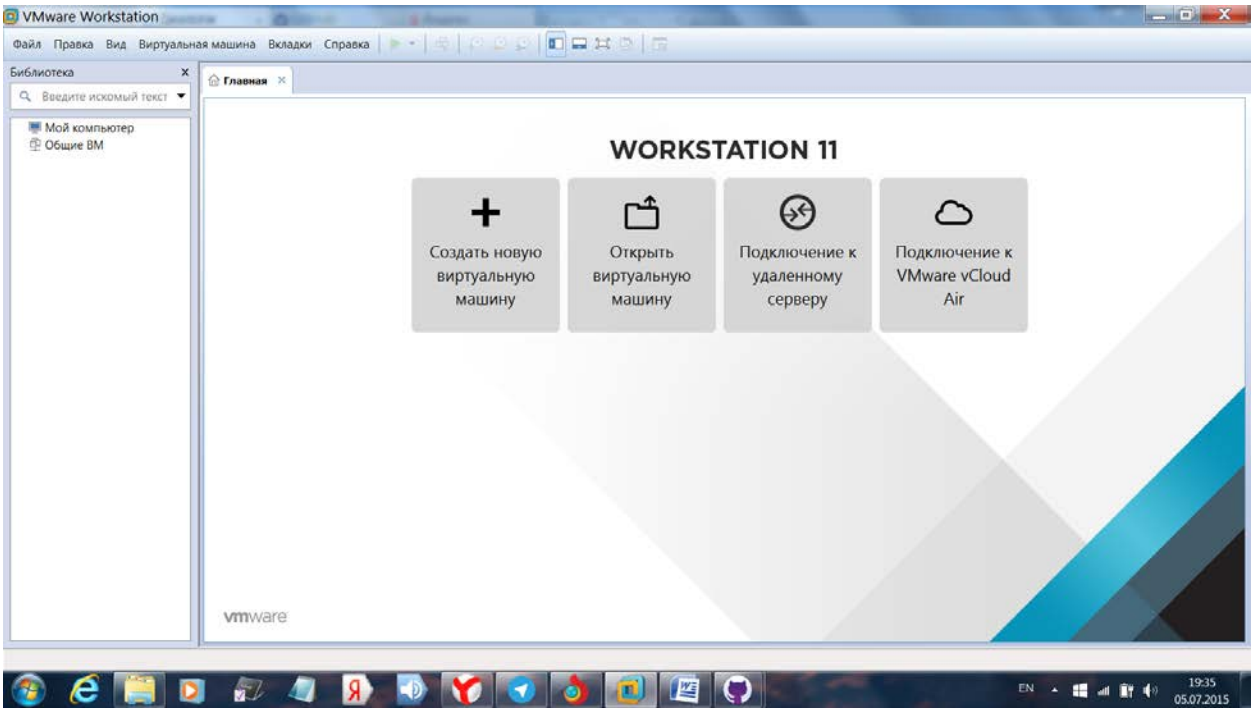
Практика

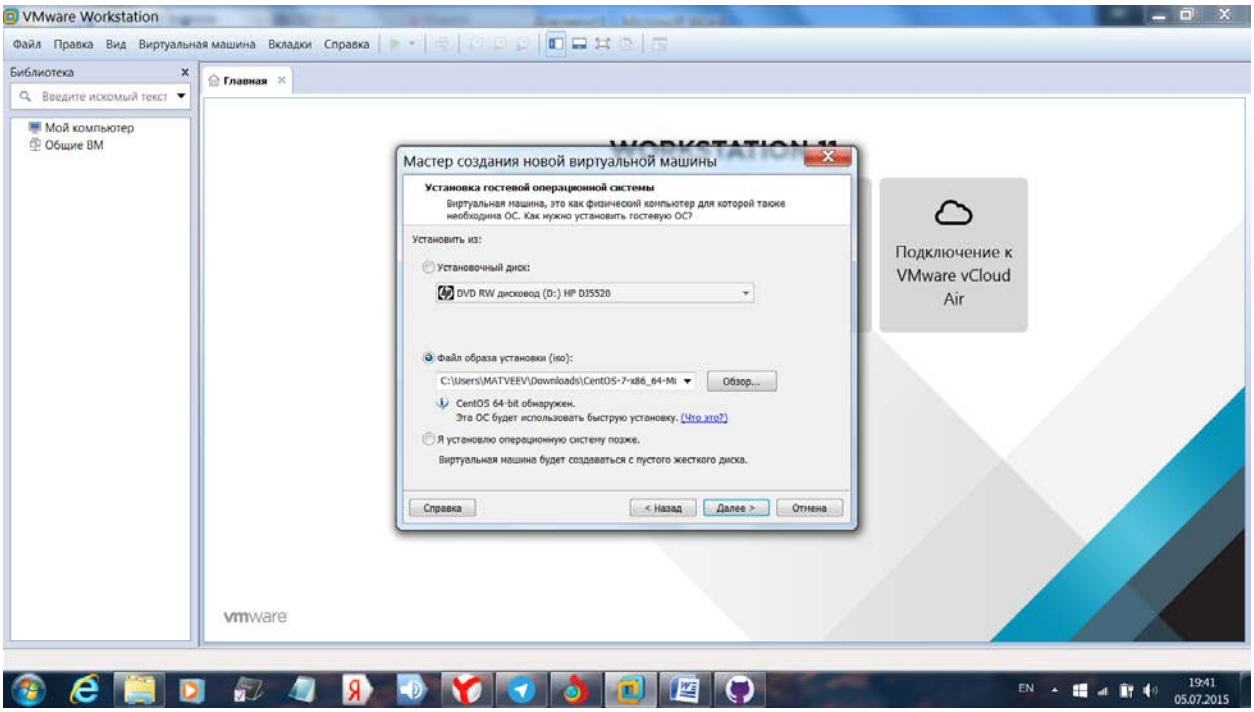
Задание 1 Матвеев М.И.

VMware Wokstation создает полностью изолированные безопасные виртуальные машины. Уровень виртуализации VVMware сопоставляет ресурсы физического оборудования с ресурсами виртуальной машины. Таким образом, каждая виртуальная машина получает собственные ЦП, память, диски и устройства ввода-вывода и является полным эквивалентом стандартного компьютера x86. Centos является дистрибутивом GNU/Linux, основанном на свободных исходных текстах коммерческого дистрибутива Red Hat Enterprise Linux компании Red Hat.

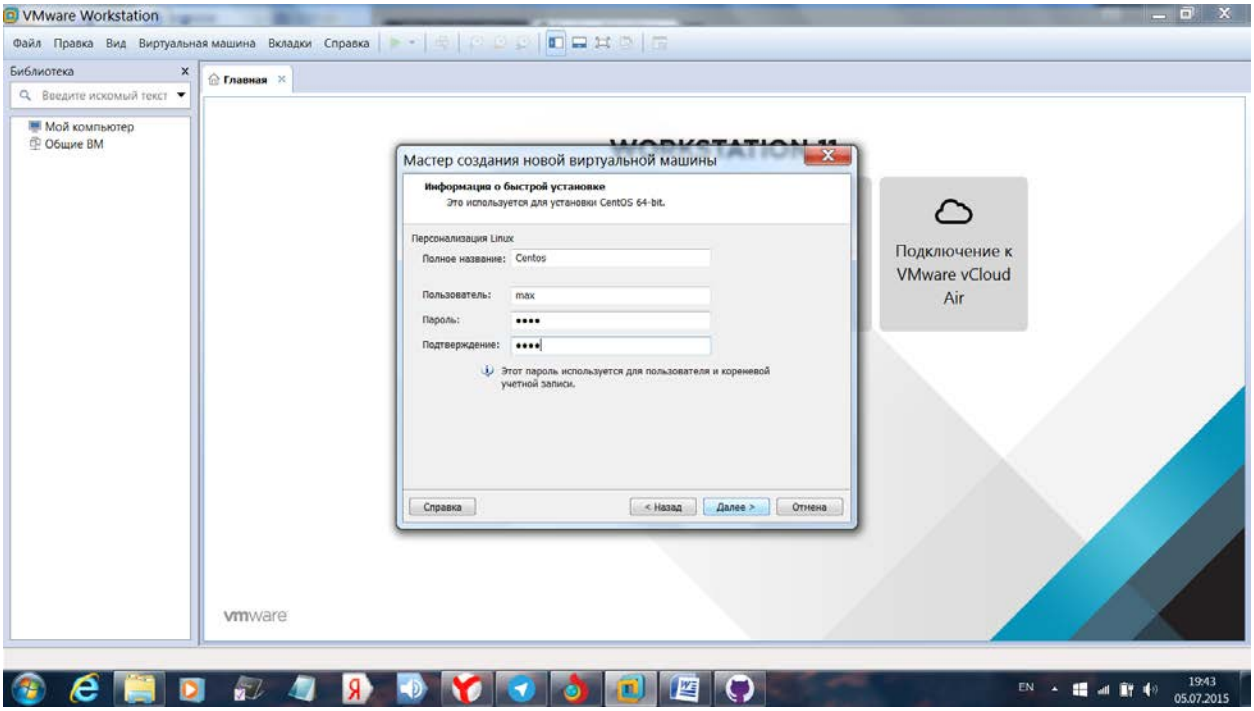
1)Установка VMware Wokstation



2) Процесс создания виртуальной машины

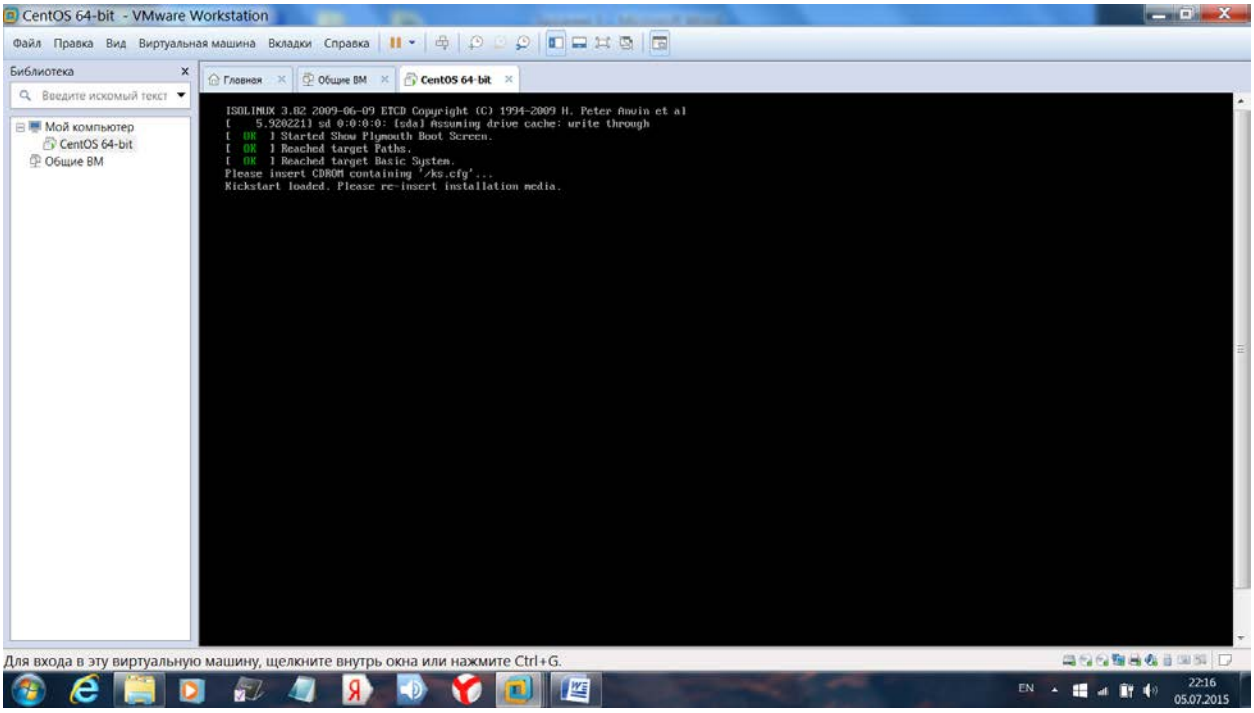


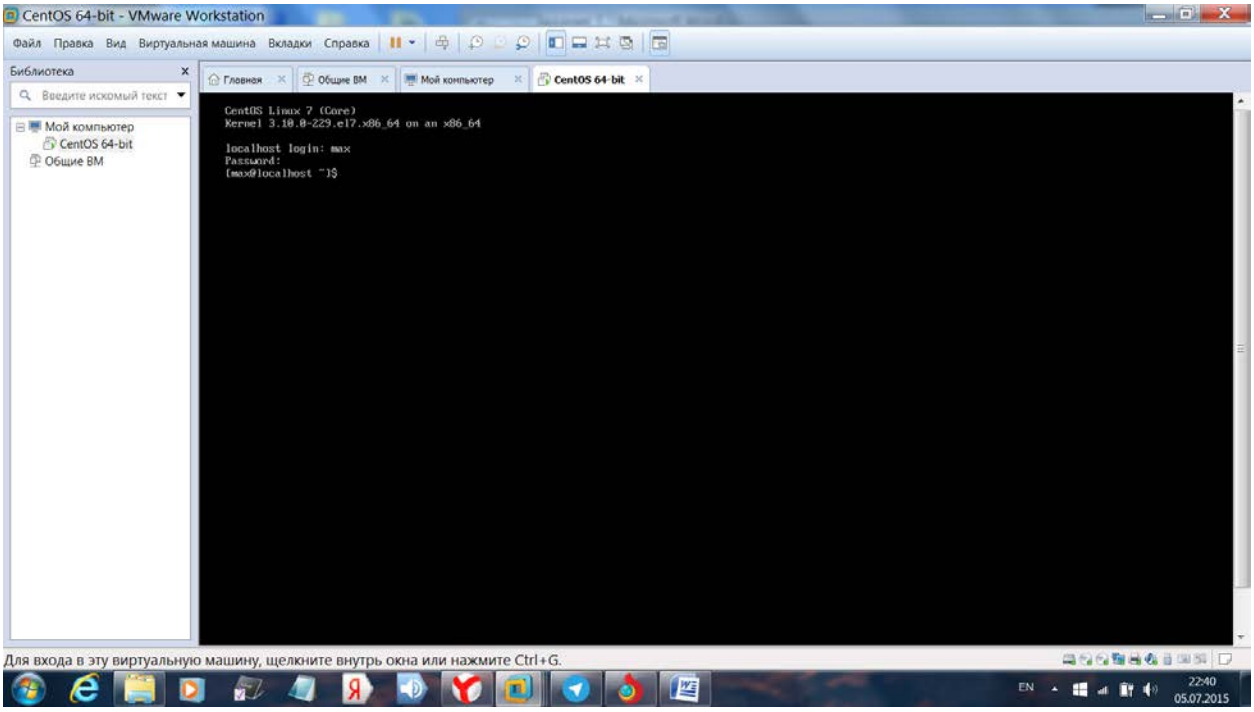
Выбор пароля и логина:



