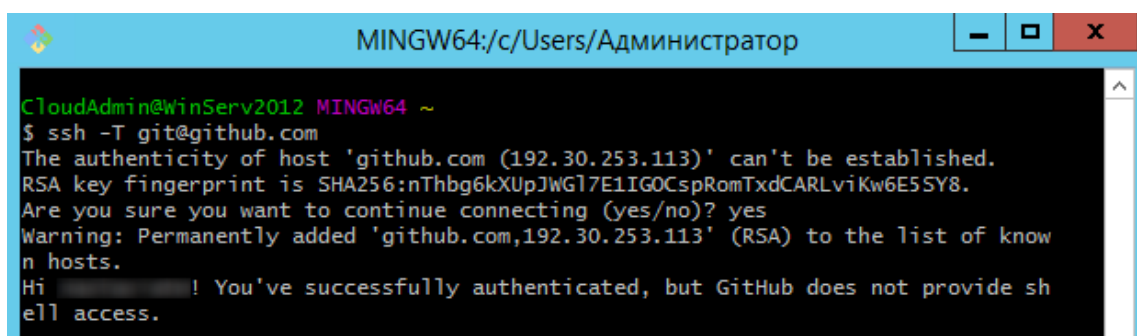


Рис. 3.16. Результат проверки

## **Задание**

Установить и настроить *GIT* на подготовленную операционную систему.

## **Содержание отчета**

1. Титульный лист
2. Цель работы.
3. Формулировка задания.
4. Подробное описание выполненных действий в процессе установки и настройки GIT.
5. Общий вывод о проделанной работе.

## **Лабораторная работа № 4**

### **УСТАНОВКА СИСТЕМЫ УПРАВЛЕНИЯ СОДЕРЖИМЫМ (CMS)**

Время выполнения – 8 часов (аудиторная работа – 4 часа, самостоятельная работа – 4 часа).

**Цель работы:** освоение основных приемов работы с системой управления содержимым (CMS), на примере работы с *CMS Drupal*.

#### **Задачи работы**

1. Получить навыки работы с системой управления содержимым.
2. Изучить основные приемы настройки программного продукта.
3. Научиться использовать системы управления содержимым для работы с *web*-ресурсами.

#### **Перечень обеспечивающих средств**

Задания лабораторной работы выполняются в операционной системе *Ubuntu*.

#### **Общие теоретические сведения**

Система управления содержимым (англ. *Content management system*, *CMS*, система управления контентом) - информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления содержимым, иначе - контентом.

Основные функции *CMS*:

- предоставление инструментов для создания содержимого, организация совместной работы над содержимым;
- управление содержимым: хранение, контроль версий, соблюдение режима доступа, управление потоком документов;
- публикация содержимого;
- представление информации в виде, удобном для навигации, поиска.

В системе управления содержимым могут находиться самые различные данные: документы, фильмы, фотографии, номера телефонов, научные данные и так далее. Такая система часто используется для хранения, управления, пересмотра и публикации документации. Контроль версий является одной из важных возможностей, когда содержимое изменяется группой лиц.

Сейчас современное *web*-программирование сильно отличается от того, что было более 20 лет назад: от статичных файлов мы пришли к микросервисной архитектуре и платформенному вебу (рис. 4.1).

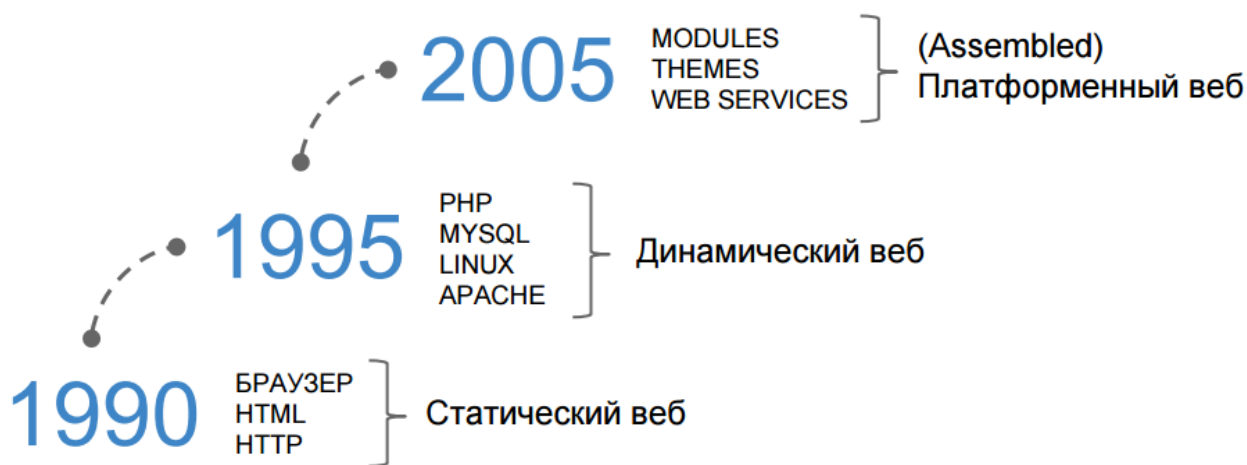


Рис. 4.1. Развитие *web* за 20 лет

За эти годы было изучено и разработано очень много инструментов, и теперь основная цель - научиться ими пользоваться, но не терять времени на то, чтобы переписывать уже имеющееся.

*Drupal* - одна из лучших систем в плане поддержки современных технологий, и в плане обучения, поэтому именно на этой платформе можно строить свою карьеру разработчика.

Основная концепция современного веб-программирования - микросервисная архитектура.

Архитектурный стиль микросервисов - это подход, при котором единое приложение строится как набор небольших сервисов, каждый из которых работает в собственном процессе и коммуницирует с остальными, используя легковесные механизмы, как правило *HTTP*. Эти сервисы построены вокруг бизнес-потребностей и развертываются независимо с использованием полностью автоматизированной среды. Существует абсолютный минимум централизованного управления этими сервисами. Сами по себе эти сервисы могут быть написаны на разных языках и использовать разные технологии хранения данных.

Отдельные сервисы в основном не решают бизнес-задачу целиком. Нужна система, которая будет связывать все отдельные элементы воедино, при этом реализовать свою собственную, уникальную для конкретного случая структуру взаимодействия. В качестве такой системы идеально подходит *Drupal*. На примере *Drupal 7* есть такие интеграции различных сервисов (рис. 4.2).



Рис. 4.2. Интеграция сервисов

Каждая отдельно взятая коммерческая система имеет свои плюсы и минусы, более развитые области, и менее развитые. *Drupal* - *open source* платформа, которая разрабатывается огромным количеством как отдельных разработчиков, так и больших компаний, причем развитие идет сразу во всех направлениях, и это большое преимущество. *Drupal* как *CMS* можно сравнить с *Linux* в области программного обеспечения (рис. 4.3).

