

## **Содержание.**

<b>Лабораторная работа 1. Установка и первый вход в систему.</b>	стр. 4
Упражнение 1. Создание виртуальной машины в Oracle VirtualBox 6.1	
Упражнение 2. Запуск виртуальной машины и установка операционной системы.	стр. 5
Упражнение 3. Локальный вход в систему	стр. 13
<b>Лабораторная работа 1-Б. Настройка подключения к системе</b>	стр. 15
Упражнение 1. Настройка параметров PuTTY	стр. 15
Упражнение 2. Настройка подключения к системе	стр. 17
Упражнение 3. Базовые команды: получение информации о системе	стр. 19
 <b>Лабораторная работа 2. Файловая система Linux</b>	стр. 21
Упражнение 1. Получение информации о файловых системах	
Упражнение 2. Получение информации об использовании дискового пространства	
Упражнение 3. Просмотр атрибутов файлов	стр. 22
Упражнение 4. Утилиты для работы с файлами	
Упражнение 5. Использование утилит чтения файлов	стр. 24
Упражнение 6. Использование ссылок в Linux	стр. 25
Упражнение 7. Поиск файлов	
 <b>Лабораторная работа 2-Б. Введение в регулярные выражения</b>	стр. 27
Упражнение 1. Использование регулярных выражений с утилитой grep	
<b>Лабораторная работа 2-В. Основы работы в текстовых редакторах (vi, vim, sed, mcedit)</b>	стр. 29
Упражнение 1. Редакторы vi и vim	
Упражнение 2. Редактор sed и регулярные выражения	
Упражнение 3. Консольный файловый менеджер Midnight Commander	стр. 31
<b>Лабораторная работа 3. Послеустановочная настройка системы.</b>	стр. 32
Упражнение 1. Настройка сети с ifupdown	
Упражнение 2 (ОПЦИОНАЛЬНО). Перевод системы на systemd-networkd	
Упражнение 3. Локализация окружения пользователя user1	стр. 33

Упражнение 4. Настройка системного времени	стр. 34
Упражнение 5. Настройка имени системы	
Упражнение 6. Управление модулями ядра	стр. 35
 <b>Лабораторная работа 4. Управление подсистемой хранения данных</b>	 стр. 37
Упражнение 1. Добавление дисков	
Упражнение 2. Создание разделов	стр. 39
Упражнение 3. Создание файловой системы	
Упражнение 4. Монтирование раздела	
Упражнение 5. Работа с томами LVM	стр. 40
 <b>Лабораторная работа 5. Процессы в Linux</b>	 стр. 42
Упражнение 1. Мониторинг процессов	
Упражнение 2. Использование сигналов	
Упражнение 3. Переключение процесса в фоновый режим	
Упражнение 4. Запуск процесса в фоновом режиме	
 <b>Лабораторная работа 6-А. Управление конфигурацией</b>	 стр. 44
Упражнение 1. Использование systemctl	
Упражнение 2. Определение переменных	
 <b>Лабораторная работа 6-Б. Введение в использование сценариев</b>	 стр. 46
Упражнение 1. Организация ввода и вывода данных	
Упражнение 2. Конструкция If-Then-Else:	
Упражнение 3. Конструкция CASE	стр. 47
Упражнение 4. Цикл While	стр. 48
 <b>Лабораторная работа 7. Управление пользователями и группами</b>	 стр. 49
Упражнение 1. Работа с пользовательскими учетными записями	

Упражнение 2. Блокировка учетных записей стр. 50

Упражнение 3. Настройте использование ключей для аутентификации ssh

**Лабораторная работа 8. Управление доступом и привилегиями** стр. 52

Упражнение 1. Настройка sudo

Упражнение 2. Разрешения в Linux/UNIX

Упражнение 3. POSIX ACL

**Лабораторная работа 9-А. Управление ПО** стр. 53

Упражнение 1. Установка и удаление программ

Упражнение 2. Загрузка пакетов из репозитория

**Лабораторная работа 9-Б (Опциональная)** стр. 54

Упражнение 1. Сборка и установка ПО из исходников

Упражнение 2. Добавление локального репозитория стр. 55

**Лабораторная работа 10-А. Резервное копирование и архивация** стр. 57

Упражнение 1. Архивирование каталога

Упражнение 2. Создание запланированной задачи

**Лабораторная работа 10-Б. Аудит доступа** стр. 58

Упражнение 1. Настройка аудита доступа к каталогу пользователя

Требования к стенду слушателей (минимальные)

Свободного пространства в ОЗУ для виртуальной машины - не менее 512 Mb

Свободное пространство на диске не менее 8 Gb

Установленный Oracle VirtualBox (<https://www.virtualbox.org/>)

Для 64 разрядной версии дистрибутива поддержка виртуализации CPU (Intel-VT или AMD-V)

Для 32-разрядной поддержка виртуализации не требуется.

Установочный образ дистрибутива Debian 10:

Для 64 разрядной версии:

<https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-10.6.0-amd64-netinst.iso>

Для 32-разрядной:

<https://cdimage.debian.org/debian-cd/current/i386/iso-cd/debian-10.6.0-i386-netinst.iso>

Или, если нет подключения к Интернет, можно использовать полный установочный образ.

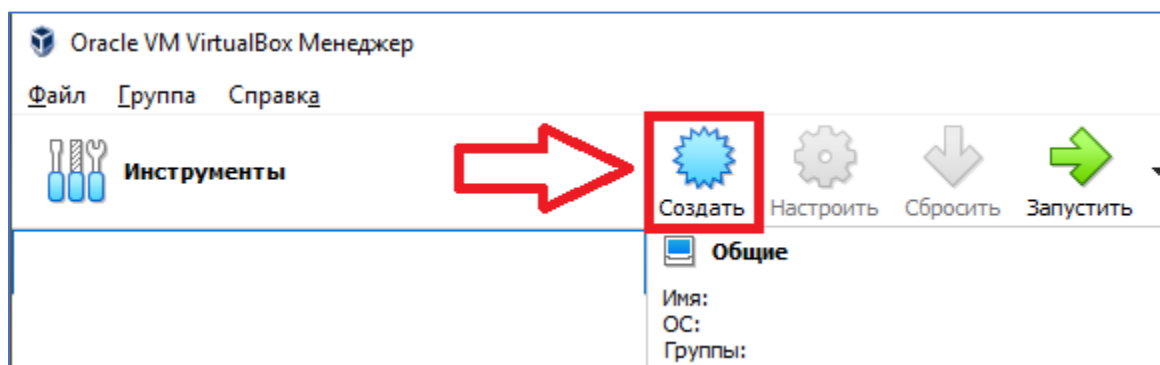
Полные установочные образы для установки без соединения с Интернет:

<https://www.debian.org/CD/http-ftp/>

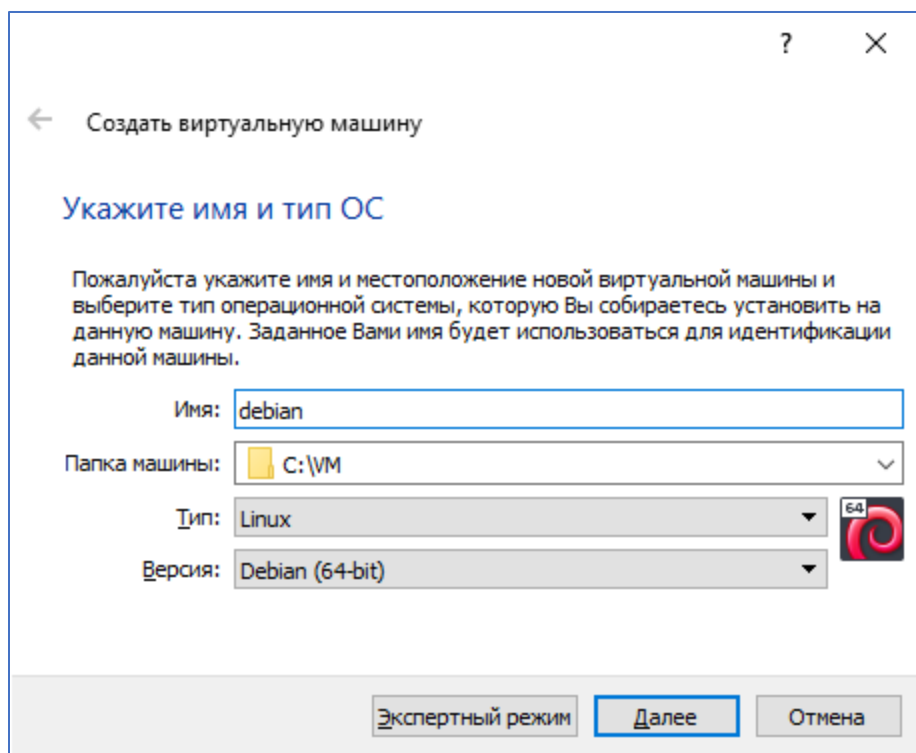
Лабораторная работа 1. Установка и первый вход в систему.

Упражнение 1. Создание виртуальной машины в Oracle VirtualBox 6.1

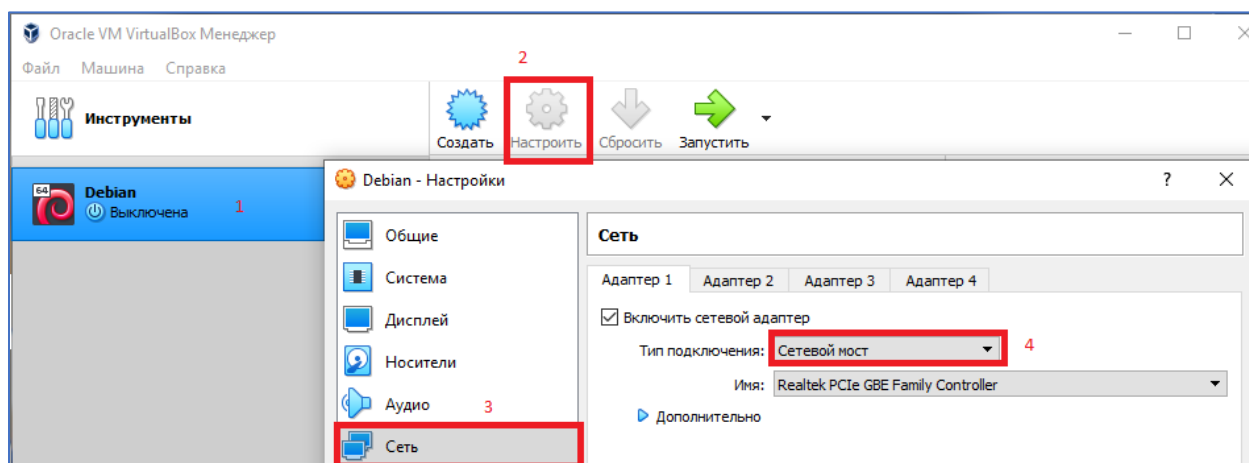
1. В панели менеджера VirtualBox нажмите «Создать»



2. Введите произвольное имя для идентификации виртуальной машины (далее VM)  
Поле «Папка машины» можно оставить по умолчанию.  
В поле «Тип» выберите Linux  
В поле «Версия» выберите Debian (64-bit).  
**Прим.** для 32-разрядных систем можно использовать не 64-bit вариант.



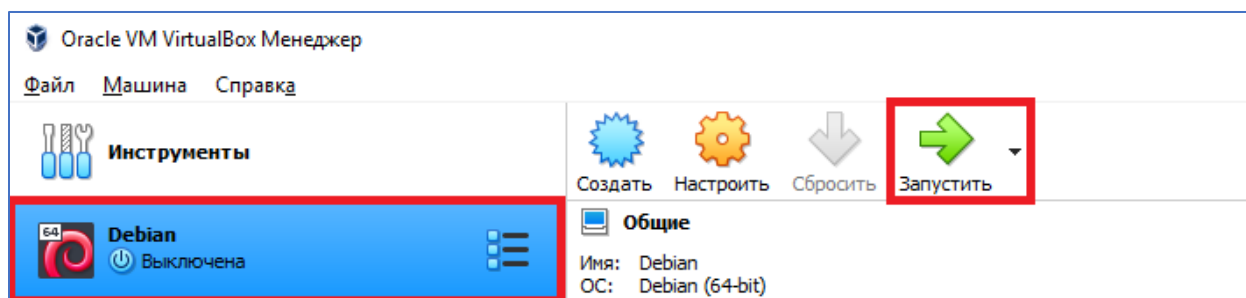
3. На остальных шагах оставьте умолчательные значения.
4. После создания VM, но до ее включения зайдите в конфигуратор ее настроек. Для этого выделите курсором VM и нажмите «Настроить» на панели.



