Круглов Вячеслав

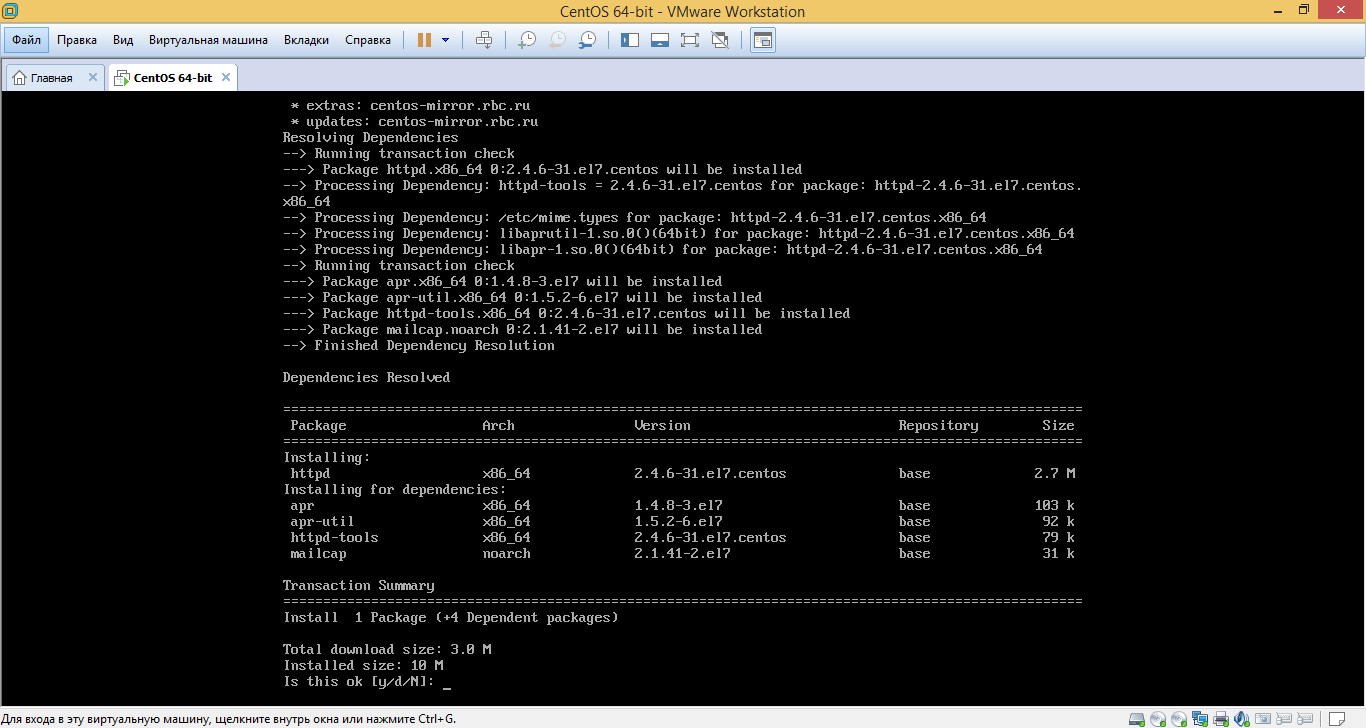
**Часть 1**

Нам необходимо установить на CentOS веб-сервер Apache, систему управления БД MySQL и интерпретатор языка PHP. В совокупности это называется аббревиатурой LAMP (Linux + Apache + MySQL + PHP).

Первый пункт (Linux) у нас уже есть. Следующим шагом будет **установка веб-сервера**.

Сделаем это, выполнив команду:

# sudo yum install httpd



После этого необходимо запустить его и добавить в автозагрузку.

# sudo systemctl start httpd.service

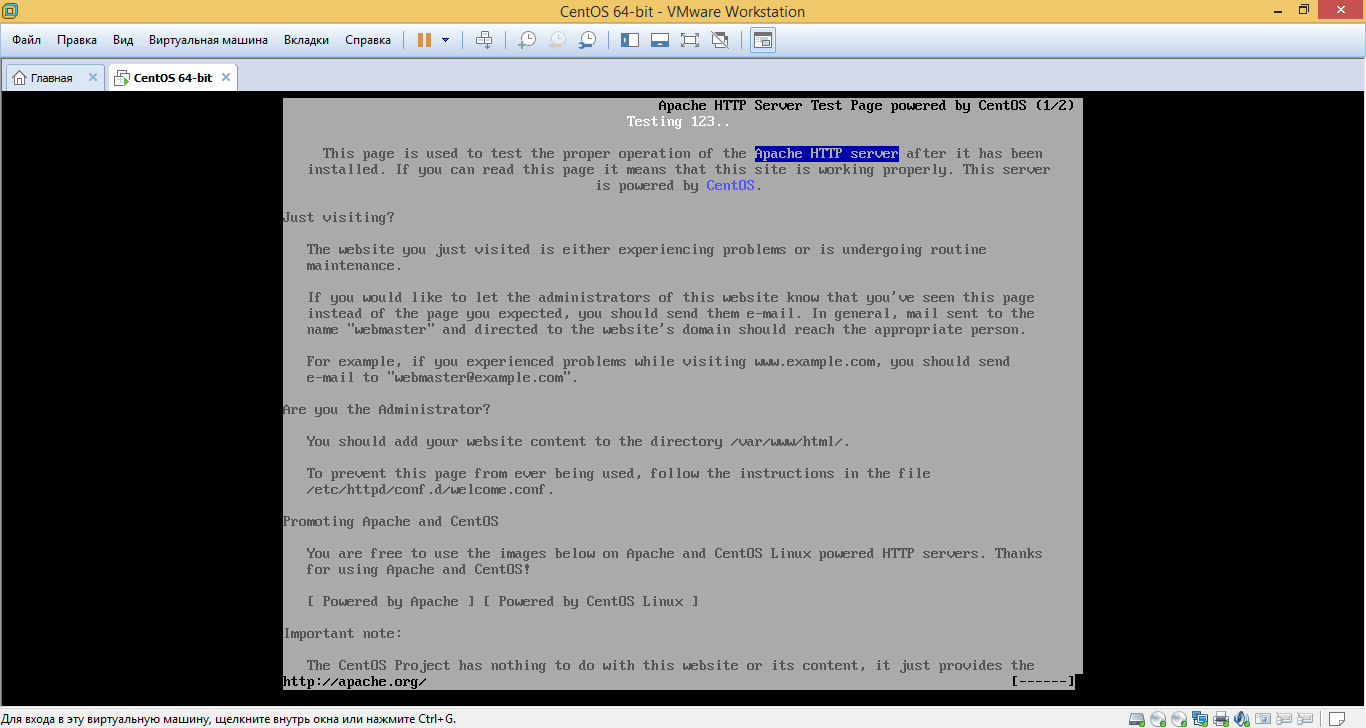
# sudo systemclt enable httpd.service

Необходимо разрешить подключение на порт 80 (протокол http). Для этого можно отключить брандмауэр вовсе, а можно следующим образом его настроить:

# firewall-cmd —zone=public —add-port=80/tcp —permanent

# firewall-cmd —reload

Откроем браузер и перейдем по адресу 127.0.0.1 (или localhost). Увидим открывшуюся тестовую страницу Apache. Значит, установка выполнена верно.



**Теперь установим MySQL.**

Мы пользуемся CentOS 7, которая вместо MySQL использует ее свободную копию – MariaDB. Чтобы установить MySQL, необходимо сперва подключить нужный репозиторий.

Установим утилиту для загрузки файлов из сети - wget:

# sudo yum install wget

Загрузим и установим файл:

# wget <http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm>

# rpm -ivh mysql-community-release-el7-5.noarch.rpm

После установки в нашей системе появился новый репозиторий, который содержит нужный нам пакет mysql-server. Установим его.

# sudo yum install mysql-server

Также запустим его и добавим в автозагрузку.

