Учебный курс

Административное управление Linux (ITC-L-адм-2)

версия 1.0

Автор:Груздев П.Ю.

г.Екатеринбург
2026

Оглавление

Требования к лабораторному стенду:

1. Компьютер со следующими характеристиками:
    1. Процессор не менее 2 ядер.
    2. ОЗУ от 12 Гб.
    3. Диск от 100Гб. свободного пространства, SSD.
2. ОС — Windows 10 или новее, GNU/Linux с графическим окружением.
3. VirtualBox 7.0 или новее. По желанию может быть использована и другая система виртуализации.
4. Установлена ОС Debian 13 на двух виртуальных машинах с графической средой (предпочтительно MATE).
    1. Имена виртуальных машин VM1 и VM2.
    2. ОЗУ — от 2 Гб
    3. Одно или два ядра.
    4. Диск от 20 Гб.
    5. Создан пользователь sa, которому дана привилегия использовать `sudo`.
5. В VirtualBox создана сеть типа NAT, в которую подключены обе виртуальные машины.
6. В лабораторных работах, в которых не указана виртуальная машина вы можете использовать любую из них.

Глава 1. Написание сценариев Bash.

1.1. Основы работы со сценариями

1. Напишите сценарий `scr1.sh` так, чтобы в нем определялась переменная `V1` и выводилось ее значение. Запустите сценарий не указывая оболочку явно.

Ответ:

```
$ mkdir ~/bin
$ cd ~/bin
$ nano scr1.sh

#!/bin/bash
V1=test
echo $V1

$ chmod a+x scr1.sh # ← Нужно сценарий сделать исполняемым
$ ./scr1.sh # ←Если Текущий каталог не находится в переменной PATH
```

2. Запустите сценарий, указывая оболочку явно.

Ответ:

```
$ bash scr1.sh
test
```

3. Перепишите сценарий `scr1.sh` таким образом, чтобы из него вызывался сценарий `scr2.sh`, который и печатал бы значение переменной `V1`.

Ответ:

```
Содержимое scr1.sh:
#!/bin/bash
V1=test
. scr2.sh

Содержимое scr2.sh
echo $V1

$ ./scr1.sh
test
```

Обратите внимание что `scr2.sh` не обязательно должен быть исполняемым и в нем нет `#!/bin/bash`.

4. Перепишите сценарий `scr1.sh` таким образом, чтобы значение переменной `V1` считывалось бы из файла `scr1rc`.

Ответ:

```
Содержимое scr1.sh:
#!/bin/bash
source scr1rc
. scr2.sh

Содержимое scr1rc:
V1=testrc

$ ./scr1.sh
testrc
```

Обратите внимание что вместо команды «`.`» была использована команда `source`. Обе команды эквивалентны. Команда `source` делает код более читабельным.

5. Напишите простой сценарий оболочки, считывающий значения трех переменных из командной строки и выводящий их значения в стандартный поток вывода. Проверьте его работу, вводя два, три и четыре значения.

Ответ:

```
Содержимое vars.sh:
#!/bin/bash
echo -n "Enter 3 values: "
read V1 V2 V3
echo V1: $V1
echo V2: $V2
echo V3: $V3

$ ./vars.sh
Enter 3 values: 1 2 3
V1: 1
V2: 2
V3: 3

$ ./vars.sh
Enter 3 values: 1 2
V1: 1
V2: 2
V3:

$ ./vars.sh
Enter 3 values: 1 2 3 4
V1: 1
V2: 2
V3: 3 4
```

6. На работу команды `read` оказывает влияние переменная окружения `IFS`. Она устанавливает символ - разделитель для ввода. По умолчанию - пробел. Как проверить работу этой переменной, не повлияв на работу интерактивной оболочки?

Ответ:

```
$ IFS=: read V1 V2
1:2

$ echo $V1 $V2
1 2
```

7. Напишите сценарий, выводящий имя команды, количество аргументов и PID процесса оболочки.

Ответ:

```
Содержимое args.sh:
#!/bin/bash
echo "Command name: $0"
echo "Number of args: $#"
echo "PID: $$"

$ ./args.sh a b c
Command name: ./args.sh
Number of args: 3
PID: 635
```

8. Проверьте работу сценария, используя явный вызов оболочки `bash`.

Ответ:

```
$ bash ./args.sh a b c
Command name: ./args.sh
```

```
Number of args: 3
PID: 636
```

9. Исследуйте работу опции `-v bash`.

Ответ:

```
$ bash -v ./args.sh a b c
#!/bin/bash
echo "Command name: $0"
Command name: ./args.sh
echo "Number of args: $#"
Number of args: 3
echo "PID: $$"
PID: 637
```

10. Измените сценарий так, чтобы он выводил все аргументы командной строки.

Ответ:

```
$ echo 'echo "All args: $@"' >> args.sh
$ ./args.sh a b c
Command name: ./args.sh
Number of args: 3
PID: 638
All args: a b c
```

11. Измените сценарий, использовав команду `shift`. Проверьте, сдвигаются ли значения позиционных параметров.

Ответ:

```
$ cat args.sh
#!/bin/bash
echo "Command name: $0"
echo "Number of args: $#"
echo "PID: $$"
echo "All args 0 shift: $@"
shift
echo "All args 1 shift: $*"
shift
echo "All args 2 shift: $*"

$ ./args.sh a b c
Command name: ./args.sh
Number of args: 3
PID: 646
All args 0 shift: a b c
All args 1 shift: b c
All args 2 shift: c
```

12. С помощью команды `set` установите в сценарии новые значения позиционных параметров.

Ответ:

```
$ cat args.sh
#!/bin/bash
echo "Command name: $0"
echo "Number of args: $#"
echo "PID: $$"
echo 'asfter: set $1 BBB CCC DDD'
set $1 BBB CCC DDD
echo "All args 0 shift: $@"
shift
echo "All args 1 shift: $*"
shift
echo "All args 2 shift: $*"
```

```
$ ./args.sh a b c
Command name: ./args.sh
Number of args: 3
PID: 650
asfter: set $1 BBB CCC DDD
All args 0 shift: a BBB CCC DDD
All args 1 shift: BBB CCC DDD
All args 2 shift: CCC DDD
```

1.2. Условные операторы

1. Как с помощью `test` проверить является файл специальным файлом символьного устройства?

Ответ:

```
$ test -c /dev/tty1
$ echo $?
0
```

2. Предполагается, что имеется переменная `STR1`, которой, вероятно, присвоено значение `str1`. Однако доподлинно это не известно и переменная может быть не инициализирована вообще. Как, используя `test`, проверить содержит ли она значение `str1`?