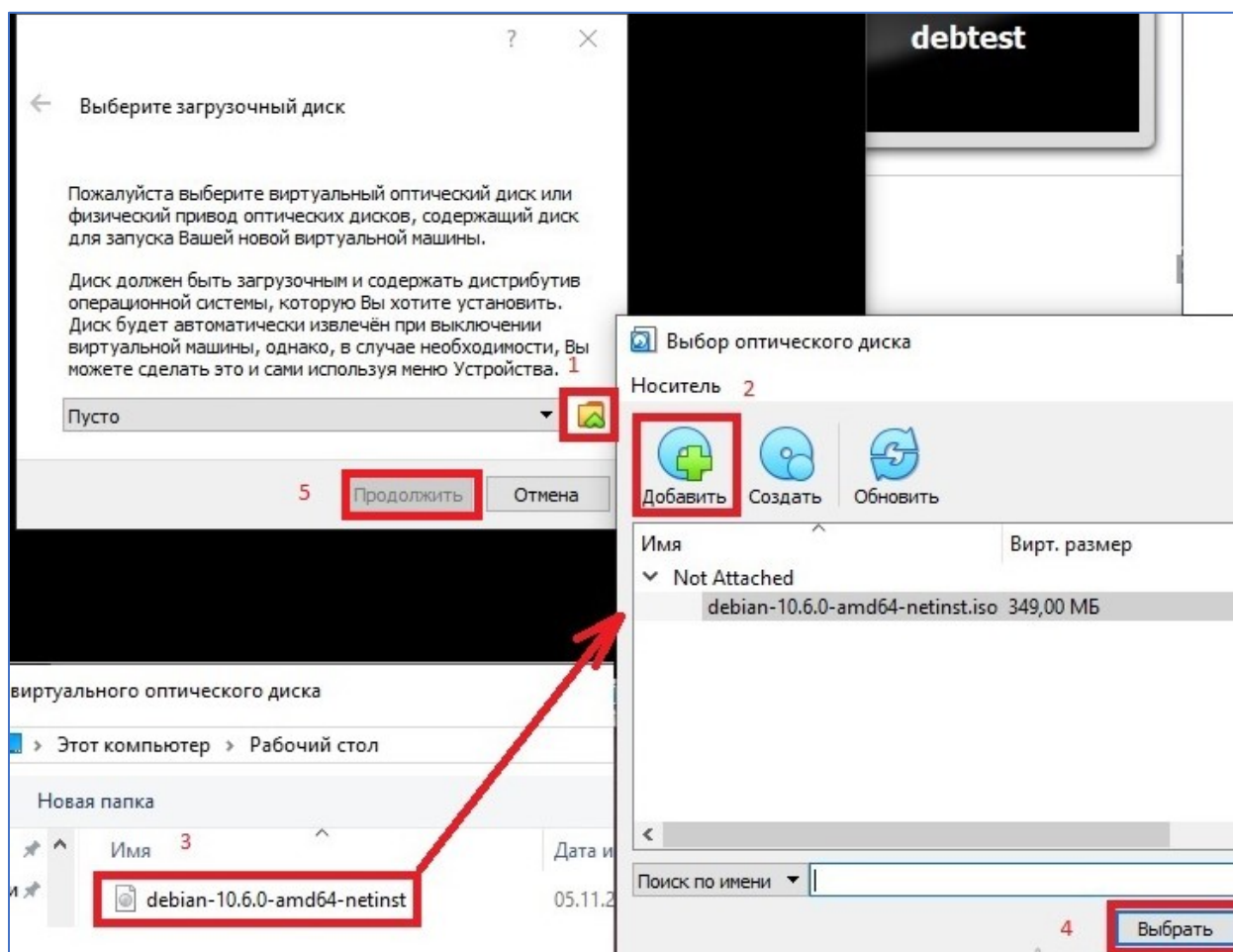
5. Выделите пункт «Сеть» и на вкладке «Адаптер 1» в выпадающем списке «Тип подключения» выберите «Сетевой мост» и нажмите кнопку «Ок»

## Упражнение 2. Запуск виртуальной машины и установка операционной системы.

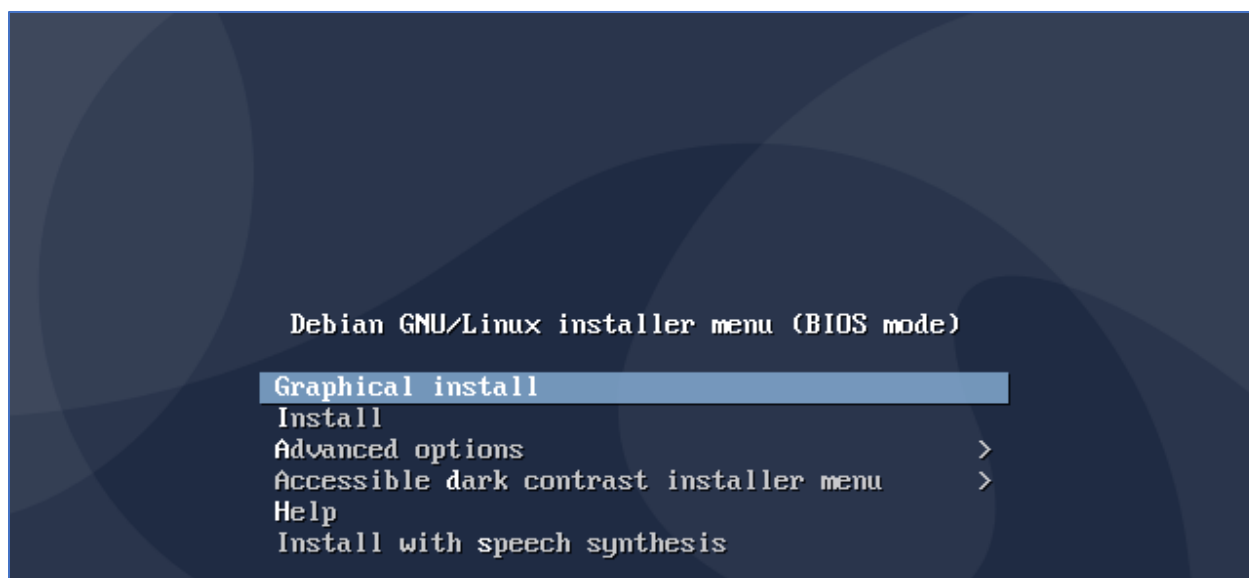
1. Выделите VM и нажмите «Запустить» на панели.



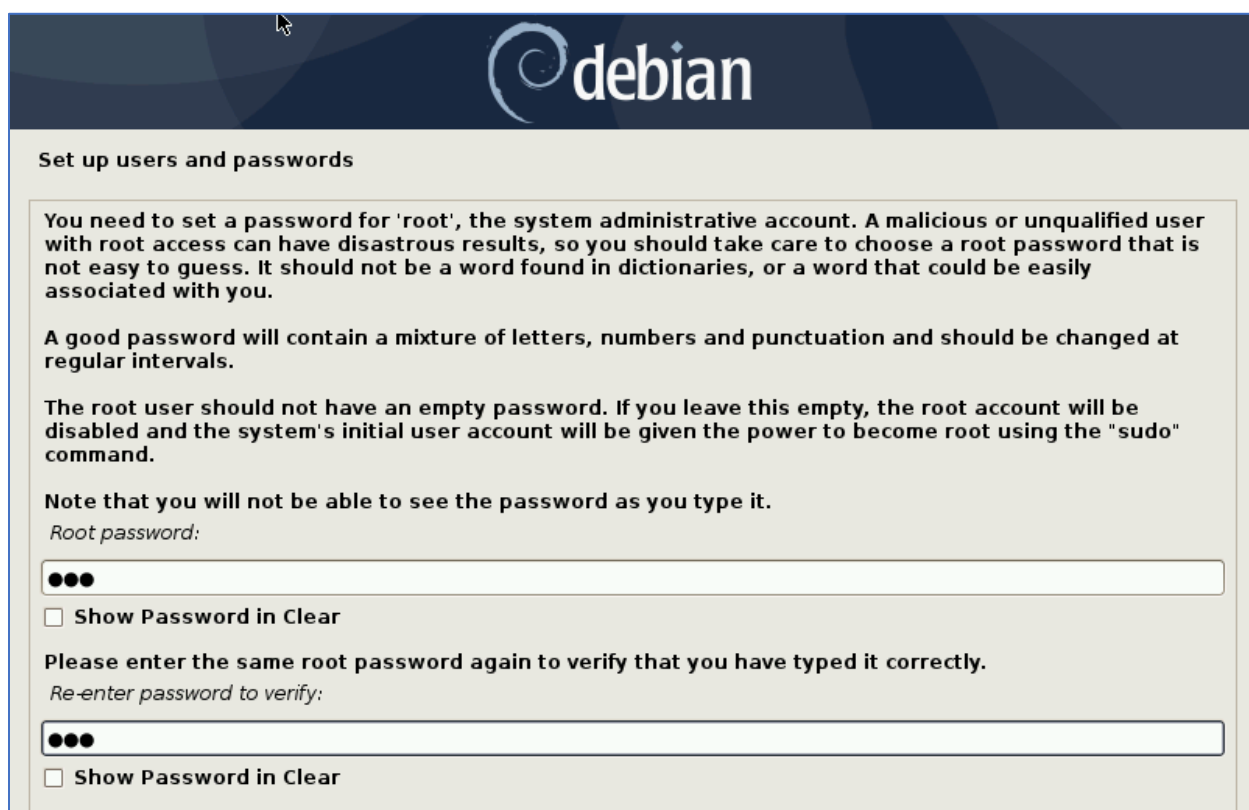
2. В появившемся приглашении выбора загрузочного диска укажите путь файлу установочного образа:



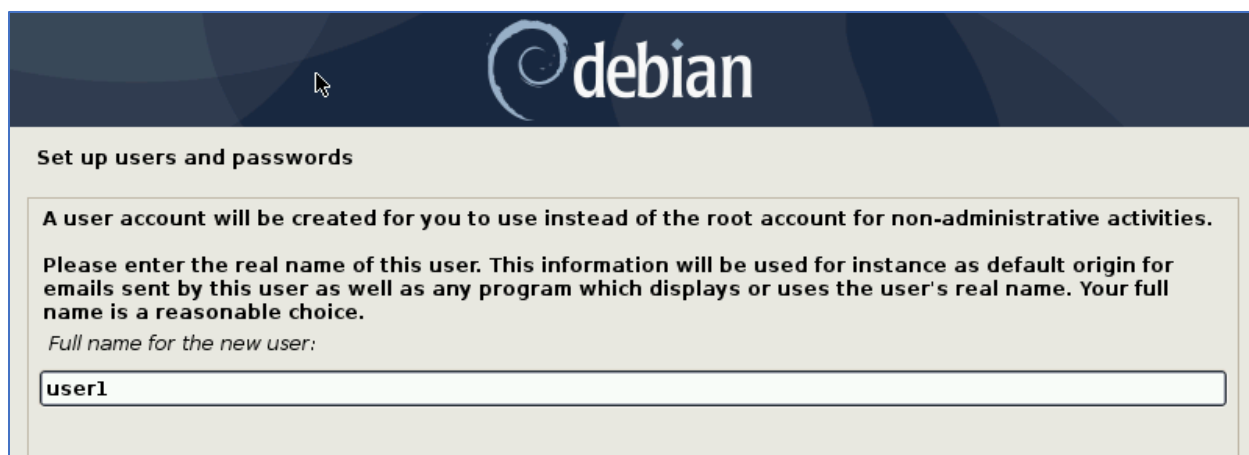
3. После запуска появится меню установщика. Выберите стрелками «Graphical Install» и нажмите «Enter»



4. На последующих экранах оставляйте значения по умолчанию и нажимайте «Continue» до появления окна с настройкой пароля для root:



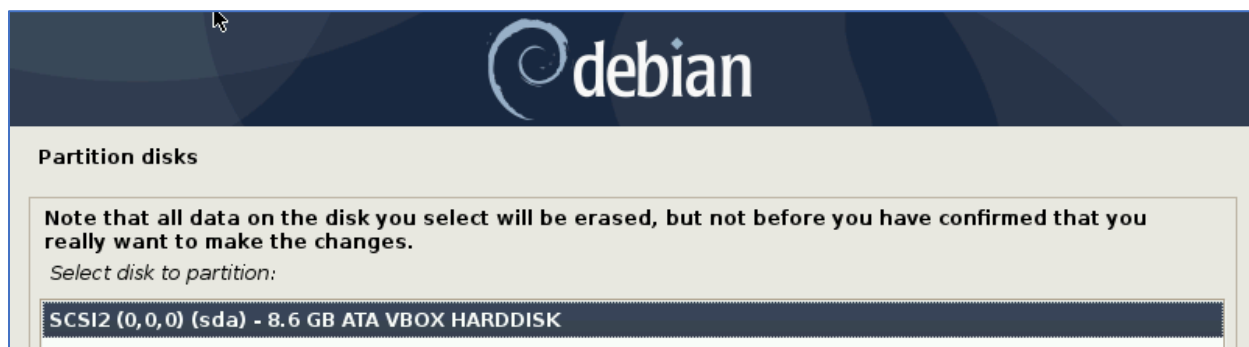
5. Введите и подтвердите во втором поле пароль
6. На следующем экране появится предложение создать пользователя – введите имя user1