Настройка дополнительных параметров:

Запуск и авторизация





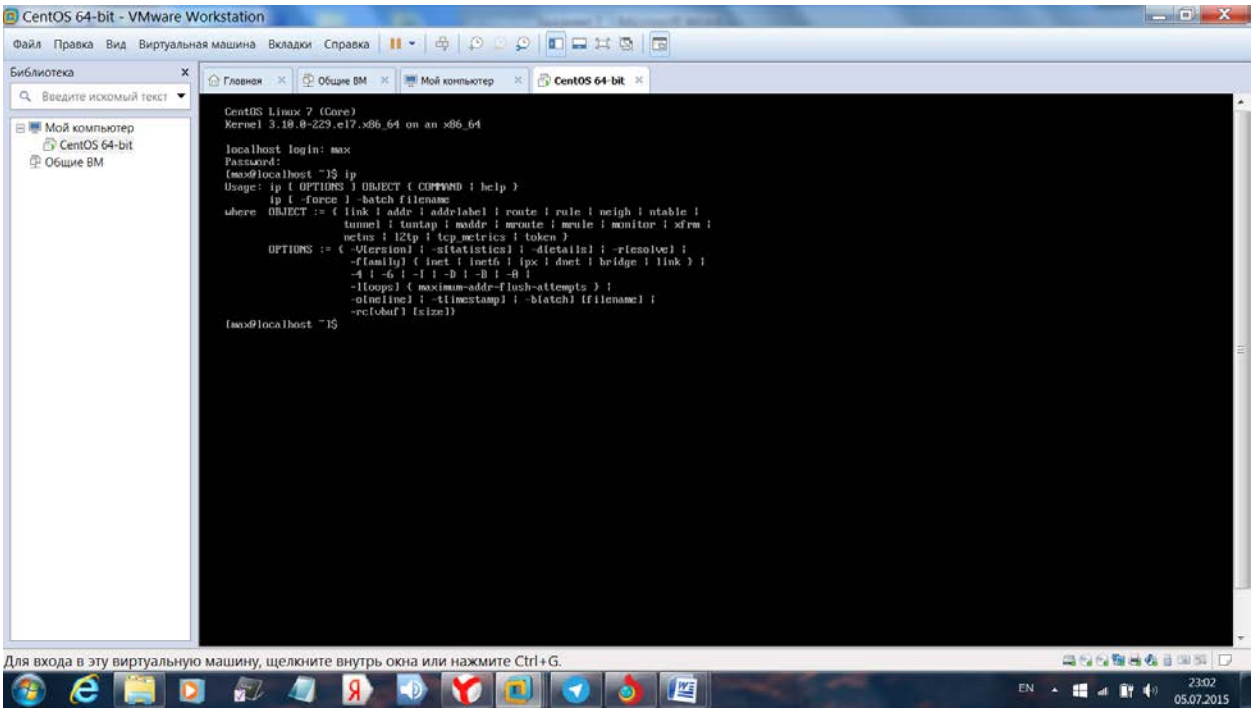
Задание 2

1)Основные команды.

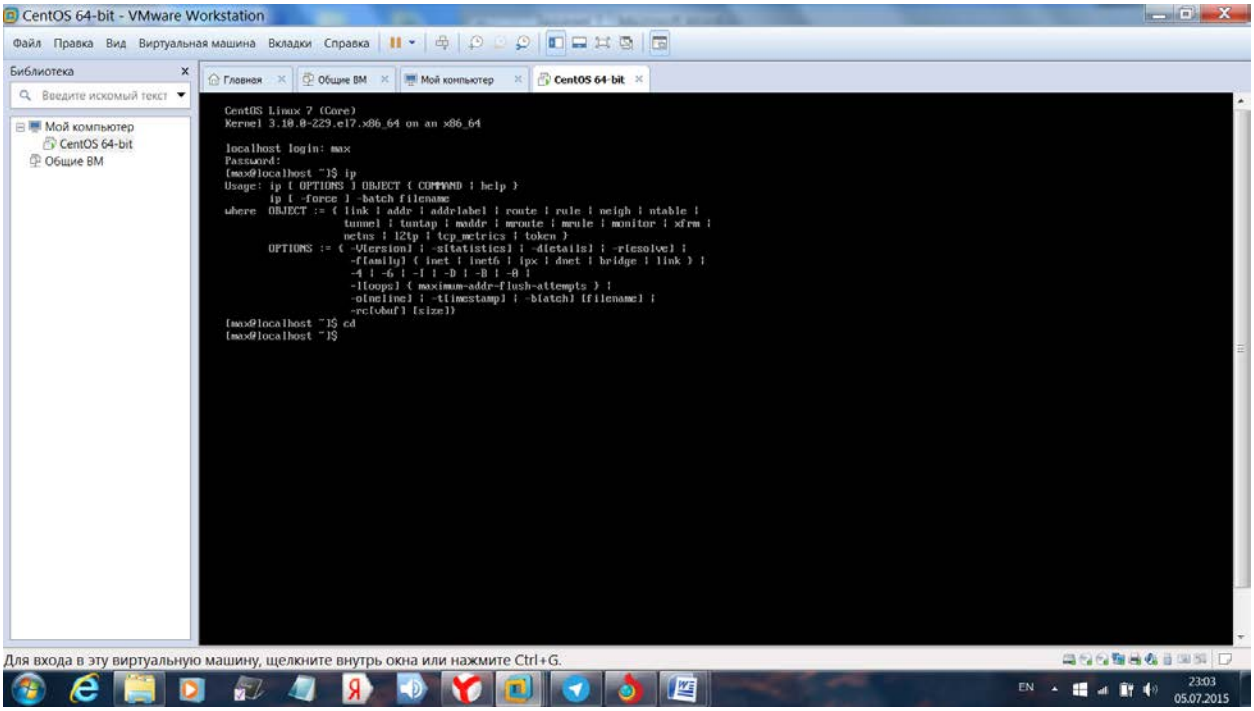
Для взаимодействия с системой, необходимо знать несколько основных команд. Одной из основных команд является команда **su**, которая делает вас пользователем “root”, у которого по умолчанию есть доступ ко всем командам. Ставь пользователем “root”, мы получаем доступ к таким командам как **poweroff**(выключение компьютера), **reboot**(перезагрузка системы),

dir(показывает список файлов в директории, отсортированных вертикально по столбцам), **mkdir**(создать директорию), **rmdir**(удалить директорию).

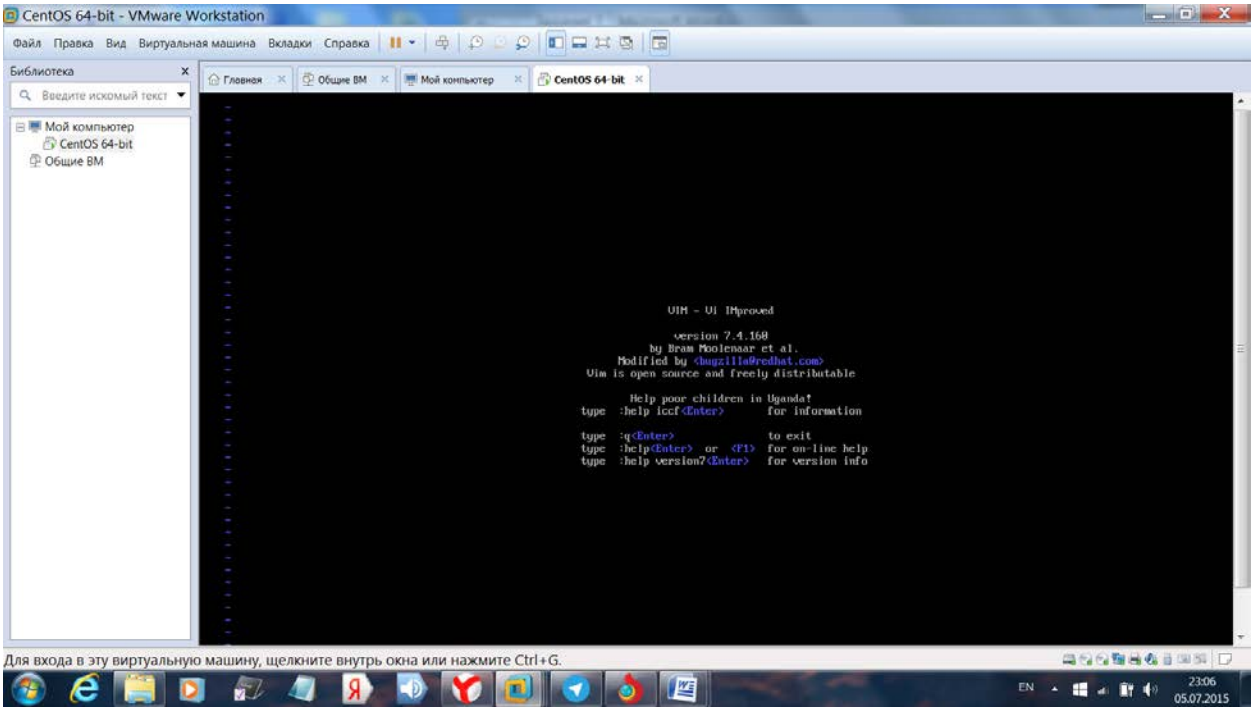
Команда ip помогает узнать информацию о пользователе:



команда cd: если вы введете эту команду без аргументов, вы попадете в свой домашний каталог; чтобы попасть в любой другой каталог, необходимо указать путь к нему.



Команда vi открывает текстовый редактор



Команда cat показывает содержимое текстовых файлов, а также "cat" можно использовать для реализации следующих инструкций:

cat /proc/cpuinfo - отобразить информацию о процессоре

cat /proc/interrupts - показать прерывания

cat /proc/meminfo - проверить использование памяти

cat /proc/swaps - показать файл(ы) подкачки

cat /proc/version - вывести версию ядра

cat /proc/net/dev - показать сетевые интерфейсы и статистику по ним

cat /proc/mounts - отобразить смонтированные файловые системы

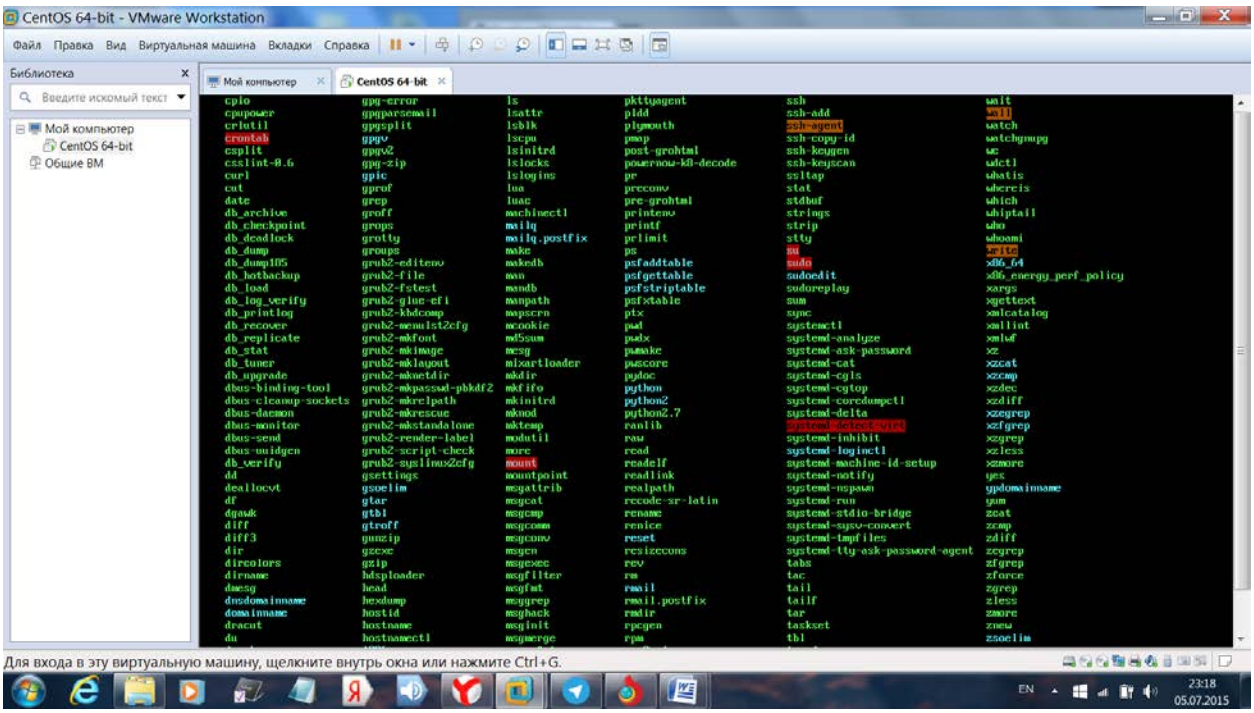
Команда cp позволяет копировать файл

-cp file1 file2 копировать файл file1 в файл file2

-cp dir/* . копировать все файлы директории dir в текущую директорию

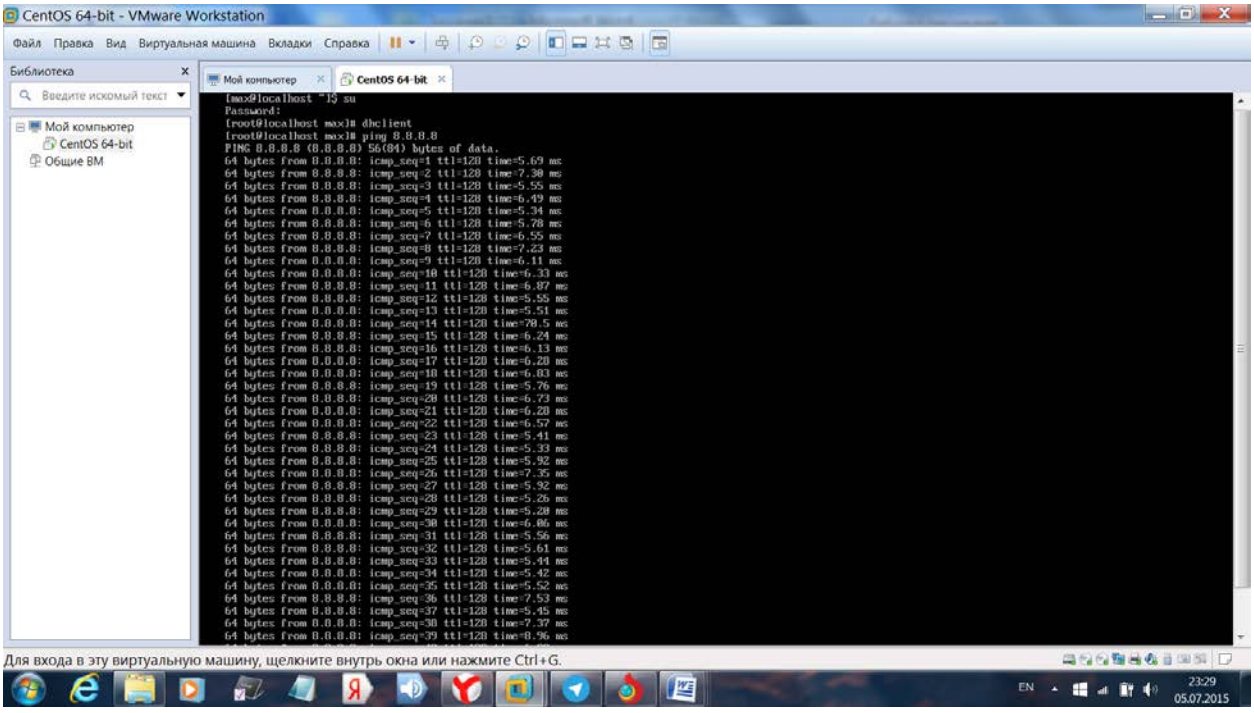
-cp -a /tmp/dir1 . копировать директорию dir1 со всем содержимым в текущую директорию