# sudo systemctl start mysqld

# sudo systemctl enable mysqld

**Приступим к установке php.**

# sudo yum install php

После установки php потребуется перезагрузка Apache:

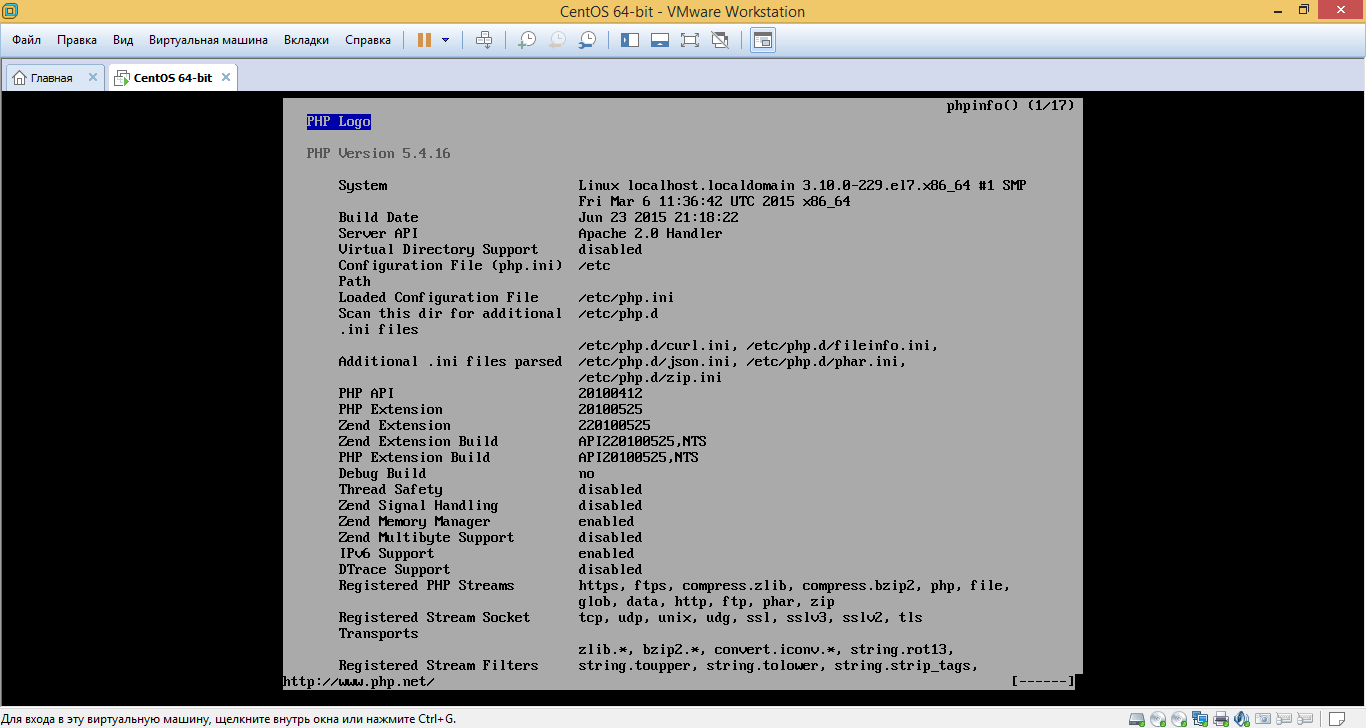
# sudo systemctl restart httpd.service

Теперь можно создать тестовый файл и поместить его в директорию веб-сервера.

# sudo nano var/www/html/test.php

Напишем в нем: ‘<?php phpinfo(); ?>’

И откроем в браузере 127.0.0.1/test.php



Теперь установим пакет php-mysql для поддержки MySQL в php и перезапустим Apache.

# sudo yum install php-mysql

# sudo systemctl restart httpd.service

Теперь нам понадобится PhpMyAdmin – программа, предоставляющая веб-интерфейс для управления БД MySQL.

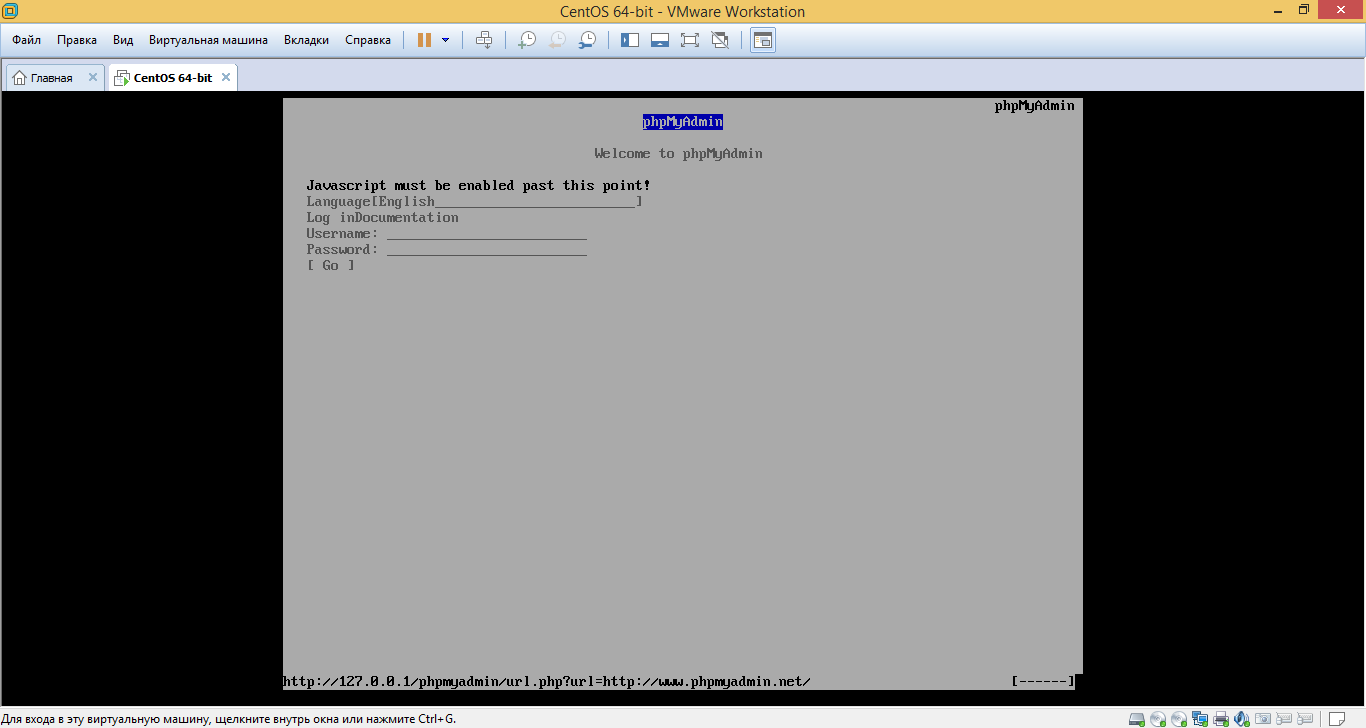
В стандартном репозитории CentOS этой программы нет. Поэтому нам надо подключить дополнительный репозиторий и выполнить установку оттуда, после чего вновь перезапустим веб-сервер.

# sudo yum install epel-release

# sudo yum install phpmyadmin

# sudo systamctl restart httpd.service

Теперь веб-интерфейс PhpMyAdmin доступен по адресу 127.0.0.1/phpmyadmin



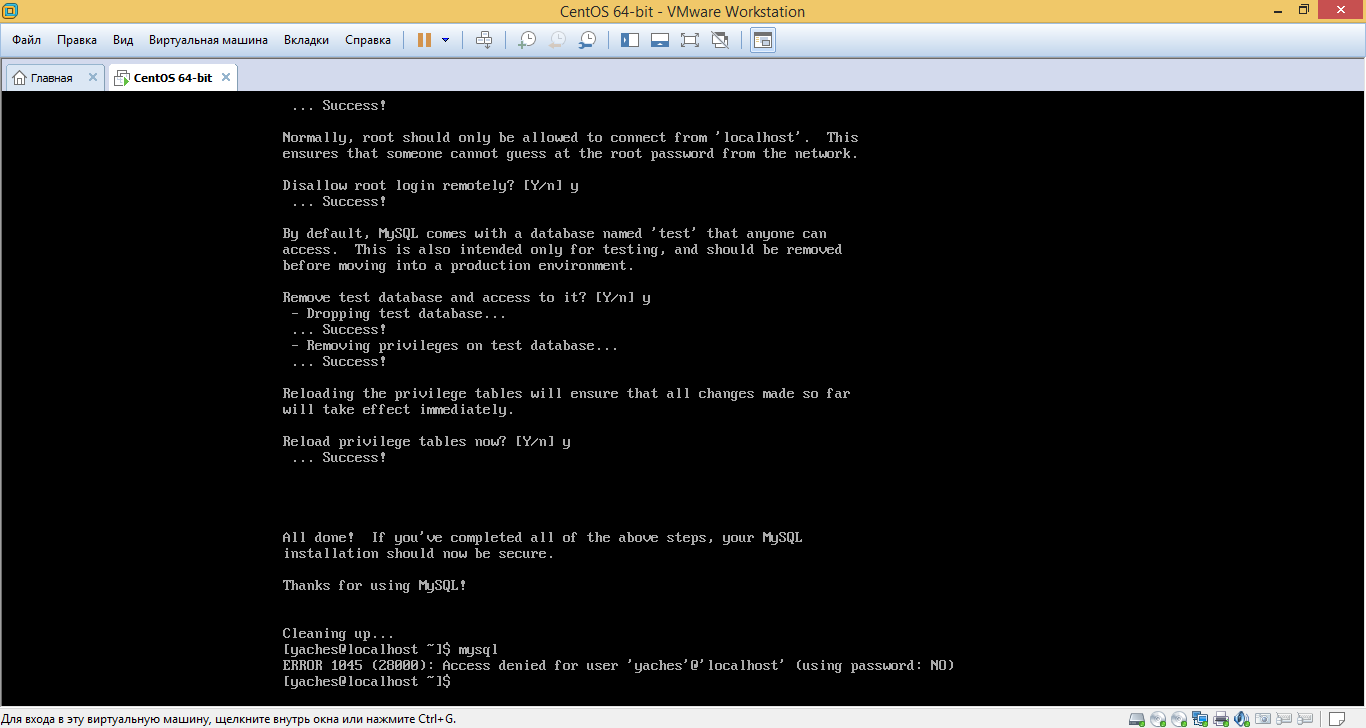
**Часть 2**

Написание регистрации на сервере с помощью php.