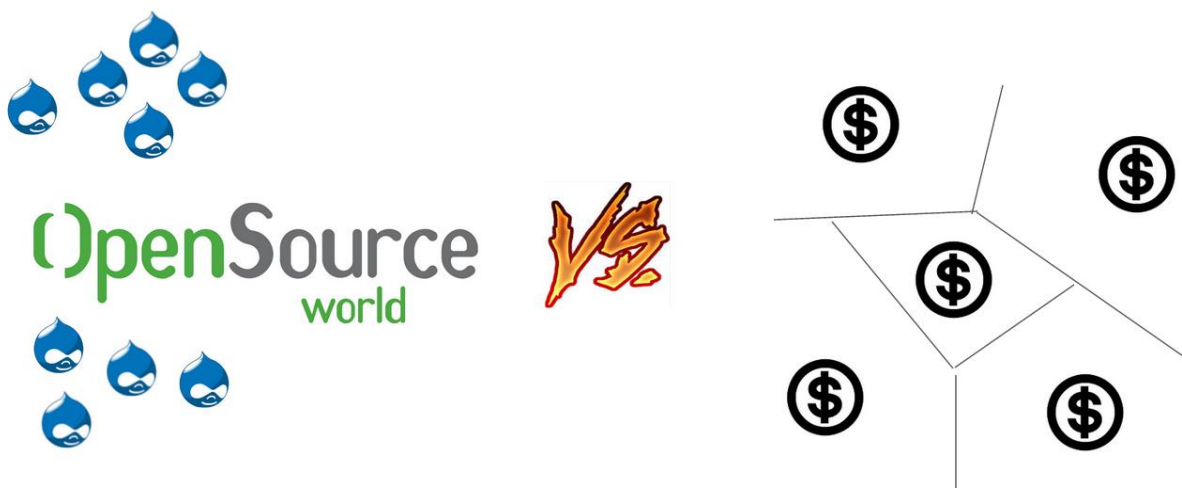


Рис 4.3. Программное обеспечение

*Drupal* состоит из следующих логических частей:

- ядро (Core);
- модули (Modules);
- темы (Theme);
- конфигурирование (Configure);
- контент (Content);
- процесс релиза (Deploy).

*Работа с контентом в Drupal*

Для того, чтобы понять, что такое контент, нужно понимать такие термины как данные, информация и суть.

Данные – что-то сырое, что пришло к нам в необработанном виде, и для нас никакой ценности пока не несет.

Информация - обработанные данные, которые без глобального контекста вероятно не помогут нам.

Суть - информация, которая подкреплена уже имеющимся контекстом, именно суть несет пользу.

Контент - это суть сайта, и это не только текст, но и все то, что используется для решения бизнес задачи.

Контент можно типизировать и структуризировать, это необходимо для более эффективной подачи (рис. 4.4).

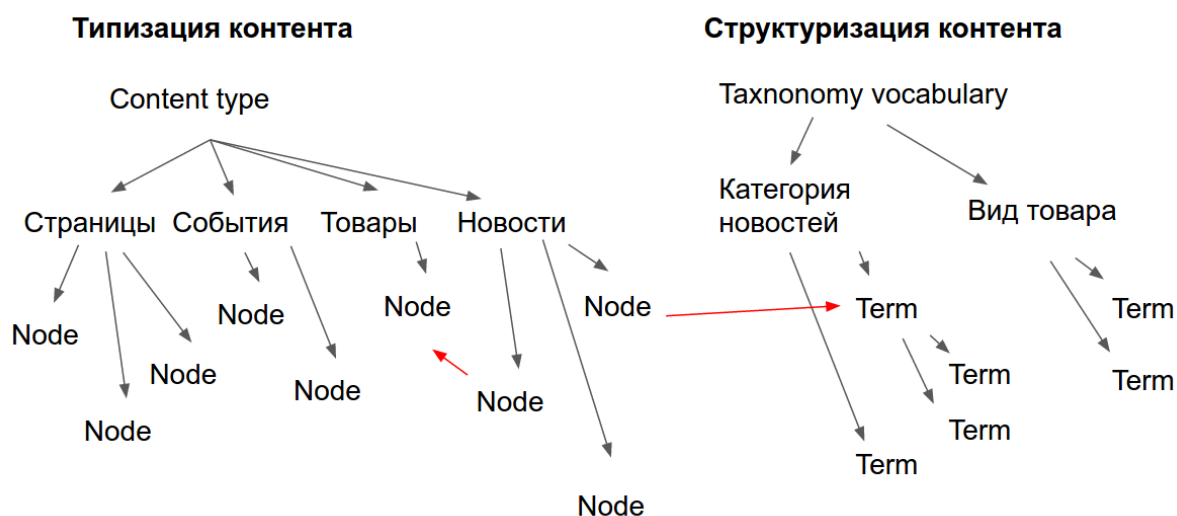


Рис. 4.4. Типизация и Структуризация контента

Типизация обусловлена необходимостью иметь для разнородной информации различный набор полей. Например, для контента типа товар нам нужна цена, но цену не имеет смысла задавать для контента типа новость.

В *Drupal* для типизации контента используется *Content type*.

<https://www.drupal.org/node/774728>

Элемент контента в *Drupal* называется *node*.

Для структуризации контента используется Таксономия.

<http://niklan.net/blog/15>

Таксономия тоже в свою очередь имеет свои типы, они называются словарями (*vocabulary*).



Элемент содержимого таксономии - *term*.

Структуризация осуществляется за счет привязки *node* к *taxonomy term*, как показано красной стрелкой справа.

Также есть возможность привязывать *node* между собой.

Для самопроверки рекомендуется сделать следующую структуру контента (рис. 4.5).

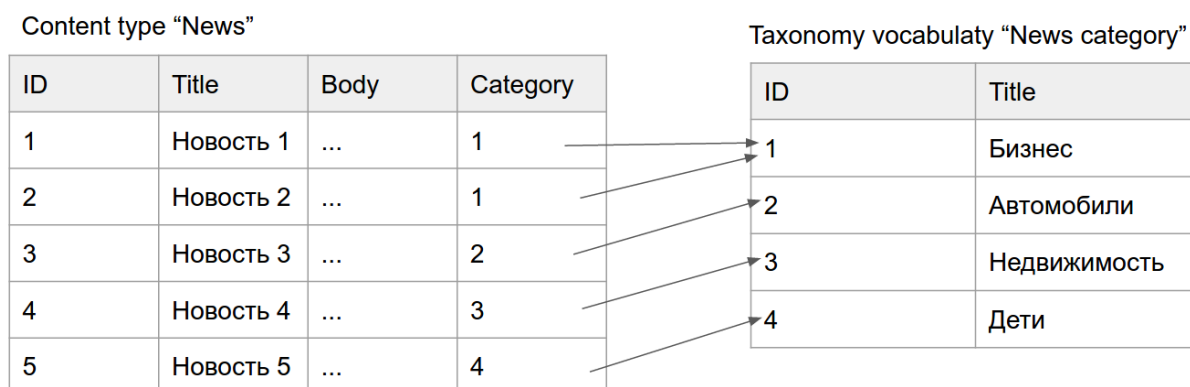


Рис. 4.5. Структура контента

На изображении схематично изображено как контент хранится в БД. Для организации такой структуры вам необходимо сделать поле типа *taxonomy term reference* у *content type* Новостей.

Еще один важный инструмент по работе с контентом – ревизионность (рис. 4.6).

ID	Title	Body	Category	Revision ID
1	Новость 1	...	3	
2	Новость 2	...	1	
3	Новость 3	...	2	
4	Новость 4	...	3	
5	Новость 5	...	4	

#### Ревизионность

- Поиск истории
- Откат изменений
- Черновики

ID	Entity ID	Title	Body	Category	Status
1	1	Новость 1	...	1	0
2	1	Новость 1	...	3	1
3	2	Новость 2	...	1	1

Рис. 4.6. Ревизионность

На рисунке видно, как *Drupal* хранит данные всех версий контента.