Команда ls показывает список фалов в текущей директории



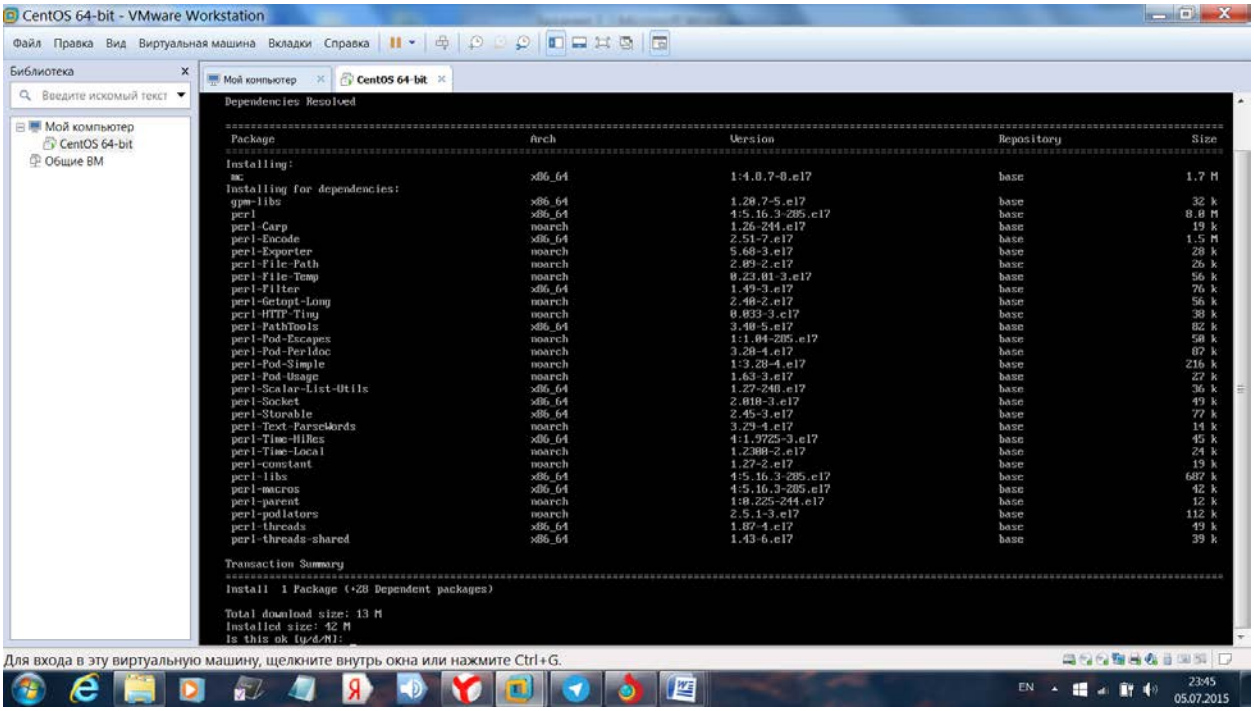
2)Подключение к сети

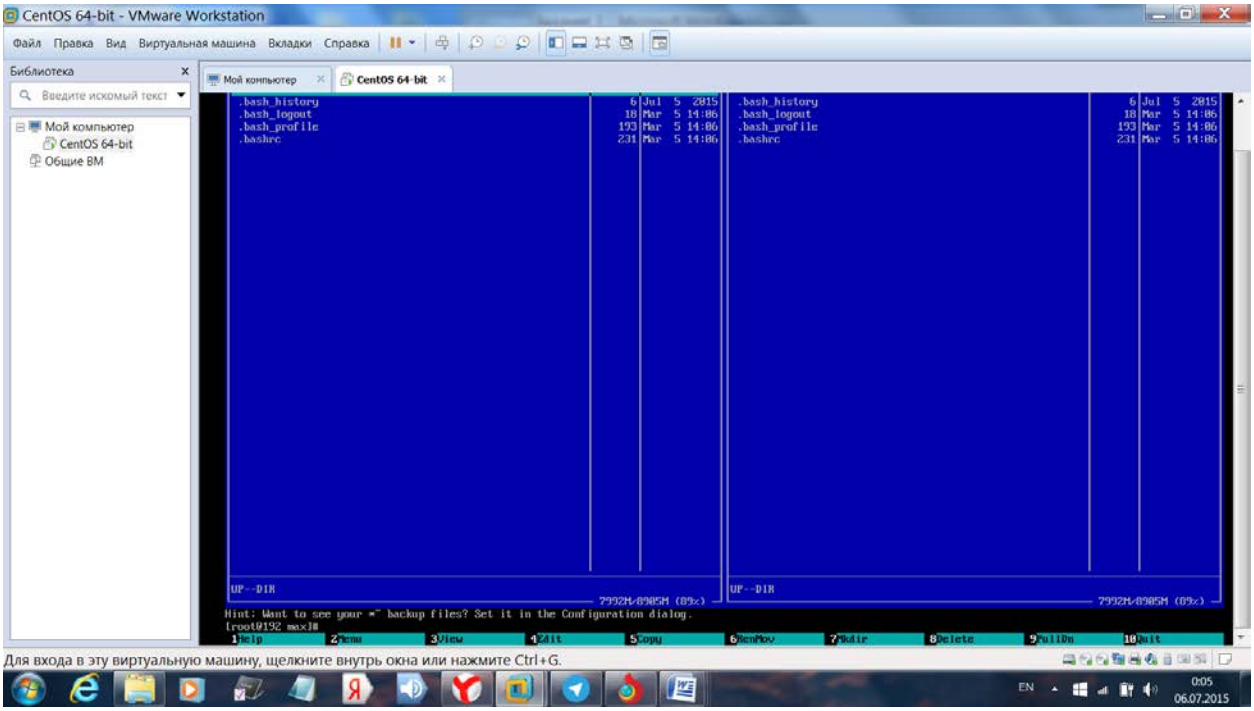
Чтобы настроить сеть, необходимо получить права рута, то есть вводим команду "su",для подключения к сети вводим "dhclient", а дальше проверяем подключение к интернету, введя "ping 8.8.8.8"



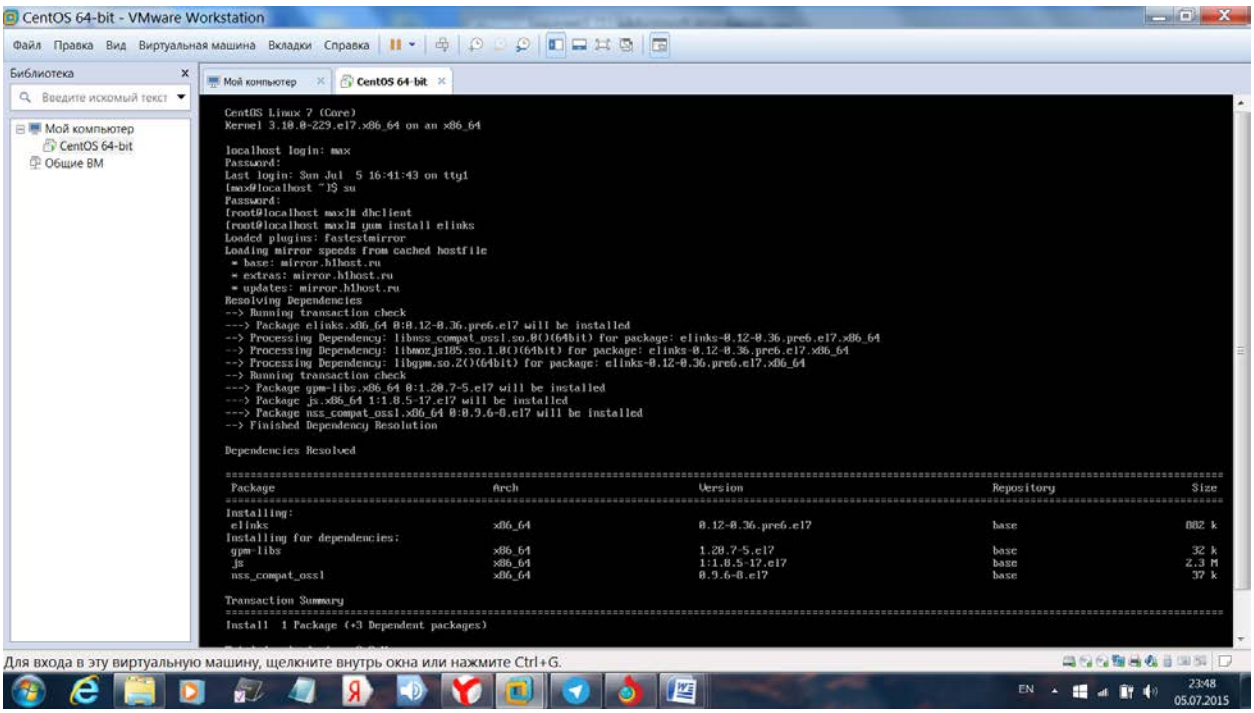
3) Установка дополнительных программ с помощью yum.

-установка mc при помощи команды “**yum install mc”:**

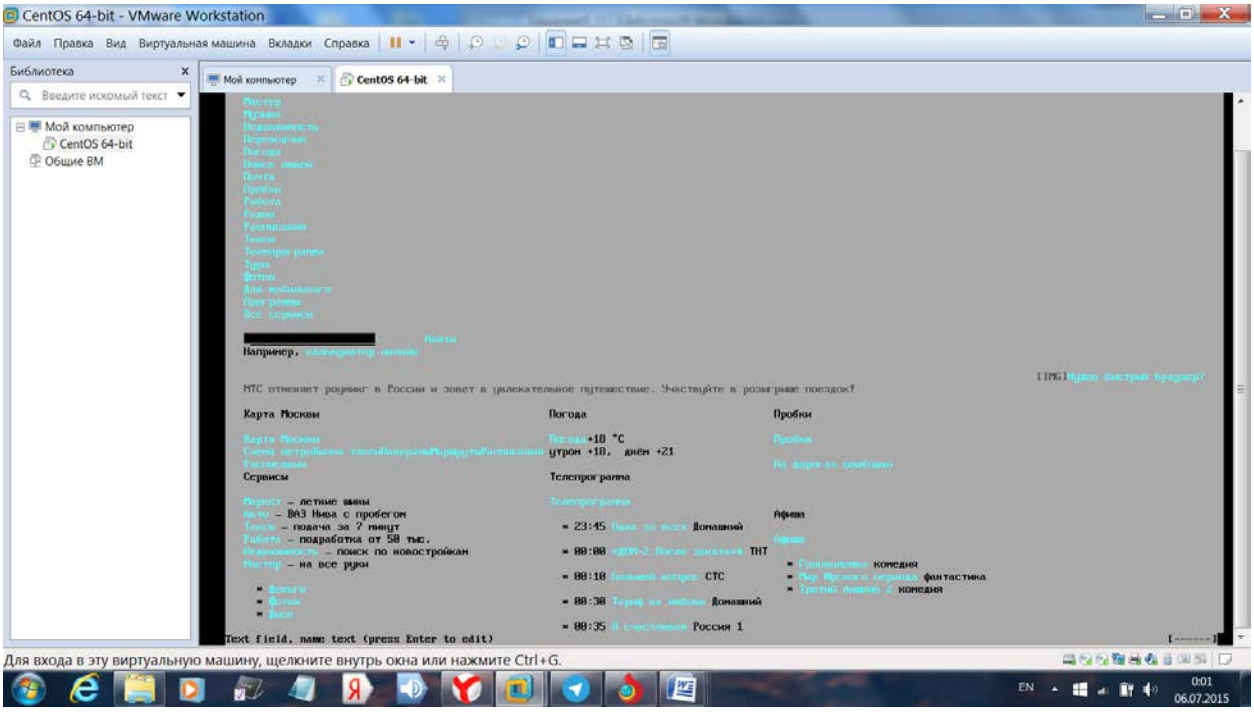




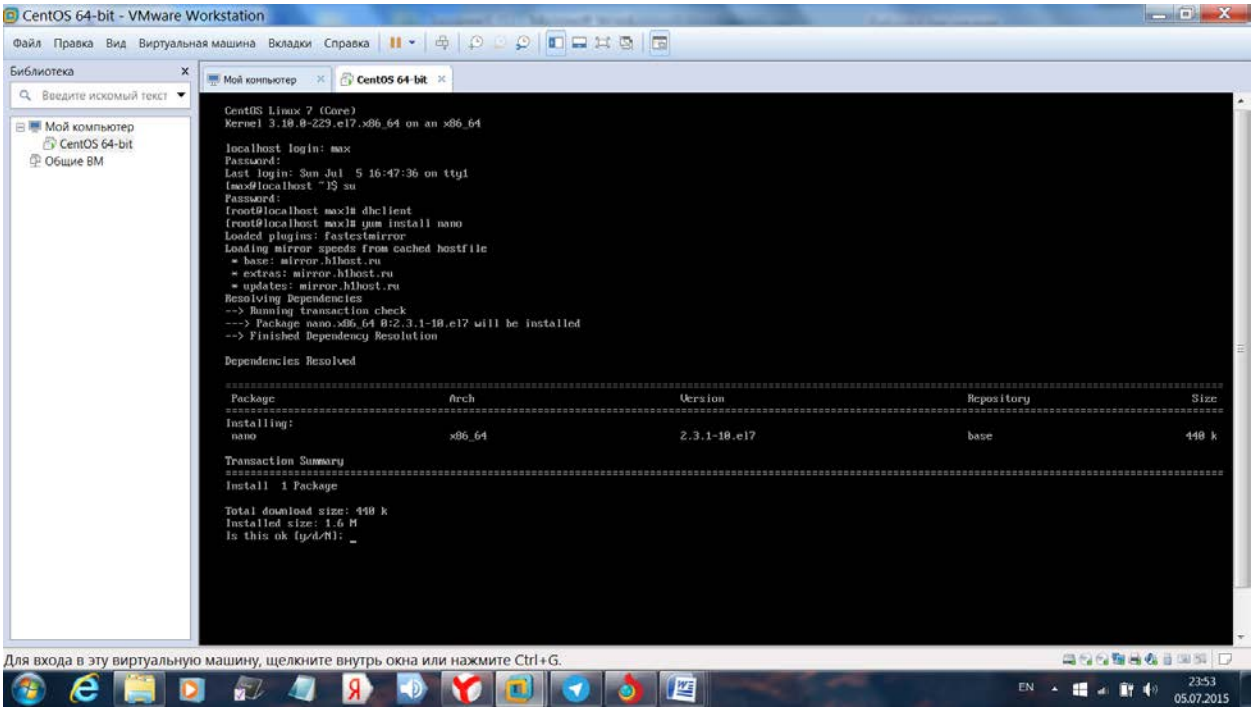
-Установка elinks с помощью команды “**yum install elinks”:**