Ответ: Обратите внимание на кавычки в команде они необходимы иначе может быть ошибка:

```
$ test "$STR1" == "str1"
$ echo $?
1
$ STR1=str1
$ test "$STR1" == "str1"
$ echo $?
0
$ unset STR1
$ test $STR1 == "str1"
-bash: test: ==: ожидается использование унарного оператора
```

3. Проверьте установлен ли в текущей оболочке запрет на использование символа конца файла C^D для выхода из сеанса.

Ответ:

```
$ test -o noclobber
$ echo $?
1
```

4. С помощью `test` проверьте имеется ли жесткая связь между `gzip` и `gunzip`.

Ответ:

```
$ test $(which gzip) -ef $(which gunzip)
$ echo $?
1
$ test $(which perlbug) -ef $(which perlthanks)
$ echo $?
0
```

5. Напишите сценарий, проверяющий имя текущего каталога и выводящий сообщение об ошибке, если оно короче пяти символов.

Ответ:

```
$ cat checkdir.sh
#!/bin/bash
if [ "$(pwd | awk -F/ '{printf length($NF)}')" -le 5 ]
then
```

```
echo "Error length shorter then 5" > /dev/stderr
exit 1
fi
$ ./checkdir.sh
$ echo $?
0
$ cd /etc
$ /home/user/scripts/checkdir.sh
Error length shorter then 5
$ echo $?
1
```

6. Требуется проверить относится ли файл, заданный в качестве аргумента, каталогом или же обычным файлом. Если верно последнее, то сценарий должен выводить имя файла и его размер. В случае, если размер файла превышает 1Кб, то размер должен выводиться в килобайтах. Если размер превышает мегабайт - в мегабайтах.

Ответ:

```
$ cat testfile.sh
#!/bin/bash
if [ $# -ne 1 ]
then
 echo "Usage: $0 {1arg}"
 exit 1
fi
if [ ! -e $1 ]
then
 echo "Can not stat file: $1"
 exit 2
fi
if [ -f $1 ]
then
 ls -lh $1 | sed -r 's/^(.+ ){4}(.+) (.+ ){3}(.+)/\2 \4/'
elif [ -d $1 ]
then
 echo $1 is directory
fi
$ ./testfile.sh /etc
/etc is directory
$ ./testfile.sh src1.sh
Can not stat file: src1.sh
$ ./testfile.sh scr1.sh
36 scr1.sh
$ ./testfile.sh /bin/vim
2,3M /bin/vim
$ ./testfile.sh
Usage: ./testfile.sh {1arg}
$ ./testfile.sh qwert asdfg
Usage: ./testfile.sh {1arg}
```

7. Напишите сценарий, анализирующий список пользователей, находящихся в настоящий момент в системе. В скрипте список пользователей должен быть преобразован в одну строку. В случае, если имеется хотя-бы один сеанс `root` должно выдаваться предупреждающее сообщение. Анализ должен быть осуществлен с помощью команды `case`.

Ответ:

```
$ cat checkroot.sh
#!/bin/bash
case $(who | awk '{printf $1}') in
*root*)
      echo "root session opened!!!"
      ;;
```

```
*)
      echo "All OK"
      ;;
esac
$ ./checkroot.sh
All OK
$ who
user     pts/1        2021-01-14 07:59 (192.168.1.198)
$ ./checkroot.sh
root session opened!!!
$ who
user     pts/1        2021-01-14 07:59 (192.168.1.198)
root     pts/2        2021-01-14 13:32 (192.168.1.198)
```

1.3. Циклы и функции

1. Напишите сценарий, позволяющий в цикле создавать в текущем каталоге символические ссылки на файлы, заданные как его аргументы. Причем для каждого файла должно запрашиваться подтверждение на создание ссылки. В этом сценарии должна использоваться команда `for`.

Ответ:

```
$ cat link.sh
#!/bin/bash
for FILE
do
echo -n "Create link for file ${FILE} in current directory(y/n): "
read ANSW
[ "${ANSW}" == "y" -o "${ANSW}" == "Y" ] && ln -s ${FILE} .
done
$ ./link.sh /etc/ ~/.bashrc
Create link for file /etc/ in current directory(y/n): n
Create link for file /home/user/.bashrc in current directory(y/n): y
$ ls -l etc .bashrc
ls: невозможно получить доступ к etc: Нет такого файла или каталога
lrwxrwxrwx. 1 user user 18 янв 14 15:13 .bashrc -> /home/user/.bashrc
```

2. Измените предыдущий сценарий так, чтобы для организации цикла использовались бы `while` или `until`.

Ответ:

```
$ cat link2.sh
#!/bin/bash
while [ $# -ne 0 ]
do
echo -n "Create link for file $1 in current directory(y/n): "
read ANSW
[ "${ANSW}" == "y" -o "${ANSW}" == "Y" ] && ln -s $1 .
shift
done
$ ./link2.sh /dev /tmp
Create link for file /dev in current directory(y/n): y
Create link for file /tmp in current directory(y/n): y
$ ls -l dev tmp
lrwxrwxrwx. 1 user user 4 янв 14 15:19 dev -> /dev
lrwxrwxrwx. 1 user user 4 янв 14 15:19 tmp -> /tmp
```

3. Напишите сценарий, который помещает идентификаторы пользователей в ассоциативный массив. Затем этот массив используется для того, чтобы напечатать идентификатор пользователя после запроса имени пользователя. Запросы имени должны поступать до тех пор пока пользователь явно не укажет, что желает завершить работу программы. Для получения сведений о пользователях удобно использовать команду

getent.

Ответ:

```
$ cat assocarr.sh
#!/bin/bash
declare -A uids
# Для создания массива используем цикл for
for name in $(getent passwd | awk -F: '{print $1}')
do
 uids[$name]=$(getent passwd $name | awk -F: '{print $3}')
done

# Обратите внимание как создан бесконечный цикл.
# Команда .. это ничего не делать
while [ .. ]
do
 echo -n 'Enter username: '
 read user
# Никогда не надейтесь что пользователи будут что-то правильно вводить
 if [ -z "${uids[${user:-noUser}]}" ]
  then
      echo 'Invalid user name'
  else
      echo "Uid of user $user is ${uids[$user]}"
 fi
 echo -n 'Do you want try another user?(Y/n)'; read answ
 answ=$(echo ${answ:=y} | cut -c1 | tr N n)
 if [ "${answ}" == n ]
 then
  echo "Goodby"
  exit 0
 fi
done
#echo ${uids[*]}
#echo ${!uids[*]}
#2 строки выше можно раскоментировать для отладки

$ ./assocarr.sh
Enter username:
Invalid user name
Do you want try another user?(Y/n)
Enter username: root
Uid of user root is 0
Do you want try another user?(Y/n)Y
Enter username: user
Uid of user user is 1000
Do you want try another user?(Y/n)foo
Enter username: qwerty
Invalid user name
Do you want try another user?(Y/n)n
Goodby
```

4. Введите непосредственно в командной строке код функции, выводящей страницу `man` для темы, указанной в качестве аргумента функции. В теле функции перед вызовом `man` используйте команду `env` для временной установки переменной окружения `LANG` в значение `ru_RU.UTF-8`. Испытайте работу функции в командной строке Bash.