Ревизионность позволяет производить поиск по истории изменений контента, откатываться на более удачные версии, создавать черновики.

Ревизионность - мощный инструмент интернет маркетинга.

### *Установка Drupal*

*Drupal* можно поставить как скачав архив с сайта <https://www.drupal.org/project/drupal>, разместив и распаковав его в корень сайта на сервере, так и воспользоваться готовыми командами специальной утилиты для консоли *Linux* - *drush*.

### *Базовые команды drush*

*drush dl* - установка *Drupal 7 stable version*.

*drush dl* [имя модуля] - установка модуля [https://www.drupal.org/project/\[имя модуля\]](https://www.drupal.org/project/[имя модуля]).

*drush en* [имя модуля] - включение модуля [имя модуля].

*drush dis* [имя модуля] - выключение модуля [имя модуля].

*drush cc* [имя кеша или *all* для всего] - чистка кеша.

*drush sql-cli* - переход в режим работы с БД

*drush eval* “код” - запуск *php*-кода из консоли.

Контрибные модули: */sites/all/modules/contrib/имя модуля*

Кастомные модули: */sites/all/modules/custom/имя модуля*

Кастомные темы: */sites/all/themes/имя темы*

Файлы: */sites/default/files*

Настройки для всех сайтов: */sites/default/settings.php*

Для того, чтобы было удобно работать с чистым *Drupal* рекомендуется установить и включить следующие модули:

– *admin\_menu*

- *adminimal\_admin\_menu*
- *ckeditor*
- *ctools*
- *devel*
- *elysia\_cron*
- *features*
- *panels*
- *panels\_everywhere*
- *pathauto*
- *revisioning*
- *rules*
- *token*
- *views*
- *module\_filter*

Также стоит отключить модуль *overlay*, чтобы не работать на сайте в административном разделе во всплывающих окнах, это не удобно и лишает вас полной кастомизации административного раздела сайта.

После того, как файлы *Drupal* лежат в корне сайта, заходим по *url* сайта и продолжаем процесс установки (рис. 4.7).

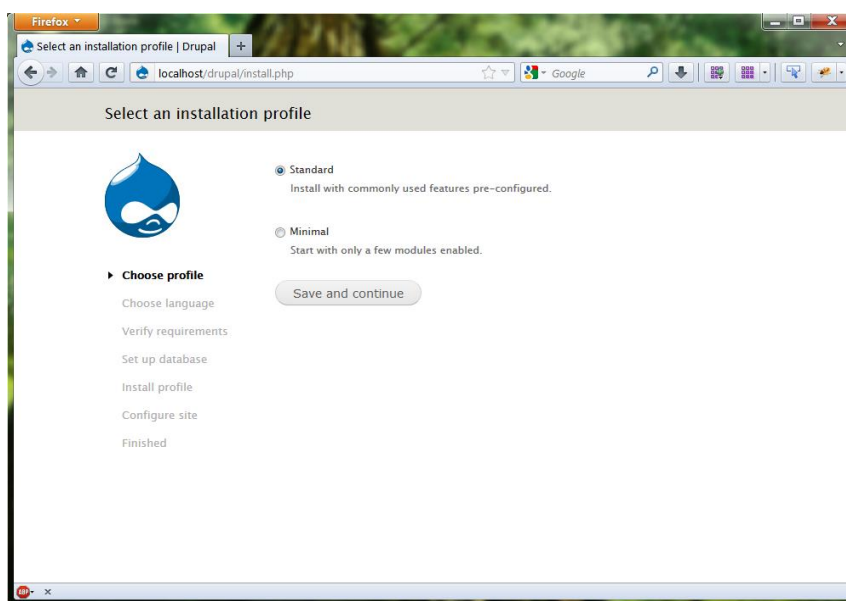


Рис. 4.7. Выбор профиля доступа

На этом этапе нужно прописать данные для вашей БД (рис. 4.8).

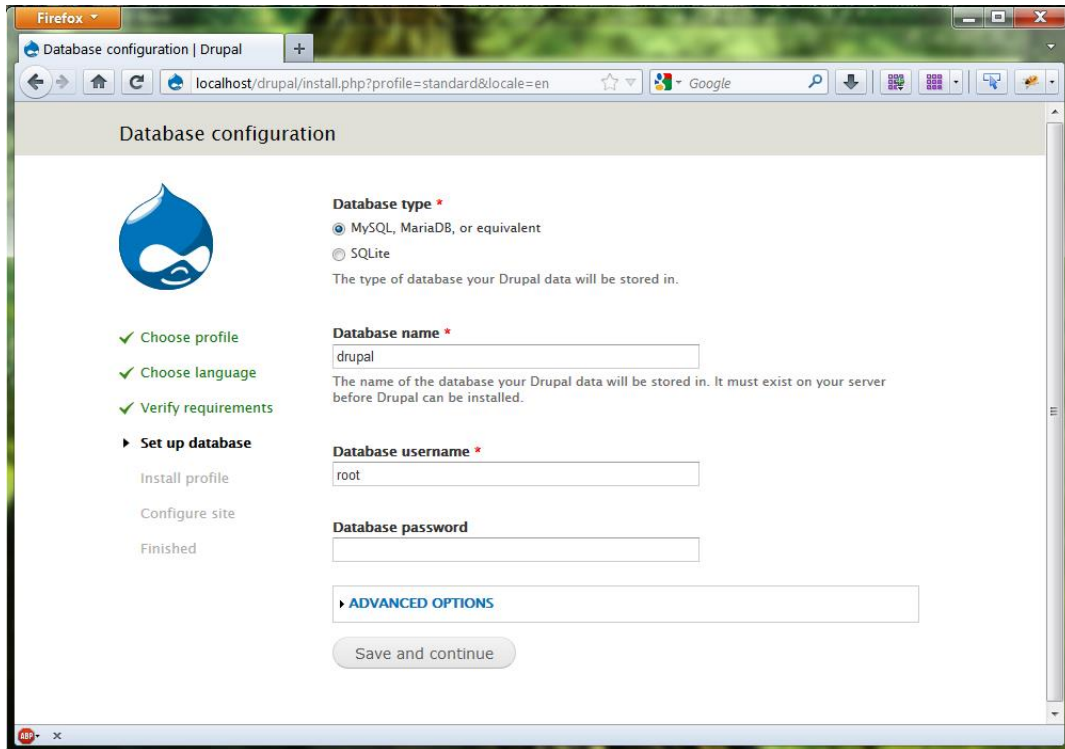


Рис. 4.8. Настройка конфигурации БД

Эти данные в дальнейшем будут записаны в *sites/default/settings.php* в виде:

```
$databases = array (  
  'default' =>  
    array (  
      'default' =>  
        array (  
          'database' => 'ИМЯ БД',  
          'username' => 'ЛОГИН',  
          'password' => 'ПАРОЛЬ',  
          'host' => 'localhost',  
          'port' => '',  
          'driver' => 'mysql',  
          'prefix' => '',  
        ),  
      ),  
    ),  
  ),  
);
```

),  
);

### Задание

1. Установить *Drupal*.
2. В директорию: *sites/all/modules/contrib*, если такой нет, необходимо создать, после чего скачать и распаковать в *contrib* следующие модули: *admin\_menu*, *adminimal\_admin\_menu*, *ckeditor*, *ctools*, *devel*, *elysia\_cron*, *features*, *panels*, *panels\_everywhere*, *pathauto*, *revisioning*, *rules*, *token*, *views*, *module\_filter*.
3. Открыть на *student.loc* страницу *Modules* и выключить *Overlay*, включить *module\_filter*.