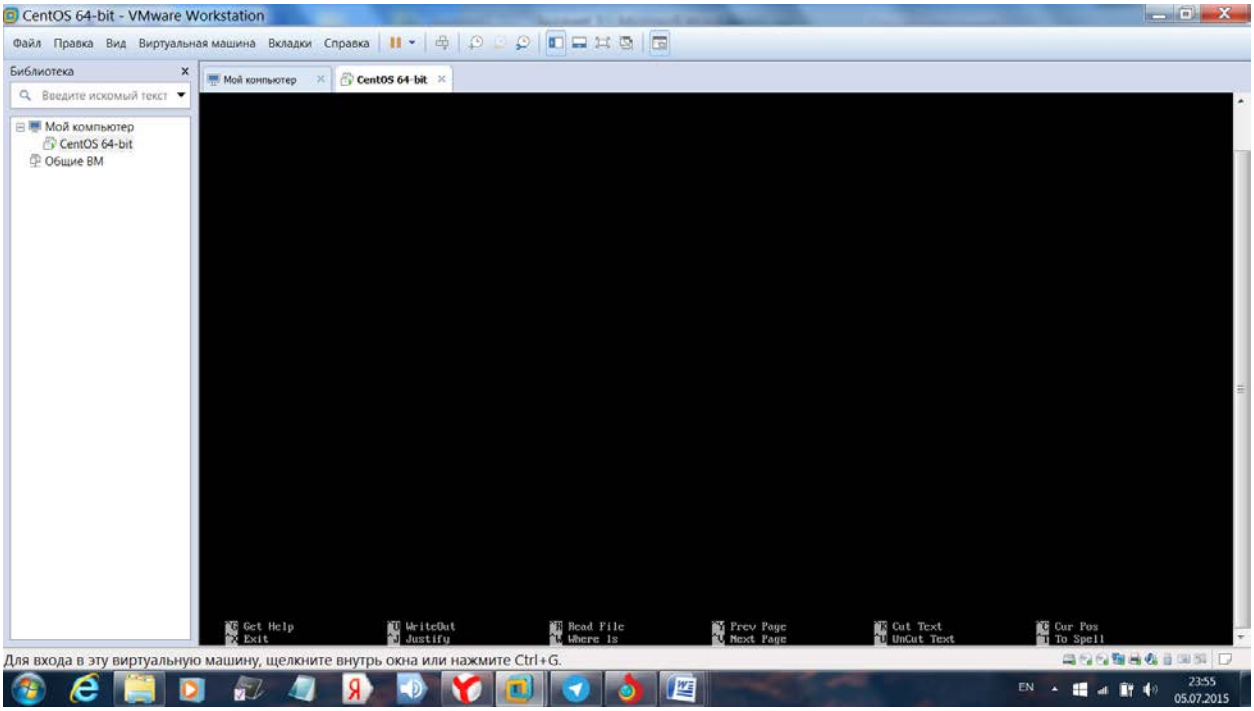

CentOS 64-bit - VMware Workstation
Файл Правка Вид Виртуальная машина Вкладки Справка
Библиотека
Введите искомый текст
Мой компьютер
CentOS 64-bit
Общие ВМ
Мой компьютер
CentOS 64-bit
CentOS Linux 7 (Core)
Kernel 3.10.0-229.el7.x86_64 on an x86_64
localhost login: max
Password:
Last login: Sun Jul 5 16:41:43 on tty1
[max@localhost ~]$ su
Password:
[root@localhost max]# dhclient
[root@localhost max]# yum install elinks
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: mirror.hlhost.ru
* extras: mirror.hlhost.ru
* updates: mirror.hlhost.ru
Resolving Dependencies
--> Running transaction check
---> Package elinks.x86_64 0:0.12-0.36.pre6.el7 will be installed
--> Processing Dependency: libnss_compat_ossl.so.0()(64bit) for package: elinks-0.12-0.36.pre6.el7.x86_64
--> Processing Dependency: libmozjs185.so.1.0()(64bit) for package: elinks-0.12-0.36.pre6.el7.x86_64
--> Processing Dependency: libgpm.so.2()(64bit) for package: elinks-0.12-0.36.pre6.el7.x86_64
--> Running transaction check
---> Package gpm-libs.x86_64 0:1.20.7-5.el7 will be installed
---> Package js.x86_64 1:1.8.5-17.el7 will be installed
---> Package nss_compat_ossl.x86_64 0:0.9.6-8.el7 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
Package Arch Version Repository Size
Installing:
elinks x86_64 0.12-0.36.pre6.el7 base 882 k
Installing for dependencies:
gpm-libs x86_64 1.20.7-5.el7 base 32 k
js x86_64 1:1.8.5-17.el7 base 2.3 M
nss_compat_ossl x86_64 0.9.6-8.el7 base 37 k
Transaction Summary
Install 1 Package (+3 Dependent packages)
Для входа в эту виртуальную машину, щелкните внутрь окна или нажмите Ctrl+G.
EN
23:48
05.07.2015

Яндекс:


CentOS 64-bit - VMware Workstation
Файл Правка Вид Виртуальная машина Вкладки Справка
Библиотека
Введите искомый текст
Мой компьютер
CentOS 64-bit
Общие ВМ
Мой компьютер
CentOS 64-bit
Карта Москвы
Погода
Пробки
Сервисы
Телепрограмма
Text field, name text (press Enter to edit)
Для входа в эту виртуальную машину, щелкните внутрь окна или нажмите Ctrl+G.
EN
0:01
06.07.2015

-Установка nano с помощью команды “**yum install nano”:**



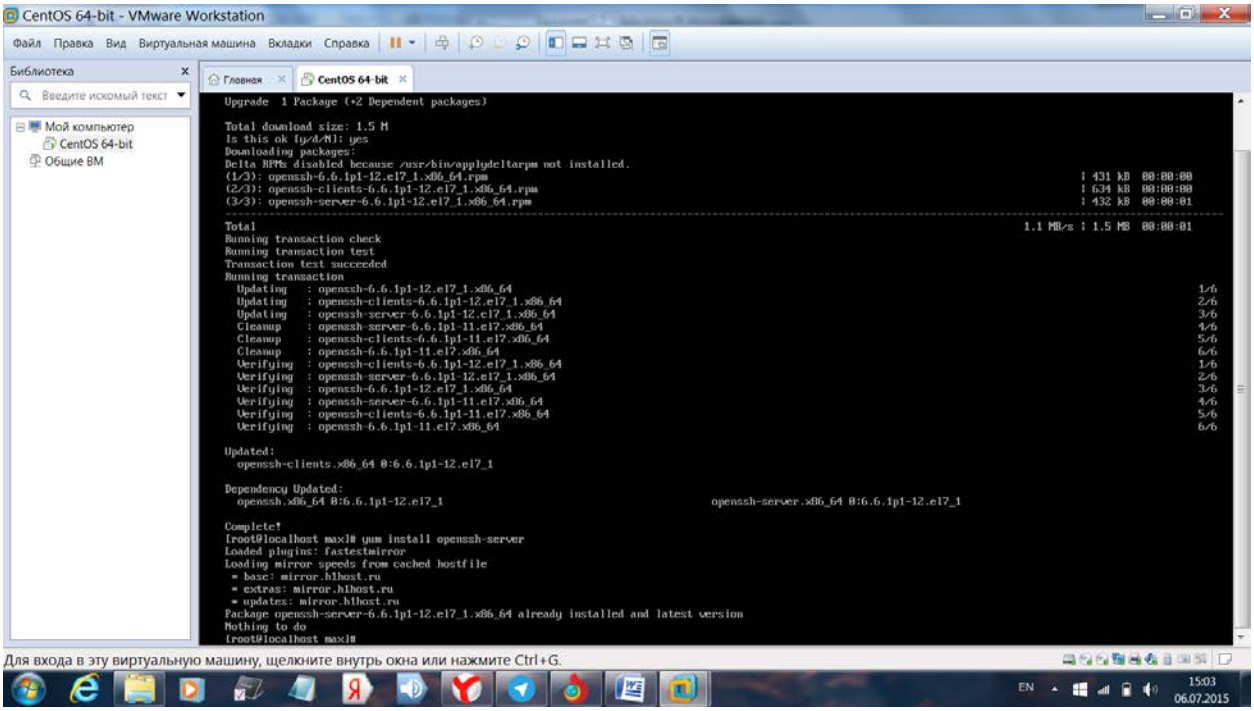


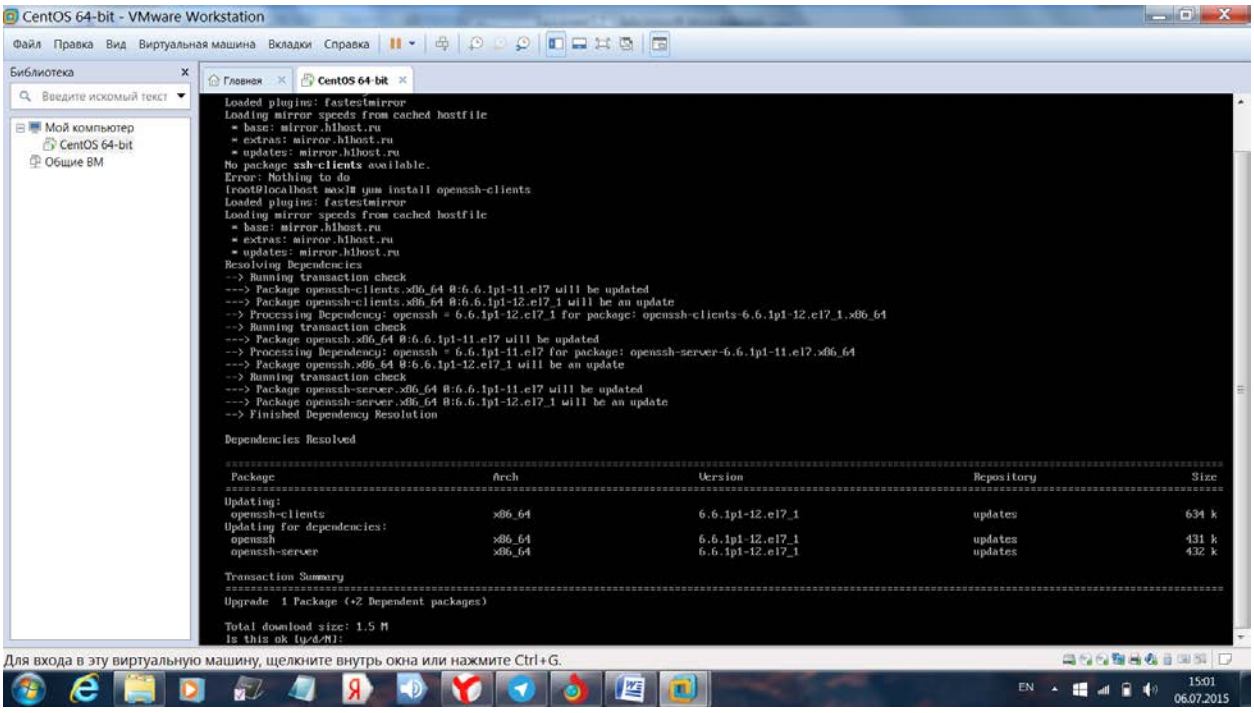
Задание 3

1)SSH это набор программ, которые позволяют регистрироваться на компьютере по сети, удаленно выполнять на нем команды, а также копировать и перемещать файлы между компьютерами. SSH организует защищенное безопасное соединение поверх небезопасных каналов связи.

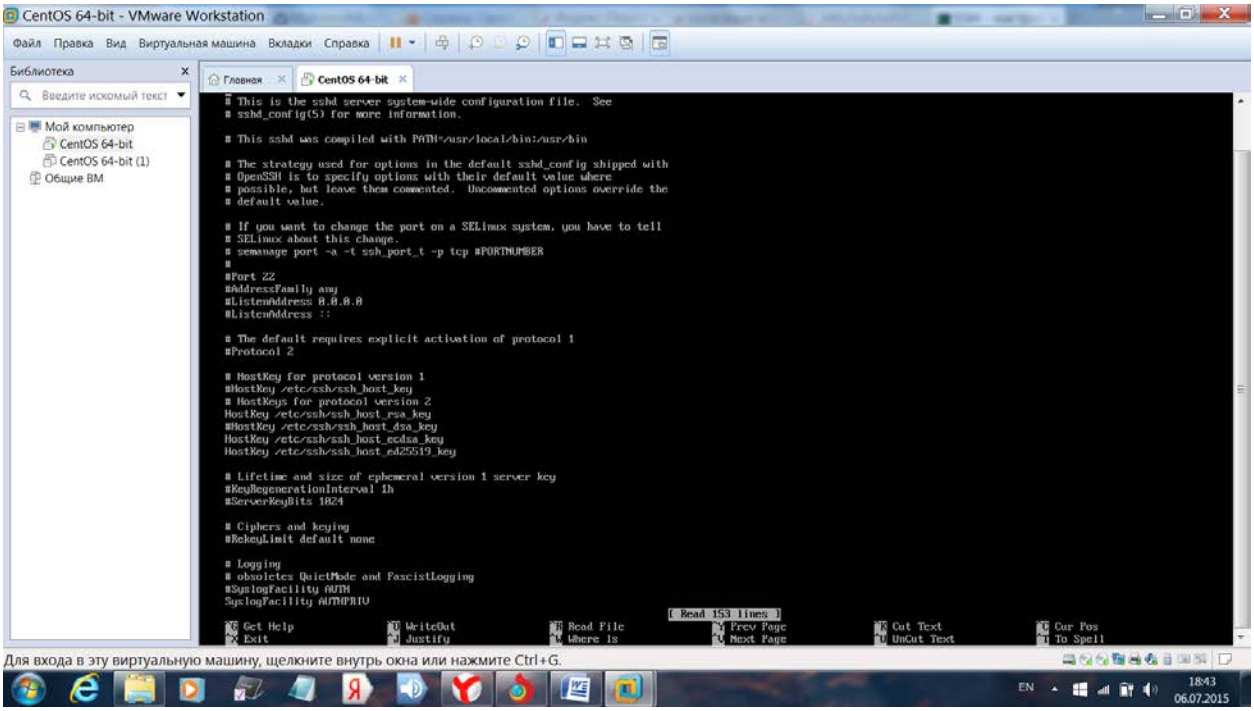
SSH предоставляет замены традиционным r-командам удаленного доступа с тем отличием, что они обладают повышенной безопасностью. Они выполняются поверх защищенных зашифрованных соединений, которые не позволяет прослушивать или подменять трафик. Кроме того, SSH может обеспечивать безопасное соединение для передачи любого другого трафика: например, почтовых сообщений или файлов.