Сперва выполним первоначальную настройку MySQL. Запустим скрипт:

# sudo mysql\_secure\_installation

В ходе его выполнения система попросит нас установить пароль root’а для входа в MySQL, удалит анонимного пользователя и тестовую БД. Соглашаемся со всем.



На скриншоте видно, что зайти в MySQL без пароля теперь нельзя. Необходимо сделать следующее:

# mysql –u root –p

После чего ввести пароль (я установил в качестве пароля 0000).

После того, как мы зашли в MySQL, создадим в нем нового пользователя yaches с привилегиями суперпользователя с тем же паролем.

mysql> GRANT ALL PRIVILEGES ON \*.\* TO yaches@localhost

-> IDENTIFIED BY ‘0000’ WITH GRANT OPTION;

Подключение к БД с сервера будем осуществлять из-под этого пользователя.

Создадим БД с именем serverdb и переключимся на работу с ней.

mysql> create database serverdb;

mysql> use serverdb;

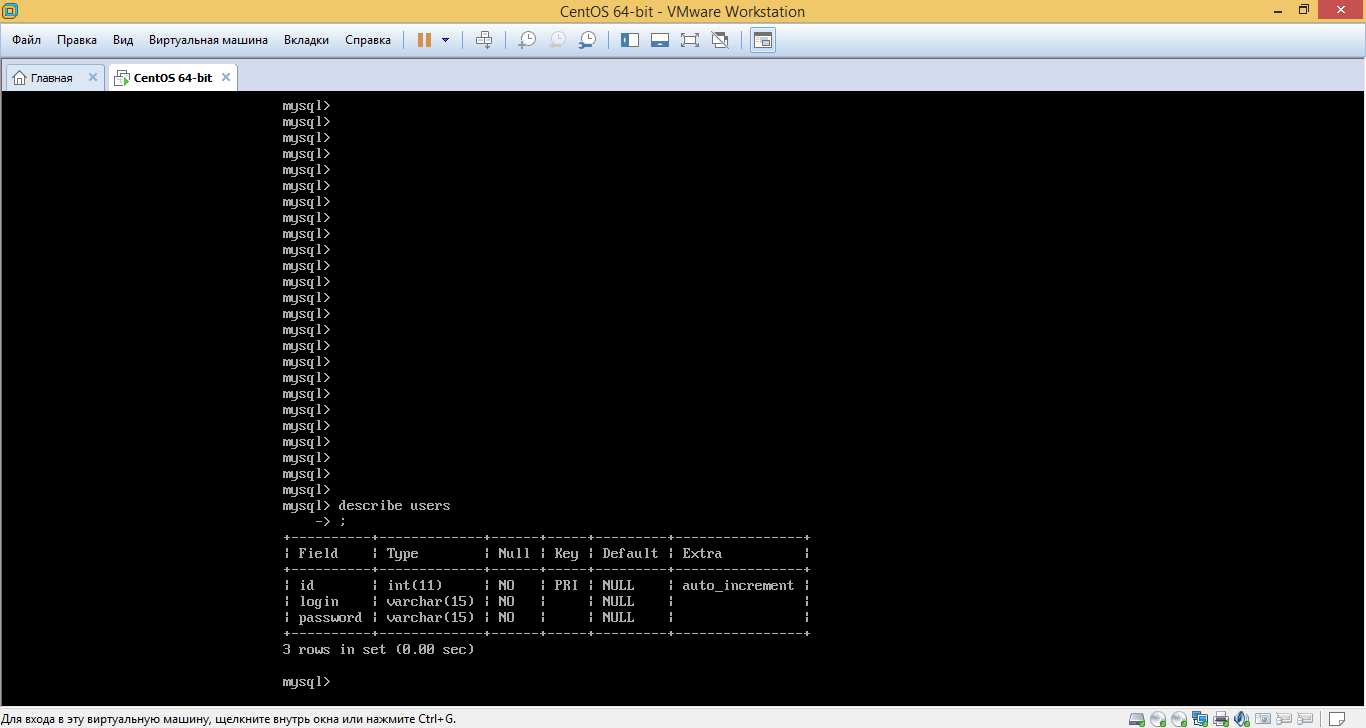
Создадим таблицу, которая будет хранить информацию о зарегистрированных пользователях: их id, логин и пароль.

mysql> create table users (id INT auto\_increment primary key, login VARCHES(15), password VARCHAR(15));

В числе параметров поля id – auto\_increment и primary key. Это значит, что для каждой новой записи в поле id будет генерироваться уникальное значение, по которому и будет определяться запись в таблице.

После сообщения об успешном создании таблицы выведем ее на экран командой:

mysql> describe users;



Выходим из MySQL командой:

mysql> \q

Приступим к написанию скриптов регистрации пользователя на сервере. Первым из них – bd.php – будет осуществлять подключение к БД.

# sudo nano /var/www/html/bd.php

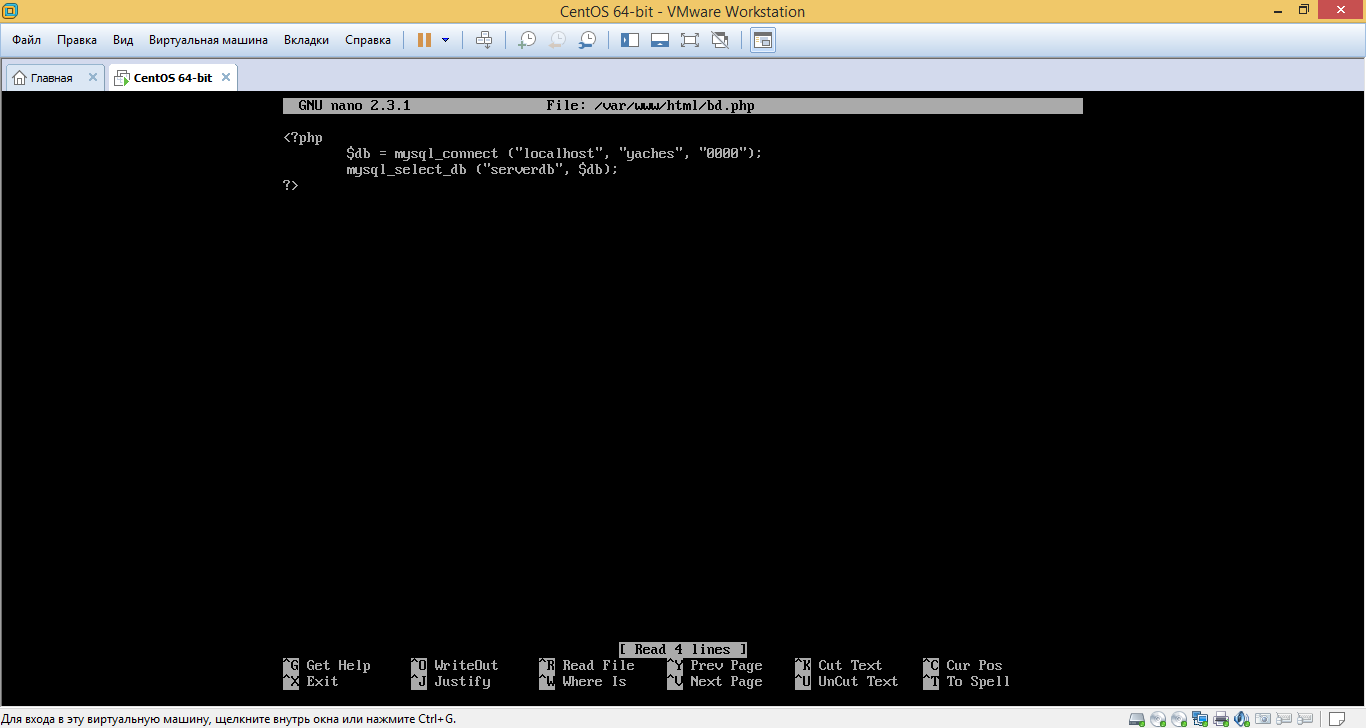
Напишем туда:

<?php

$db = mysql\_connect ("localhost", "yaches", "0000"); // Адрес MySQL-сервера, имя пользователя и пароль соответственно

mysql\_select\_db ("serverdb", $db); // Имя БД

?>



Сохранить, закрыть. Теперь создадим страничку регистрации reg.php.