### Содержание отчета

1. Титульный лист
2. Цель работы.
3. Формулировка задания.
4. Подробное описание выполненных действий в процессе выполнения задания.
5. Общий вывод о проделанной работе.

## Лабораторная работа № 5

### МОДУЛЬ VIEWS

Время выполнения – 8 часов (аудиторная работа – 4 часа, самостоятельная работа – 4 часа).

**Цель работы:** освоение основных приемов работы с основными модулями системой управления содержимым *Drupal*.

#### Задачи работы

1. Получить навыки работы с модульной системой.
2. Изучить основные приемы персонализации программного продукта.
3. Научиться использовать модуль views для работы с *web*-ресурсами.

#### Перечень обеспечивающих средств

Задания лабораторной работы выполняются в *CMS Drupal*.

#### Общие теоретические сведения

Основные понятия: представление, вид отображения, тема, теминг, меню, административная тема, дашборд.

Для чего на сайтах делают несколько вариантов отображения для одного и того же контента:

- конверсия;
- вовлечение;
- User-experience;
- программирование восприятия.

На примере сайта *ngs.ru* считаем количество вариантов отображения одной и той же сущности *Новость* с точки зрения программиста (рис. 5.1а, 5.1б)

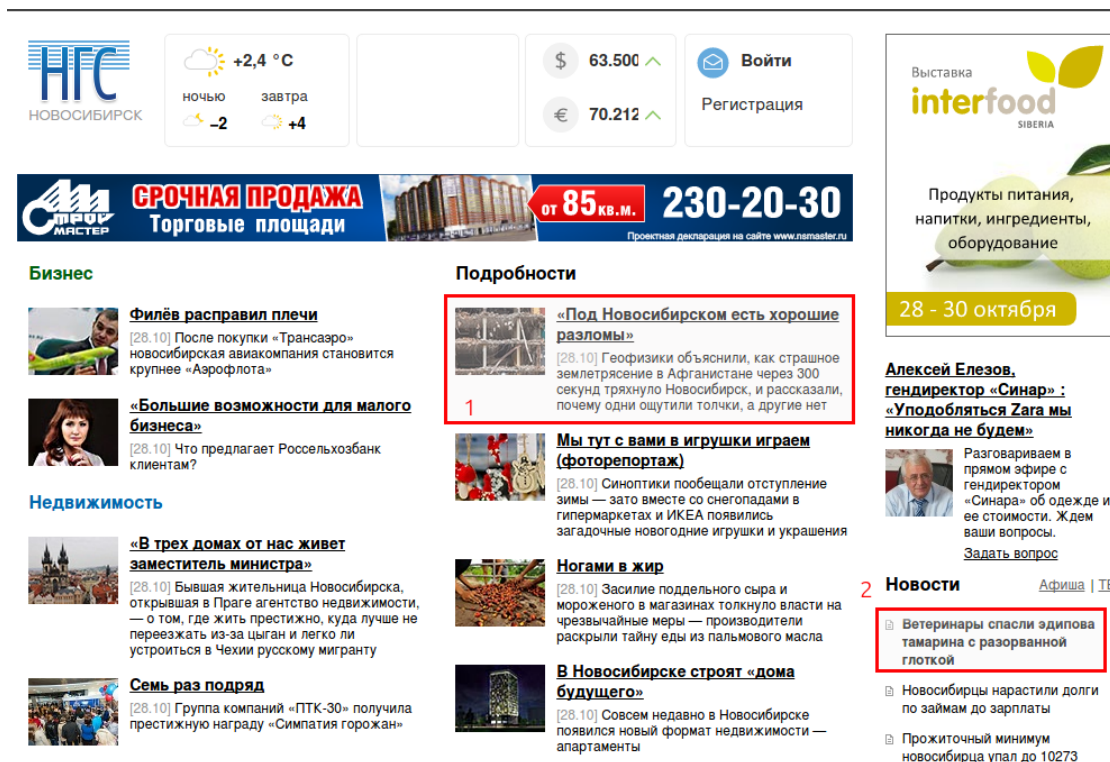


Рис. 5.1 а. Варианты отображения сущности «Новость»

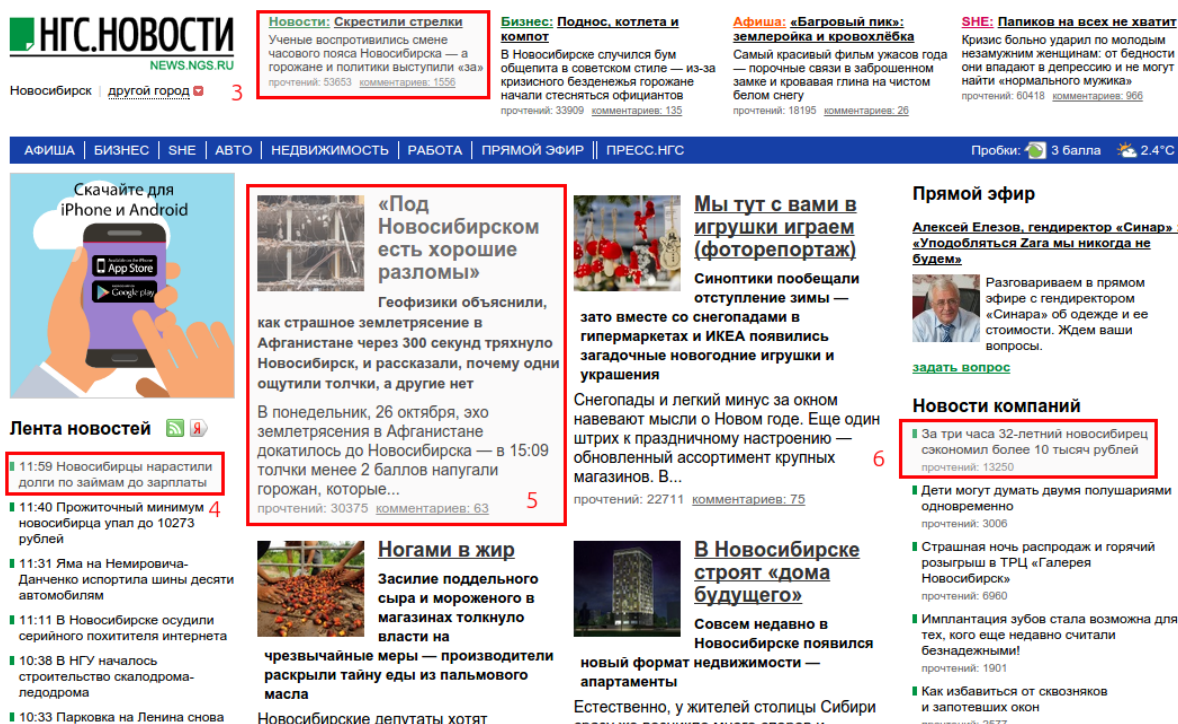


Рис. 5.1 б. Варианты отображения сущности «Новость»

+ страница самой новости и полное описание, итого **7**.

В *Drupal* есть уже готовые инструменты: модуль *views* и *view modes* (рис. 5.2).

