**Справочный материал. Основные команды в консоли Linux  
ls** — выдать список файлов в текущем каталоге

**bash** — запуск новой оболочки bash

**exit** — выход из текущей оболочки

**type** — определить тип команды

**echo** — отобразить строку на экране  
**cat /etc/os-release –** версия Линукс **uname –** тип ОС **lscpu** — просмотр информации о процессоре  
**free** — использование памяти.  
l**shw - просмотр общей информации об оборудовании**  
**ln -s** <на\_что\_сделать\_ссылку> <имя\_ссылки> — создать символическую ссылку.  
**rm** <файлы> — удалить файлы.  
**mkdir** <каталог> — создать новый каталог.  
**rmdir** <каталог> — удалить пустой каталог.  
**pwd** — вывести имя текущего каталога.  
**whoami** — вывести имя под которым Вы зарегистрированы.  
**date** — вывести дату и время.  
**time** <имя программы> — выполнить программу и получить информацию о  
времени, нужном для ее выполнения.  
**who** — определить кто из пользователей работает на машине.  
**uptime** — количество времени, прошедшего с последней перезагрузки  
операционной системы.

**ping <имя\_узла>** — проверка доступности узла

**lsblk** — список блочных устройств

**Df -h**  — сведения ою используемом пространстве

**mount** — монтирование устройств

**ss** — просмотр сетевых соединений

**top, htop** — просмотр процессов

# Лабораторные работы

Лабораторная работа 1: systemd

Задача: Создать unit типа mount для /mnt/disk1.

Добавьте строку **/dev/sdb1 /mnt/disk1 ext4 defaults 0 2** в файл **/etc/fstab**

systemctl list-units --type mount #юниты типа mount

sudo systemctl cat mnt-disk1.mount #просмотр юнита. Скопировать все содержимое

cd /etc/systemd/system

sudo nano mnt-disk1.mount #создаем файл юнита

[Unit]

Documentation=man:fstab(5) man:systemd-fstab-generator(8)

SourcePath=/etc/fstab

Before=local-fs.target

Requires=systemd-fsck@dev-sdb1.service

After=systemd-fsck@dev-sdb1.service

After=blockdev@dev-sdb1.target

[Mount]

Where=/mnt/disk1

What=/dev/sdb1

Type=ext4

sudo nano /etc/fstab #закоментируйте последнюю строку

sudo umount /mnt/disk1

sudo systemctl daemon-reload #перечитывание файлов systemd

df -h #/mnt/disk1 нет в списке

sudo systemctl start mnt-disk1.mount #запуск демона

df -h #появился /mnt/disk1

sudo systemctl stop mnt-disk1.mount #найдите /mnt/disk1

Задача: Создать unit типа automount для /mnt/disk1

sudo systemctl cat mnt-disk1.mount #просмотр юнита

cd /etc/systemd/system

sudo nano mnt-disk1.automount

[Unit]

Documentation=man:fstab(5) man:systemd-fstab-generator(8)

SourcePath=/etc/fstab

Before=local-fs.target

Requires=systemd-fsck@dev-sdb1.service

After=systemd-fsck@dev-sdb1.service

After=blockdev@dev-sdb1.target

[Automount]

Where=/mnt/disk1

TimeoutIdleSec=10

[Install]

WantedBy=multi-user.target

sudo systemctl daemon-reload #перечитать конфиги

sudo systemctl stop mnt-disk1.mount #стопаем mnt-disk1.mount

sudo systemctl enable --now mnt-disk1.automount #включаем и запускаем mnt-disk1.automount

df -h #нет /mnt/disk1

df -h /mnt/disk1 #нет /mnt/disk1

sudo systemctl disable --now mnt-disk1.automount #выключаем и стопаем mnt-disk1.automount

sudo reboot #перезагружаем и проверяем статус mnt-disk1.automount и mnt-disk1.mount

Задача: Написать простой скрипт, который будет копировать любую директорию в каталог /mnt/disk1 и сделать скрипт исполняемым файлом.

mkdir dir-mount

touch dir-mount/file1

nano script.sh #создать script.sh

#!/bin/bash

cp -R /home/имя\_пользователя/dir-mount /mnt/disk1

chmod +x script.sh #делаем script.sh исполняемым

Задача: Создать unit типа service на основе созданного выше скрипта.