7. Далее, когда появится экран задания пароля пользователю – задайте и подтвердите пароль.
8. Дальнейшие шаги с настройками по умолчанию до появления экрана с разбиением диска на разделы. Выберите «Guided – use entire disk»

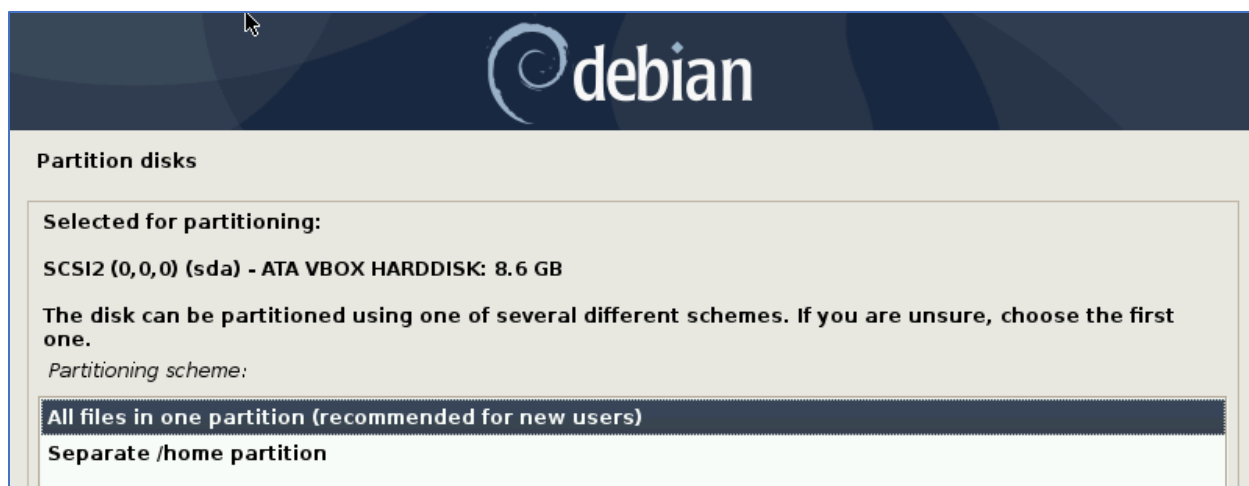


9. Далее – выбираете диск, на котором создавать разделы:



10. Выбираете схему разделов, как показано на скриншоте:






11. Указываете завершить процедуру создания разделов:



12. Отметить «Yes» для записи изменений на диск:



### Partition disks

If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.


The partition tables of the following devices are changed:  
SCSI2 (0,0,0) (sda)

The following partitions are going to be formatted:  
partition #1 of SCSI2 (0,0,0) (sda) as ext4  
partition #5 of SCSI2 (0,0,0) (sda) as swap

Write the changes to disks?

☐ No  
☒ Yes

13. Далее указываете не использовать следующий CD или DVD:



### Configure the package manager

Your installation CD or DVD has been scanned; its label is:

Debian GNU/Linux 10.6.0\_Buster\_ - Official amd64 NETINST 20200926-10:16

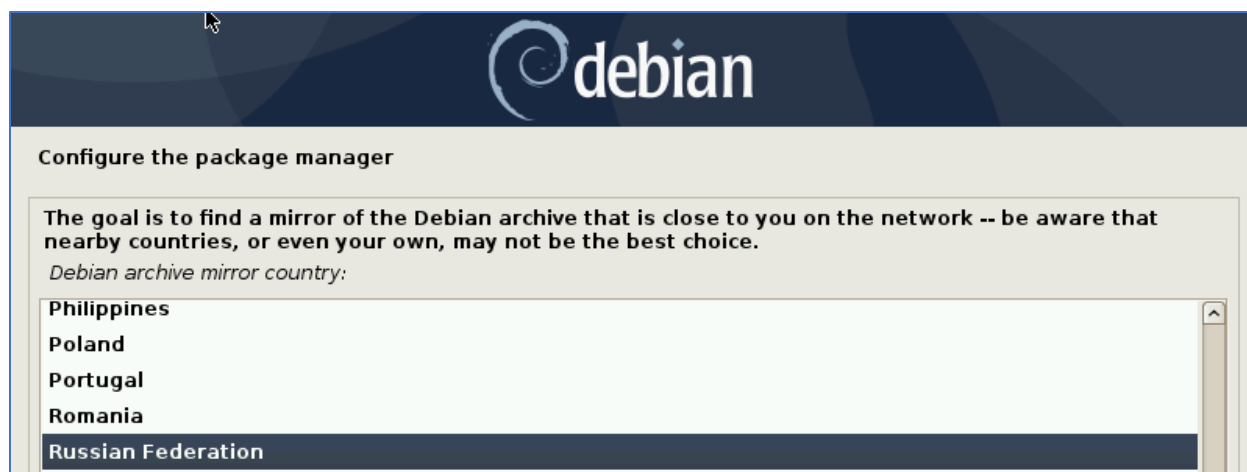
You now have the option to scan additional CDs or DVDs for use by the package manager (apt). Normally these should be from the same set as the installation CD/DVD. If you do not have any additional CDs or DVDs available, this step can just be skipped.

If you wish to scan another CD or DVD, please insert it now.

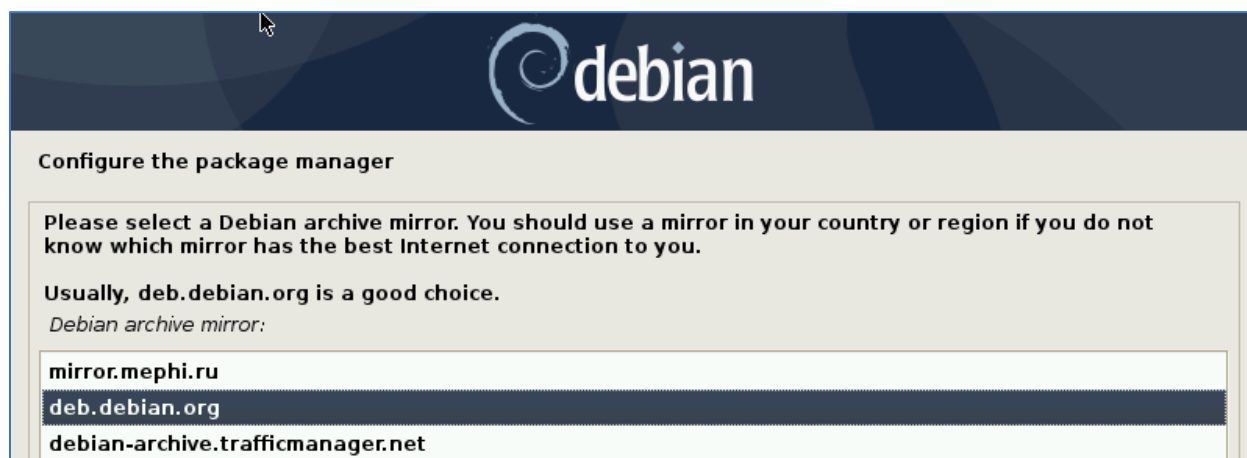
Scan another CD or DVD?

☒ No  
☐ Yes

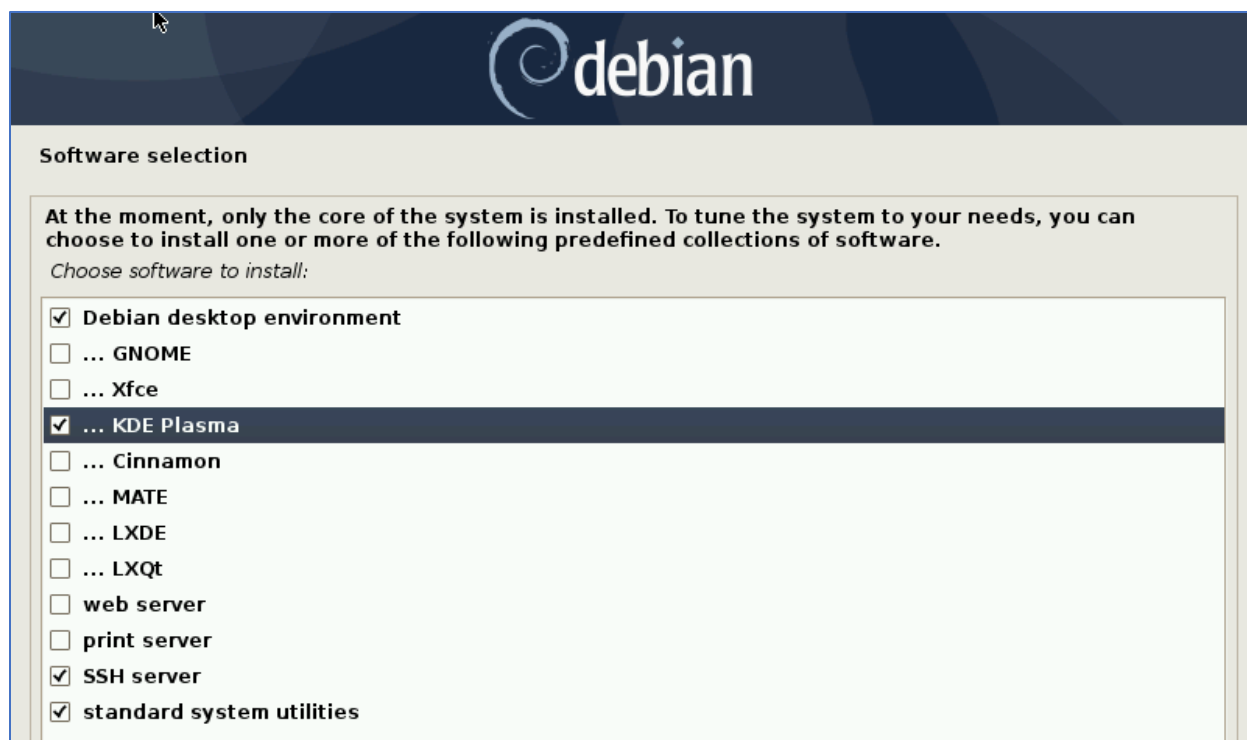
14. Далее, выбираете «Russian Federation» как расположение репозитория для установки ПО:



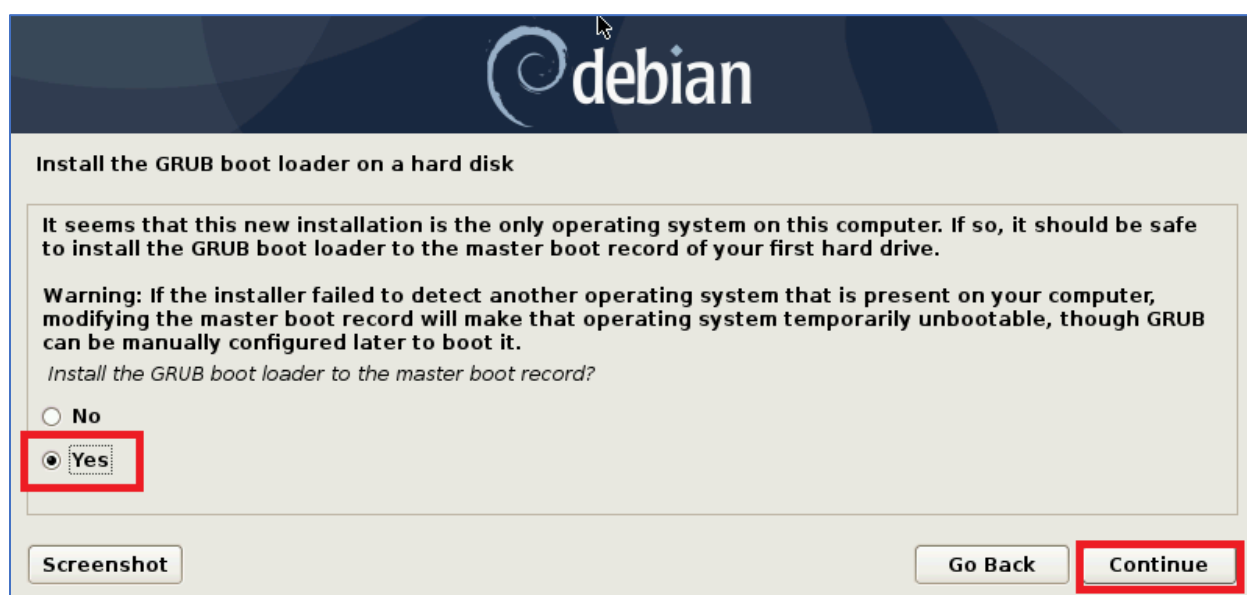
15. И репозиторий `deb.debian.org`



16. Далее, по умолчанию, до появления экрана с выбором устанавливаемого ПО. Отметьте, кроме умолчательных опций также KDE Plasma и SSH Server, а отметку print server уберите, как показано на скриншоте:



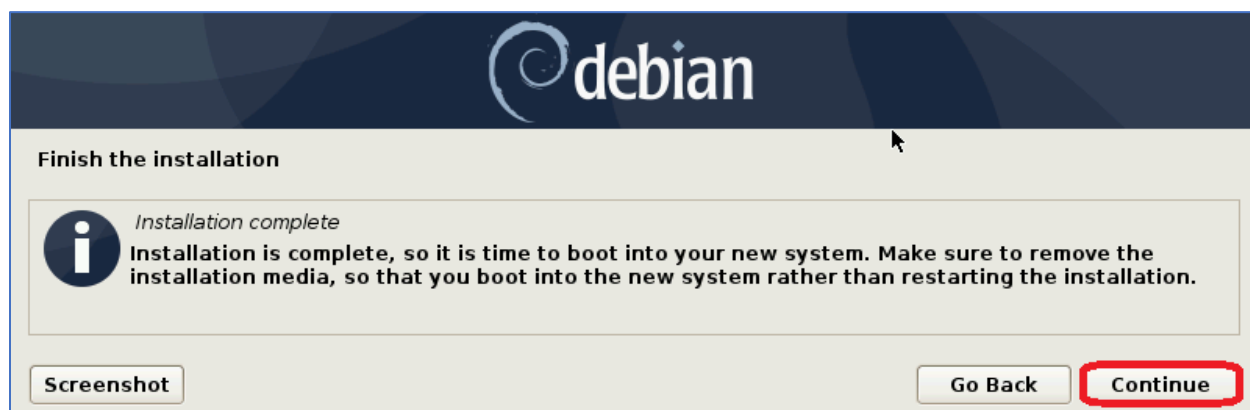
17. Удостоверьтесь, что указано «Yes» для установки загрузчика в MBR на жесткий диск:



18. И укажите на какой диск, выбрав строку «/dev/sda»:



19. Установка завершена, нажмите «Continue» для перезагрузки:



### Упражнение 3. Локальный вход в систему.

**Внимание!** Для освобождения курсора мыши из ВМ нажмите правую клавишу Ctrl.

1. Используя комбинацию клавиш Ctrl + Alt + F1 войдите в текстовую консоль как пользователь root с паролем 111.
2. Для передачи пользователю административных полномочий добавьте его в системную группу sudo:

```
gpasswd -a user1 sudo
```

### Упражнение 4. Начало работы в консоли.

**Цель работы:** Получить практические навыки по основным моментам, связанным с работой в консоли.

1. Перейдите в текстовую консоль (виртуальный терминал)

2. Под каким пользователем вы находитесь в терминале: привилегированным или обычным?
3. Узнайте ip-адрес своей системы (в случае VirtualBox: строка 2: enp0s3 после inet)

#### Решение.

1. Войдите в текстовую консоль (виртуальный терминал) нажатием комбинации клавиш Ctrl + Alt + F3
2. Обратите внимание на формат приглашения оболочки: \$ или #
3. Выполните команду `ip a` и запишите ip-адрес интерфейса

```
Debian GNU/Linux 10 debian tty1
debian login: root
Password:
Linux debian 4.19.0-12-amd64 #1 SMP Debian 4.19.152-1 (2020-10-18) x86_64

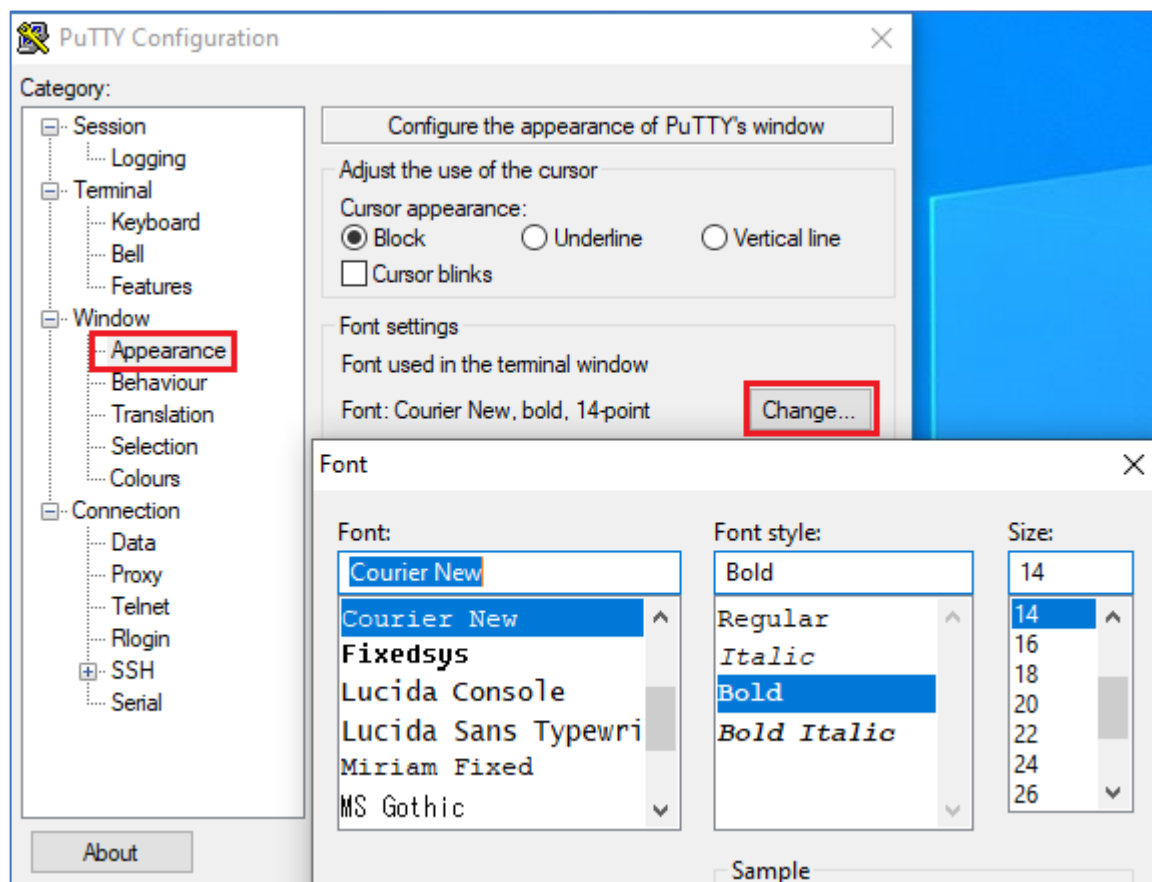
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:f0:8e:51 brd ff:ff:ff:ff:ff:ff
    inet 10.5.4.94/24 brd 10.5.4.255 scope global dynamic noprefixroute enp0s3
        valid_lft 85330sec preferred_lft 85330sec
    inet6 fe80::a00:27ff:fef0:8e51/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
root@debian:~# _
```

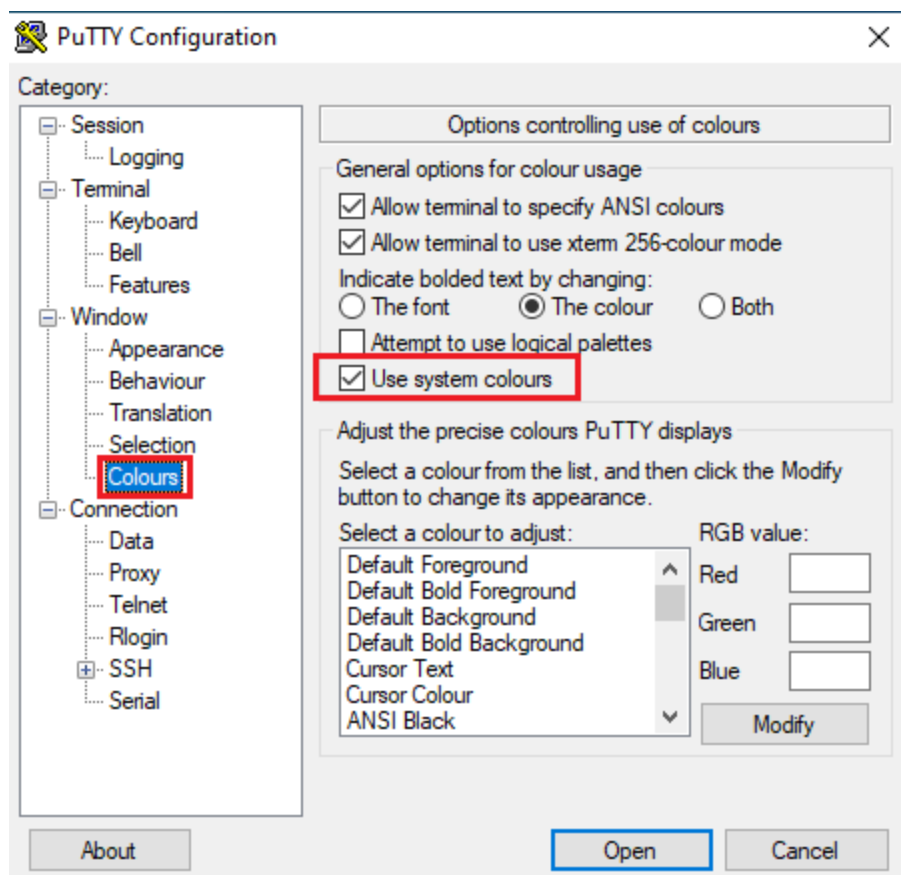
## Лабораторная работа 1-Б. Настройка подключения к системе

### Упражнение 1. Настройка параметров PuTTY.

1. Запустите PuTTY и при желании настройте шрифт и фон терминала
2. Для этого в поле Category выберите Appearance и нажмите кнопку Change как показано на скриншоте ниже.

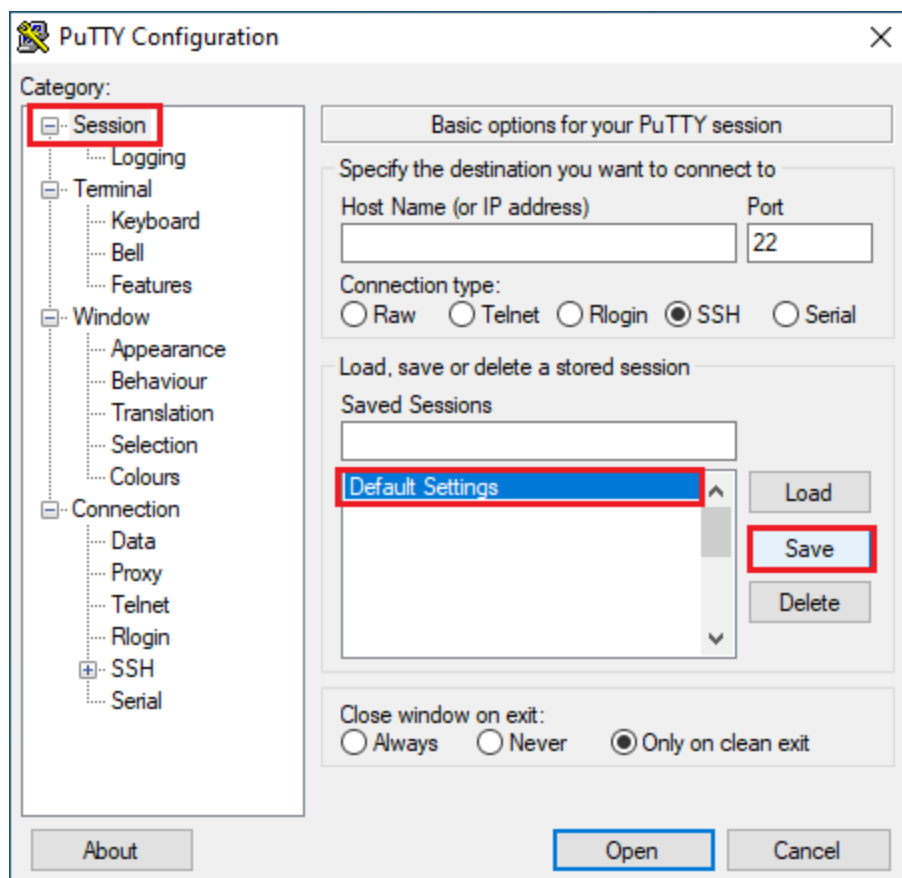


3. Выберите подходящий размер и начертание шрифта. Опционально можно выбрать и сам шрифт, но в данном примере в этом нет необходимости.
4. Далее, в поле Category перейдите на Colours и отметьте флажок Use system colours как показано на следующем скриншоте