#### Представления (*views*) и варианты отображения (*view mode*)

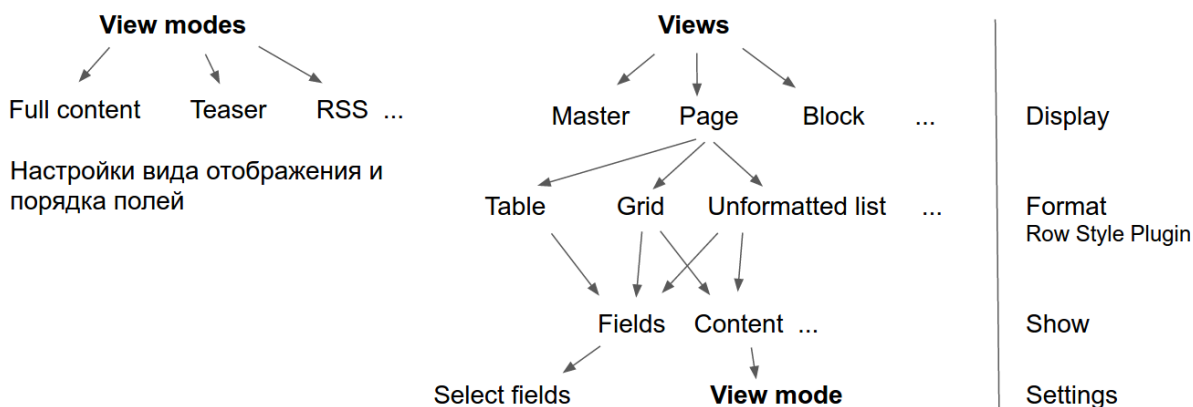


Рис. 5.2. Модули *View* и *View modes*

*View mode* - это характеристика *content type*. По сути *view mode* - это уже готовый *HTML node*. Задается для всего типа контента в целом (рис. 5.3).

Пример *view mode*, которые уже есть в *Drupal* по умолчанию: *Teaser* (краткое описание), *Full* (полное описание), *RSS*- тиль вывода для *RSS*-ленты.

*Views* - модуль для построения и вывода списка содержимого.

<https://www.drupal.org/project/views>

*Display* - где будет отображаться наш список (отображаться страницей по конкретному *url*, будет встроен куда-то как блок).

*Format* - как выводить эту информацию. Таблица - каждая запись выводится в одной строке таблицы, а столбцы - это характеристики контента. Сетка - это таблица, каждая ячейка которой является единицей контента (например, *node*). Неформатированный список - по сути Сетка с одним столбцом и множеством строк.

*Show* - что показывать (либо выбранный набор полей, либо уже подготовленный *view mode*).



*Settings* - настройки, которые зависят от выбранных ранее вариантов.

The screenshot shows the Drupal administration interface for the Views module. The breadcrumb trail is 'Home » Administration » Structure » Views'. The 'Views' tab is active, showing a list of existing views. At the top, there are links to 'Add new view', 'Add view from template', and 'Import'. The table below lists five default views: Archive, Backlinks, Front page, Glossary, and Recent comments. Each row includes the view name, its description, the tag (all are 'default'), the path, and an 'Enable' button.

VIEW NAME	DESCRIPTION	TAG	PATH	OPERATIONS
Archive Displays: Block, Page In code Type: Content	Display a list of months that link to content for that month.	default	/archive	Enable
Backlinks Displays: Block, Page In code Type: Content	Displays a list of nodes that link to the node, using the search backlinks table.	default	/node/%/backlinks	Enable
Front page Displays: Feed, Page In code Type: Content	Emulates the default Drupal front page; you may set the default home page path to this view to make it your front page.	default	/frontpage, /rss.xml	Enable
Glossary Display: Page In code Type: Content	A list of all content, by letter.	default	/glossary	Enable
Recent comments Displays: Block, Page In code Type: Comment	Contains a block and a page to list recent comments; the block will automatically link to the page, which displays the comment body as well as a link to the node.	default	/comments/recent	Enable

Рис. 5.3. Главная страница модуля *views*: */admin/structure/views*

Создание *views* (рис. 5.4, 5.5).

The screenshot shows the 'Add new view' form in the Drupal administration interface. The breadcrumb trail is 'Home » Administration » Structure » Views'. The 'Add new view' tab is active. The form includes a 'View name' field, a 'Description' checkbox, and a 'Show' dropdown set to 'Content'. Below this, there are filters for 'of type' (set to 'All'), 'tagged with', and 'sorted by' (set to 'Newest first'). The 'Create a page' section is checked and contains fields for 'Page title', 'Path' (pre-filled with 'http://drupal\_test.loc/'), 'Display format' (set to 'Unformatted list'), 'Items to display' (set to '10'), and checkboxes for 'Use a pager', 'Create a menu link', and 'Include an RSS feed'. The 'Create a block' section is unchecked. At the bottom, there are buttons for 'Save & exit', 'Continue & edit', and 'Cancel'.

Рис. 5.4. Создание *Views*

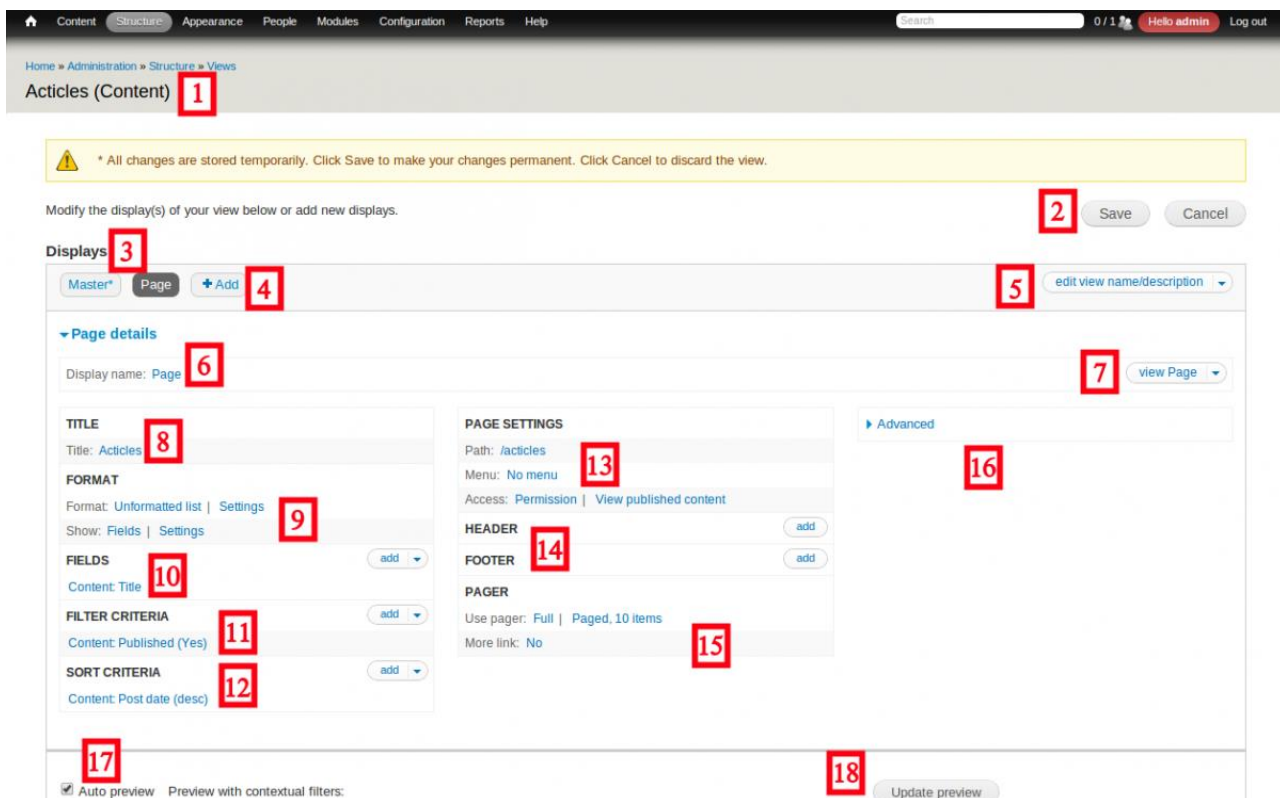


Рис. 5.5. Схема настройки *views*

1. Заголовок
2. Кнопки сохранения и отмены
3. Дисплеи
4. Создать дисплей
5. Опции (общие операции)
6. Название дисплея
7. Опции для дисплея
8. Заголовок страницы
9. Настройки формата
10. Поля
11. Критерии фильтрации
12. Критерии сортировки
13. Настройки страницы
14. Вставка Header и Footer
15. Настройка постраничной навигации