# sudo nano /ver/www/html/reg.php

<html>  
<body>  
<h2>Registration</h2>  
<form action="save\_user.php" method="post">

<!—данные из этой формы отправятся в файл save\_user.php, который мы создадим позднее -->  
<p>  
<label>Login:<br></label>  
<input name="login" type="text" size="15" maxlength="15">

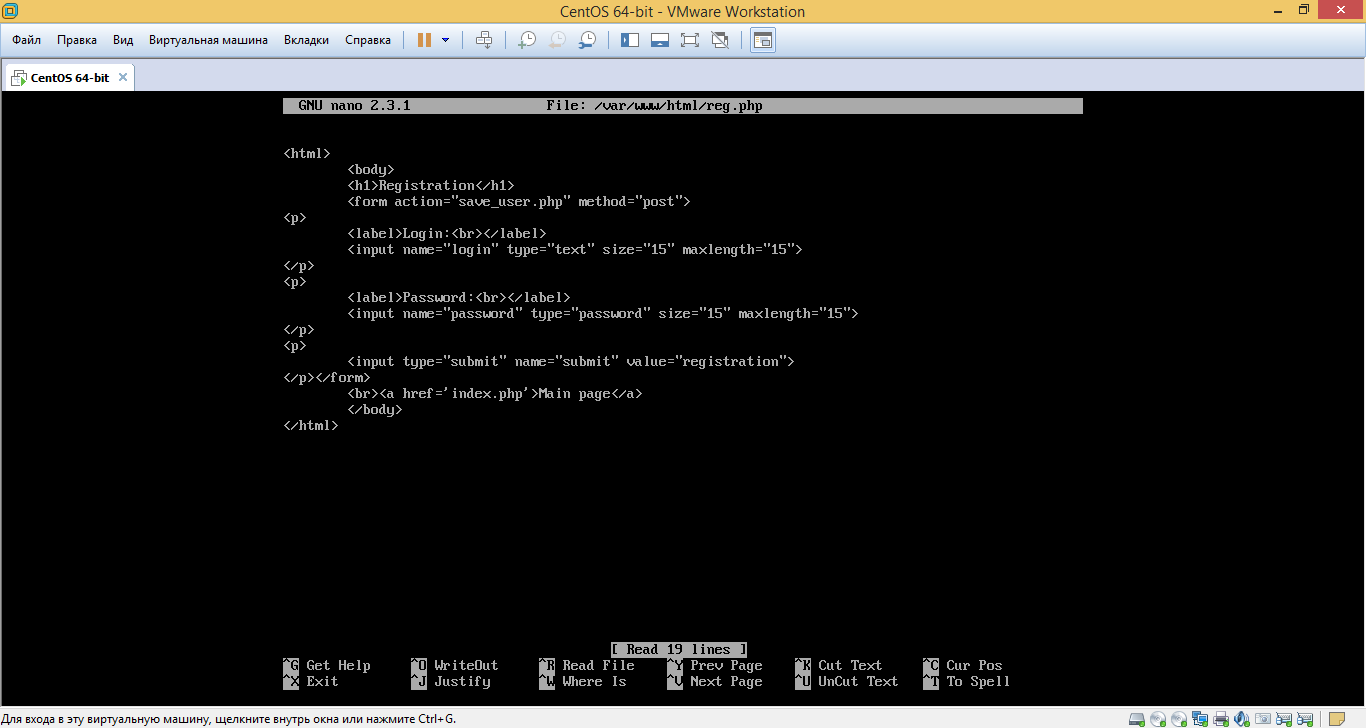
<!—Поле для логина -->  
</p>

<p>  
<label>Password:<br></label>  
<input name="password" type="password" size="15" maxlength="15">  
<!—Поле для пароля -->

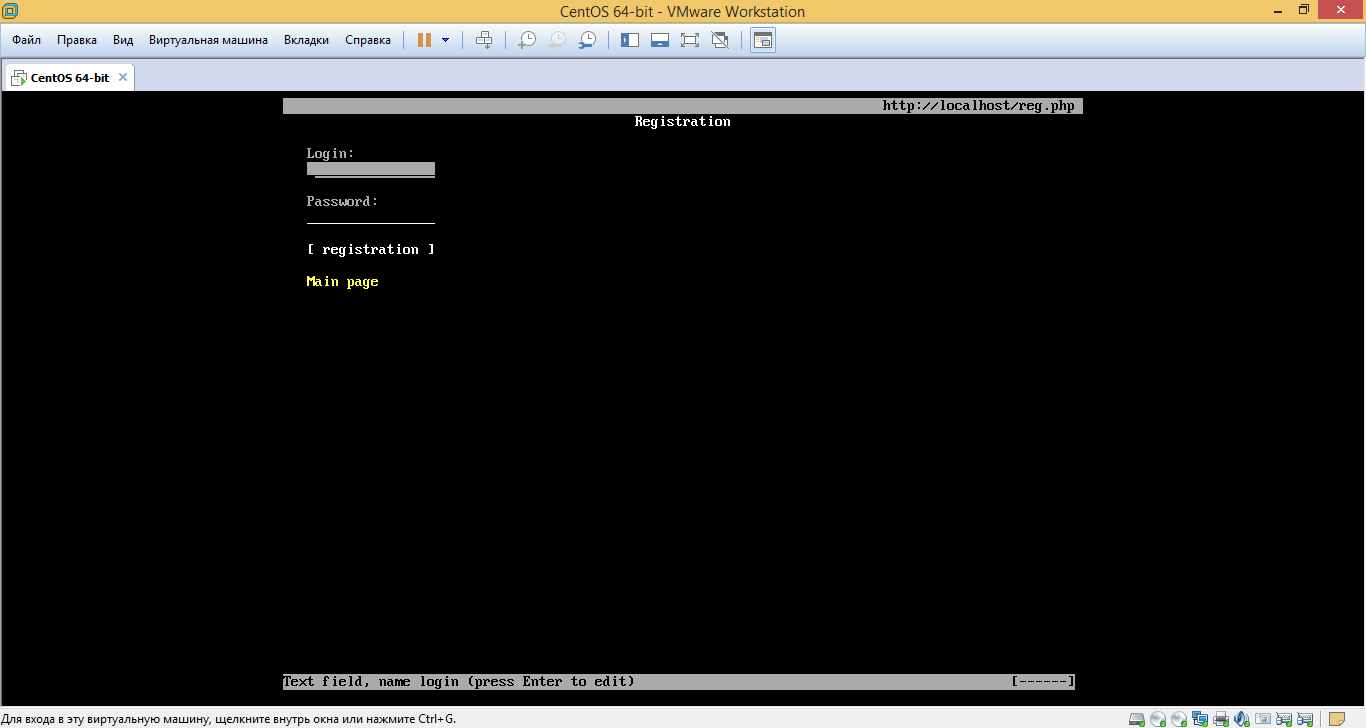
</p>  
<p>  
<input type="submit" name="submit" value="registration">  
<!—Кнопка, отправляющая данные в save\_user.php -->

</p></form>  
</body>

<br><a href='index.php'>Main page</a>  
</html>



Посмотрим, что получилось.



Приступим к написанию скрипта save\_user.php, который будет сохранять введенные логин и пароль в БД.