cd /etc/systemd/system

sudo nano mnt-disk1.service #создаем файл сервиса

[Unit]

Description=copy /home/user/dir

Requires=mnt-disk1.mount

After=mnt-disk1.mount

[Service]

Type=oneshot

ExecStart=/home/имя\_пользователя/script.sh

StandardOutput=syslog

StandardError=syslog

SyslogIdentifier=mycopy

[Install]

WantedBy=multi-user.target

sudo systemctl daemon-reload #перечитать конфиги

sudo systemctl start mnt-disk1.service #запускаем mnt-disk1.service

sudo systemctl statust mnt-disk1.service

df -h /mnt/disk1

Лабораторная работа 2: Управление ПО

Задача: Установить postgresql.

sudo apt update #обновление списка пакетов

sudo apt search --names-only postgresql-15 #поиск пакета

sudo apt-key adv --fetch-keys 'https://apt.postgresql.org/pub/repos/apt/ACCC4CF8.asc' #добавляем ключ

sudo add-apt-repository 'deb https://apt.postgresql.org/pub/repos/apt/ jammy-pgdg main'

#добавляем репозиторий

sudo apt update #обновление списка пакетов

sudo apt search --names-only postgresql-15 #поиск пакета

sudo apt show postgresql #просмотр информации

sudo apt show postgresql-contrib #просмотр информации

sudo apt install -y postgresql postgresql-contrib #установка нужных пакетов

sudo systemctl status postgresql.service #проверка

sudo -i -u postgres #вход вв аккаунт postgres

psql #запуск клиента psql

exit #выход из консоли psql

exit #выход из postgres

Задача: открыть порт для postgresql

sudo ufw status #просмотр состояния ufw

sudo ufw enable #включить ufw

sudo ufw allow 5432/tcp #открыть TCP порт 5432

ss -t state listening #просмотр ожидающих соединений. 5432 порт слушается только на 127.0.0.1

sudo nano /etc/postgresql/16/main/postgresql.conf

Раскомментируйте строку

#listen\_addresses = 'localhost'

И приведите к виду

listen\_addresses = '\*'

sudo systemctl restart postgresql.service #перезапуск сервиса postgresql

ss -t state listening #0.0.0.0:postgresql (5432)

Задача: Просмотрите все зависимости для nginx. Установите nginx

sudo apt depends nginx #просмотр зависимостей

sudo apt install -y nginx #установка пакета nginx

Задача: Удалите nginx

sudo systemctl status nginx.service #статус сервиса nginx

sudo systemctl stop nginx.service #остановка

sudo apt remove -y nginx #удаление

Лабораторная работа 3: Управление пользователями

Задача: Создайте нового пользователя newuser и дайте ему права root

sudo adduser newuser #создание пользователя

sudo usermod -aG sudo newuser #добавление в группу

sudo cat /etc/passwd | grep newuser #проверьте что запись о пользователе появилась в passwd

su newuser #переключение под newuser

passwd #измените пароль

sudo systemctl status postgresql.service #проверка состояния сервиса с вводом нового пароля newuser

Задача: Предоставьте новому пользователю newuser права на запуск команд systemctl и apt без ввода пароля sudo

sudo gpasswd -d newuser sudo #удаляем newuser из группы sudo

type systemctl #определяем тип и путь к файлу

type apt #определяем тип и путь к файлу

sudo nano /etc/sudoers #редактируем sudoers

Ищем в файле раздел # Cmnd alias specification и добавляем строку алиаса для набора команд

**Cmnd\_Alias SRVCAPT = /usr/bin/apt, /usr/bin/systemctl**

Где:

Cmnd\_Alias #тип алисаса

SRVCAPT #имя алисаса

/usr/bin/apt, /usr/bin/systemctl #набор команд

Ниже, в разделе правил, добавляем строку, описывающую поведение sudo для пользователя newuser

**newuser ALL=(ALL) NOPASSWD: SRVCAPT** #разрешаем пользователю newuser, подключенному с любого хоста запускать без ввода пароля команды из набора SRVCAPT через sudo