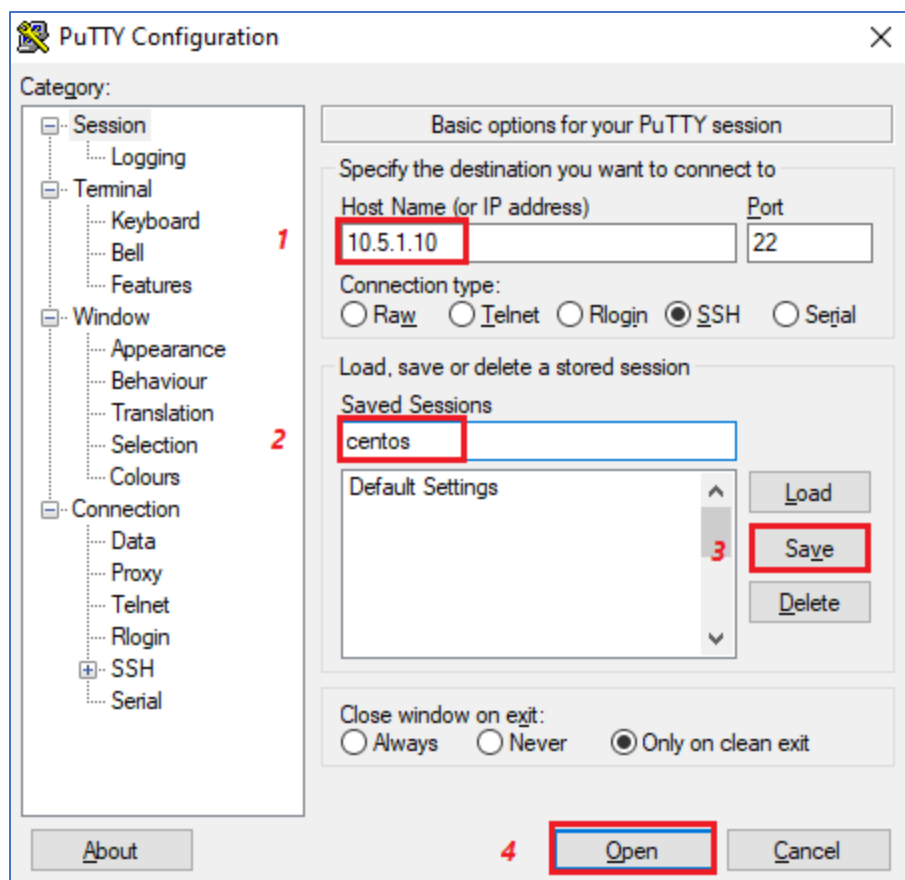
5. Для сохранения сделанных настроек в качестве умолчательных в поле Category перейдите на Session и в поле Saved Sessions выделите курсором мыши Default Settings, затем нажмите кнопку Save (Сохранить) как показано ниже:





## Упражнение 2. Настройка подключения к системе.

1. В PuTTY, в поле Category убедитесь, что выбрано Session
2. В поле Host Name (or IP address) введите ранее отображенный командой «ip a» адрес установленной системы и перейдя в поле Saved Sessions введите произвольное имя, под которым Вы сохраните это подключение (например, server), затем нажмите кнопку «Save» теперь, для подключения нажмите Open (или просто дважды щелкните на имени сохраненного подключения) как показано на скриншоте ниже.



3. Подключитесь к системе: подтвердите в появившемся окне подлинность (значение ключа) подключения и введите свои логин и пароль (user1 и 111)

### Основы интерфейса командной строки

echo \$SHELL – отобразит используемую оболочку

Приглашение командной строки bash:

\$ Обозначает контекст непривилегированного пользователя

# контекст привилегированного пользователя (root)

### Структура команд и формат ключей

Linux - регистрозависимая ОС.

Командная строка в bash составляется из имени команды, за которым могут следовать ключи (опции) - указания, модифицирующие поведение команды. Ключи начинаются с символа - или --

Пример: ls --all и ls -a

Далее могут следовать аргументы (параметры) - имена объектов, над которыми должна быть выполнена команда.

Клавиша “Tab” используется для автозаполнения при вводе команд и файловых путей в Bash.

Реализуется пакетом bash-completion

Комбинации клавиш.

Прим. Клавиша Ctrl на письме обозначается символом “^”.

Ctrl + r – реверсивный поиск по истории

Ctrl + u - удалить введенный текст в командной строке слева от курсора

Ctrl + a – перевод курсора на начало строки

Ctrl + e – перевод курсора в конец строки

Ctrl + c – прерывает текущий процесс (соответствует сигналу 2 – SIGINT)

Ctrl + d – обрабатывается как EOF при вводе с клавиатуры. В сеансе соответствует выходу (logout)

Поддерживается история команд.

В сессии сохраняется в памяти, по завершении пишется в файл **.bash\_history**

Выводится командой **history**.

### **Работа с документацией: man, info**

man (от англ. manual — руководство) — команда, предназначенная для форматирования и вывода справочных страниц. Поставляется почти со всеми UNIX-подобными дистрибутивами.

Синтаксис: man [опции] [раздел] manpage

Система помощи info

Также разработана сообществом GNU.

Пример: info ls

Опция --help

Для получения краткой информации о программе, написанной сообществом GNU, можно использовать опцию --help

Пример: ls --help

### **Упражнение 3. Базовые команды: получение информации о системе и подключенных к ней пользователях**

Введите команду who и посмотрите результат

# who

Передайте вывод команды who другой команде (wc) для последующей обработки

```
# who | wc -l
```

```
# wc (от англ. word count — «подсчет слов»)
```

где l - число строк в файле;

w - число слов в файле;

c - число символов в файле

Получите сведения о вошедшем пользователе, выполнив следующие команды:

```
id
```

```
whoami
```

Получите сведения о дате и времени

```
$ date
```

Получите сведения об имени и адресе системы:

```
$ hostnamectl
```

```
$ ip a
```

```
$ ifconfig -a
```

Используйте команду для вывода всех встроенных инструкций Bash

```
$ help
```

```
$ help | less
```

Выведите справочную информацию по команде man на русском языке

```
$ man -L ru man
```

Одновременный запуск нескольких команд:

```
whoami; hostname; date
```

Работа с несколькими аргументами:

```
ls /var /usr
```

## Глава 2. Структура ФС в Linux

### Стандарт иерархии файловой системы (FHS)

**ls /** отобразит каталоги верхнего уровня

**man hier** описание иерархии файловой системы

В системах Unix и Linux каждый объект файловой системы является частью единого дерева директорий файловой системы с одним общим корнем.

Для доступа к файлу необходимо знать полный путь к этому файлу, начинающийся с корневой директории.

При добавлении нового экземпляра файловой системы, требуется предоставить доступ к ней из какой-либо директории дерева файловой системы. Эта директория называется точкой монтирования.

### Лабораторная работа 2. Файловая система Linux

#### Упражнение 1. Получение информации о файловых системах

Выведите все задействованные файловые системы и посмотрите на какие директории они смонтированы. Для этого используйте команду **df** (disk free) - отчет об использовании дискового пространства файловой системой:

```
df -Th
```

```
df -Tha
```

```
df -Thx tmpfs
```

Подробности об опциях команды **df** можно посмотреть в документации:

```
man df
```

#### Упражнение 2. Получение информации об использовании дискового пространства

1. Выполните перечисленные ниже команды и посмотрите вывод:

**du /home** - подсчитывает и выводит размер, занимаемый директорией

**du -h /home** (- h оптимизирует отображение единиц измерения в выводе)

**du -h -s /home** (суммарный показатель объема занимаемого каталогом)

**du --max-depth=1 /home**

**du -d1 /home**

### Упражнение 3. Просмотр атрибутов файлов

Используйте команду `ls` для вывода информации о типе файла и его атрибутов:

`ls -l`

-           - регулярный файл (жесткая ссылка)

`d`           - директория (каталог)

`l`           - символическая ссылка

Получите информацию об инодах файлов, например, в директории `/etc`

`ls -li /etc`

### Упражнение 4. Утилиты для работы с файлами

`pwd` – отображение текущего каталога

`ls` - отображение содержимого каталога

`touch` - для создания пустого файла можно воспользоваться командой `touch`, изначально предназначенной для установки времени последнего изменения файла или доступа в текущее время

Для чтения файла может использоваться утилита `cat` предназначенная для вывода указанного файла или файлов в поток вывода.

`cat` - объединяет файлы и печатает на стандартный вывод

Примеры:

Читать со стандартного ввода и выводить в терминал

`cat -`

Читать со стандартного ввода и перенаправить вывод в файл:

`cat > myfile1`

`my sample text`

Нажать `Enter`, затем комбинацию `Ctrl + d`

Читать (ввод) из файла и выводить в терминал

`cat myfile1`

Создайте каталог `myfolder`

`mkdir myfolder`

Создайте дерево каталогов:

```
mkdir -p test1/test2/test2
```

Используйте команду перехода между директориями:

```
cd test1/test2/test2
```

```
cd .. перейти в директорию уровнем выше
```

```
cd ../../ перейти в директорию двумя уровнями выше
```

```
cd перейти в домашнюю директорию
```

```
cd ~user1 перейти в домашнюю директорию пользователя user1
```

```
cd - перейти в директорию, в которой находились до перехода в текущую директорию
```

Удалите пустые каталоги:

```
rmdir -p test1/test2/test2
```

**Создайте каталог и добавьте в него файлы.**

**Затем удалите каталог вместе с файлами.**

rm – для удаления файла

rm -r - для удаления каталога рекурсивно, т.е. с содержимым.

```
mkdir sampledir
```

```
echo 'Test string' > sampledir/file1
```

```
rmdir sampledir
```

```
rm -r sampledir
```

Используйте утилиту копирования файлов – cp

Синтаксис: cp <источник> <место назначения>

Скопируйте файл /etc/passwd в текущий каталог

```
cp /etc/passwd .
```

Проверьте результат:

```
ls
```

Используйте в своем домашнем каталоге операции переименование, перемещение, в том числе под другим именем.

Синтаксис: mv <источник> <место назначения>

```
mv passwd testpasswd
```

```
mv testpasswd myfolder
```

Проверка:

```
ls
```

```
ls myfolder
```

или

```
ls ~ myfolder
```

а также:

```
ls . myfolder
```

### **Упражнение 5. Использование утилит чтения файлов**

cat - используется для вывода на экран содержимого файла.

Выполните `cat /etc/passwd`

Выполните команду `more`:

```
more /etc/ssh/sshd_config
```

Используйте Enter для прокрутки вниз

И пока не вышли из файла, клавишу b для прокрутки вверх

Также выполните команду `less`:

```
less /etc/ssh/sshd_config
```

`less` - скроллинг вверх и вниз, PgUp / PgDwn также работает

Для выхода из команды нажмите q

В терминальном подключении (PuTTY) выполните команду:

```
sudo tail -n 20 /var/log/auth.log
```

Затем, выполните команду:

```
sudo tail -fn0 /var/log/auth.log
```

Войдите в систему локально (в текстовую виртуальную консоль)

В терминальном подключении посмотрите появившиеся новые записи в журнале

Для выхода нажмите комбинацию клавиш Ctrl + C

### **Индексный дескриптор (inode):**

```
df -Ti -x devtmpfs -x tmpfs
```



```
ls -li
```

### Упражнение 6. Использование ссылок в Linux

Жесткие ссылки указывают на inode.

Символические ссылки (l) - указывают на имя файла, который имеет соответствующий индексный дескриптор

Каждая ассоциация файла и инода известна как жесткая ссылка. Дополнительные ссылки могут быть созданы с помощью ln

Создайте текстовый файл:

```
echo 'My first file' > ~/myfolder/testfile1
```

Добавьте жесткую ссылку на него:

```
ln ~/myfolder/testfile1 ~/myhardlink
```

Убедитесь, что оба имени ссылаются на один inode:

```
ls -li ~/myfolder/testfile1 ~/myhardlink
```

Теперь создайте символическую ссылку на него:

```
ln -s /home/user1/myfolder/testfile1 ~/mysoftlink
```

Проверьте результат:

```
ls -l
```

### Упражнение 7. Поиск файлов

Найдите файл testfile1 в дереве директорий, начиная с /home:

```
find /home -name testfile1
```

Найдите все жесткие ссылки для файла myhardlink:

```
find /home -samefile myhardlink
```

locate - позволяет найти файл, указанный по шаблону (части имени)

```
sudo apt install locate -y
```

```
locate testfile1
```

locate работает с индексированной бд

```
sudo updatedb
```

```
locate testfile1
```

Поиск местонахождения исполняемых файлов:

which – показывает путь к каталогу указанного исполняемого файла

```
which vi
```

## Лабораторная работа 2-Б. Введение в регулярные выражения Использование фильтрации, утилита `grep`

`man 7 regex (man re_format)`

Символы базовых регулярных выражений:

символ "точка" (.) соответствует любому одиночному символу;

\* – любое количество повторов предшествующего знаку «\*» символа/набора символов, в том числе и нулевое;

^ - соответствует началу строки, но, иногда, в зависимости от контекста, инвертирует символьный класс в регулярных выражениях.