<?php

// заносим введенные пользователем логин и пароль в переменные $login и $password, если они пустой, то уничтожаем переменные  
if (isset($\_POST['login'])) {

$login = $\_POST['login'];

if ($login == '') { unset($login);}

}

if (isset($\_POST['password'])) {

$password=$\_POST['password'];

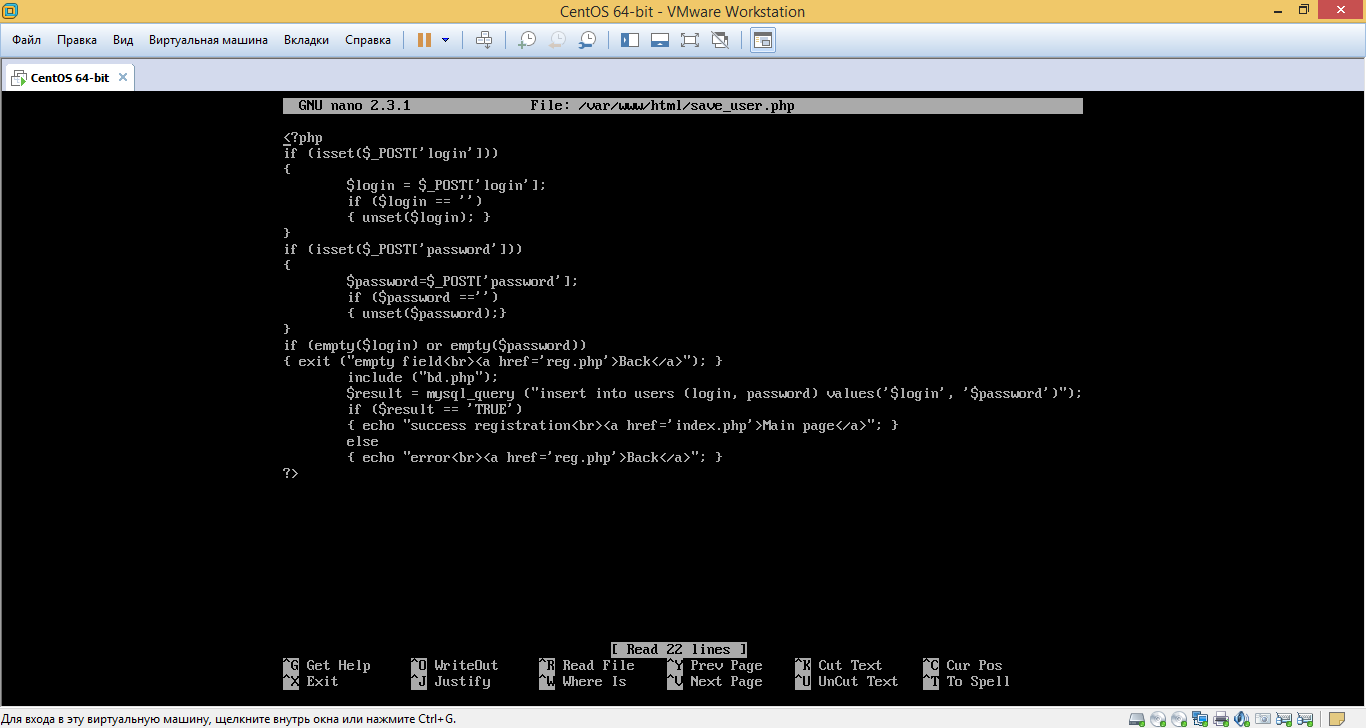
if ($password =='') { unset($password);}

}

if (empty($login) or empty($password)) { exit ("empty field<br><a href='reg.php'>Back</a>"); }

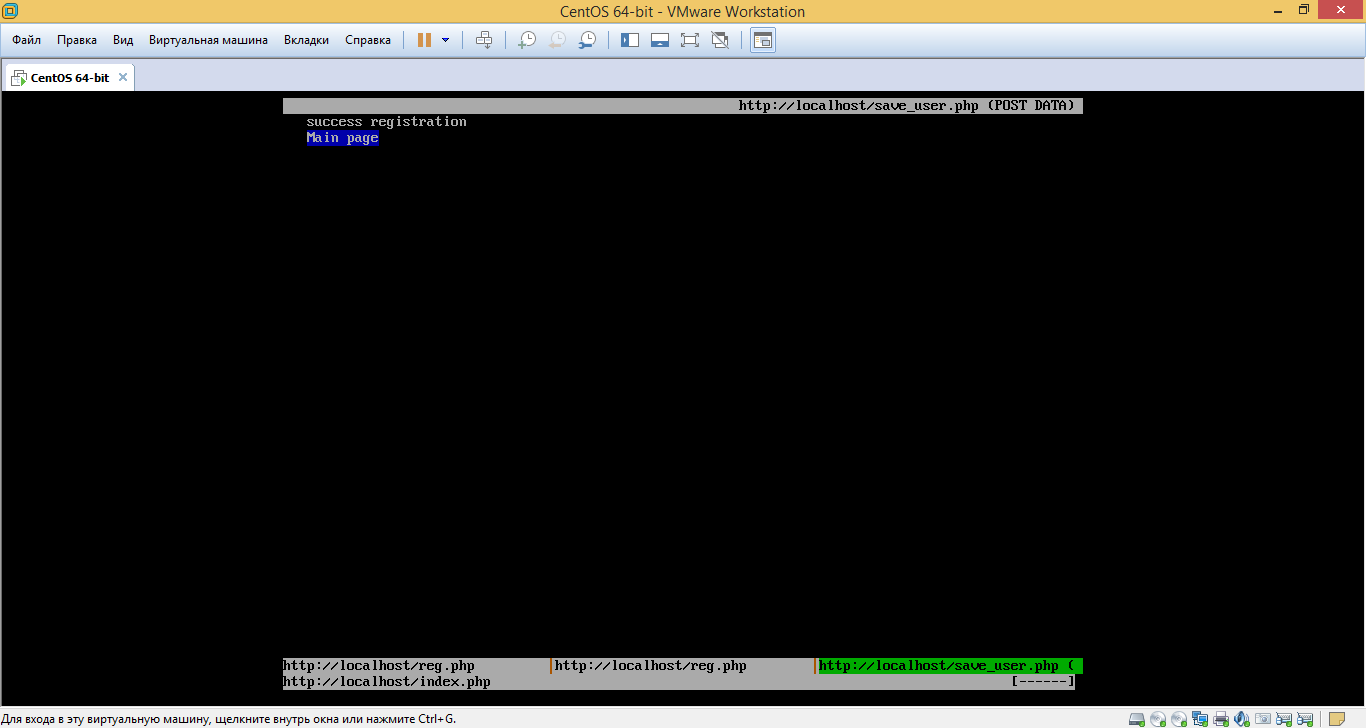
include ("bd.php"); // Подключаемся к базе  
$result = mysql\_query ("INSERT INTO users (login,password) VALUES('$login','$password')");  
// Проверяем, есть ли ошибки  
if ($result2=='TRUE')  
{ echo "success registration<br><a href='index.php'>Main page</a>"; }  
else

{ echo "error<br><a href='reg.php'>Back</a>"; }  
?>



Сохранить, закрыть.

Снова открываем в браузере страницу 127.0.0.1/reg.php и вводим свои данные. Например, test1 и pwd1.



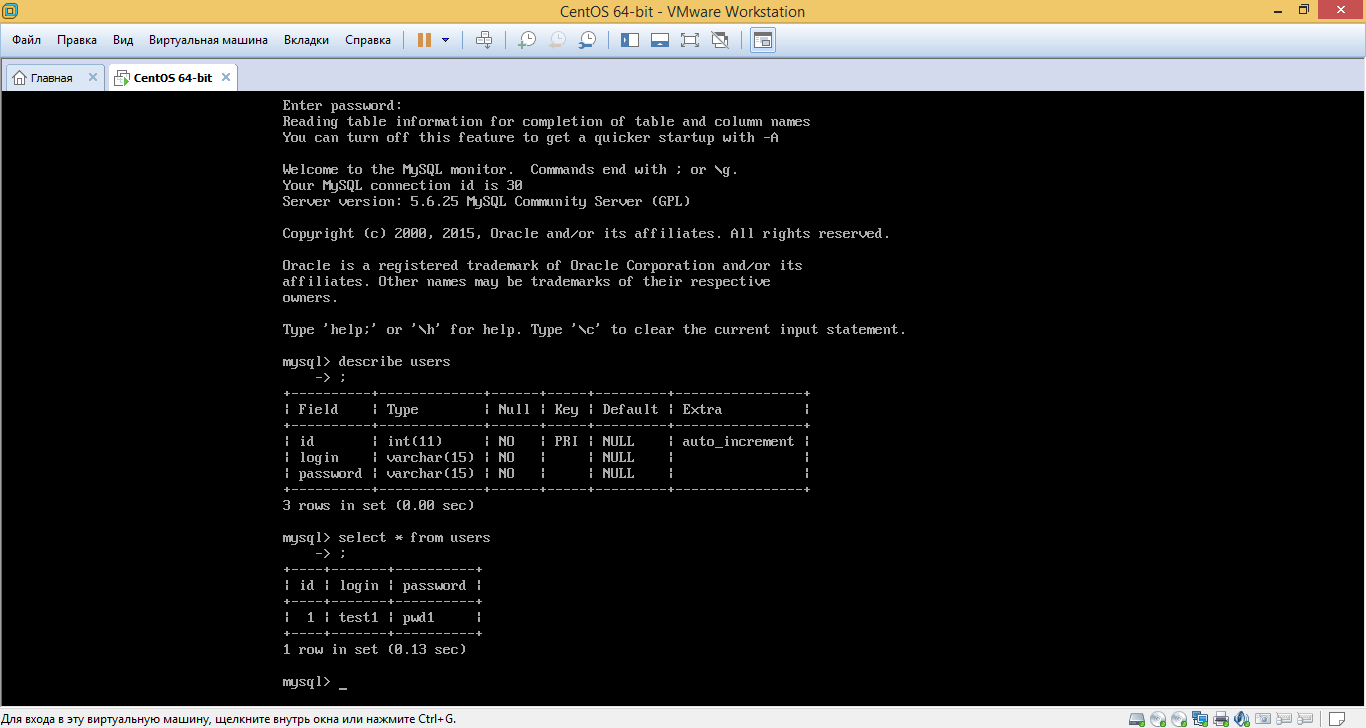
Регистрация прошла успешно. Для проверки откроем таблицу в MySQL.