2) Установка клиента и сервера с помощью команд “yum install openssh-clients” и “yum install openssh-server”





Заходим в файл с настройками с помощью команды **“nano /etc/ssh/sshd_config “**



Видим следующие директивы:

Port 22 — указывает порт по которому сервер будет ожидать входящего соединения. Заменим его на 2496

AddressFamily-семейство адресов которое должна использовать служба sshd(8). Допустимые значения: ``any'' ``inet'' (только IPv4) и ``inet6'' (только IPv6). Значение по умолчанию - ``any''

Далее идут строки:

"ListenAddress 0.0.0.0 "

"ListenAddress : :"

Эти строки отвечают за настройку разграничений по сетевым интерфейсам, сетевому адресу или имени компьютера. По умолчанию сервер «слушает» (принимает подключения) на всех сетевых интерфейсах.

Protocol — позволяет выбрать версию протокола 1 или 2. Рекомендуется протокол 2.

Строки HostKey необходимы для второй версии протокола SSH и отвечают за названия файлов

ключей и их расположение, эти ключи используются при аутентификации с ключом хоста

Следующие строки относятся к версии протокола 1 :

KeyRegenerationInterval 1h

ServerKeyBits 1024

Следующая группа параметров относится к аутентификации, первый параметр(**LoginGraceTime)** означает, что соединение будет разорвано через указанное количество секунд, если пользователь не войдёт в систему **.**

Второй параметр(**PermitRootLogin)** разрешает или запрещает вход по SSH под суперпользователем(root)

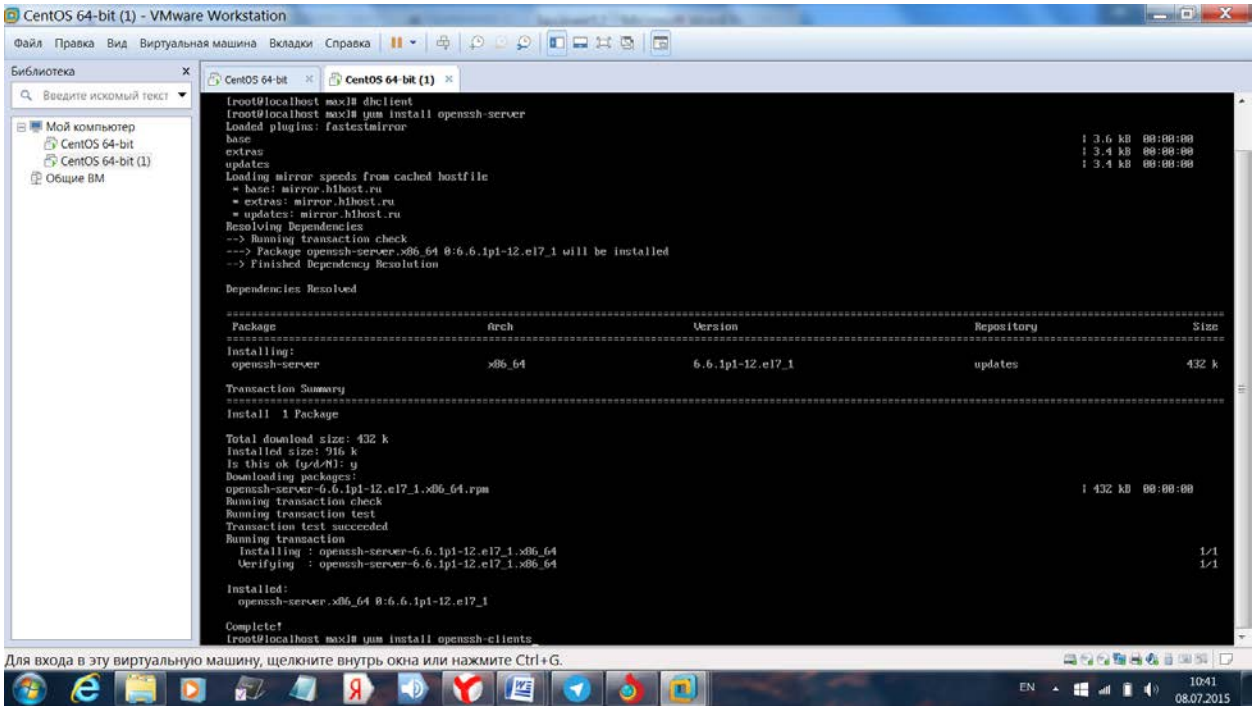
Третий параметр (**StrictModes**) включает проверку демоном ssh прав и владение домашним каталогом пользователя, который пытается получить удалённый доступ к компьютеру.

3) Заходим с одной виртуальной машины на другую используя ssh-соединение.

Для этого:

- создаём вторую виртуальную машину, производим соответсвующие настройки сети и установку софта, аналогично первой ВМ

- для настройкии ssh-соединения используем openssh-server, устанавливаем на обе ВМ пользуюсь командами “**yum install openssh-server”** и **“yum install openssh-clients”**



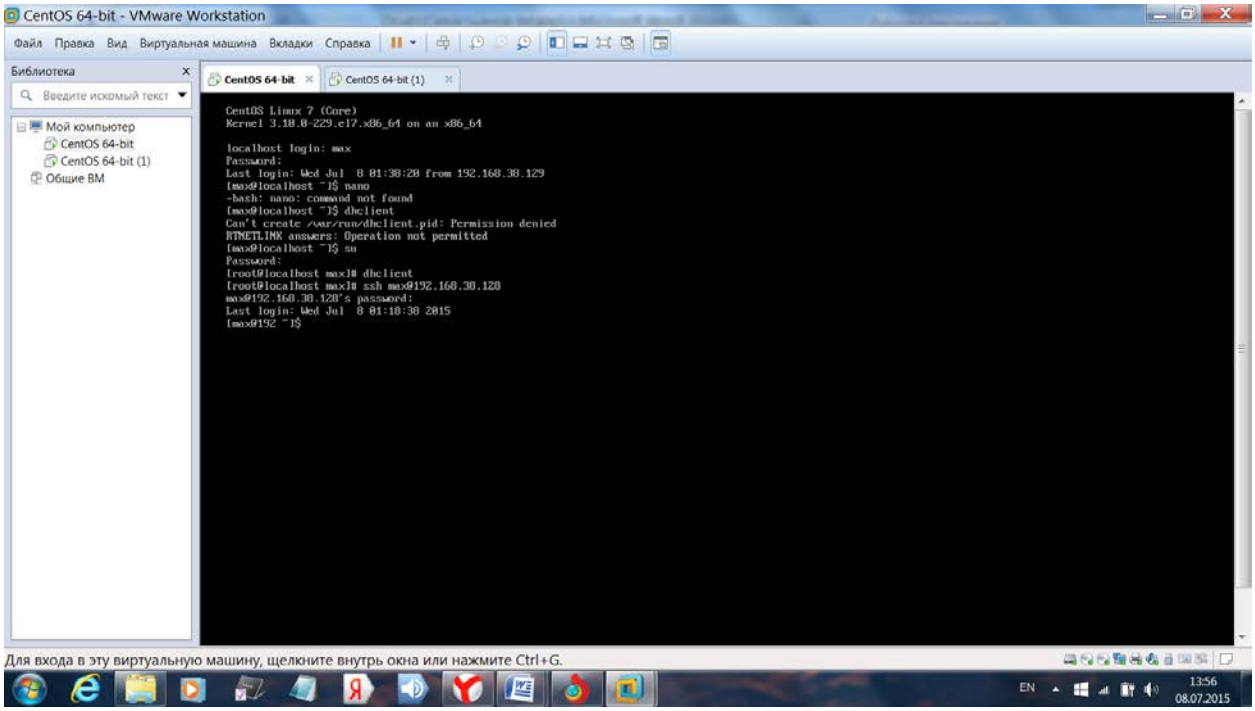
- с помощью текстового редактора nano и команды “nano /etc/ssh/sshd_config” открываем и редактируем конфигурационные файлы.

- узнаем ip с помощью команды **“ip a”**. Ip: 192.168.38.128

- теперь с основной системы соединяемся удалённым сервером с помощью команды **“ssh max@ip”**

- Затем получаем предупреждение, что подлинность хоста не установлена, видим ESCDA ключ и подтверждаем подключение, получаем уведомление о том, что наш айпи запомнен, вводим пароль

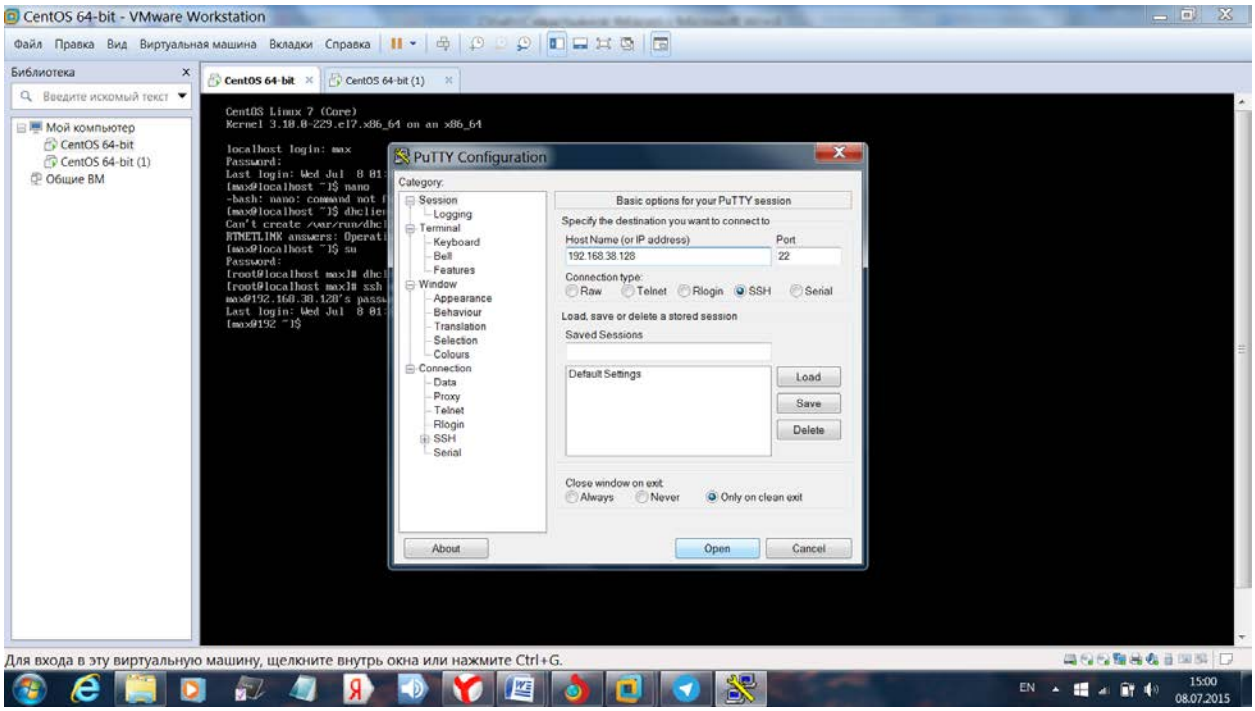
удалённого сервера и таким образом получаем доступ к удаленному хосту.



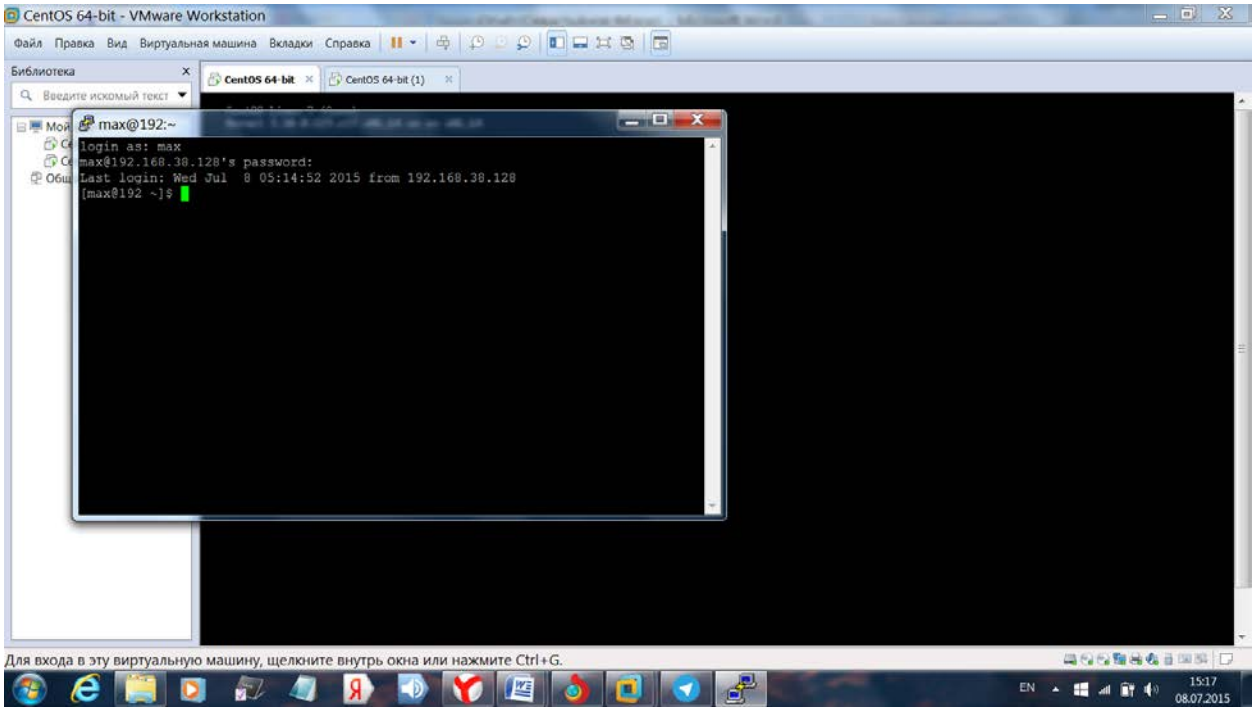
4) Научиться с помощью putty управлять виртуальной машиной с реальной машины