Ответ:

```
$ function manru () {
> env LANG=ru_RU.UTF-8 man $@
> }
```

или если в одну строку

```
function manru () { env LANG=ru_RU.UTF-8 man $@; }

$ LANG=en_US.UTF-8
$ man -P head man
MAN(1)                   Manual pager utils                MAN(1)

NAME
       man - an interface to the system reference manuals

SYNOPSIS
       man [man options] [[section] page ...] ...
       man -k [apropos options] regexp ...
       man -K [man options] [section] term ...
       man -f [whatis options] page ...
$ manru -P head man
MAN(1)           Утилиты просмотра справочных страниц       MAN(1)

НАЗВАНИЕ
       man - доступ к системным справочным страницам

СИНТАКСИС
       man [параметры man] [[раздел] страница ...] ...
       man -k [параметры apropos] регвыр ...
       man -K [параметры man] [раздел] термин ...
       man -f [whatis параметр
```

5. Создайте сценарий, который будет проверять установлены ли в системе некоторые программы, например (curl, git, python3). Проверка установленной программы должна быть реализована в виде функции. Скрипт выдает сообщение что все хорошо или сообщение о том, какие программы отсутствуют. Скрипт должен завершаться с ошибкой если какой-то программы нет.

Ответ:

$ cat check_progs.sh

```
#!/bin/bash

# Функция проверяет, установлена ли программа в системе
check_dependency() {
    local program_name=$1

    # command -v ищет путь к программе. Если находит — возвращает код 0.
    if command -v "$program_name" > /dev/null 2>&1; then
        echo "[ OK ] Утилита '$program_name' установлена."
        return 0  # Успех
    else
        echo "[ ОШИБКА ] Утилита '$program_name' НЕ НАЙДЕНА в системе!"
        return 1  # Сбой
    fi
}

# --- ОСНОВНАЯ ЧАСТЬ СКРИПТА ---

echo "=== Старт проверки системных требований ==="

# Переменная для подсчета ошибок
errors_count=0

# Вызов 1: Проверяем наличие curl
check_dependency "curl"
errors_count=$((errors_count + $?))

# Вызов 2: Проверяем наличие git
check_dependency "git"
```

```
errors_count=$((errors_count + $?))

# Вызов 3: Проверяем наличие python3
check_dependency "python3"
errors_count=$((errors_count + $?))

echo "========================================="

# Анализируем суммарный результат работы функции
if [ "$errors_count" -eq 0 ]; then
    echo "Все зависимости на месте. Можно запускать основной проект!"
    exit 0
else
    echo "Ошибка: Обнаружено пропущенных утилит: $errors_count."
    echo "Пожалуйста, установите недостающие программы перед запуском."
    exit $errors_count
fi
```

6. * Измените предыдущий скрипт так, чтобы «OK» был зеленого цвета, а «ОШИБКА» - красного.

Глава 2. Процесс загрузки и уровни выполнения.

2.1. Управление загрузкой.

1. Перейдите на текстовую виртуальную консоль. И посмотрите как отображается баннер входа.

   Ctrl+Alt+F5

2. Отредактируйте содержимое файла `/etc/issue`, так чтобы на экране выводилось имя терминала и IP адрес.

```
# vi /etc/issue
TTY - \l
IP - `hostname -I`
```

3. Отредактируйте файл `/etc/motd` и проверьте, выводится ли на экран его содержимое при входе в сеанс.

```
# vi /etc/motd
PRIVET
# logout
```

4. Изучите содержимое файла конфигурации GRUB2. Какие пункты меню загрузки там определены? Как описывается раздел, с которого производится загрузка ядра? Как устанавливается раздел для корневого каталога?

   Какие пункты меню загрузки там определены?

   Пунктов меню может и не быть вместо этого используется Boot Loader Specification (BLS), тогда в конфигурации будут строки:

```
insmod blscfg
blscfg
```

   В случае использования классической конфигурации, пункты меню описываются в разделах:

```
$ info grub2 menuentry
$ info grub2 submenu
```

   Как описывается раздел, с которого производится загрузка ядра?

```
set root='hd0,gpt2'
```

   Как устанавливается раздел для корневого каталога?

   Как опция загрузки ядра `root=UUID=...`

5. Выполните команду `grub-mkconfig` (`grub2-mkconfig` для Rocky Linux) и сравните вывод с файлом `/boot/grub/grub.cfg`.

   Вывод `grub2-mkconfig` и файла `/boot/efi/EFI/rocky/grub.cfg` совпадают

6. Как описывается загрузка консольного порта в файле `/etc/default/grub`?

```
GRUB_TERMINAL_OUTPUT="console"
```

7. Через встроенную оболочку GRUB задайте параметр загрузки ядра `init=/bin/bash`. Запустите систему. Что произошло? Получите список всех процессов. Загрузите систему в обычном режиме.

Такой способ позволяет получить доступ к системе минуя нормальную инициализацию. Сразу после загрузки ядра запускается оболочка, которая имеет полный доступ к ОС.

2.2. Управление инициализацией systemd.

1. Определите состояние службы `sshd`.

```
# systemctl status sshd.service
```

2. Определите в какую цель по умолчанию загружается система. Как изменить эту цель?

```
# systemctl get-default
# systemctl set-default
```

3. Установит цель загрузки по умолчанию в `multi-user.target`. Перезагрузите систему.

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

4. Прервите загрузку в меню выбора ОС. И, через встроенную оболочку GRUB, загрузите систему в цель `basic.target`. Можете ли вы взаимодействовать с компьютером?

   В конце строчки `linux ($root)/vmlinuz …..rhgb quiet` дописать `systemd.unit=basic.target`

   Консоль не доступна. Systemd не инициализирует интерфейс взаимодействия с пользователем.

5. Какая команда показывает только текущее состояние сервиса?

```
# systemctl is-active sshd.service
```

6. Проверьте нормально ли работает система в целом. Если состояние системы не соответствует нормальному (`running`), то как определить в чем причина?

```
# systemctl is-system-running
# systemctl --failed
```

7. Остановите и запретите запуск службы `firewalld`.

```
# systemctl disable firewalld.service
# systemctl mask firewalld.service
```

Глава 3. Управление устройствами.

3.1. Модули ядра.