[] - любой символ из числа заключенных в скобки (символы могут быть разделены запятыми, указание диапазона - [0-9]);

Предназначены для задания подмножества символов. Квадратные скобки, внутри регулярного выражения, считаются одним символом, который может принимать значения, перечисленные внутри этих скобок. Метасимвол ^ означает отрицание множества:

[^] - любой символ кроме тех что указаны в скобках;

\ - Служит для экранирования специальных символов, т.е. отменяет спец. значение следующего за ним метасимвола;

Пример: `grep 'bin\sh' /etc/passwd`

-- \<...\> -- Экранированные угловые скобки - задают границы слова (не работает в `sed`).

Пример: `grep -R '\<sed\>' /usr/share`

-- \( \) --Экранированные круглые скобки

Предназначены для выделения групп регулярных выражений. Они полезны при использовании с оператором «|» и при извлечении подстроки.

-- \{ \} -- Экранированные фигурные скобки

Задают число вхождений предыдущего выражения. Для уточнения количества повторений наборов символов применяется модификатор `\{min,max\}`.

Пример: `grep '\(ro.*\)\{2\}' /etc/passwd`

Символы расширенных регулярных выражений

Многие символы, экранируемые в базовых выражениях – ( ) { } | – но не – <> – используются без экранирования.

Знак вопроса -- ? --

Означает, что предыдущий символ или регулярное выражение встречается 0 или 1 раз.

```
$ grep -E '^r?o' /etc/passwd
```

Знак "плюс" -- + --

Указывает на то, что предыдущий символ или выражение встречается 1 или более раз

### Утилита **grep**

```
grep [параметры] регулярное_выражение [файл]
```

-c - задает отображение только числового значения, сколько строк соответствует шаблону;

-i - игнорировать регистр;

-h - подавляет вывод имен файлов, включающих найденные строки;

-l - только отображение имен файлов, содержащих найденные строки;

-n - нумерация выводимых строк;

-s - подавляет вывод сообщений о несуществующих или нетекстовых файлах и ошибках;

-v - отображение строк, не соответствующих шаблону;

-E - включение расширенных регулярных выражений;

### Упражнение 1. Использование регулярных выражений с утилитой **grep**

**Задача:** отобразить из конфигурационного файла /etc/ssh/sshd\_config все закомментированные строки.

```
grep '^#' /etc/ssh/sshd_config
```

отобразите теперь только не закомментированные строки:

```
grep -v '^#' /etc/ssh/sshd_config
```

Уберите из вывода пустые строки для удобочитаемости:

```
grep -v '^#\\|^$' /etc/ssh/sshd_config
```

## Лабораторная работа 2-В. Основы работы в текстовых редакторах (vi, vim, sed, mcedit)

### Упражнение 1. Редакторы vi и vim

vi – есть везде.

vim при необходимости можно установить командой `sudo apt install vim -y`

vimtutor – встроенный учебник vim

**Отредактируйте файл ~/myfolder/testfile1**

`vi ~/myfolder/testfile1`

удалите имеющуюся строку (двукратное нажатие d)

переведите редактор в режим вставки (нажатием i) и введите текст string1

выйдите с сохранением

Нажмите Esc, затем :, затем введите wq и нажмите Enter

Или

Esc + Shift + ZZ

Создайте и отредактируйте файл ~/testfile2

`vi ~/testfile2`

введите текст

string1

string2

string3

выйдите с сохранением

### Упражнение 2. Редактор sed и регулярные выражения

Формат команды

sed команды\_редактирования [имя\_файла]

-i – редактировать файл

-e – добавить к команде скрипт (регулярное выражение)

в файлах ~/testfile2 и ~/myfolder/testfile1 замените строку "string1" на "string2", результат вывести на стандартное устройство вывода:

```
sed -e 's/string1/string2/' ~/testfile2 ~/myfolder/testfile1
```

теперь проверьте файлы, открыв их командой cat:

```
cat ~/testfile2 ~/myfolder/testfile1
```

Примените изменения к файлам:

```
sed -i 's/string1/string2/' ~/testfile2 ~/myfolder/testfile1
```

Снова проверьте файлы, открыв их командой cat:

```
cat ~/testfile2 ~/myfolder/testfile1
```

Создайте для экспериментов копию конфигурационного файла /etc/ssh/sshd\_config под именем example.txt в своем домашнем каталоге:

```
cp /etc/ssh/sshd_config ~/example.txt
```

удалите пустые строки из вывода файла example.txt

```
sed -e '/^$/d' example.txt
```

удалите комментированные строки из вывода файла example.txt

```
sed -e '/ *#/d' example.txt
```

Примените все эти изменения к файлу:

```
sed -i '/ *#/d; /^$/d' example.txt
```

Опционально: Система контроля версий rcs

```
sudo apt install rcs -y
```

```
mkdir -p ~/myproject/RCS
```

```
echo 'My first string' > ~/myproject/projectfile1
```

```
cd ~/myproject
```

```
ci -l projectfile1
```

Введите текст This my initial copy и нажмите Enter

Затем введите символ «.» и нажмите Enter

Внесите изменения в файл:

```
echo 'Second string' >> ~/myproject/projectfile1
```

И сохраните очередную версию

```
ci -l projectfile1
```

Внесите ещё изменения и сравните с архивной версией:

```
rcsdiff projectfile1
```

Откатите изменения к исходной версии файла:

```
co -r1.1 projectfile1
```

Проверьте файл

```
cat projectfile1
```

Посмотрите журнал:

```
rlog ./projectfile1
```

Верните последнюю версию:

```
co projectfile1
```

Проверьте файл

```
cat projectfile1
```

### **Упражнение 3. Консольный файловый менеджер Midnight Commander (MC) и редактор mcedit**

Запустите Midnight Commander:

```
mc
```

выберите файл example.txt

нажмите F4 для редактирования

Выберите редактор mcedit вводом цифры 3

Добавьте текст «test string» в конец файла

нажмите F2 для сохранения

Выйдите, нажав F10

Создайте директорию sampledir

Откройте ее в правой панели (для перехода между панелями используйте Tab)

В левой панели выделите файл example.txt и нажмите F5 для копирования

Выйдите, нажав F10

## Глава 3. Настройка (конфигурирование) ОС.

### Лабораторная работа 3. Послеустановочная настройка системы.

В контексте сервера инструментом настройки сети является ifupdown. В отличие от NetworkManager он использует конфигурационный файл /etc/network/interfaces

#### Упражнение 1. Настройка сети с ifupdown

Узнайте и запишите текущую конфигурацию ip адреса, маршрута по умолчанию и сервер разрешения имен

```
ip a; ip route show; cat /etc/resolv.conf
```

Используйте ее для задания статической адресной настройки в файле /etc/network/interfaces

Добавьте в конец файла /etc/network/interfaces строки, как в примере:

```
auto enp0s3
```

```
iface enp0s3 inet static
```

```
address 192.168.0.26/24
```

```
gateway 192.168.0.1
```

```
dns-nameservers 192.168.0.1
```

Имя интерфейса, адрес/маску, а также шлюз и сервер dns укажите свои, ранее записанные из вывода ip a

#### Упражнение2 (ОПЦИОНАЛЬНО). Перевод системы на systemd-networkd

Останавливаем и отключаем NetworkManager

```
# systemctl stop NetworkManager
```

```
# systemctl disable NetworkManager
```

#### Убираем файл /etc/network/interfaces

```
mv /etc/network/interfaces /etc/network/interfaces.save
```

Включаем systemd-networkd и systemd-resolved:

```
systemctl enable systemd-networkd
```

```
systemctl enable systemd-resolved
```



**Создаем конфиг сети, в котором также используем адрес/маску, шлюз и сервер dns свои, ранее записанные из вывода ip a:**

```
vi /etc/systemd/network/enp0s3.network
```

```
[Match]
```

```
Name=enp0s3
```

```
[Network]
```

```
Address=<Ваш ip адрес>/24
```

```
Gateway=<адрес шлюза>
```

```
DNS=<адрес dns>
```

**Примените настройки:**

```
systemctl restart systemd-networkd
```

```
systemctl start systemd-resolved
```

```
rm /etc/resolv.conf
```

```
ln -s /run/systemd/resolve/resolv.conf /etc/resolv.conf
```

### **Упражнение 3. Локализация окружения пользователя user1**

```
locale -a
```

Расскомментируйте в файле /etc/locale.gen строку ru\_RU.UTF-8 UTF-8:

```
sudo vi /etc/locale.gen
```

Или запустить (сделает то же самое):

```
sudo dpkg-reconfigure locales
```

Выполните:

```
sudo localedef ru_RU.UTF-8 -i ru_RU -f UTF-8
```

localedef считывает указанные файлы charmap и input, компилирует их в форму, используемую функциями locale

Задайте в настройках профиля пользователя локализацию по умолчанию:

```
echo export LANG=ru_RU.utf8 >> .bashrc
```

Проверьте:

```
locale -a
```

**Прим.** Настроить раскладку клавиатуры по умолчанию и комбинацию клавиш для переключения в файле:

```
/etc/default/keyboard
```

#### **Упражнение 4. Настройка системного времени**

##### **Настройка синхронизации системного времени:**

```
cat /etc/systemd/timesyncd.conf
```

```
systemctl status systemd-timesyncd.service
```

##### **Настройка представления системного времени:**

Посмотрите вывод команды:

```
timedatectl
```

Поменяйте настройки времени на Europe/Madrid:

```
timedatectl set-timezone Europe/Madrid
```

Верните Europe/Moscow

```
timedatectl set-timezone Europe/Moscow
```

#### **Упражнение 5. Настройка имени системы**

```
hostnamectl
```

```
sudo hostnamectl set-hostname vm1
```

##### **Настройка файла /etc/hosts**

Приведите первые две строки файла к виду:

```
sudo vi /etc/hosts
```

```
127.0.0.1    localhost vm1
```

```
127.0.1.1    vm1
```

## Упражнение 6. Управление модулями ядра

Просмотр загруженных модулей:

```
lsmod
```

Вывод информации о модуле:

```
sudo modinfo e1000e
```

Загрузка модуля

```
sudo modprobe e1000e
```

Проверка результата

```
lsmod | grep e1000e
```

Выгрузка модуля из памяти:

```
sudo modprobe -r e1000e
```

```
lsmod | grep e1000e
```

### Версия ядра:

```
uname -r
```

Модули хранятся в каталоге `"/lib/modules/<версия ядра>"` в виде файлов с расширением «ko».

```
# find /lib/modules/`uname -r` -name *.ko
```

### Управление параметрами ядра «на лету»

Команда `sysctl` используется для изменения параметров ядра во время выполнения

Проверьте, идет ли `ping` на собственную систему.

Проверьте текущее значение параметра `net.ipv4.icmp_echo_ignore_all`, который используется для включения и выключения ответа вашей системы на `ping`.

```
sysctl net.ipv4.icmp_echo_ignore_all
```

Значение 0 позволяет вашей системе реагировать на `ping`.