-Скачиваем putty с http://www.putty.org/

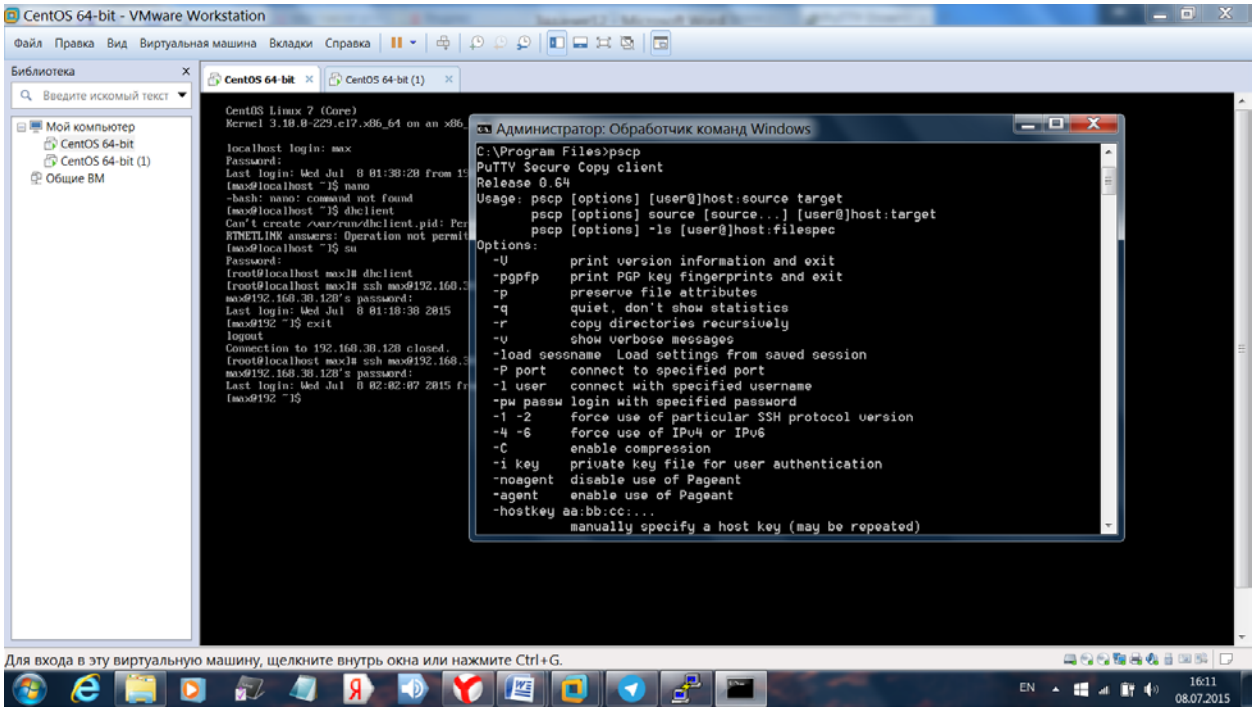
-Открываем putty и вводим ip виртуальной машины



-Открывается окно, в которое мы вводим логин и пароль для подключения к вм

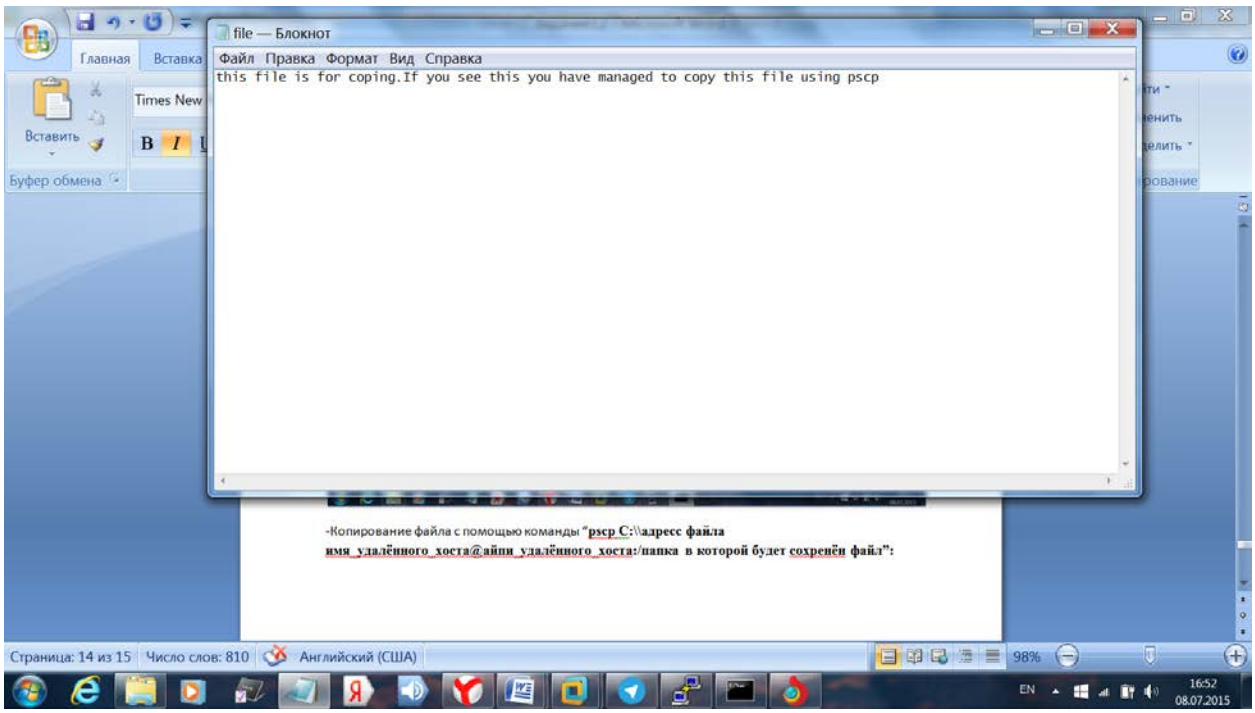


-Скачиваем утилиту pscp и открываем ее через Обработчик команд Wndows

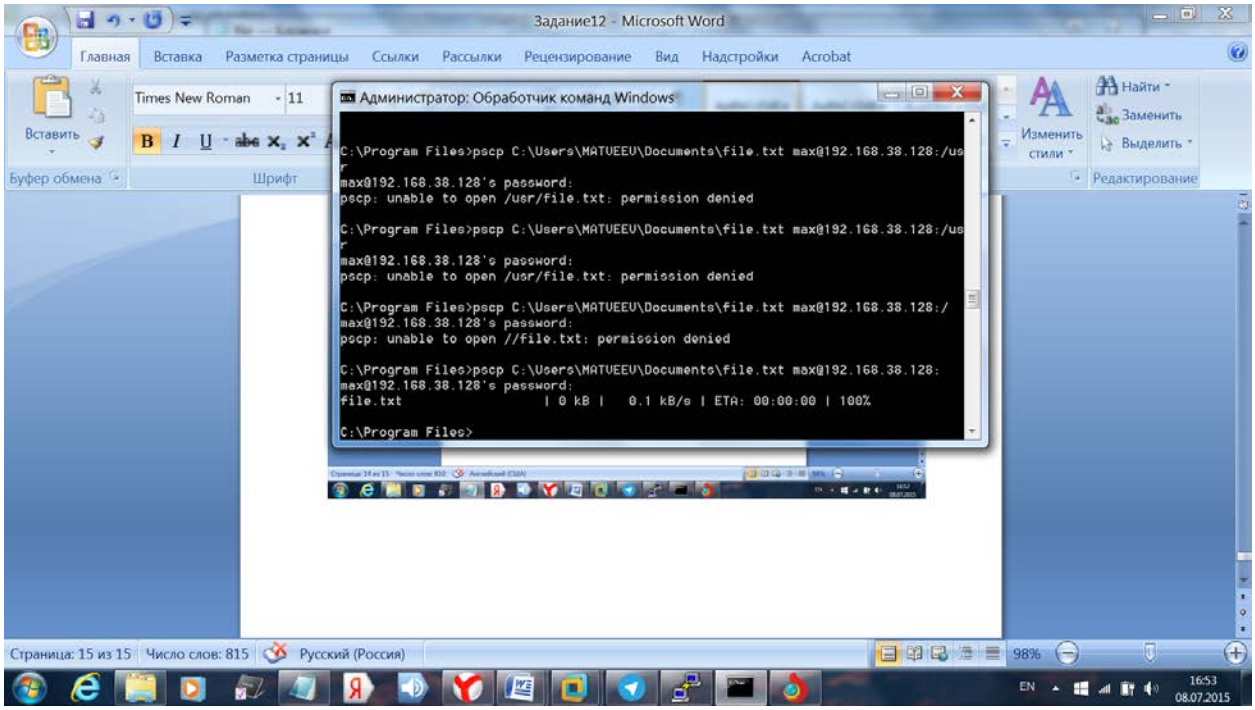


-Копирование файла с помощью команды “**pscp C:\\адресс файла имя_удалённого_хоста@айпи_удалённого_хоста:/папка в которой будет сохренён файл”:**

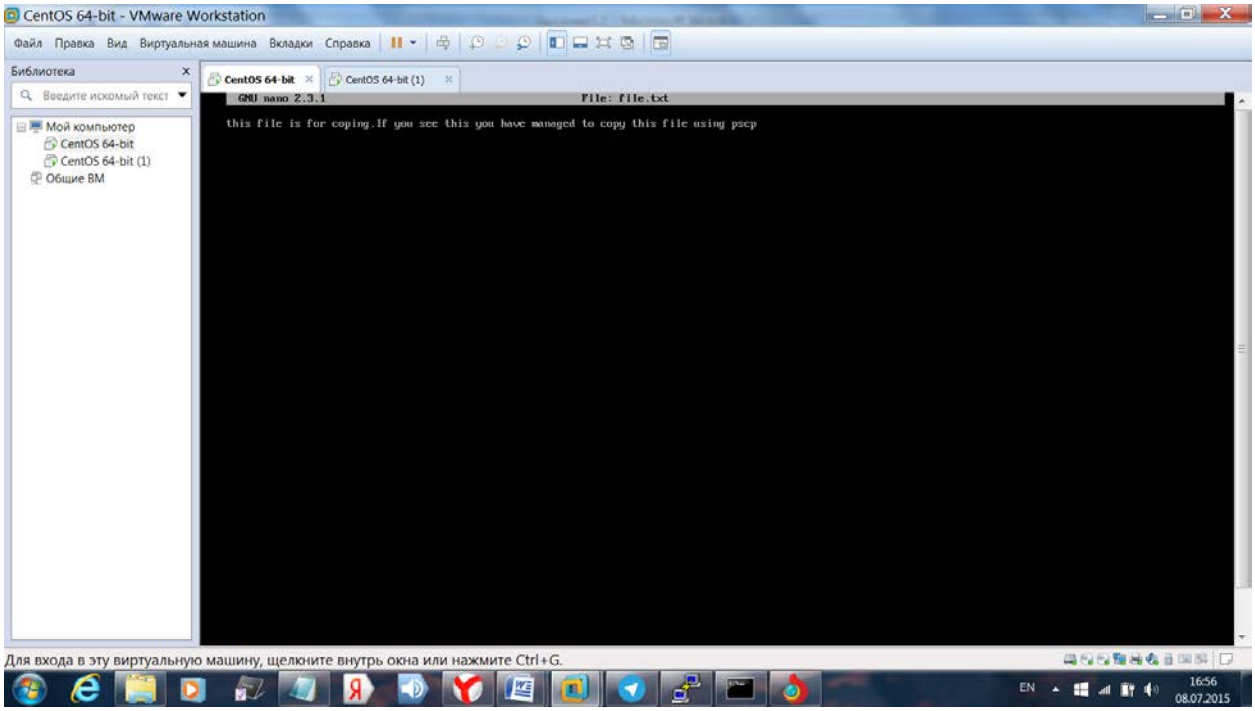
Создание файла на реальной машине:



Далее в командной строке вводим нашу команду

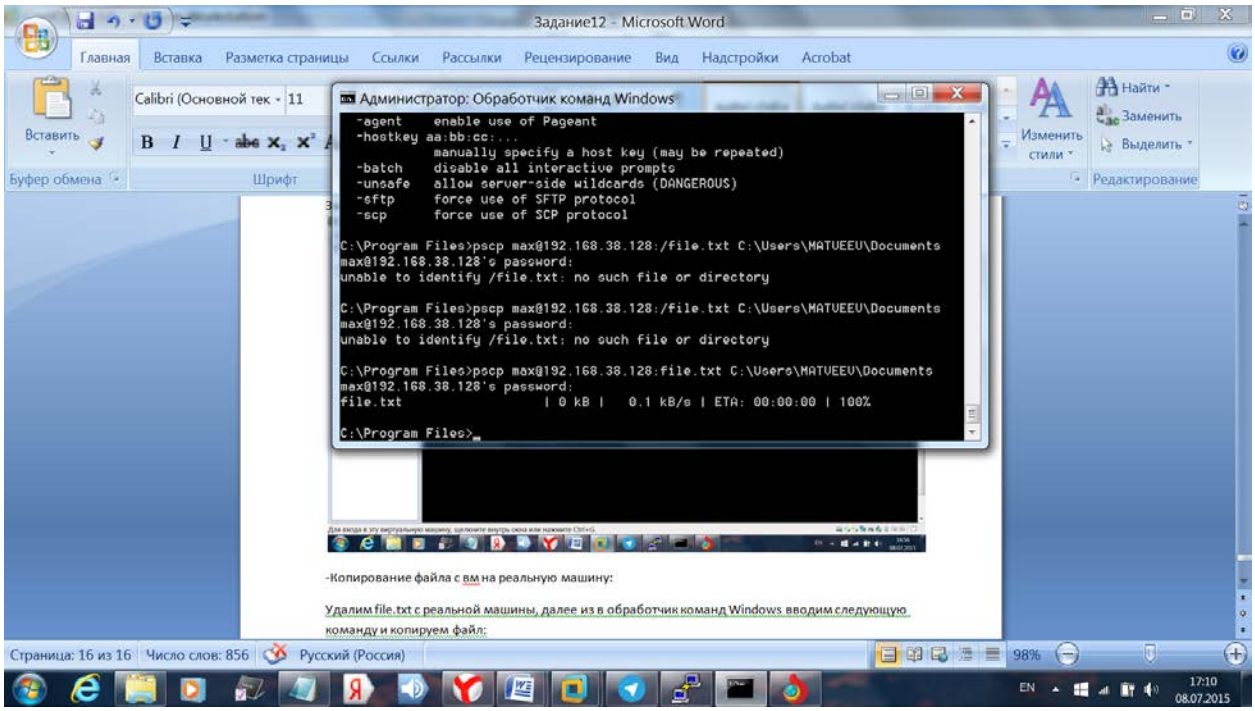


Заходим с вм и открываем скопированный файл с помощью nano:

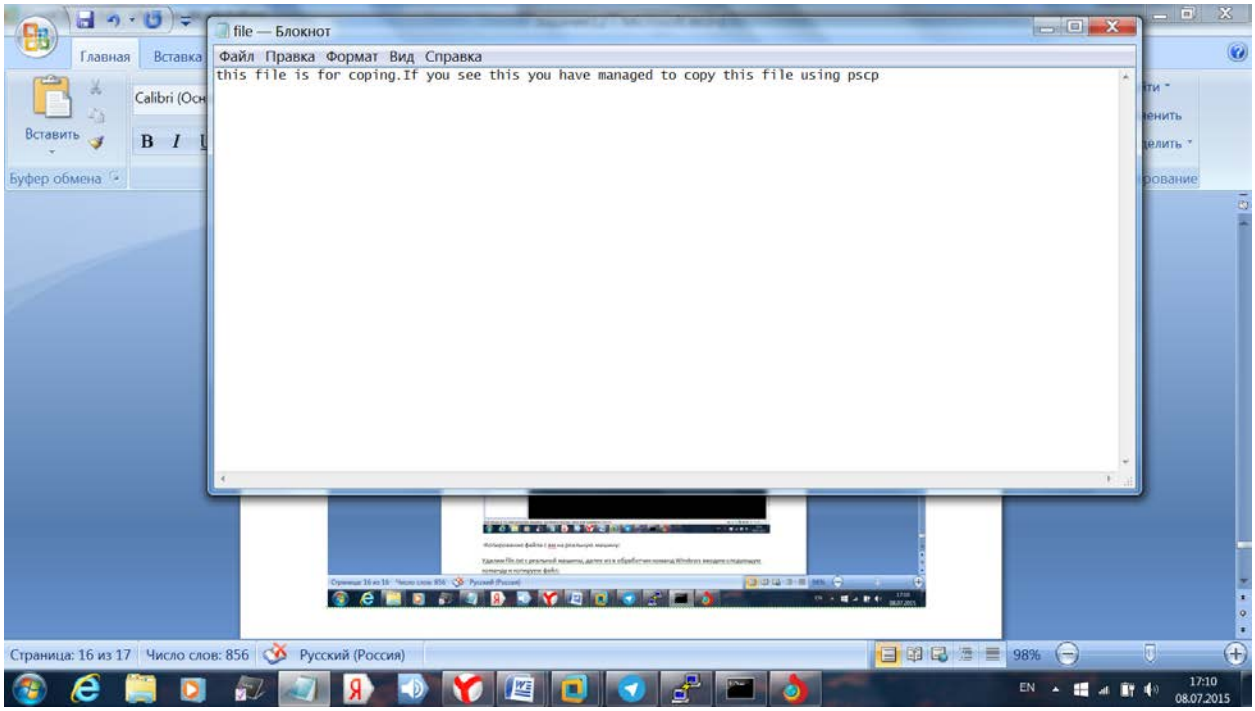


-Копирование файла с вм на реальную машину помощью команды **“pscp имя_удалённого_хоста@айпи_удалённого_хоста:/папка в которой харанится файл адрес нового расположения файла на реальной машине”**

Удалим file.txt с реальной машины, далее из в Обработчика команд Windows вводим следующую команду и копируем файл:



Затем открываем скопированный файл:



5) SSH ключи

Ключи SSH служат средством идентификации вас при подключении к серверу SSH с использованием **криптосистемы с открытым ключом** и **аутентификации вызов-ответ**. Одним из непосредственных достоинств этого метода перед традиционной идентификацией с помощью пароля является то, что вы можете быть авторизованы на сервере без регулярной необходимости отсылать ваш пароль через сеть. Даже если кто-либо будет прослушивать ваше соединение, у него не будет возможности перехватить и взломать ваш пароль, поскольку фактически он никогда не передается. Также использование для идентификации ключей SSH устраняет риск, связанный с брут-форс (brute-force) атаками, за счет существенного уменьшения шанса атакующего угадать правильные учетные данные.

Идентификация при помощи ключей SSH предоставляет дополнительную безопасность, а также может быть более удобным способом, чем традиционная идентификация при помощи пароля. При использовании вместе с программой, называемой агентом SSH, ключи SSH могут подключать вас к серверу или нескольким серверам без необходимости помнить и вводить ваш пароль отдельно для каждой системы.

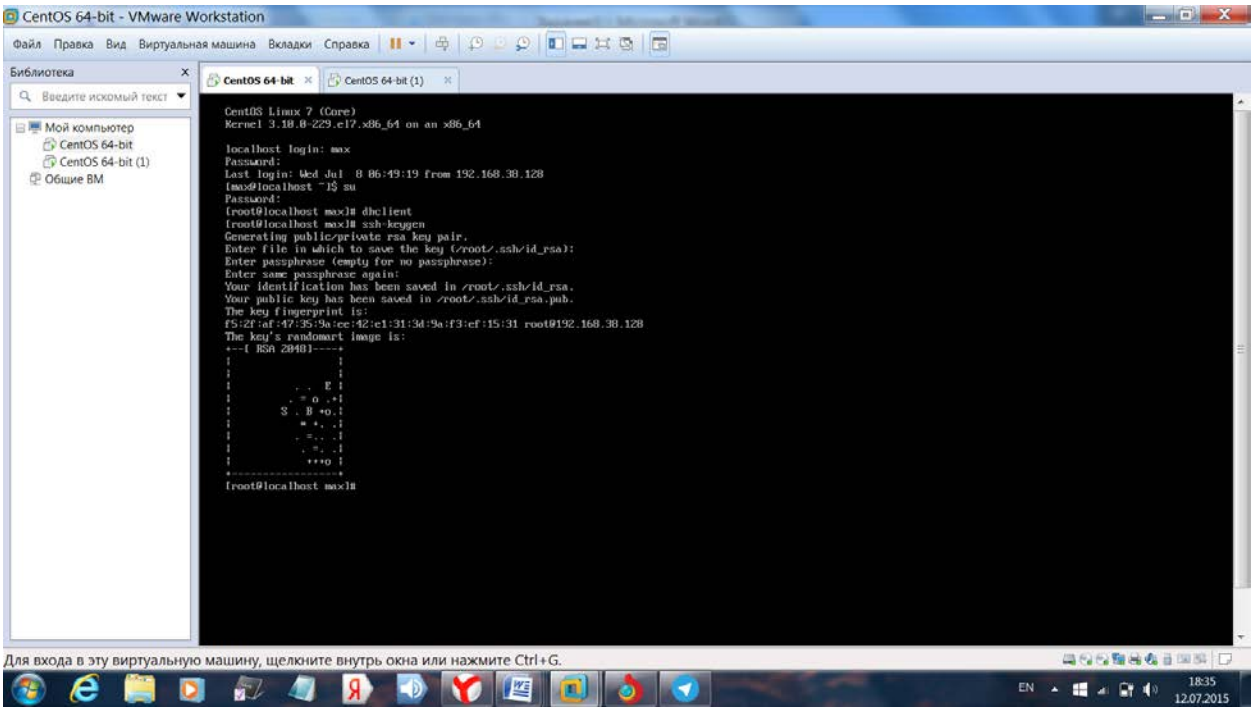
Ключи SSH являются парными: один из них - закрытый, другой - открытый. Закрытый ключ известен только вам, и он должен быть в безопасности. С другой стороны, открытый ключ может свободно раздаваться с любого сервера SSH, к которому вы хотите подключиться.

Когда у сервера SSH есть ваш открытый ключ в файле, и он видит, что вы запрашиваете соединение, он использует этот открытый ключ, чтобы создать и отправить вам т.н. вызов. Этот вызов является чем-то вроде зашифрованного сообщения, на которое должен поступить соответствующий ответ, чтобы сервер предоставил вам доступ. Безопасным это сообщение делает тот факт, что оно может быть прочитано только кем-то, у кого есть закрытый ключ. Открытый ключ может быть использован для зашифровки сообщения, но расшифровать то же самое сообщение он не сможет. Только вы, держатель закрытого ключа, будете иметь возможность корректно принять вызов и создать соответствующий ответ.

Этот этап вызов-ответ проходит незаметно для пользователя. До тех пор, пока у вас есть закрытый ключ, который обычно хранится в каталоге ~/.ssh/, ваш клиент SSH будет иметь возможность отправить правильный ответ серверу.

Поскольку закрытые ключи считаются конфиденциальной информацией, обычно они хранятся на диске в зашифрованном виде. По этой причине, когда запрашивается закрытый ключ, необходимо ввести пароль для расшифровки этого ключа. Внешне это может быть похоже на ввод пароля непосредственно на сервере SSH, но это не так: этот пароль используется только для расшифровки закрытого ключа в вашей локальной системе. Этот пароль не передается и не должен передаваться через сеть.

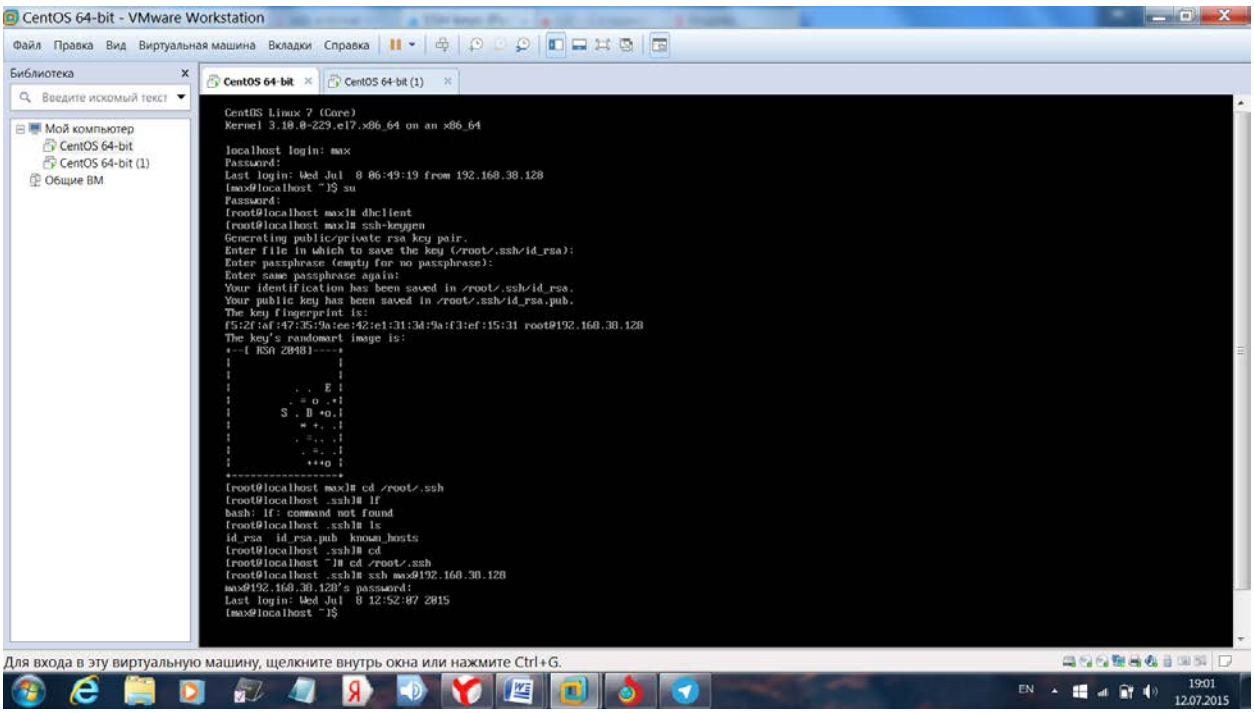
Пара ключей SSH может быть сгенерирована при помощи команды **ssh-keygen**:



Пара ключей сгенерирована

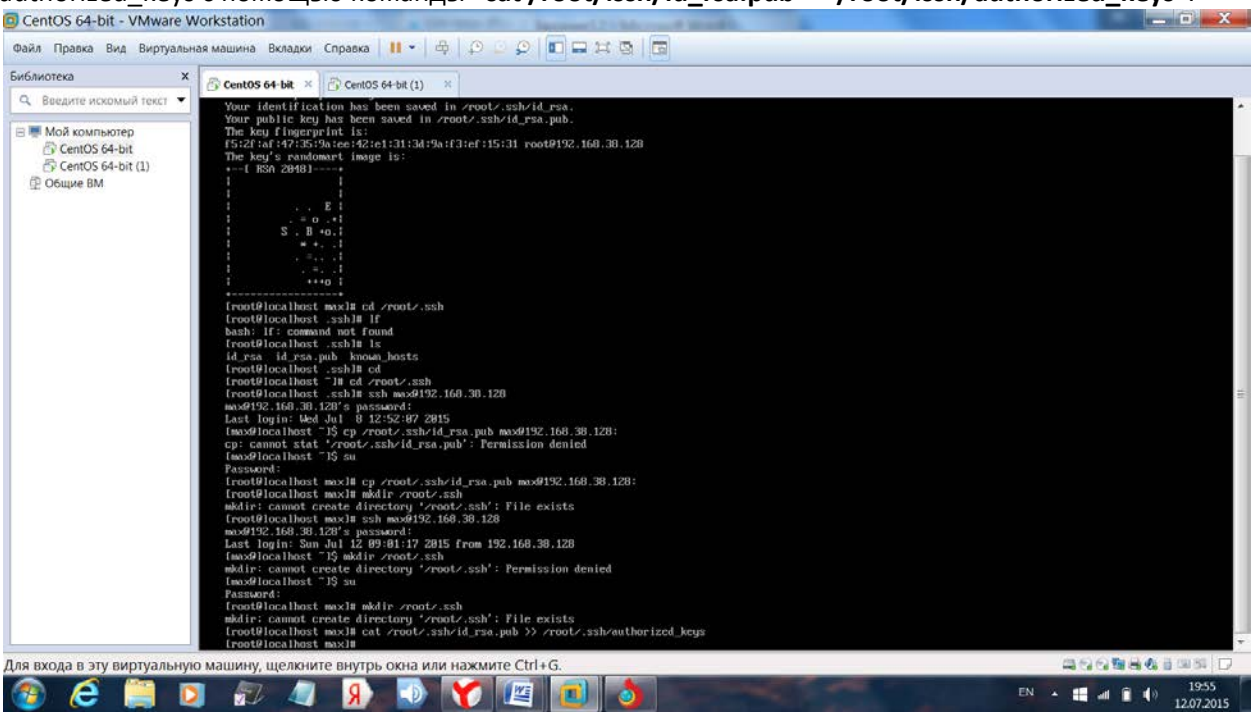
Далее зайдем в директорию /root/ssh и сможем увидеть три файла : id_rsa id_rsa.pub и known_hosts.