Сохраняем и закрываем файл. Пробуем подключиться под newuser и запустить несколько команд через sudo

su newuser #вход под newuser

sudo systemctl status sshd.service #проверка sshd

sudo apt show nginx #просмотр сведений о пакете nginx

sudo nano /etc/apt/sources.list #список репозиториев. Обратите внимание на запрос пароля и последующую ошибку

sudo apt edit-sources #список репозиториев

Лабораторная работа 4: Права доступа

Задача: Создайте новый файл и просмотрите кто является владельцем файла, какая группа у файла?

su имя\_пользователя #вход под другим пользователем

cd ~(имя\_пользователя) #переход в домашний каталог

sudo mkdir dirAR #создание каталога dirAR

sudo touch fileAR #создание файла fileAR

ls -la /home #просмотр содержимого каталога

drwxr-xr-x 5 root root 4096 Apr 16 13:54 .

drwxr-xr-x 20 root root 4096 Apr 9 13:38 ..

drwxr-x--- 2 admin admin 4096 Mar 24 18:26 admin

drwxr-x--- 9 new-ubuntu-admin new-ubuntu-admin 4096 Apr 16 13:53 new-ubuntu-admin

drwxr-x--- 3 rusanov rusanov 4096 Apr 16 14:00 rusanov

Обратите внимание на владельцев директорий в выводе. Также обратите внимание на права доступа директории /home (в выводе **.** - это права доступа просматриваемой директории /home, **..** - это права доступа вышестоящей директории /) Благодаря каким правам вы смогли просмотреть содержимое /home?

ls -la #просмотр содержимого каталога

Обратите внимание на права доступа файлов в выводе.

drwxr-x--- 4 rusanov rusanov 4096 Apr 16 14:10 .

drwxr-xr-x 5 root root 4096 Apr 16 13:54 ..

-rw------- 1 rusanov rusanov 51 Apr 16 14:00 .Xauthority

-rw------- 1 rusanov rusanov 35 Apr 16 14:00 .bash\_history

-rw-r--r-- 1 rusanov rusanov 220 Apr 16 13:54 .bash\_logout

-rw-r--r-- 1 rusanov rusanov 3771 Apr 16 13:54 .bashrc

drwx------ 2 rusanov rusanov 4096 Apr 16 14:00 .cache

-rw-r--r-- 1 rusanov rusanov 807 Apr 16 13:54 .profile

drwxrwxr-x 2 root root 4096 Apr 16 14:10 dirAR

-rw-rw-r-- 1 root root 0 Apr 16 14:10 fileAR

Задача: Измените владельца файла fileAR на вашего пользователя. Откройте файл fileAR для редактирования под своим пользователем, введите команду для вывода на экран надписи “Hello, World”. Запустите файл как исполняемый.

sudo chown имя\_пользователя fileAR #изменение владельца

ls -la #просмотр содержимого каталог

nano fileAR #редактирование файла

echo "Hello, World" #Добавьте и сохраните изменения

./fileAR #выполнение файла

или

/Path/to/fileAR #выполнение файла

chmod ?+x fileAR #кому надо добавить x?

./fileAR #выполнение файла

Задача: Скопируйте каталог dirAR в /opt и добавьте права доступа на чтение и редактирование своему пользователю через группу

sudo cp -r dirAR /opt #копирование в /opt

ls -la /opt #просмотр содержимого каталога

sudo addgroup groupar #создание группы groupar

sudo usermod -aG groupar имя\_пользователя #добавление в группу

sudo usermod -aG groupar newuser #добавление в группу

sudo chown :groupar /opt/dirAR #смена группы

sudo chmod g+rw /opt/dirAR/ #права группе на чтение и запись

su newuser #вход под newuser

echo 'Write to /opt/dirar' > /opt/dirAR/filear #создание файла /opt/dirAR/filear и добавление содержимого

ls -la /opt/dirAR/ #просмотр содержимого каталога

exit

nano /opt/dirAR/filear #на редактирование нет прав

sudo chmod 666 /opt/dirAR/filear #права на чтение и запись всем

ls -la /opt/dirAR/ #просмотр содержимого каталога

nano /opt/dirAR/filear #редактирование файла

Задача: Задайте права владельцу на запись, чтение и выполнений; группе - на чтение и запись; остальным - только чтение через числовое представление?