Установите значение 1 с помощью утилиты командной строки `sysctl`, а затем проверьте, приходит ли ответ на `ping`.

```
sudo sysctl net.ipv4.icmp_echo_ignore_all=1
```

```
ping localhost
```

Установите значение обратно в 0 и проверьте исходное поведение в восстановленном состоянии.

```
sudo sysctl net.ipv4.icmp_echo_ignore_all=0
```

```
ping localhost
```

**Прим.** Прервать ping комбинацией Ctrl + C

## Глава 4. Управление подсистемой хранения данных

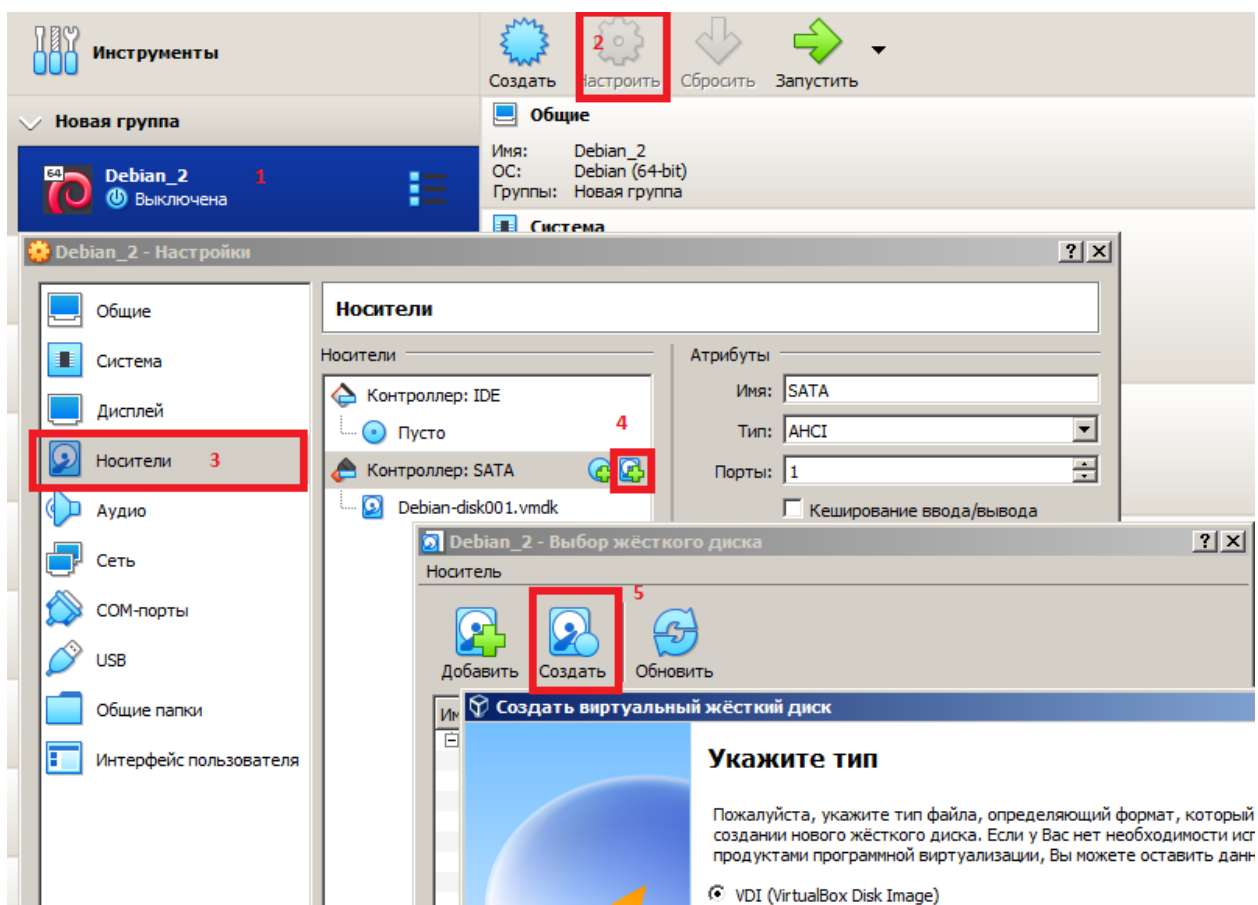
### Лабораторная работа 4. Управление подсистемой хранения данных

#### Упражнение 1. Добавление дисков

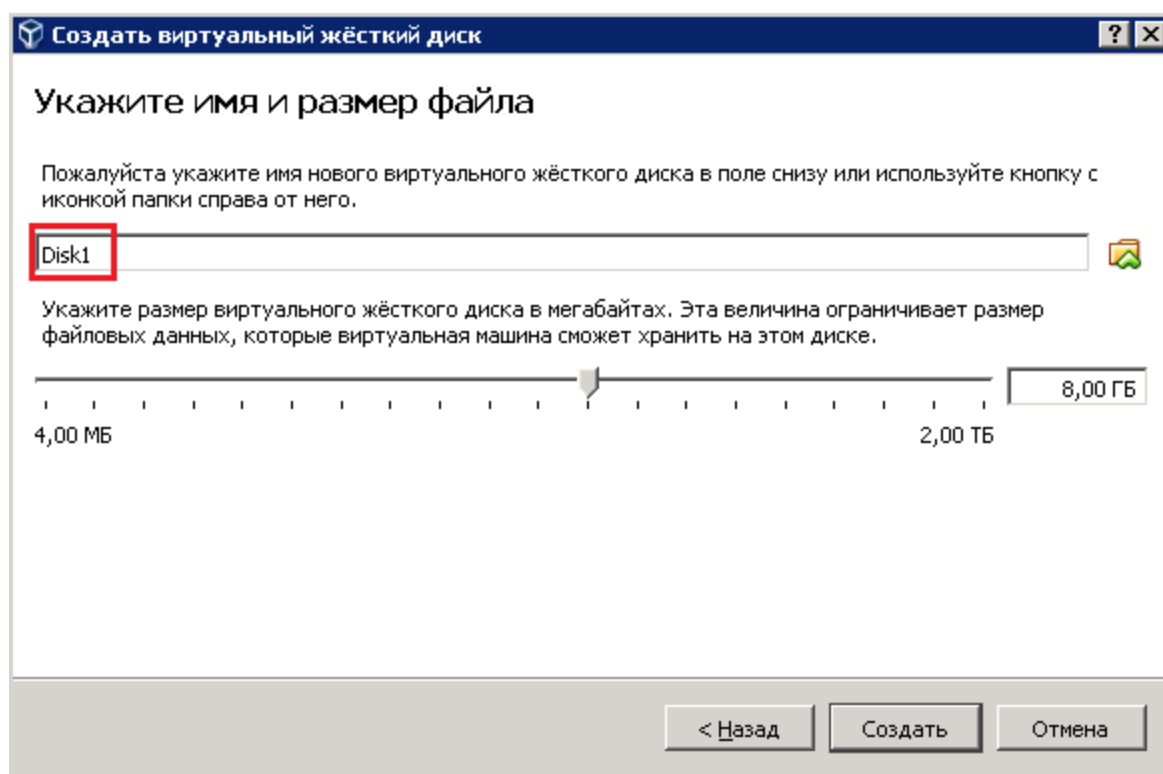
Выключите систему

```
sudo poweroff
```

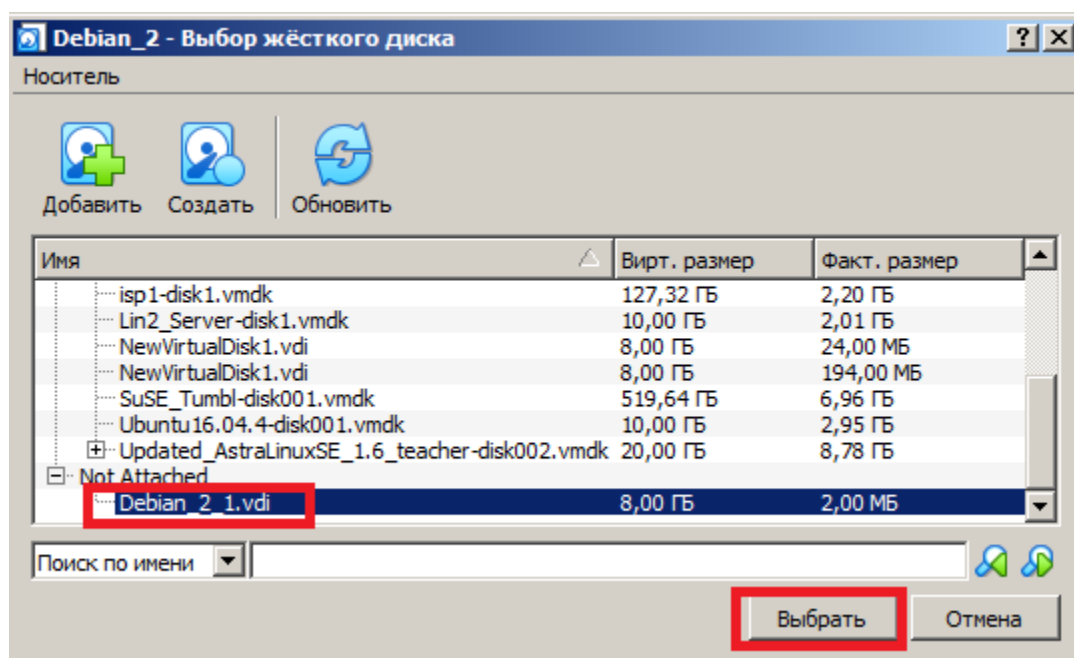
Войдите в настройки виртуальной машины и в меню Носители добавьте жесткий диск к контроллеру SATA:



Примите все параметры по умолчанию:



Выберите в списке появившийся диск и нажмите **Выбрать**:



Повторите процедуру для подключения еще двух дисков.

Запустите виртуальную машину и подключитесь к системе по ssh (PuTTY).

Выполните команды для отображения информации о новых дисках:

```
dmesg | grep sd
```

```
sudo lshw -C disk
```

### **Упражнение 2. Создание разделов**

Создайте на устройстве /dev/sdb раздел размером 100 мегабайт

```
sudo fdisk /dev/sdb
```

Чтобы создать раздел, нажмите n

Оставьте по умолчанию P – primary partition

Оставьте предложенный по умолчанию номер раздела 1

Далее по умолч (Enter)

В финале нажмите w – записать изменения

### **Упражнение 3. Создание файловой системы**

Отформатируйте раздел /dev/sdb1 файловой системой ext4

```
sudo mkfs.ext4 /dev/sdb1
```

Посмотрите сведения о файловой системе на разделе /dev/sdb1:

```
sudo file -s /dev/sdb1
```

### **Упражнение 4. Монтирование раздела**

Создайте директорию для монтирования раздела с ФС:

```
mkdir /disk2
```

Смонтируйте раздел ФС

```
sudo mount /dev/sdb1 /disk2
```

Проверьте доступность ФС для записи. Создайте в ней файл:

```
touch /disk2/file1.txt
```

```
ls -a /disk2
```

Размонтируйте файловую систему:

```
umount /disk2
```

```
ls -a /disk2
```

Воспользуйтесь командой

```
sudo blkid
```

Запишите UUID раздела /dev/sdb1

Добавьте запись в конец файла /etc/fstab

```
UUID=<Идентификатор раздела без кавычек> /disk2 ext4 defaults 0 0
```

Закройте файл с сохранением

Проверьте результат

```
sudo mount /disk2
```

### Упражнение 5. Работа с томами LVM

Создайте физические тома на устройствах /dev/sdc /dev/sdd

```
sudo pvcreate /dev/sdc /dev/sdd
```

Проверьте результат:

```
sudo pvdisplay
```

Создайте группу томов vg1 из ранее созданных физических томов:

```
sudo vgcreate vg1 /dev/sdc /dev/sdd
```

Проверьте результат:

```
sudo vgdisplay
```

Создайте раздел с именем «vol1», размером 1Gb

```
sudo lvcreate -n vol1 -L 1G vg1
```

Посмотрите информацию по созданным логическим томам

```
sudo lvdisplay
```

Отформатируйте том файловой системой ext4:

```
sudo mkfs.ext4 -L volume1 /dev/vg1/vol1
```



Создайте директорию и смонтируйте том:

```
sudo mkdir /volume1
```

```
sudo mount /dev/vg1/vol1 /volume1
```

```
sudo umount /volume1
```

добавляем строку в /etc/fstab, например:

```
/dev/vg1/vol1 /usr/volume1 ext4 defaults 0 0
```

Проверьте результат:

```
mount /volume1
```

```
ls -a /volume1
```

```
umount /volume1
```

```
ls -a /volume1
```

## Глава 5. Процессы в Linux

### Лабораторная работа 5. Процессы в Linux

#### Упражнение 1. Мониторинг процессов

Отобразите состояние процессов в системе:

```
ps -aux
```

```
ps -auxf
```

Выполните мониторинг текущего состояния процессов:

```
top
```

Выведите информацию, каким процессорным ядром был обработан каждый процесс.

Для этого, не выходя из программы нажмите клавишу F, перейдите в меню на P и нажмите пробел. Выйдите из меню кнопкой Esc

Выйдите из top – для этого нажмите клавишу q

#### Упражнение 2. Использование сигналов

Выведите список сигналов. Для этого выполните команду `kill -l`

Подключитесь по ssh(PuTTY) под пользователем user1

Вернитесь в терминал root'a и выполните `ps -aux | grep user1`

Скопируйте pid.