# mysql –u yaches –p

mysql> use serverdb;

mysql> select \* from users;

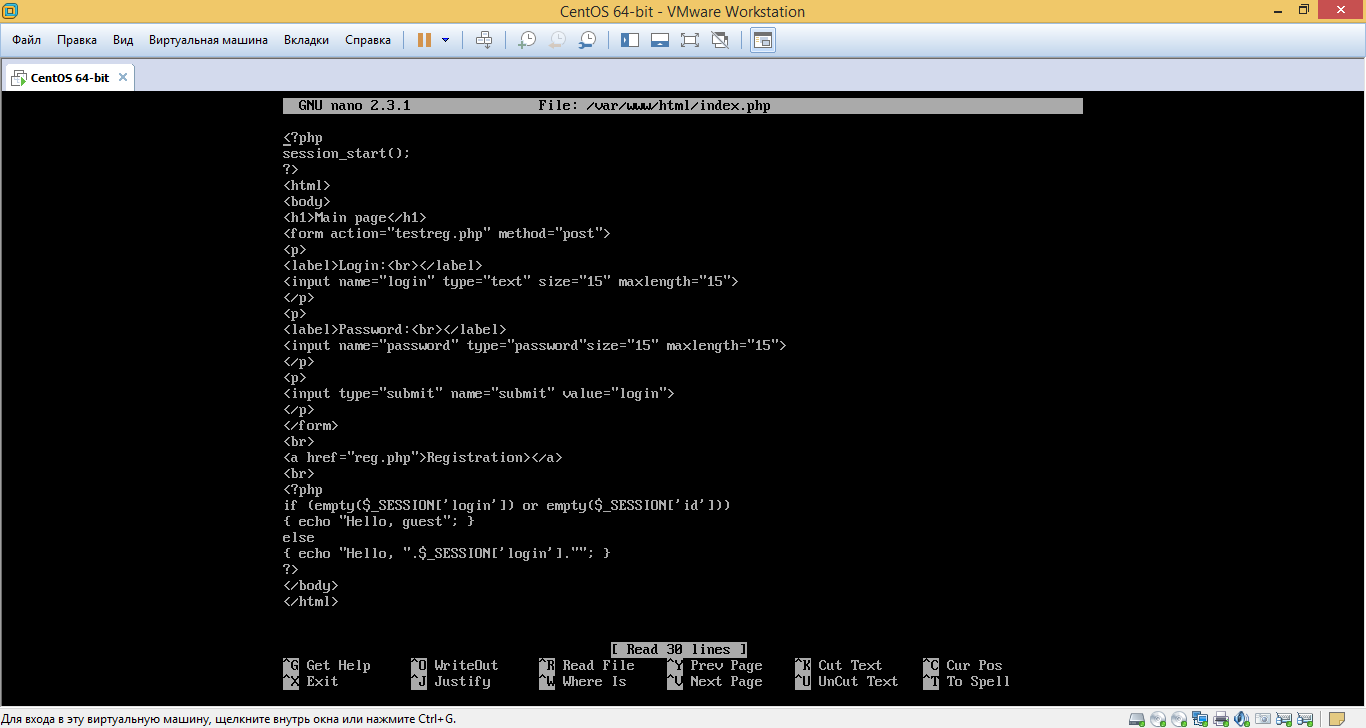
Увидим внесенные данные о пользователе test1.



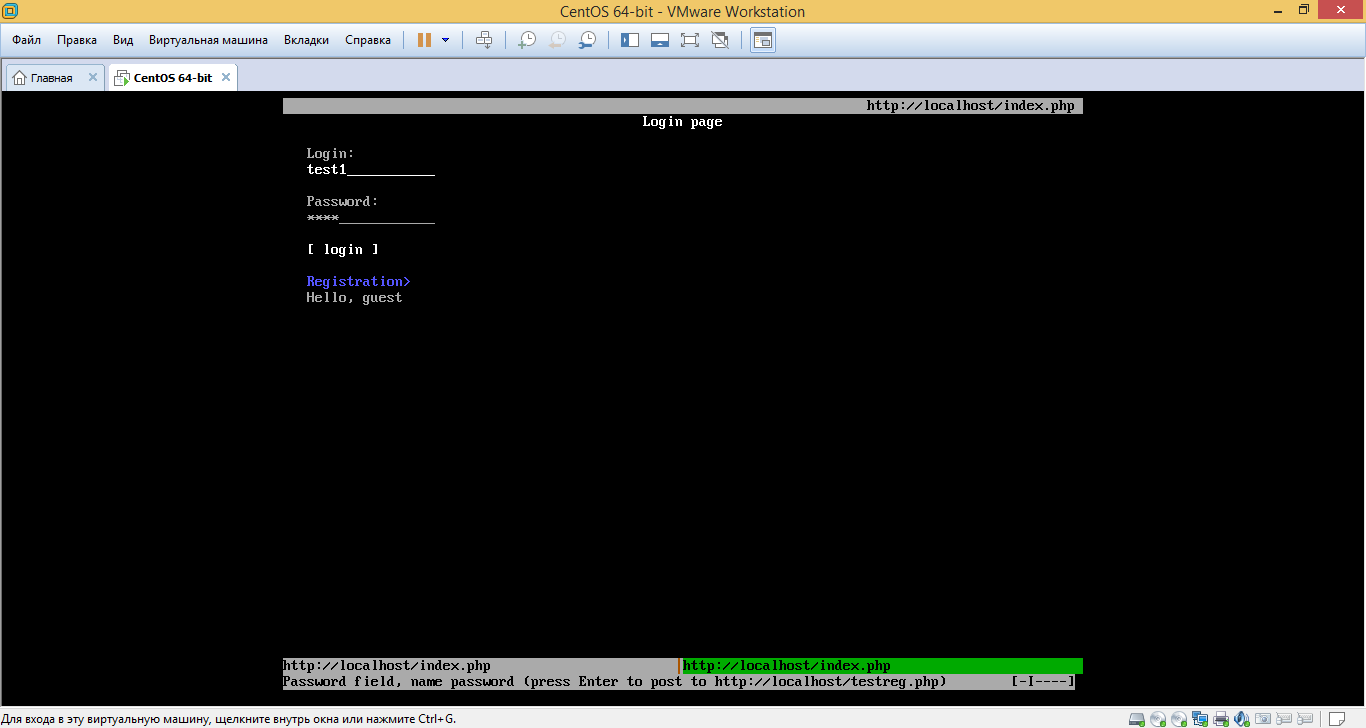
Следующим шагом создадим форму для входа уже зарегистрированных пользователей index.php.

<?php  
 session\_start();

?>  
 <html>  
 <body>  
 <h1>Main page</h1>  
 <form action="testreg.php" method="post">  
 <!—Отправляем введенные данные в файл testreg.php -->  
 <p>  
 <label>Login:<br></label>  
 <input name="login" type="text" size="15" maxlength="15">  
 </p>  
 <p>  
 <label>Password:<br></label>  
 <input name="password" type="password" size="15" maxlength="15">  
 </p>  
 <p>  
 <input type="submit" name="submit" value="Войти">   
<br>  
 <!-- Cсылка на регистрацию -->   
<a href="reg.php">Registration</a>   
 </p></form>  
 <br>  
 <?php  
 // Проверяем, пусты ли переменные логина и id пользователя  
 if (empty($\_SESSION['login']) or empty($\_SESSION['id']))  
 {  
 // Если пусты, то приветствуем его как гостя  
 echo "Hello, guest";  
 }  
 else  
 {  
 // Если не пусты, то по имени  
 echo "Hello, ".$\_SESSION['login']."";  
 }  
 ?>  
 </body>  
 </html>



Открываем браузер.



Последний этап. Напишем файл testreg.php, который будет проверять, есть ли пользователь с введенными данными в БД или нет.