1. Получите список модулей, загруженных в ядро в вашей системе. Имеется ли в списке модуль e1000?

```
# lsmod
```

2. Узнайте, кто является автором модуля e1000.

```
# modinfo e1000
```

3. Определите зависимости модуля e1000 от других модулей.

```
# less /lib/modules/4.18.0-331.el8.x86_64/modules.dep | grep e1000e
```

4. Если модуль e1000 загружен, то попробуйте его выгрузить. Что при этом происходит?

```
# modprobe -r e1000
```

5. Вновь загрузите модуль e1000, какова реакция системы?

```
# modprobe e1000
```

6. Для какого оборудования предназначен модуль igb?

```
# modinfo igb
```

7. Как реализована поддержка USB в вашей системе? В виде модулей или статической части ядра?

```
# grep HCI_HCD /boot/config-$(uname -r)
```

3.2. Работа с устройствами.

1. Получите список PCI устройств.

```
# lspci
```

2. Какой модуль ядра используется для работы с видеоадаптером.

```
# lspci -vs
```

3. Получите список подключенных USB устройств.

```
#lsusb
```

4. Создайте правило `udev`, которое создает символьную ссылку для подключаемого USB Flash диска с именем `flashka`.

Создайте файл `/etc/udev/rules.d/99-fleshka.rules` со следующим содержимым:

```
ACTION=="add", SUBSYSTEM=="block", ENV{ID_USB_SERIAL}=="VBOX_HARDDISK-0[:]0",
SYMLINK+="myflesh%n"
```

Выполните команду перечитывания конфига:

```
# udevadm control --reload
```

Проверьте работу правила:

```
# udevadm test -a add /sys/class/block/имя_диска
```

5. Определите через файловую систему `/sys` какие блочные устройства установлены в вашу систему и куда они подключены.

```
# ls /sys/class/block/
```

6. Установите пакет smartmontools и получите сведения о состоянии дисков.

```
# dnf install smartmontools
# smartctl -a /dev/sdb
```

7. Проверьте с помощью команды `lspci` установлен ли в системе PCI Ethernet адаптер.

```
# lspci | grep Ether
```

8. С помощью команды `loginctl` определите в каких терминалах запущены пользовательские сессии. Сделайте то же самое командой `who`.

```
# loginctl
# who
```

9. Как открыть еще одну сессию в терминале типа tty?

   Ctrl+F2

Глава 4. Настройка сетевых параметров.

4.1. Просмотр текущего состояния.

Выполняется на VM1.

1. Выполните `ip addr show` и запишите:
   - Названия всех интерфейсов
   - MAC-адрес каждого интерфейса
   - IP-адрес и маску подсети
   - Состояние (UP/DOWN)
2. Выполните `ip route show`. Запишите default gateway.
3. Выполните `ss -tuln`. Какие TCP/UDP порты слушаются на машине?
4. Выполните `sudo ss -tulpan`. Посмотрите какие приложения слушают порты и какие установили соединения и с кем?
5. Проверьте файл `/etc/resolv.conf`. Какие DNS-серверы настроены на вашей машине?

4.2. Временная настройка (до перезагрузки).

Выполняется на VM1.

1. Назначьте дополнительный IP-адрес интерфейсу (eth0 или что-то еще, название интерфейса возьмите из предыдущей части).

```
sudo ip addr add 192.168.100.50/24 dev eth0
```

2. Проверьте, что адрес появился.

```
ip addr show eth0
```

3. Удалите временный адрес.

```
sudo ip addr del 192.168.100.50/24 dev eth0
   ```
```

4. Измените MTU на 1400 и проверьте.

```
sudo ip link set dev eth0 mtu 1400
ip link show eth0
```

5. Верните MTU обратно (обычно 1500).

4.3. Настройка через systemd-networkd.

Выполняется на VM1.

1. Проверьте какие службы настройки сети сейчас включены:

```
systemctl is-active NetworkManager systemd-networkd networking
```

2. Если служба NetworkManager работает, то деактивируйте ее и замаскируйте.

```
sudo systemctl disable --now NetworkManager
sudo systemctl mask NetworkManager
```
```

3. Создайте файл `/etc/systemd/network/My.network`.

```
[Match]
Name=eth0

[Network]
DHCP=yes
```

4. Включите и запустите systemd-networkd

```
sudo systemctl enable --now systemd-networkd
```

5. Проверьте какие настройки получены.

```
networkctl status
● Interfaces: 1, 2, 3, 4, 5, 6
       State: routable
Online state: online
     Address: 172.27.7.145 on eth0
              fe80::5054:ff:fecd:fa9f on eth0
     Gateway: 172.27.7.1 on eth0
         DNS: 172.27.7.1
```

6. Настройте полученный от DHCP адрес статически.

```
[Match]
Name=eth0

[Network]
DHCP=no
Address=172.27.7.145/24
Gateway=172.27.7.1
DNS=172.27.7.1
```

7. Перезапустите службу systemd-networkd и проверьте полученные настройки.

```
sudo systemctl restart systemd-networkd
networkctl status
```

4.4. Настройка через NetworkManager (nmcli, nmtui, nm-connection-editor).

Выполняется на VM2.

1. Проверьте какие службы настройки сети сейчас включены:

```
systemctl is-active NetworkManager systemd-networkd networking
```

2. Отключите службу systemd-networkd и включите NetworkManager, если NetworkManager не активен.

```
sudo systemctl disable --now systemd-networkd
sudo systemctl enable --now NetworkManager
```

3. Проверьте состояние устройств и подключений.

```
nmcli device status
nmcli connection show
```

4. Сначала удалите все подключения, кроме lo. Затем создайте подключение с именем LAN и настройкой статического адреса командой nmcli. Адрес тот же что и в задании с systemd-networkd.

```
nmcli connection delete Wired\ connection\ 1
nmcli connection add con-name LAN type ethernet iface eth0 \
ipv4.method manual ipv4.addresses 172.27.7.145/24 \
ipv4.gateway 172.27.7.1 ipv4.dns 172.27.7.1
```

5. Активируйте созданное подключение.

```
nmcli connection up LAN
```

6. Переключите соединение LAN на использование DHCP через nmtui.