16. Расширенные настройки
17. Автоматический предпросмотр результатов
18. Обновить предпросмотр

### **Задание**

Настроить отображение содержимого контента, созданного на прошлой лабораторной работе, используя модуль Views.

Для этого понадобится модуль Views.

### **Содержание отчета**

1. Титульный лист
2. Цель работы.
3. Формулировка задания.
4. Подробное описание выполненных действий в процессе работы над заданием.
5. Общий вывод о проделанной работе.

## **Лабораторная работа № 6**

### **МОДУЛИ SLACK, TRELLO, ADDTHIS, DGIS\_MAPS, SIMPLENEWS, LOGINZA**

Время выполнения – 8 часов (аудиторная работа – 4 часа, самостоятельная работа – 4 часа).

**Цель работы:** освоение основных приемов работы с основными модулями системой управления содержимым *Drupal*.

#### **Задачи работы**

1. Получить навыки работы с модулями CMS Drupal для использования сервисов slack, trello, addthis, dgis\_maps, simplenews, loginza.
2. Изучить основные приемы настройки программного продукта.
3. Научиться использовать модули CMS Drupal для работы сервисов slack, trello, addthis, dgis\_maps, simplenews, loginza.

#### **Перечень обеспечивающих средств**

Задания лабораторной работы выполняются в *CMS Drupal*.

#### **Общие теоретические сведения**

Slack - корпоративный мессенджер. Запущен в тестовом режиме в августе 2013 года, публичный релиз состоялся 12 февраля 2014 года. В первый день тестирования зарегистрировались 8 тысяч компаний. По данным компании на июнь 2015 Slack ежедневно используют 1,1 миллиона пользователей.

Бесплатный аккаунт slack дает возможности (применительно для системы умного дома):

- 1) хранение архива 10000 сообщений с вложениями;
- 2) возможность отправки текстовых сообщений, изображений, ссылок, кнопок и много других возможностей;
- 3) различные клиенты для всех платформ;
- 4) нет необходимости использовать VPN (спасибо РКН);
- 5) возможность отправлять сообщения в системный чат (на примере имеющего модуля телеграм) - в планах;
- 6) широчайшие возможности интеграции (хороший api).

Что реализовано модулем в настоящий момент:

- 1) подписка на системный чат в зависимости от требуемого приоритета (меню настройки);

- 2) отправка текстовых сообщений через webhook:

```
include_once(DIR_MODULES . 'slack/slack.class.php');  
$slack_module = new slack();  
$slack_module->sendMessageToAll ("test message to all");
```

- 2) отправка сообщений с вложением (картинка) через webhook. Картинка должна быть доступна по внешней ссылке (из локальной ссылки почему-то не работает).

```
include_once(DIR_MODULES . 'slack/slack.class.php');  
$slack_module = new slack();  
$slack_module->sendImageToAll  
("http://192.168.1.xx/img.jpg", "test image");
```

Для отправки снапшотов с камеры, пока приходится их выкладывать во временную папку по внешней ссылке.

Настройка:

Отправка сообщений через webhook:

а) Регистрируемся <https://api.slack.com/>

б) Создаем новое приложение <https://api.slack.com/apps/new AppName> - название приложения.

*Short description* - краткое описание приложения.

*Describe what your app does on Slack* - полное описание приложения.

*Link to clear instructions on how to install your Slack app.*

*Link to support for your Slack app* - две ссылки на страницы с описанием установки и использования данного приложения.

в) Получаем ссылку на *Webhook*. Для этого на вкладке *Basic Information* выбираем *Add features and functionality*, далее *Incoming Webhooks*, и включаем *Activate Incoming Webhooks*.

Ссылка *Webhook* имеет формат: [https://hooks.slack.com/services/T000000 ... XXXXXXXXXXXX](https://hooks.slack.com/services/T000000...XXXXXXXXXX)

Эту ссылку добавляем на странице *Настройки* модуля.

г) настраиваем необходимый приоритет отправляемого системного чата.

Меню настройки (рис. 6.1, 6.2).