Выполните `kill -15 <pid>`

#### Упражнение 3. Переключение процесса в фоновый режим

1. Создайте процесс переднего плана, запустив команду:  
`sudo tail -fn0 /var/log/syslog`
2. Используйте комбинацию Ctrl + Z чтобы послать сигнал остановки процессу
3. Введите команду `jobs` для вывода состояния процессов
4. Запустите процесс в фоновом режиме:  
`bg`
5. Введите команду `jobs` для проверки
6. Переведите процесс на передний план командой:  
`fg`
7. Завершите процесс используя комбинацию Ctrl + C

#### Упражнение 4. Запуск процесса в фоновом режиме

1. Переключите оболочку в контекст root:  
`sudo -i`
2. Запустите процесс в фоновом режиме (с помощью символа & в конце команды):

```
tail -fn0 /var/log/syslog &
```

3. Введите команду `jobs` для вывода состояния процессов:  
`jobs`
4. Переведите процесс на передний план командой:  
`fg`
5. Завершите процесс используя комбинацию `Ctrl + C`

## Глава 6. Управление конфигурацией

### Лабораторная работа 6-А . Управление конфигурацией

#### Упражнение 1. Использование systemctl

Получите информацию о текущем target. Для этого выполните команду

```
systemctl get-default
```

Выведите список всех загруженных юнитов

```
systemctl list-units --type target
```

Измените target по умолчанию

```
systemctl set-default multi-user.target
```

Проверьте результат:

```
systemctl get-default
```

Верните первоначальную настройку

```
systemctl set-default graphical.target
```

Переключите систему в multi-user.target

```
systemctl isolate multi-user.target
```

Переключите систему назад в graphical.target

```
systemctl isolate graphical.target
```

#### Упражнение 2. Определение переменных

Проверьте текущее значение переменной LANG

```
set | grep LANG
```

Отмените значение переменной в текущей сессии

```
unset LANG
```

```
set | grep LANG
```

Экспортируйте новое значение переменной в сессию:

```
export LANG=ru_RU.UTF-8
```

Проверьте результат

```
date
```

Обратите внимание на язык вывода команды date

Создайте переменную окружения LOG

```
nano /etc/environment
```

```
LOG="/var/log"
```

Переподключите сессию и выполните:

```
echo $LOG
```

```
cd $LOG
```

## Лабораторная работа 6-Б. Введение в использование сценариев

### Упражнение 1. Организация ввода и вывода данных

Создайте файл сценария

```
touch myscript.sh
```

Сделайте его исполняемым

```
chmod +x myscript.sh
```

чтобы сформировать для пользователя приглашение ввести данные, используйте команду `read`.

```
nano myscript.sh
```

Введите в файл следующие строки:

```
#!/bin/bash
```

```
echo -n "Enter Name: "
```

```
read p
```

```
echo Hello "$p !!!"
```

Запустите скрипт на исполнение:

```
./myscript.sh
```

### Упражнение 2. Конструкция If-Then-Else:

Перезапишите файл командой:

```
cat > myscript.sh
```

вставив нижеследующее содержимое:

```
#!/bin/bash
```

```
if
```

```
test -d test
```

```
then
```

```
    echo "Directory exists"
```

```
else
```

```
    mkdir test
```

```
fi
```

Запустите скрипт на исполнение:

```
./myscript.sh
```

### **Упражнение 3. Конструкция CASE:**

Перезапишите файл командой:

```
cat > myscript.sh
```

вставив нижеследующее содержимое:

```
#!/bin/bash
```

```
echo "Please, select editor:"
```

```
echo "1 nano"
```

```
echo "2 vim"
```

```
echo "3 sed"
```

```
echo "4 Exit"
```

```
read selector
```

```
case $selector in
```

```
1)
```

```
/bin/nano
```

```
;;
```

```
2)
```

```
/usr/bin/vim
```

```
;;
```

```
3)
```

```
/bin/ed
```

```
;;
```

```
4)
```

```
exit 0
```

```
;;
```

```
*)
```

```
echo "incorrect action"
```

```
esac
```

Запустите скрипт на исполнение:

```
./myscript.sh
```

#### Упражнение 4. Цикл **While**

Перезапишите файл командой:

```
cat > myscript.sh
```

вставив нижеследующее содержимое:

```
# !/bin/bash
```

```
x=1
```

```
while [ $x -lt 5 ]
```

```
do
```

```
mkdir test$x
```

```
echo "Directory created: test$x"
```

```
x=$(( $x + 1 ))
```

```
done
```

Запустите скрипт на исполнение:

```
./myscript.sh
```



## Глава 7. Управление пользователями

### Лабораторная работа 7. Управление пользователями и группами

#### Упражнение 1. Работа с пользовательскими учетными записями

`sudo -i`

Добавление учетной записи пользователя

```
useradd user2 -m -s /bin/bash
```

Удалите пользователя с каталогом и открытым сеансом:

```
userdel user2 -rf
```

Добавьте пользователей:

```
useradd user2 -m -s /bin/bash
```

```
passwd user2
```

и

```
useradd -m -s /bin/bash backupuser
```

```
passwd backupuser
```

111

Изменить параметры пользователя

```
usermod user2 -c 'Ivanov Ivan Ivanovich,239,45-67,499-239-45-33'
```

Задайте пароль:

```
passwd user2
```

Посмотрите информацию о пользователе:

```
id user2
```

или

```
cat /etc/passwd | grep user
```

Создайте группы

```
groupadd -g 1500 mygroup1
```

```
groupadd -g 1501 mygroup2
```

Проверьте:

```
cat /etc/group | grep mygroup
```

Добавьте user2 в группы

```
usermod -aG mygroup1,mygroup2 user2
```

Удалите его из групп

```
usermod -G '' userX
```

Чтобы увидеть к каким группам вы принадлежите используйте команду groups:

```
$ groups
```

Задать членство в группе mygroup1 для множества пользователей так, чтобы в ней были только пользователи user1 и backupuser, а пользователя user2 не было:

```
root@server:~# gpasswd -M user1, backupuser mygroup1
```

**Прим.** Ключ -M явно задает членство в группе, переопределяя существующее

Для добавления к существующему членству используйте опцию -a

## Упражнение 2. Блокировка учетных записей

1. nologin shell

Выполните замену шелла на /sbin/nologin:

```
# vim /etc/passwd
```

```
user2:x:1002:1002:Petrov Petr:/home/petrov:/sbin/nologin
```

Попробуйте подключиться по ssh как user2

Верните шелл /bin/bash

Попробуйте подключиться по ssh снова

Отключите учетную запись:

```
$ sudo usermod -L user2      ##(-U – unlock)
```

Заставьте пользователя user2 сменить пароль при следующем входе:

```
$ sudo chage -d 0 user2
```

## Упражнение 3. Настройте использование ключей вместо паролей для аутентификации ssh подключения

Зайдите как backupuser

Генерируем ключевые пары

```
backupuser@server:~$ ssh-keygen
```

Копируем ключ на сервер:

```
$ ssh-copy-id server
```

\$ ssh server – подключение без запроса ввода пароля.

## Глава 8. Управление доступом и привилегиями

### Лабораторная работа 8. Управление доступом и привилегиями

#### Упражнение 1. Настройка sudo

Настройте разрешение использовать sudo пользователю backupuser для выполнения резервного копирования утилитой tar:

В директории /etc/sudoers.d/ создайте файл с именем пользователя

Прим. Файлы в /etc/sudoers.d / должны иметь доступ 0440 и принадлежать root:root.

Выполните:

```
cat > /etc/sudoers.d/backupuser
```

```
backupuser    ALL= NOPASSWD: /usr/bin/tar
```

Проверьте под backupuser:

```
sudo tar -cvf etc.tar /etc
```

```
ls -l ~
```

#### Упражнение 2. Разрешения в Linux/UNIX

```
touch myfile1
```

```
ls -l
```

```
chmod 666 myfile1
```

```
ls -l
```

#### Упражнение 3. POSIX ACL

Назначьте пользователю user1 и mygroup1 разрешения чтения и записи на myfile1.

```
setfacl -m u:user1:rw file.txt
```

Назначьте группе mygroup1 права на чтение

```
setfacl -m g:mygroup1:r file.txt
```

Просмотр ACL:

```
getfacl file.txt
```

## Глава 9. Управление ПО

### Лабораторная работа 9-А. Управление ПО

#### Упражнение 1. Установка и удаление программ.

1. Выполните синхронизацию с репозиториями  
`apt update`
2. Запросите сведения о наличии пакета `htop`  
`apt search htop`
3. Запросите подробные сведения о пакете:  
`apt info htop`
4. Установите пакет из репозитория:  
`apt install htop`
5. Удалите установленные пакеты:  
`apt remove htop -y`

#### Упражнение 2. Загрузка пакетов из репозитория

1. Если работали под `root`, то войдите в систему под обычным пользователем (`user1`)
2. Запросите загрузку пакета `htop` из репозитория  
`apt download htop`
3. Проверьте результат – наличие пакета (`htop_<версия>.deb`) в рабочем каталоге:  
`ls`

## Лабораторная работа 9-Б (Опциональная)

### Упражнение 1. Сборка и установка ПО из исходников

1. Установите необходимые для сборки пакеты:

```
sudo apt-get install -y libncurses5-dev autoconf automake libtool autotools-dev dpkg-dev fakeroot dh-make
```

2. Скачайте архив с исходным кодом приложения для приведенного примера:

```
wget ftp://ftp.invisible-island.net/lynx/tarballs/lynx2.9.0dev.5.tar.gz
```

3. Распакуйте архив в своем домашнем каталоге:

```
tar -xvf lynx2.9.0dev.5.tar.gz
```

4. Перейдите в распакованную директорию:

```
cd lynx2.9.0dev.5
```

5. Выполните «дебианизацию» для подготовки к сборке deb пакета:

```
dh_make --createorig -p lynx_2.9.0
```

```
less debian/control
```

```
dh_install
```

6. Произведите сборку пакета:

```
dpkg-buildpackage -rfakeroot --no-sign
```

7. Установите пакет

```
cd
```

```
sudo dpkg -i lynx_2.9.0-1_amd64.deb
```

8. Удалите программу:

```
sudo dpkg -r lynx
```

## Упражнение 2. Добавление локального репозитория

1. Установите утилиту reprepro:

```
sudo -i
```

```
apt install reprepro
```

2. Создайте директорию для будущего репозитория:

```
mkdir -p /var/repo/debian/conf
```

3. Создайте конфигурационный файл distributions:

```
touch /var/repo/debian/conf/distributions
```

```
cat > /var/repo/debian/conf/distributions
```

```
Codename: buster
```

```
Suite: stable
```

```
Version: 10.x
```

```
Origin: Debian
```

```
Label: Debian 10.x
```

```
Description: Debian Stable Updates Repository
```

```
Architectures: amd64 source
```

```
Components: main
```

```
DebIndices: Packages Release . .gz .bz2
```

```
DscIndices: Sources Release . .gz .bz2
```

```
Contents: . .gz .bz2
```

4. Выполните инициализацию репозитория:

```
cd /var/repo/debian
```

```
reprepro createsymlinks
```

5. Добавьте пакет в локальный репозиторий:

```
cp ~/user1/lynx_2.9.0-1_amd64.deb /var/repo/debian/pool
```

```
reprepro -b /var/repo/debian --ask-passphrase includedeb buster /var/repo/debian/pool/lynx_2.9.0-1_amd64.deb
```

6. Добавьте путь к локальному репозиторию в конфигурационный файл :

```
vi /etc/apt/sources.list
```

```
deb [trusted=yes] file:/var/repo/debian buster main
```

7. Установите пакет из репозитория:

```
apt update
```

```
apt install lynx
```



## Глава 10. Резервное копирование

### Лабораторная работа 10-А. Резервное копирование и архивация

**Упражнение 1. Заархивируйте каталог /usr/include без сжатия и с использованием разных алгоритмов сжатия.**

```
tar -C /usr -cf include.tar include
```

```
tar -C /usr -zcf include.tar.gz include
```

```
tar -C /usr -jcf include.tar.bz2 include
```

```
tar -C /usr -Jcf include.tar.xz include
```

**Обратите внимание на эффективность сжатия между тремя методами:**

```
$ ls -lh include.tar*
```

### Упражнение 2. Создание запланированной задачи

1. Переключите оболочку в контекст root  
sudo -i
2. Создайте сценарий архивирования каталога /etc

```
nano /etc/cron.hourly/backup_conf.sh
```

Пример сценария:

```
#!/bin/sh
```

```
echo Backup conf
```

```
CMD="/usr/bin/tar"
```

```
RCMD="ssh backupuser@vm1"
```

```
DIRS="/etc/"
```

```
cd /; $CMD -czf - $DIRS | $RCMD "cat > `hostname`.backup_conf.`date '+%Y%m%d'`.tbz"
```

3. Сделайте файл исполняемым:

```
chmod +x /etc/cron.hourly/backup_conf.sh
```

4. Поставьте задачу в системный crontab  
cat >> /etc/crontab  
\*/1 \* \* \* \* root run-parts /etc/cron.hourly
5. Для проверки войдите как backupuser и выполните ls:  
su - backupuser  
ls

## Лабораторная работа 10-Б. Аудит доступа

### Упражнение 1. Настройка аудита доступа к каталогу пользователя

Добавьте правило аудита доступа к домашнему каталогу user1 в файл /etc/audit/rules.d/audit.rules

```
nano /etc/audit/rules.d/audit.rules
```

```
# Добавим правило для отслеживания доступа (r и w) к директории /home/user1:
```

```
-w /home/user1 -p rwa -k testlog
```

Изменения конфигурации вступят в силу после перезапуска демона auditd:

```
# systemctl restart auditd
```

Имитируем запись файла в отслеживаемую директорию:

```
# touch ~user1/hello_world
```

Анализ журнальных файлов: утилита aureport

Все журнальные файлы сохраняются в /var/log/audit/audit.log.

```
# tail /var/log/audit/audit.log | grep hello
```

Смотрим на ouid=0 (uid автора изменения)

Их можно детализировать с помощью утилиты aureport.

```
ausearch -ui 0 --interpret | grep hello
```