   Запустите: `sudo nmtui`

   - Выберите «Редактировать подключение» (Edit a connection)→ «LAN» → «Изменить»
   - Параметры
     - «КОНФИГУРАЦИЯ IPv4» → «<Автоматически>»
     - «Адреса» → «<Удалить>»
     - «Серверы DNS» → «<Удалить>»
   - Выберите «OK» и «Активировать подключение» для применения
   - Выйдите из nmtui и проверьте получившиеся настройки.

```
nmcli
eth0: подключено к LAN
        "Red Hat Virtio 1.0"
        ethernet (virtio_net), 52:54:00:CD:FA:9F, аппаратное обеспечение, mtu 1>
        ip4 по умолчанию
        inet4 172.27.7.144/24
        route4 172.27.7.0/24 metric 100
        route4 default via 172.27.7.1 metric 100
        inet6 fe80::108d:6fcf:4e83:ad8d/64
        route6 fe80::/64 metric 1024

lo: подключено (внешнее) к lo
        "lo"
        loopback (unknown), 00:00:00:00:00:00, программное обеспечение, mtu 655>
        inet4 127.0.0.1/8
        inet6 ::1/128

DNS configuration:
        servers: 172.27.7.1
        domains: vclass7
        interface: enp1s0
```

7. Настройка через nm-connection-editor (если есть GUI):
   - Запустите: `nm-connection-editor`
     - Удалите соединение LAN знак «-».
     - Нажмите «+» для добавления нового подключения.
     - Выберите тип «Ethernet»
       - На вкладке «Ethernet» укажите
         - Имя соединения - «LAN-GUI»
         - Устройство — eth0
       - На вкладке «Параметры IPv4» укажите:
         - Метод — Автоматически (DHCP)
8. Проверьте сведения о соединениях если есть в области уведомлений индикатор подключения к сети.

4.5. Диагностика сети

Выполняется на VM2.

1. Проверьте связь со шлюзом: `ping -c 4 <IP шлюза>`
2. Проверьте связь с машиной VM1: `ping -c 2 <IP VM1>`
3. Запустите трассировку до ya.ru командой `tracepath` (возможно потребуется установить пакет iputils-tracepath)

```
tracepath ya.ru
```

4. Проверьте содержимое ARP-таблицы.

```
ip neighbor show
```

5. Проверьте статистику интерфейса.

```
ip -s link show eth0
```

6. *Установите пакет tcpdump и проследите трассировку до ya.ru. Ответьте на следующие вопросы:
    1. Какой протокол используется в трассировке?
    2. Какие еще протоколы используются во время трассировки?
7. *Просканируйте открытые порты TCP на машине VM2 утилитой `nmap`.

Глава 5. Настройка удаленного доступа.

5.1. SSH

Выполняется на VM1 и VM2.

1. Проверьте запущена ли служба SSH на VM1 и VM2.

```
systemctl status ssh
ss -ltnp | grep sshd
```

2. На VM1 и VM2 измените порт сервера с 22 на 22001 для VM1 и 22002 для VM2.

```
nano /etc/ssh/sshd_config
...
Port 22001
...
systemctl restart ssh
ss -ltnp | grep sshd
```

3. Создайте пару ключей на VM1.

```
ssh-keygen
```

4. Скопируйте публичный ключ с VM1 на VM2 командой scp.

```
scp ~/.ssh/id_ed25519.pub 10.X.200.1:/home/sa/vm1.sshkey.pub -P 22002
```

5. Подключитесь по ssh с VM1 на VM2 и добавьте ранее скопированный ключ в список доверенных для авторизации ключей.

```
ssh -p 22002 IP_vm2
cat vm1.sshkey.pub >> .ssh/authorized_keys
```

6. Выйдете из сеанса и вновь подключитесь к VM2 с использованием публичного ключа.
7. Создайте пару ключей на VM2.
8. Установите на публичный ключ VM2 на VM1 командой `ssh-copy-id`.

```
ssh-copy-id -p 22001 IP_vm1
```

9. Настройте на VM1 пользовательский конфигурационный файл ssh. В файле подключения опишите с использованием имен машин и для VM1 и для VM2.

```
nano .ssh/config
Host  vm1
   HostName IP_vm1
   Port 22001

Host  vm2
   HostName IP_vm2
   Port 22002
```

10. Проверьте удается ли вам подключиться к VM1 и VM2 по именам, выполнив на удаленных машинах команду hostname.

```
for TG in vm{1.2}
do
  ssh $TG hostname
done
```

11. Добавьте собственный ключ в список авторизованных и проверьте что команда из предыдущего пункта не требует ввода пароля пользователя.
12. Повторите пункты 9-11 для VM2.

5.2. RDP.

1. Установите пакеты xrdp и xorgxrdp на VM1.
2. Замените сертификат для защиты подключения по RDP.

```
$ cd /etc/xrdp
$ sudo rm -i cert.pem key.pem
$ sudo openssl req -x509 -newkey rsa:2048 -nodes -keyout key.pem -
out cert.pem -days 3650
```

3. В файле отключите подключение каталогов к удалённой машине в секции Chansrv

```
[Chansrv]
; drive redirection
; See sesman.ini(5) for the format of this parameter
#FuseMountName=/run/user/%u/thinclient_drives
#FuseMountName=/media/thinclient_drives/%U/thinclient_drives
#FuseMountName=thinclient_drives
; this value allows only the user to access their own mapped drives.
; Make this more permissive (e.g. 022) if required.
#FileUmask=077
; Can be used to disable FUSE functionality - see sesman.ini(5)
EnableFuseMount=false
; Uncomment this line only if you are using GNOME 3 versions 3.29.92
; and up, and you wish to cut-paste files between Nautilus and Windows. Do
; not use this setting for GNOME 4, or other file managers
#UseNautilus3FlistFormat=true
```

4. Создайте пользователя с именем remote и настройте ему запуск сессии MATE (рабочая среда MATE должна быть установлена).

```
$ adduser remote
$ echo "mate-session" | sudo tee ~remote/.xsession
$ sudo chown remote:remote ~remote/.xsession
```

5. Перезапустите службы xrdp и xrdp-sesman
6. На VM2 установите пакеты remmina и remmina-plugin-rdp.
7. Создайте в remmina подключение типа RDP к VM1 проверьте, что оно работает.

5.3. VNC для удаленного сеанса.

1. На VM2 установите пакет tigervnc-standalone-server.
2. Создайте пользователя remote и сопоставление его с виртуальным дисплеем в файле `/etc/tigervnc/vncserver.users`.

```
$ sudo adduser remote

$ cat /etc/tigervnc/vncserver.users
# TigerVNC user assignment
#
# This file assigns users to specific VNC display numbers.
# The syntax is <display>=<username>. E.g.:
#
# :2=andrew
# :3=lisa
:11=remote
```

3. Скопируйте файл systemd юнита для tigervncserver@.service и добавить в него опции автоматического перезапуска.

```
$ sudo cp /usr/lib/systemd/system/tigervncserver@.service /etc/systemd/system
```

```
$ sudo nano /etc/systemd/system/tigervncserver@.service
$ sudo grep -A6 Serv /etc/systemd/system/tigervncserver@.service
[Service]
Type=forking
ExecStart=/usr/libexec/tigervncsession-start %i
PIDFile=/run/tigervncsession-%i.pid
SELinuxContext=system_u:system_r:vnc_session_t:s0
Restart=always
RestartSec=10

$ sudo systemctl daemon-reload
```

4. Включите и запустите службу для пользователя remote.

```
$ sudo systemctl enable tigervncserver@:11
$ sudo systemctl start tigervncserver@:11

$ sudo systemctl is-active tigervncserver@:11
active
$ sudo systemctl is-enabled tigervncserver@:11
enabled
```

5. Войдите в сеанс пользователя remote и задайте пароль VNC и настройте скрипт запуска графической оболочки.

```
$ sudo -u remote -i
$ vncpasswd
…
$ cat ~/.config/tigervnc/xstartup
#!/bin/sh
xrdb $HOME/.Xresources
vncconfig -iconic &
dbus-launch --exit-with-session mate-session

$ chmod +x ~/.config/tigervnc/xstartup
```

6. На VM1 установите пакеты remmina и remmina-plugin-vnc.
7. Настройте подключение по протоколу VNC к порту 5911 на VM2.