APIURL: ⓘ

Значение по умолчанию: https://hooks.slack.com/services/xxxx/

Enable: ⓘ

☒ Yes☐ No

Значение по умолчанию: Yes

MSGLEVEL: ⓘ

Значение по умолчанию: 2

Legacy token: ⓘ

Значение по умолчанию: хохр-xxxxxxxx-xxxxxxxxxxxx-xxxxxxxx-xxxxxxxx

Рис. 6.1. Меню настройки *Slack*

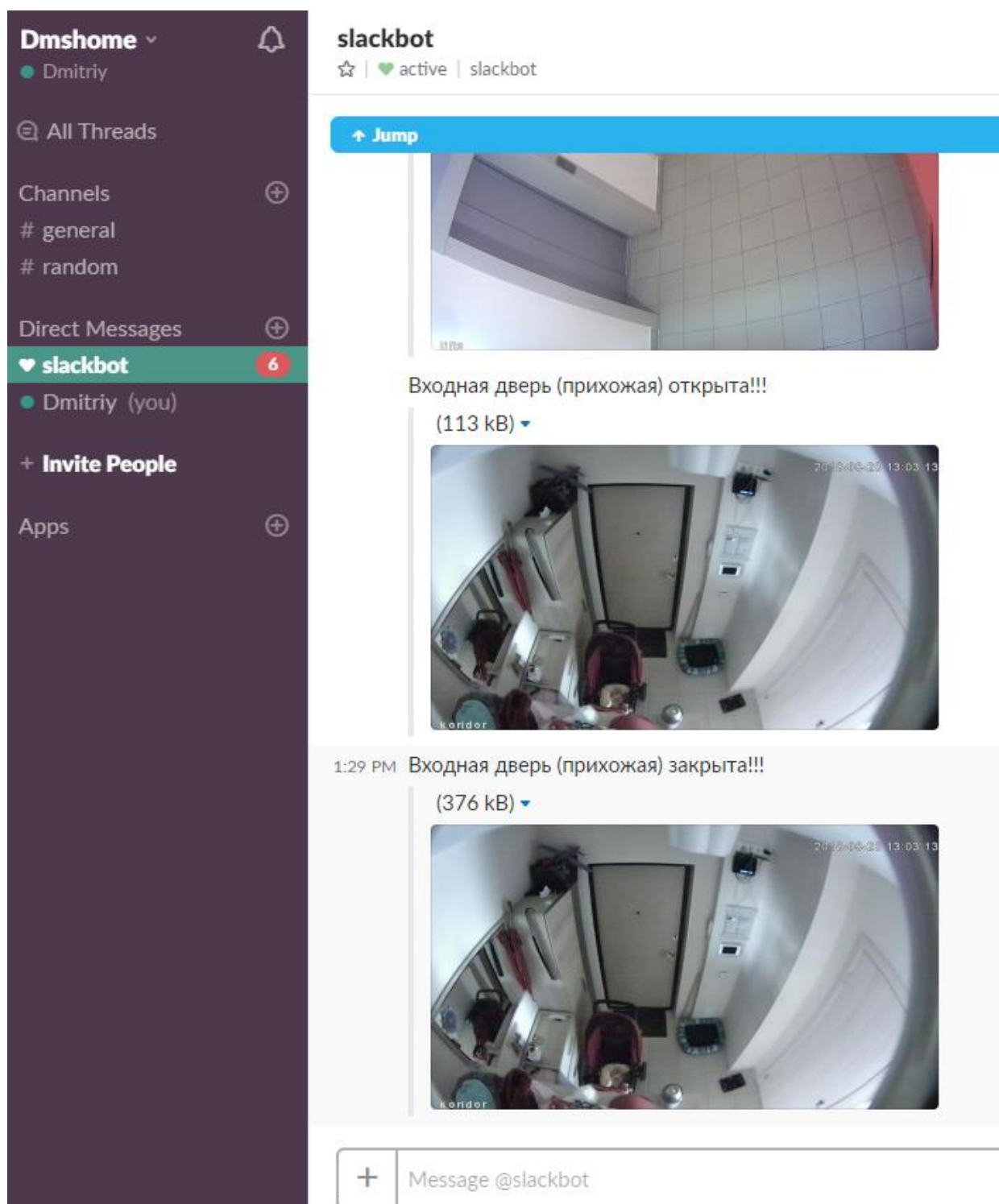


Рис. 6.2. Чат с программой

*Trello* - облачная программа для управления проектами небольших групп, разработанная *Fog Creek Software* (рис. 6.3).



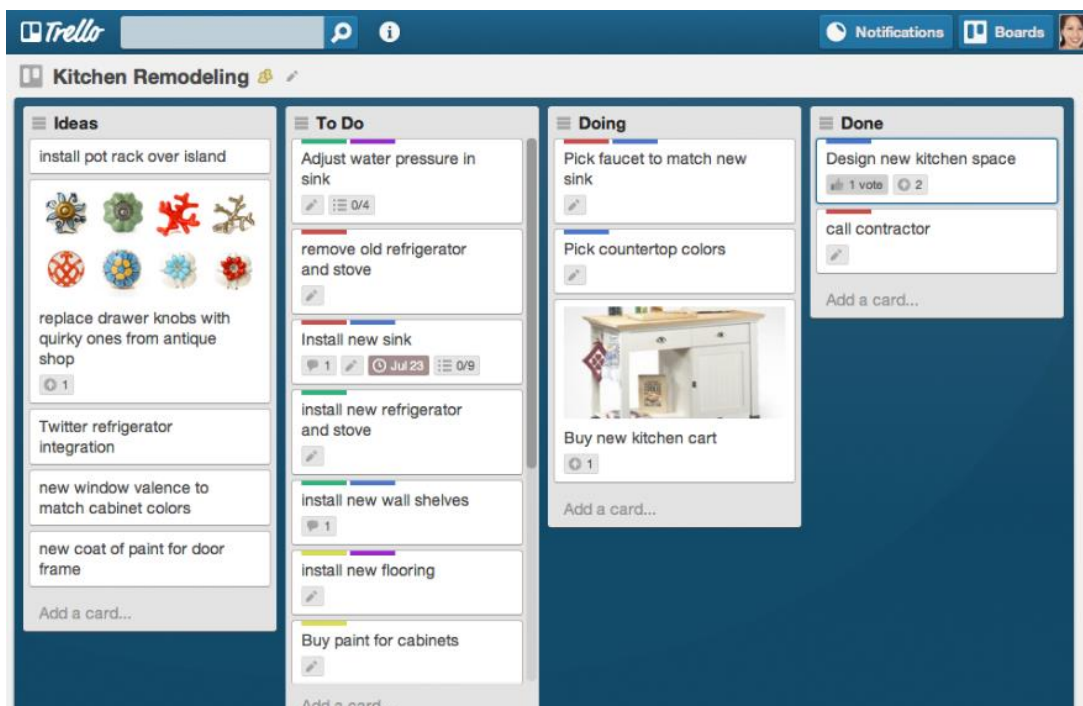


Рис. 6.3. Модуль управления проектами *Trello*

*AddThis.com* представляет собой набор инструментов, позволяющий пользователям делиться контентом с сайтами социальных сетей, используя *AddThis*.

*AddThis* запущен в сентябре 2006 года, является модулем № 1 для создания закладок и публикации в Интернете. *AddThis* распространяет контент по сети, облегчая вашим посетителям возможность делать закладки и делиться ими с другими людьми. Эта простой, но мощный модуль прост в установке и предоставляет ценную аналитику о закладках и активности пользователей.

Обратите внимание, что этот модуль поддерживает только часть функций, предоставляемых *AddThis*.

Модуль *2gis maps* добавляет свой тип поля для работы с картой 2ГИС, через который можно добавлять карты в объекты. Кроме того, он добавит блок, в котором показывается карта.

Используется *2GIS Maps API* и для каждой карты можно определить центр и масштаб, добавлять и удалять маркеры (рис. 6.4).

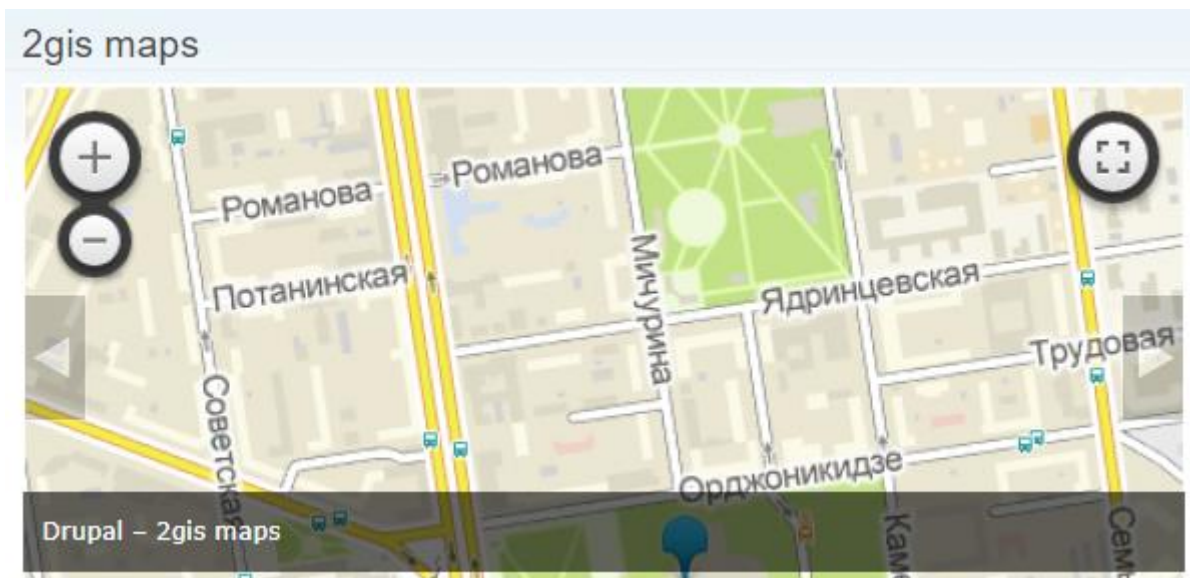


Рис. 6.4. Модуль *2gis maps*

Модуль *Simplenews* публикует и рассылает информационные бюллетени по спискам подписчиков.