Подключаемся к удаленному хосту:

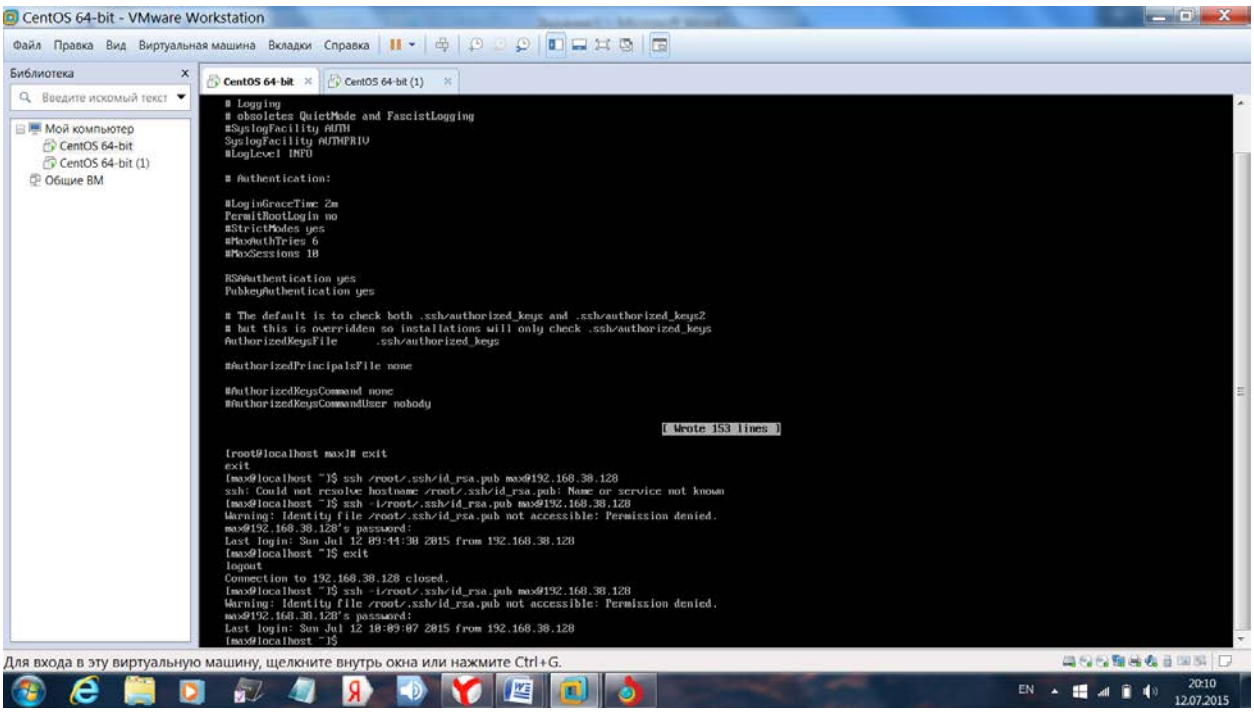


-С помощью команды “**cp /root/.ssh/id_rsa.pub max@192.168.38.128 :”** копируем файл id_rsa.pub в домашний каталог на удаленный сервер

-Создаем каталог /root/.ssh(если его нет) на удаленном сервере и добавляем id_rsa.pub в authorized_keys с помощью команды **“cat /root/.ssh/id_rsa.pub >> /root/.ssh/authorized_keys”**.



Теперь можно подключаться следующим образом:



6) FTP

FTP — стандартный протокол, предназначенный для передачи файлов по TCP-сетям (например, Интернет). Использует 21й порт. FTP часто используется для загрузки сетевых страниц и других документов с частного устройства разработки на открытые сервера хостинга.

Протокол построен на архитектуре «клиент-сервер» и использует разные сетевые соединения для передачи команд и данных между клиентом и сервером. Пользователи FTP могут пройти аутентификацию, передавая логин и пароль открытым текстом, или же, если это разрешено на сервере, они могут подключиться анонимно. Можно использовать протокол SSH для безопасной передачи, скрывающей (шифрующей) логин и пароль, а также шифрующей содержимое.

Первые клиентские FTP-приложения были интерактивными инструментами командной строки, реализующими стандартные команды и синтаксис. Графические пользовательские интерфейсы с тех пор были разработаны для многих используемых по сей день операционных систем. Среди этих интерфейсов как программы общего веб-дизайна вроде Microsoft Expression Web, так и специализированные FTP-клиенты (например, FileZilla).

FTP является одним из старейших прикладных протоколов, появившимся задолго до HTTP, и даже до TCP/IP, в 1971 году. В первое время он работал поверх протокола NCP[1]. Он и сегодня широко используется для распространения ПО и доступа к удалённым хостам.

Доступ к файлам на удаленном компьютере по протоколу FTP осуществляется с помощью программ, которые называются **FTP-клиентами** (в качестве примитивного FTP-клиента может использоваться www-браузер, например Opera, Firefox или Microsoft Internet Explorer). Практически все современные операционные системы включают также FTP-клиент для работы в командной строке, который так и называется «ftp».
Если у вас есть интернет, то вы можете получить доступ к большому количеству информации, расположенной в различных уголках Сети. Для использования FTP необходим так называемый FTP-клиент, подключающийся к FTP-серверу (сервер, откуда скачиваются данные).*Анонимный FTP* позволяет подключаться к серверу даже не будучи на нем зарегистрированным (не имея на нем логина и пароля). Как правило, в качестве логина (имени пользователя) указывается anonymous, а в качестве пароля — ваш e-mail. Это делается на больших серверах для того, чтобы каждый мог скачать, к примеру, бесплатный дистрибутив Linux или какие-нибудь другие полезные программы.

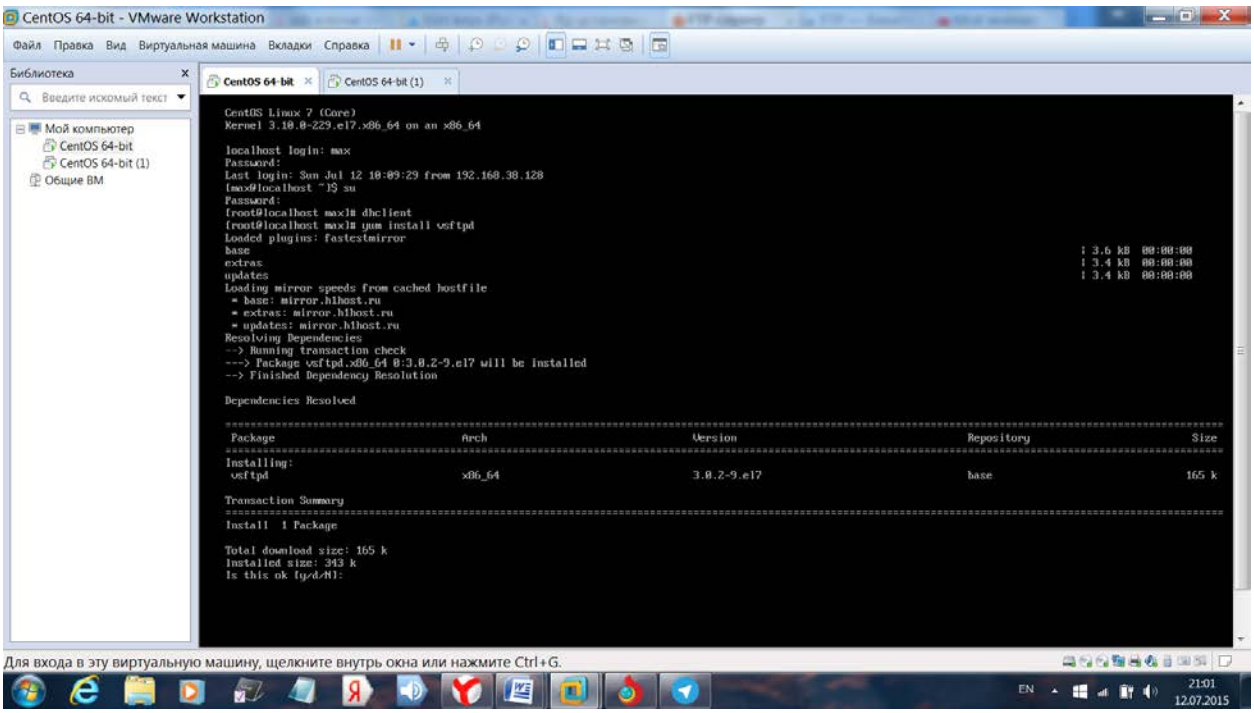
FTP-клиент общается с FTP-сервером при помощи специальных FTP-команд (в зависимости от сервера они могут незначительно отличаться, но в целом набор команд более-менее стандартен). Тогда почему бы не дать возможность пользователю вводить эти команды, чтобы без посредника просматривать содержимое FTP-сервера, закачивать файлы, устанавливать режимы передачи. Именно так и было на заре интернета. Первопроходцы глобальной паутины торопливо набирали команды для FTP-сервера прямо из консоли. Существует такая возможность и поныне. Правда, если для пользователей Unix-систем такая манера общения с FTP весьма привычна, то обладатели Windows наверняка даже и не подозревают, что их система позволяет это делать. Надо отметить, что использовать консольный вариант FTP-клиента поначалу очень даже увлекательно, а в некоторых случаях и чрезвычайно полезно.

SFTP (SSH File Transfer Protocol) является сетевым протоколом, который обеспечивает функциональность передачи файлов и манипулирование по любому надежному потоку данных. Он обычно используется с протоколом SSH-2 (TCP port 22) для обеспечения безопасной передачи файлов, но предназначен и для работы с другими протоколами.

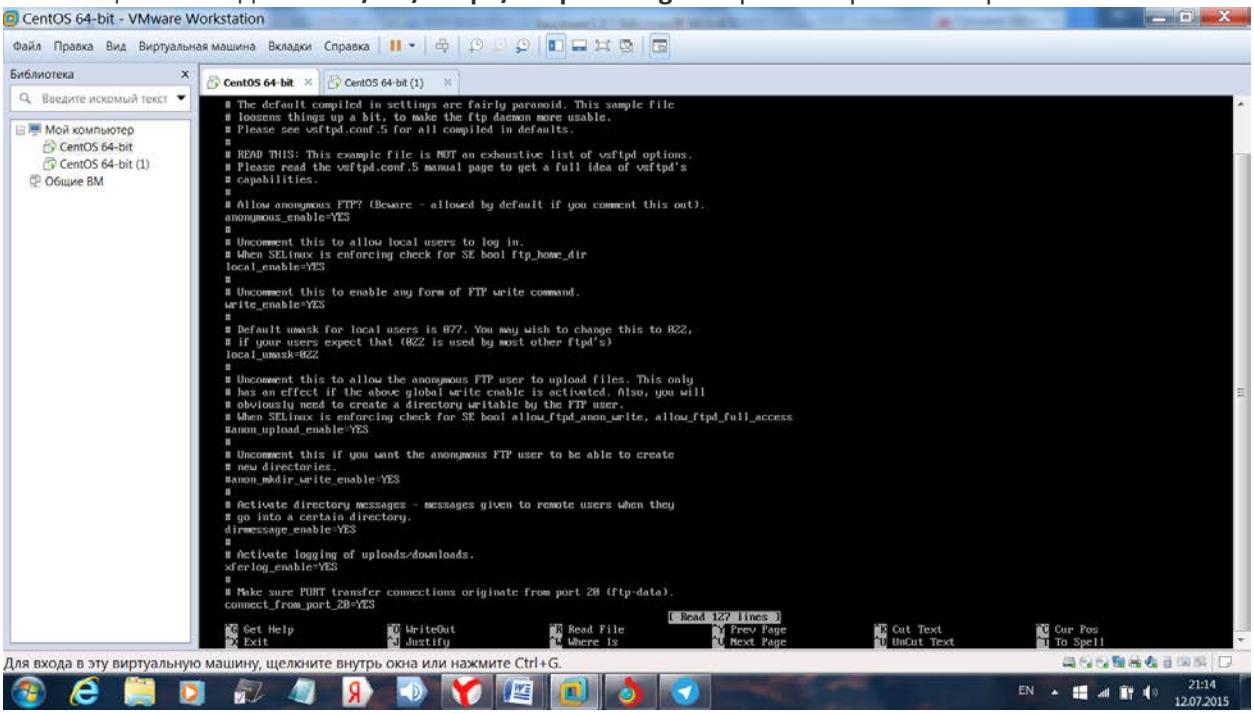
Плюсы: Хороший стандарт, который строго определяют большинство (если не все) аспектов деятельности. Имеет только одно подключение (нет необходимости подключения к DATA). Подключение всегда защищено .Список каталогов является однородным и машиночитаемым Протокол включает операции для разрешения и манипулирования атрибутом, захвата файла и отличается большей функциональностью

Минусы: Передача двоичная и не удобна для чтения пользователем. Ключами SSH труднее управлять и проверять. Стандарты определяют определенные вещи, как опциональные или рекомендованные, что приводит к определенным проблемам совместимости между разными названиями программного обеспечения от разных производителей. Нет копирования сервер-сервер и рекурсивных операций по удалению каталога Нет встроенной SSH / SFTP поддержки в VCL и .NET фреймворках.

-Устанливаем FTP с помощью команды **“yum install vsftpd”:**



-С помощью команды **“nano /etc/vsftpd/vsftpd.config”** открываем файл с настройками:



Внесем следующие изменения :

Опция	Описание
anonymous_enable=NO	Запрещаем анонимный доступ

local_enable=YES	Разрешаем доступ локальным пользователям
write_enable=YES	Даем пользователям FTP права на запись
connect_from_port_20=NO	Отключаем 20 порт, уменьшает привилегии VSftpd
chroot_local_user=YES	Chroot всех пользователей
local_umask=022	Устанавливаем маску 022, чтобы быть уверенными в том, что для всех файлов (644) и папок (755) которые мы закачиваем, устанавливаются соответствующие права

-Задаем пароль и логин для этого пользователя

- Также установим FTP клиент, для подключения к FTP серверу