sudo chmod ??? /opt/dirAR/filear

Задача: Создайте нового пользователя testar и предоставьте ему права на чтение и запись без включения в группу grouparи смены прав остальным пользователям

sudo apt install -y acl #установка пакета acl

sudo setfacl -m u:testar:rw /opt/dirAR/filear #права rw

ls -la /opt/dirAR/filear #просмотр прав доступа

su testar #вход под testar

nano /opt/dirAR/filear #откройте и добавьте содержимое. Сохраните файл с тем же именем

exit #выход из testar

sudo setfacl -b /opt/dirAR/filear #удаление всех ACL

ls -la /opt/dirAR/filear #просмотр прав доступа

Лабораторная работа 5: Выполнение задач по расписанию

Данная лабораторная работа будет выполняться на основе выполнения резервного копирования тестовой базы данных postgresql 16. Для выполнения задания необходимо задать пароль для пользователя postgres и изменить настройки подключения к postgresql, чтобы при локальном обращении не вводить парольот postgres.

Задача: Изменить настройки подключения для postgresql и создать тестовыю базу данных и таблицу в ней.

sudo nano /etc/postgresql/16/main/pg\_hba.conf

Найдите в файле блок для локального подключения по IPV4 и замените следующей строкой

# IPv4 local connections:

host all all 127.0.0.1/32 trust

sudo systemctl restart postgresql.service #перезапуск

psql -h 127.0.0.1 -U postgres -c 'CREATE DATABASE test' #Создание бд test

psql -h 127.0.0.1 -U postgres -d test -c 'CREATE TABLE my(some\_id serial PRIMARY KEY, some\_text text);'

#Создание таблицы

Задача: Создать резервную копию БД test и автоматизировать задачу через cron

pg\_dump -h 127.0.0.1 -U postgres test > $(date "+%Y-%m-%d\_%H:%M:%S").sql #Создание бекапа с датой в имени

На основе команды бекапа создадим скрипт в домашнем каталоге для использования его в расписании задач

touch backupdb.sh #создание файла

nano backupdb.sh #редактирование файла

#имя BD

**db\_name=test**

#пользователя для подключения

**db\_user=postgres**

#хост подключения

**db\_host=127.0.0.1**

#каталог для размещения бекапов (/home/имя\_пользователя/backup/имя\_БД)

**backupfolder=~/backup/$db\_name**

#имя файла резервной копии с датой и временем создания в названии в формате .sql

**sqlfile=$backupfolder/$db\_name-$(date +%d-%m-%Y\_%H-%M-%S).sql**

#имя файла сжатой резервной копии с датой и временем создания в названии

**zipfile=$backupfolder/$db\_name-$(date +%d-%m-%Y\_%H-%M-%S).sql.gz**

#Сначала нужной создать каталог для хранения резервный копий

**mkdir -p $backupfolder**

#команда для запуска резервной копии с использованием созданных ранее переменных

**pg\_dump -U $db\_user -h $db\_host $db\_name | gzip > $zipfile**

chmod +x backupdb.sh #права на запуск

Для проверки и наглядности работы расписания установим выполнения бекапа БД каждые 5 мин

crontab -e #редактирование crontab

\*/5 \* \* \* \* /home/имя\_пользователя/backupdb.sh

Убедитесь, что каждые 5 минут в каталоге /home/имя\_пользователя/backup/имя\_БД создается сжатая резервная копия

ll ~/backup/test/ #проверка каталога

Задача: Закомментируйте в crontab строку выполнения скрипта каждые 5 минут. И создайте свое расписание, по которому скрипт выполнится в **текущий день недели, текущего месяца, текущего числа, текущего часа в ближайшие минуты**

sudo ls -la /var/spool/cron/crontabs/ #просмотр файлов расписаний crontab всех пользователей

sudo nano /var/spool/cron/crontabs/имя\_пользователя

**m h dom mon dow command**

**минуты часы число месяц день\_недели команда**

**минуты 0-59** #минута запуска

**часы 0-23** #час запуска

**число 1-31** #число месяца запуска

**месяц 1-12** #месяц запуска

**день\_недели 0-6** #день недели запуска (0 - воскресенье)

**команда** #команда или скрипт с указанием абсолютного пути