5.4. noVNC для доступа к текущему сеансу (Удаленный помощник).

1. На VM1 установите пакеты novnc и x11vnc.
2. Создайте сценарий, который будет:
   - Создавать одноразовый пароль и показывать его на экране.
   - Создавать сертификат, если его еще нет для этого пользователя и компьютера.
   - Запускать VNC сервер на случайном порту.
   - Запускать noVNC через веб-сокет.

```
$ sudo vi /usr/local/bin/remassistant
$ sudo chmod +x /usr/local/bin/remassistant

$ cat /usr/local/bin/remassistant
#!/bin/bash
# Создаем случайный пароль
PASS=$(shuf -i 100000-999999 -n 1)
# И печатаем пароль на экран красным цветом
echo -e "Однократный пароль для подключения: \e[31m${PASS}\e[0m"
echo "Этот пароль нужно будет сообщить удаленному помощнику"
```

```
echo -e '\e[34mНе закрывайте это окно!!!\e[0m'

#LOG=~/.remassist.log
LOG=/dev/null
echo "---- $(date +'%Y-%m-%d %H:%M:%S') ----" > ${LOG}

WPORT=6080

VPORT=$(shuf -i 50000-59999 -n 1)

# Создаем сертификат пользователя, если его нет или он не тот:
CERTDIR=~/.config/remassistant
test -d ${CERTDIR} || mkdir -p ${CERTDIR}

CERTFILE="${CERTDIR}/$(whoami).pem"
if [ "$(openssl x509 -in  ${CERTFILE} -noout -text &>>${LOG} | grep -o UPN:$
(whoami)@$(hostname))" !=  "UPN:$(whoami)@$(hostname)" ]
then
 [ -r "${CERTFILE}" ] && rm -f "${CERTFILE}"
 openssl req -x509 -newkey rsa:4096 -keyout ${CERTFILE} -out ${CERTFILE} -sha256
-days 3650 -nodes -subj "/CN=$(hostname)" -addext "subjectAltName=DNS:$
(hostname),otherName:1.3.6.1.4.1.311.20.2.3;UTF8:$(whoami)@$(hostname)" &>>$
{LOG}
fi

# Создаем каталог ~/.vnc если его нет
test -d ~/.vnc || mkdir -p ~/.vnc

# Записываем пароль в файл
x11vnc -storepasswd "$PASS" ~/.vnc/passwd &>>${LOG}

# Запускаем VNC сервер и noVNC
x11vnc -localhost -usepw -rfbport $VPORT &>>${LOG} &
websockify --web /usr/share/novnc/ --ssl-only --cert ${CERTFILE} $WPORT
localhost:$VPORT &>>${LOG}
```

Обратите внимание, что команды if, openssl и websockify это одна строка вплоть до (hostname)"] или ${LOG}

3. Создайте обще системный ярлык для запуска скрипта.

```
$ cat /usr/share/applications/remassistant.desktop
#!/usr/bin/env xdg-open
[Desktop Entry]
Type=Application
Version=1.0
Name=Remote Assiatant
Name[ru]=Удаленный помощник
GenericName=Remote Assistant
GenericName[ru]=Удаленный помощник
Comment=Run program for remote help
Comment[ru]=Запустить программу для удалленой помощи
Icon=help
Exec=/usr/local/bin/remassistant
Terminal=true
Categories=Utility;RemoteAccess;Network;
```

4. Запустите приложение через ярлык «Удаленный помощник» в меню «Стандартные» или «Интернет». Ярлык так же можно разместить на рабочем столе.
5. На VM2 подключитесь через браузер по URL https://IP_vm1:6080/vnc.html
6. Используйте пароль, который был показан на экране для подключения к сеансу.

Глава 6. Печать в GNU/Linux

6.1. Управление принтерами

1. Если еще не установлен, то установите пакет cups.
2. Разрешите удаленный доступ и удаленное администрирование `cupsd`.

```
# cupsctl –remote-admin --remote-any
```

3. Проверьте для какой группы или групп разрешено управление `cups` в вашей системе.

   В браузере http://localhost:631 с данного пункта

4. Проверьте является ли ваша учетная запись администратором `cupsd`. Если нет, до добавьте себя в группу администраторов `cupsd`.

```
# grep SystemGroup /etc/cups/cups-files.conf
```

5. Если у вас виртуальная машина, то разрешите в ней последовательный порт. Установите Generic Text Printer с помощью WEB интерфейса. В качестве порта укажите `/dev/ttyS0` (COM1). Подключитесь к этому порту.
6. Установите ограничение для заданий на печать так, чтобы максимальное количество печатаемых страниц не превышало 100, а максимальный объем заданий на печать не превышал 10Мб
7. Напечатайте файл `/etc/passwd` отправив его на ваш принтер.
8. Каким образом поставить задание на печать так, чтобы оно было напечатано через пять минут.
9. Установите ограничение на максимальный размер задания на печать в 10Мб.
10. Запретите всем пользователям, кроме входящих в группу lp, печать на принтере.
11. Поменяйте драйвер принтера на Generic PDF Printer, через веб интерфейс.
12. Установите минимальное разрешение печати.
13. Распечатайте любой текстовый файл.
14. Запретите постановку заданий в очередь. Продолжается ли печать текста?
15. Запретите печать. Продолжается ли печать текста?
16. Снимите задание с печати. Продолжается ли печать текста?

Глава 7. Системные журналы.