<?php

session\_start();

// Проверяем, введены ли все данные

if (isset($\_POST['login'])) {

$login = $\_POST['login'];

if ($login == '') { unset($login);}

}

if (isset($\_POST['password'])) {

$password=$\_POST['password'];

if ($password =='') { unset($password);}

}

if (empty($login) or empty($password))

{

exit ("empty field<br><a href='index.php'>Main page</a>");

}

include ("bd.php");

// Проверяем существование в БД такого логина

$result = mysql\_query("SELECT \* FROM users WHERE login='$login'",$db);

$myrow = mysql\_fetch\_array($result);

if (empty($myrow['password']))

{

exit ("wrong login<br><a href='index.php'>Main page</a>");

}

else {

// Сравниваем пароли

if ($myrow['password']==$password) {

$\_SESSION['login']=$myrow['login'];

$\_SESSION['id']=$myrow['id'];

echo "Hello, ".$\_SESSION['login']."<br>Your id is ".$\_SESSION['id']."<br><a href='index.php'>Main page</a>";

}

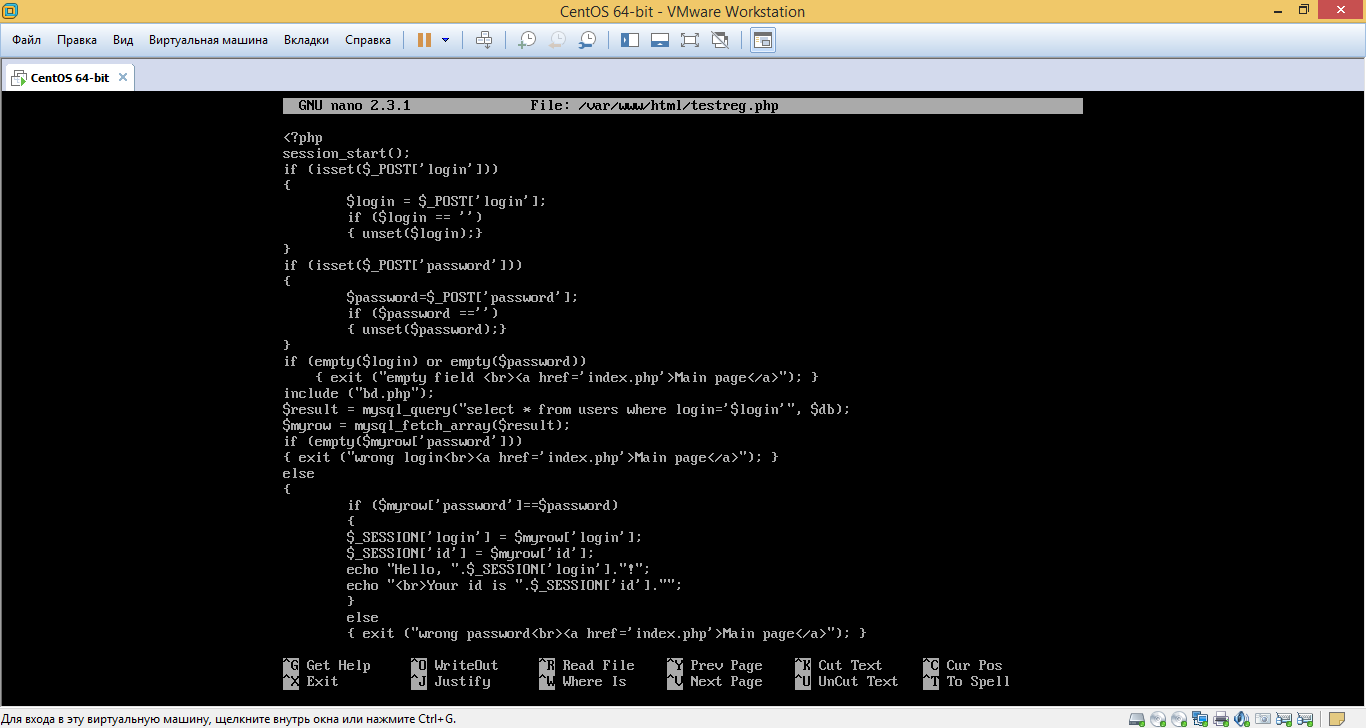
else {

exit ("wrong password<br><a href='index.php'>Main page</a>");