Информационные письма можно рассылать как анонимным, так и аутентифицированным пользователям в вашей системе. Письмо в формате *HTML* можно отправить с помощью модуля *Mime Mail* (рис. 6.5).

Home » Administration » Content

**SHOW ONLY NEWSLETTERS WHICH**

Subscribed to:

**UPDATE OPTIONS**

<input type="checkbox"/>	TITLE	NEWSLETTER CATEGORY	CREATED	PUBLISHED	SENT	SUBSCRIBERS	OPERATIONS
<input type="checkbox"/>	<a href="#">French Newsletter</a>	Drupal 7 newsletter	Fri, 01/06/2012 - 16:04		Not yet sent	35	<a href="#">edit</a>
<input type="checkbox"/>	<a href="#">English Newsletter</a>	Drupal 7 newsletter	Fri, 01/06/2012 - 16:03		Not yet sent	35	<a href="#">edit</a>
<input type="checkbox"/>	<a href="#">Newsletter</a>	Drupal 7 newsletter	Wed, 01/04/2012 - 12:04		Not yet sent	35	<a href="#">edit</a>
<input type="checkbox"/>	<a href="#">Newsletter</a>	Drupal 7 newsletter	Wed, 01/04/2012 - 12:04		Not yet sent	35	<a href="#">edit</a>
<input type="checkbox"/>	<a href="#">Another newsletter</a>	Drupal 7 newsletter	Thu, 12/15/2011 - 17:20		Not yet sent	35	<a href="#">edit</a>
<input type="checkbox"/>	<a href="#">New product</a>	Drupal 7 newsletter	Wed, 12/14/2011 - 15:09			35	<a href="#">edit</a>
<input type="checkbox"/>	<a href="#">New test</a>	Drupal 7 newsletter	Wed, 12/14/2011 - 14:52			35	<a href="#">edit</a>
<input type="checkbox"/>	<a href="#">really unpublished</a>	Test	Thu, 12/08/2011 - 14:21			0	<a href="#">edit</a>

Рис. 6.5. Модуль информационной рассылки *simplenews*

*Loginza* – это система идентификации, обеспечивающая единый доступ к популярным *web*-сервисам, которая представляет собой интерактивный виджет на *JavaScript*.

Данная система предоставляет широкий список вариантов аутентификации через учетные записи распространенных *web*-порталов и сервисов, таких Яндекс, *Google* и многие другие (рис. 6.6).

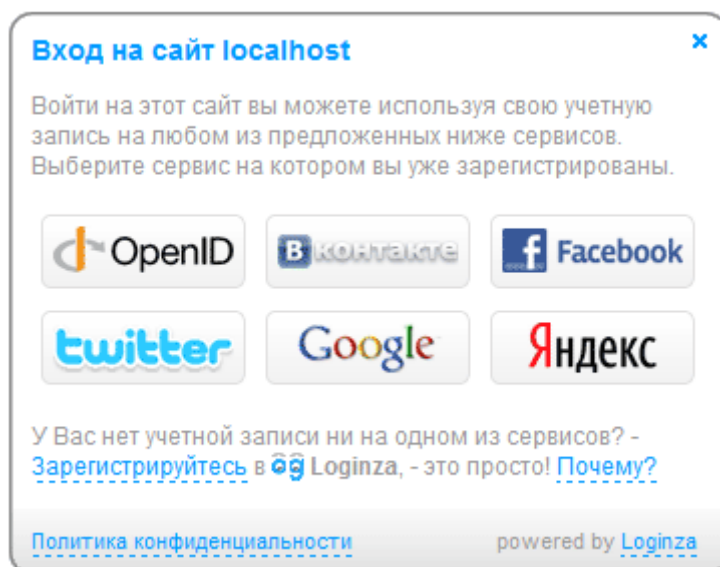


Рис. 6.6. Виджет *Loginza*

### Задание

Настроить имя сайта: *Drupal First Test*. Слоган: *FirstD*.

Создать словарь таксономии *News Category*.

1. Создать контент тайп *News* с филдом *category* и привязкой к терму из задачи 1. Комментарии включены.
2. Настроить вывод категории новости на детальной странице новости.
3. Сделать красивые *URL* для новостей вида *news/title*.
4. Зарегистрироваться в *slack* и сделать пересылку новостей в приватный чат.
5. Зарегистрироваться на *trello*, сделать пересылку новости в столбец *New news*.

6. Добавить возможность делиться новостью в социальных сетях, используя модуль *addthis*.

7. Добавить возможность показывать на новости метки на карте *2gis*, используя модуль *dgis\_maps*. Поле не обязательное вывод только на детальном описании (*full view mode*).

8. Сделать подписку *News*, зарегистрировать пользователя с *email drupaltest@yopmail.com*, подписать его и отправить рассылку.

9. Сделать авторизацию через соцсети, используя модуль *loginza*, авторизоваться.

Для этого понадобятся следующие модули: *slack*, *trello*, *addthis*, *dgis\_maps*, *simplenews*, *loginza*.

### Содержание отчета

1. Титульный лист
2. Цель работы.
3. Формулировка задания.
4. Подробное описание выполненных действий в процессе выполнения задания.
5. Общий вывод о проделанной работе.

## **БИБЛИОГРАФИЧЕСКИЙ СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ**

1. Архитектурные решения информационных систем : учебник / А.И. Водяхо, Л.С. Выговский, В.А. Дубенецкий, В.В. Цехановский. — 2-е изд., перераб. — Санкт-Петербург : Лань, 2017. — 356 с. — Режим доступа: <http://e.lanbook.com>. —Загл. с экрана.

2. Гуриков, С.Р. Интернет-технологии [Электронный ресурс]: учеб. пособие / С.Р. Гуриков. — М. : ФОРУМ : ИНФРА-М, 2017. — 184 с. — Режим доступа: <http://znanium.com>.

3. Информационные технологии [Электронный ресурс]: Учебное пособие / Л.Г. Гагарина, Я.О. Теплова, Е.Л. Румянцева и др.; Под ред. Л.Г. Гагариной - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2015. - 320 с. - Режим доступа: <http://znanium.com>.

4. Информационные технологии [Электронный ресурс]: Учебное пособие / Л.Г. Гагарина, Я.О. Теплова, Е.Л. Румянцева и др.; Под ред. Л.Г. Гагариной - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2015. - 320 с. - Режим доступа: <http://znanium.com>.

5. Кацко, С. Ю. Мультимедийные технологии [Электронный ресурс] : учеб. пособие / С. Ю. Кацко. - Новосибирск : СГУГиТ, 2015. - 139 с. - Режим доступа: <http://lib.sgugit.ru>.

6. Ли, П. Архитектура интернета вещей / П. Ли ; перевод с английского М.А. Райтман. — Москва : ДМК Пресс, 2019. — 454 с. — Режим доступа: <http://e.lanbook.com>. —Загл. с экрана.