7.1. Управление журналированием rsyslog

1. Создайте в конфигурации демона `rsyslogd` строку, которая заставит его записывать сообщения от источника auth с уровнем важности не ниже info в файл `/var/log/mylog` . Не забудьте перезапустить службу, чтобы изменения вступили в силу.

```
auth.info                       /var/log/mylog
```

2. Проверьте, записываются ли в этот файл сообщения при входе в сеанс и выходе из него пользователей.

```
$ logout
```

3. Создайте аналогичный журнал `/var/log/mylogpriv` для записи сообщений от источника `authpriv` . Проверьте его работоспособность. Записываются ли в него те же сообщения, что и в предыдущий журнал?

```
authpriv.info                   /var/log/mylog
```

4. Посмотрите с помощью `journalctl` журнал загрузки системы.

```
# journalctl -b
```

5. Настройте `systemd` вести постоянные журналы.

```
# vim /etc/systemd/journald.conf
Storage = persistent
```

6. Создайте настройки ротации для журналов `/var/log/mylog` и `/var/log/mylogpriv`.
   Файлы должны ротироваться ежедневно.
   Ротация первой копии должна осуществляться два раза.
   Ротируемые копии должны сжиматься утилитой gzip.

```
# vim /etc/logrotate.conf
/var/log/mylog {
      daily
      rotate 2
      create
      compress
}
```

Аналогично для mylogpriv

7. Для файла `/var/log/mylog` установите в качестве дополнительного условия ротации достижение им размера 10 Кб.

Добавить строчку внутрь скобок
```
size  10K
```

7.2. Journald

1. Сделайте журналы постоянными.

   Для этого надо проверить есть ли каталог /var/log/journal, и если его не, то создать его.

2. Установите максимальный срок хранения журналов 6 месяцев.

```
$ grep Reten /etc/systemd/journald.conf
MaxRetentionSec=6month

$ sudo systemctl restart systemd-journald
```

3. Посмотрите события в журнале за последний час.

```
$ sudo journalctl --since "1 hour ago"
```

4. Проверьте какие события происходили во всех юнитах связанных со службой печати. События начинать просматривать с последнего и в подробном формате.

```
$ sudo journalctl -xeu cups*
```

5. Сделайте то же самое, но вывод должен быть в удобном для человека JSON формате.

```
$ sudo journalctl -xeu cups* -o json-pretty
```

6. Откройте журнал событий для службы печати в режиме вновь поступающих сообщений и перезапустите ее на другом терминале. Проверьте как прошел процесс перезапуска. Что еще перезапускалось вместе с cups.service?

```
$ sudo journalctl -fu cups*
```

7. Посмотрите 10 последних событий пользовательского сенса.

```
$ journalctl --user -n10
```

Глава 8. Мониторинг системы

8.1. Утилиты *top

1. Запустите терминал и выполните команду `top`.
2. Изучите верхние строки вывода. Найдите следующие параметры:
   - Время работы системы (uptime)
   - Среднюю загрузку (load average)
   - Количество задач (Tasks): total, running, sleeping
   - Загрузку процессора (%Cpu): us, sy, id, wa
   - Использование памяти (MiB Mem, MiB Swap)
3. Выполните top в пакетном режиме для однократного вывода получите только сводную статистку по ОС.

```
$ top -b -n1 | head -5
```

4. Сравните содержимое файла /proc/loadavg и первой строкой в выводе команды top.

```
$ cat /proc/loadavg ; top -b -n1 | head -1
```

5. Установите пакеты htop и btop. Запустите их и посмотрите какие сведения о загруженности системы они выдают и в каком виде.
6. Установите пакет atop и включите сбор сведений о производительности в журналы atop.

```
$ sudo apt install atop
$ sudo systemctl enable --now atop
```

7. Проверьте что записалось в журнал утилитой atop.

```
$ sudo atop -B -r /var/log/atop/atop_$(date +%Y%m%d)
```

Команда t переключается между измерениями вперед, а T назад. Интервал сбора сведений раз в 10 минут по умолчанию.

8. Установите пакет stress-ng и загрузите все ядра минимум на 10 минут. После проверьте снова сведения о загруженности системы из журнала atop.

```
$ sudo stress-ng -c8 -t 600
```

Параметр -c8 означает 8 потоков по одному на ядро.

8.2. Мониторинг процессора

1. Установите пакет sysstat (если он ещё не установлен).
2. Выполните команду для просмотра средней загрузки процессора:

```
$ iostat -c
```

3. Запишите значения в отчёт и сделайте вывод о загруженности ЦП.
4. Выполните команду

```
$ sar -q LOAD 1 10
```

5. Найдите параметр runq-sz. Каково его значение? Справляется ли процессор с нагрузкой?

Если он больше удвоенного числа ядер, это признак того, что процессор — узкое место.

8.3. Мониторинг оперативной памяти

1. Выполните команду:

```
$ free -h
```

2. Определите: общий объём ОЗУ, объём использованной и свободной памяти, объём swap.
3. Выполните команду:

```
$ vmstat
```

4. Найдите в выводимых данных столбцы: swpd (использовано swap), free (свободная память), buff (буферы), cache (кэш).
5. Запишите результаты в отчёт. В идеале swap должен быть равен 0 — сделайте вывод.

8.4. Мониторинг дискового пространства

1. Выполните команду для просмотра смонтированных файловых систем:

```
$ df -h
```

2. Исключите из вывода файловые системы типа devtmpfs и tmpfs.

```
$ df -h -x devtmpfs -x tmpfs
```

3. Определите сколько свободного места на диске, на котором находится ваш домашний каталог.

```
$ df -h ~
```

4. Выполните команду для просмотра информации о блочных устройствах:

```
$ lsblk -o NAME,SIZE,FSUSE%,MOUNTPOINTS
```

5. Изучите информацию о дисковых разделах.
6. Установите пакет iotop и изучите дисковую нагрузку от процессов.

```
$ sudo apt install iotop
$ sudo iotop
```

7. Получите общую загрузку дисков командой iostat.

```
$ iostat -d
```

8.5. Мониторинг сети

1. Определите имя сетевого интерфейса

```
$ ip link show
```

2. Выполните команду для просмотра статистики по сетевому интерфейсу (замените enp0s3 на имя вашего интерфейса):

```
$ ip -s link show dev enp0s3
```

3. Найдите в выводе:
   - RX bytes/packets — полученные данные
   - TX bytes/packets — отправленные данные
   - errors, dropped — ошибки и потери
4. *Получите данные, которые вы искали в выводе статистики в JSON формате.
5. *Установите пакеты iftop и nethogs. Изучите данные, которые предоставляют одноименные утилиты.

8.6. Анализ событий

1. Изучите формат записи событий в журнале `/var/log/syslog`.
2. Утилитой grep найдите все события в журнале `/var/log/syslog` за промежуток времени с 12:00 до 14:00. (Если событий таких еще нет, используйте другой промежуток времени).

```
$ sudo grep "^$(date +'%Y-%m-%d')T1[23]:[0-5][0-9]:" /var/log/syslog
```

3. Установите пакет logwatch и создайте сводный отчет о сегодняшних событиях.

```
$ sudo logwatch --range Today
```

4. *Используя приложение 1 из учебника настройте реакцию системы на неправильное использование команды `sudo` в виде блокировки учетной записи и завершения сеанса пользователя.