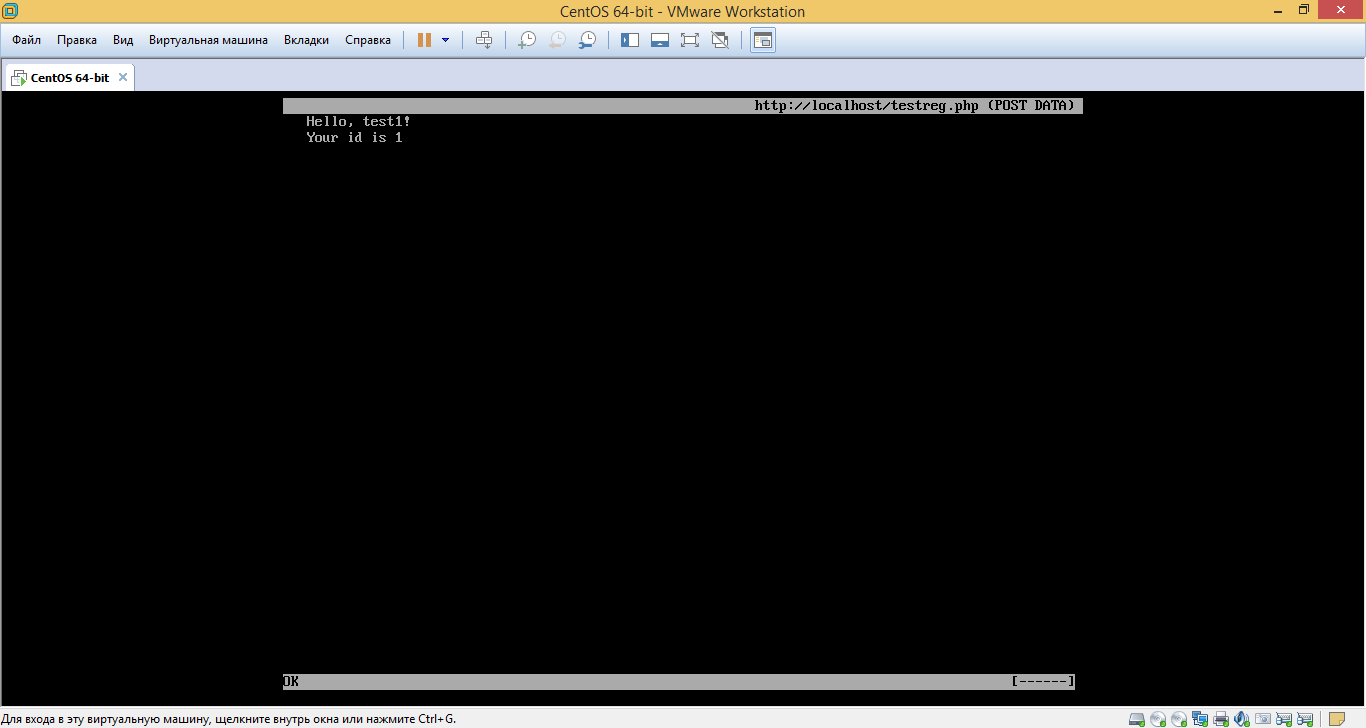
}

}

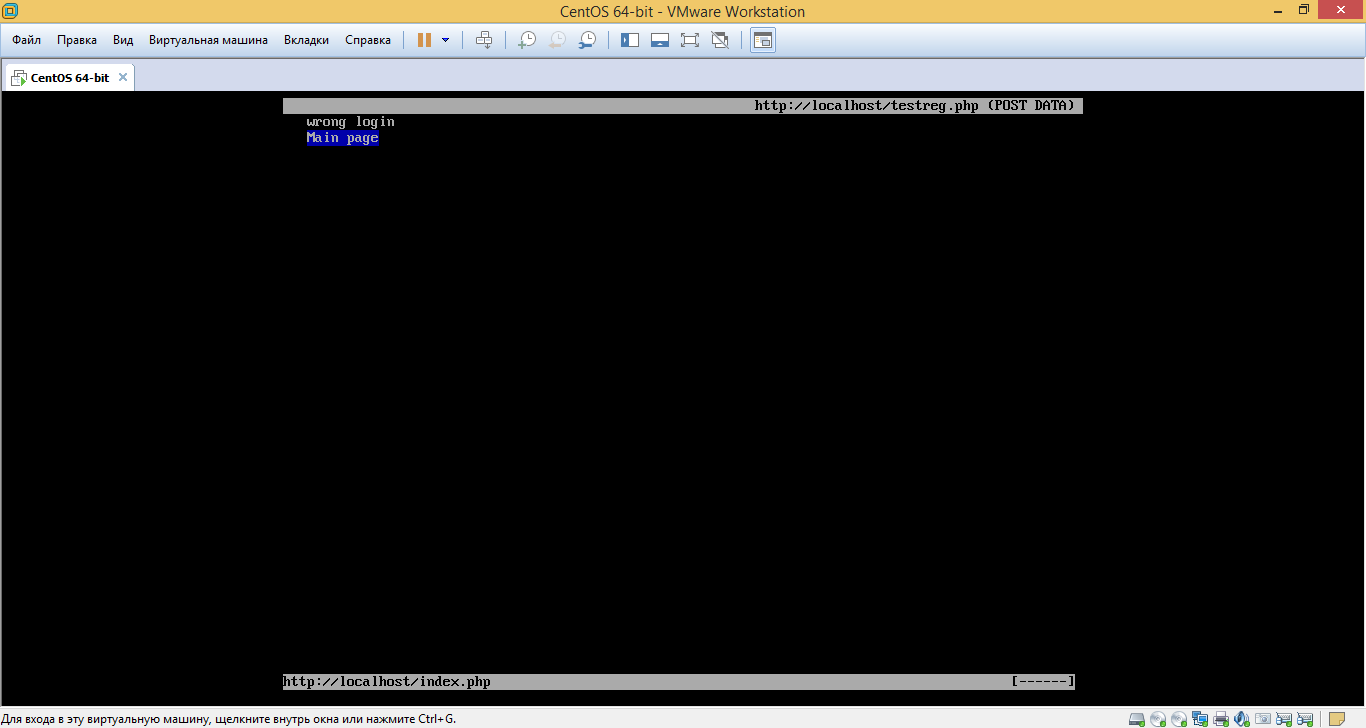
?>

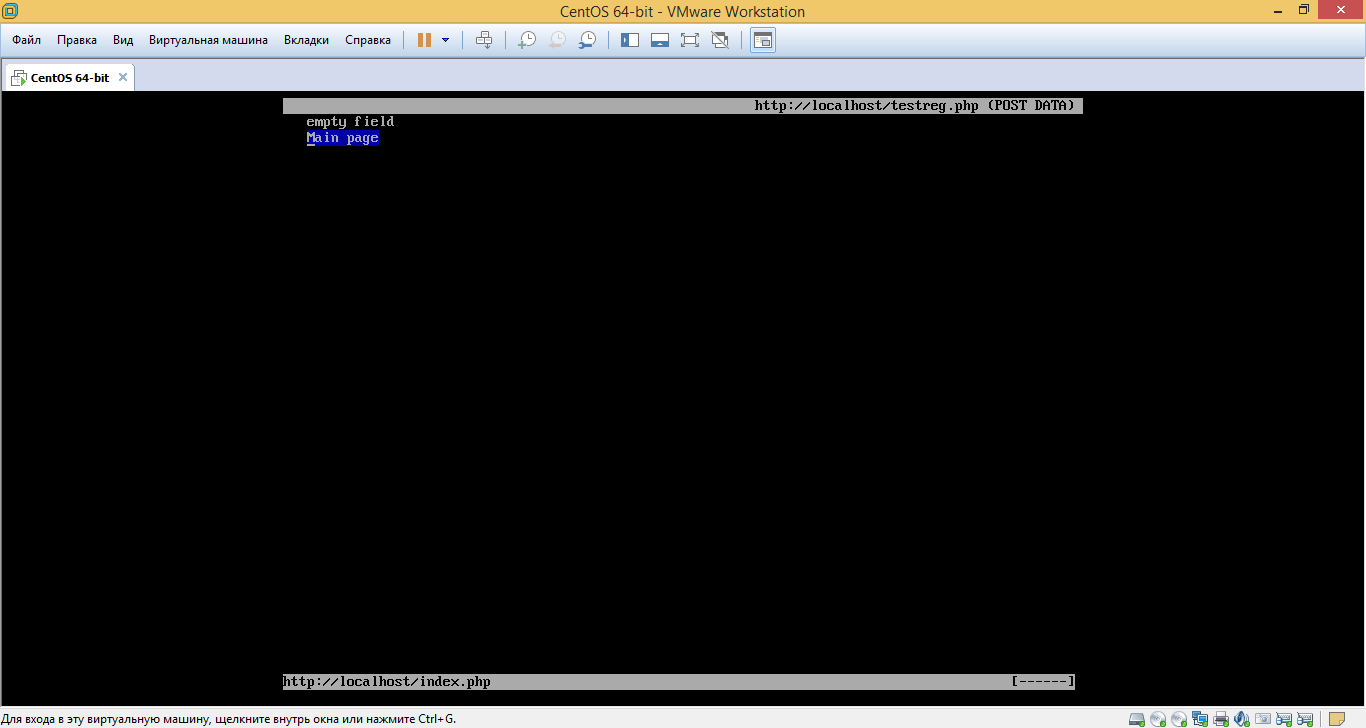


И снова открываем браузер.



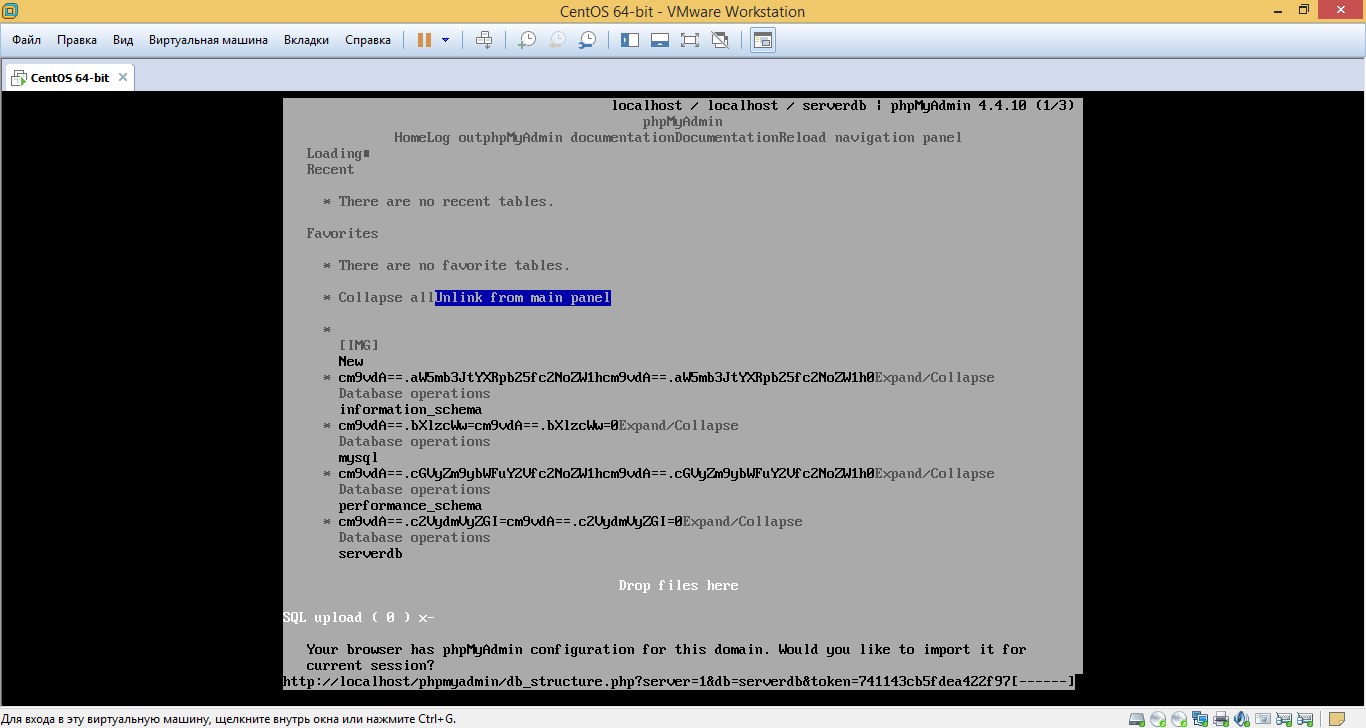
Попробуем вести несуществующий логин или не ввести его вовсе.





Все прекрасным образом работает.

Стоит сказать несколько слов о PhpMyAdmin (это было в задании). Это набор php-скриптов, позволяющих осуществлять управление БД с помощью веб-интерфейса. Доступен этот интерфейс по адресу <http://localhost/phpmyadmin>. После ввода своих данных попадаем на страничку с элементами управления



Интерфейс этого сервиса в консольном браузере представляется мне не очень удобным. Именно поэтому работу с БД я осуществлял из консоли mysql.