Глава 9. Отложенное и регулярное выполнение заданий

9.1. Отложенные задания

1. Создайте файл, содержащий команду, позволяющую послать суперпользователю электронное письмо, содержащее сведения о средней загрузке системы.

```
mail -t root < uptime
```

2. Выполните команду `at` так, чтобы предыдущий файл был исполнен через 3 дня.

```
# at now +3 days
```

3. Получите список заданий `at` .

```
# atq
```

9.2. Периодические задания

1. Каждое утро в 10:30 требуется получать в виде электронного письма список из десяти пользователей, входивших в сеанс в последнее время. Создайте соответствующую таблицу `cron` для обычного пользователя.

```
$ echo '30 10 * * * sa last -10' > crontab.eg
$ crontab crontab.eg
```

2. Будучи суперпользователем выведите таблицу, которая создана в предыдущем пункте.

```
# crontab -l -u sa
```

3. От имени суперпользователя отредактируйте эту таблицу так, чтобы создавалось еще одно электронное письмо с информацией о пользователях, находящихся в сеансе в 9:30 утра каждого рабочего дня.

```
# crontab -e crontab.eg
```

4. Проверьте установлен ли у вас пакет anacron, если не установлен, то установите.

```
# apt install anacron
```

5. Изучите как `anacron` запускает ежедневные, еженедельные и ежемесячные задания в вашей системе.

```
$ cat /etc/anacrontab
```

6. Каким образом запускается сам `anacron`?

Anacron выполняет задания с заданной периодичностью, но при этом не предполагает что машина постоянно включена. Запуск anacron осуществляется посредством systemd. Для проверки выполнить команду:

```
systemctl list-timers | grep anacron
```

или cron

```
# cat /etc/cron.hourly/0anacron
```

Глава 10. Резервное копирование.

10.1. dd.

1. Скопируйте первые 512 байт из файла устройства жесткого диска, с которого загружается операционная система, в файл first_sect.img и определите тип его содержимого.

```
# dd if=/dev/sda of=./first_sect.img count=1 bs=512
```

2. Получите шестнадцатеричный дамп первых четырех байт файла first_sect.img.

```
# hexdump -n 4 first_sect.img
```

10.2. Сжатие файлов.

1. Создайте пустой файл и сожмите его `gzip` . Уменьшился ли размер этого файла? Почему?

```
# touch empty
# gzip empty
```

2. Получите шестнадцатеричный дамп этого файла.

```
# hexdump touch.gz
```

3. Получите в домашнем каталоге сжатые `gzip`, `xz` и `bzip2` копии файла `/bin/mount` . Исходный файл не должен быть изменен. Сравните результаты сжатия.

```
# gzip -k empty
# bzip2 -k empty
# xz -k empty
# ls -l
```

4. Проверьте целостность сжатых файлов.

   Опция `-t` у утилит сжатия

5. Как с помощью `bzip2` сделать сжатую копию некоторого файла с расширением, характерным для `gzip` ?

```
# bzip2 -S .gz
```

10.3. Резервное копирование.

1. От имени суперпользователя создайте в его домашнем каталоге полный архив файлов в каталоге `/etc` . Архив должен называться `etc.tar.gz`.

```
# tar czf etc.tar.gz /etc
```

2. Получите содержимое архива для просмотра списка файлов в нем.

```
# tar tzfv etc.tar.gz
```

3. Извлеките из созданного архива единственный файл `/etc/hosts` так, чтобы он был сохранен в каталоге `/tmp` с тем же именем.

```
# tar xzfvg etc.tar.gz
```

4. Разделите созданный архив на части, размером по 1Мб, используя для этого утилиту `split` . Каким образом из полученных частей собрать целый архив?

```
# split -b 1M etc.tar.gz
# cat файлы_кусков_архива > etc.tar.gz
```

5. Создайте `cpio` каталога `/etc`.

```
# find -true /etc |  cpio -ov > etc.cpio
```

6. Выведите содержимое этого архива на экран.

```
# cpio -t < etc.cpio
```

7. Извлеките из архива в домашний каталог файлы, в имени которых есть строка sysctl.

```
# cpio -iF etc.cpio -E sysctl
```

8. Найдите опцию `cpio` , которая при создании архива позволяет не изменять время доступа к файлам, помещаемым в архив.

```
--preserve-modification-time
```

Глава 11. Программные системы хранения

Перед выполнением лабораторной работы добавьте к виртуальной машине не менее 3 дисков, размером не менее 8Гб.

11.1. LVM

1. Добавьте по одному LVM разделу размером 3Гб, на 2 диска.

```
# fdisk
```

2. Создайте логический том с зеркальным отображением. Размер тома 1 Гб. Файловая система ext4.

```
# pvcreate /dev/sdb1
# pvcreate /dev/sdc1
 /dev/sdb1 /dev/sdc1
# lvcreate -m 2 -L 1G /dev/mvg mlv
# mkfs -t ext4 /dev/mvg/mlv
```

3. Увеличьте размер зеркального тома и файловой системы на нем на 1 Гб.

```
# lvextend -L +1G /dev/mvg/mlv -r
```

4. Создайте логический том размером 2 Гб с чередованием. Логический том должен иметь файловую систему swap.

```
# lvcreate -i 2 -L 2G /dev/mvg slv
```

5. Получите информацию о состоянии группы и использовании дисков в ней.

```
# vgdisplay -v
```

6. Опишите подключение зеркального тома к каталогу `/mirror1` в `/etc/fstab`.

```
/dev/mvg/mlv      /mirror1    ext4  defaults    0     0
```

7. Подключите еще один диск размером 8Гб к виртуальной машине. Добавьте этот диск к группе целиком.

```
# pvcreate /dev/sde
# vgextend mvg /dev/sde
```

8. Перенесите на новый диск данные с одного из разделов и удалите освобожденный раздел из группы.

```
# pvmove /dev/sdb1 /dev/sde
# vgreduce mvg /dev/sdb1
# pvremove /dev/sdb1
```

11.2. Программный RAID

1. Создайте RAID 5 из трех разделов, размером по 3 Гб каждый. Файловая система ext4.

```
# mdadm -C -l 5 -n 3 myraid /dev/sdb2 /dev/sdc2 /dev/sdd2
# mkfs -t ext4 /dev/myraid
```

2. Получите информацию о состоянии созданного RAID массива.

```
# cat /proc/mdstat
```

И/или

```
# mdadm --detail /dev/md/myraid
```

3. Каков размер файловой системы, созданной в RAID массиве?

   Немного меньше 3 Гб

4. Опишите подключение RAID 5 к каталогу `/raid5` в `/etc/fstab`.

```
/dev//md/myraid   /raid5      ext4  defaults    0     0
```

5. *Замените один из дисков в RAID массиве.