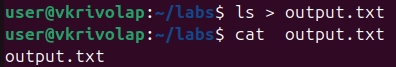
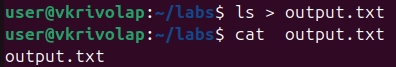
**Цель работы:** Изучить основные принципы работы с текстом и текстовыми файлами в Linux.

**1 Ознакомиться и изучить на практике программы и утилиты, предназначенные для работы с текстом в операционных системах Linux и Windows.**

**2. При помощи механизма перенаправления записать в файл output.txt содержимое домашнего каталога.**



**3. Вывести на экран содержимое файла output.txt.**



**4. Произвести замену знаков препинания (,.-) на знак + в файле output.txt. Результат сохранить в файле result.txt. При замене знаков в Linux следует воспользоваться утилитой sed и перенаправить результат в новый файл.**

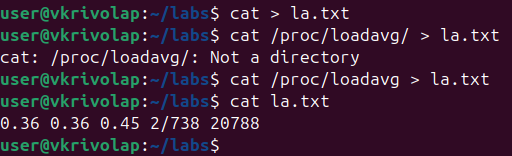
****

**5. Результат замены вывести на экран.**

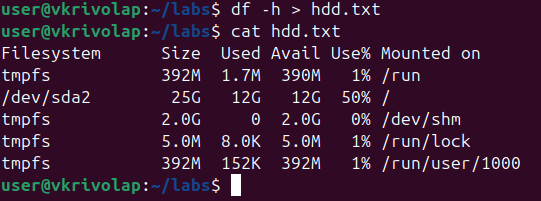
****

**6. Сбор системной информации:**

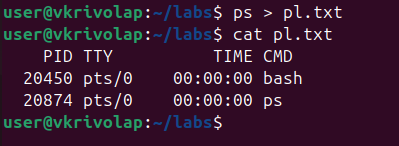
**6.1 Сохранить текущий Load Average в файл la.txt:**

****

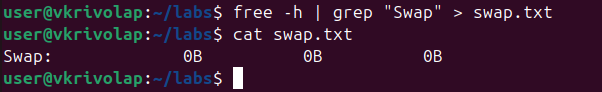
**6.2 Сохранить текущую загрузку дисков (размеры) в файл hdd.txt:**

****

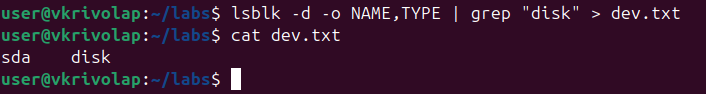
**6.3 Сохранить текущий список процессов пользователей в файл pl.txt:**

****

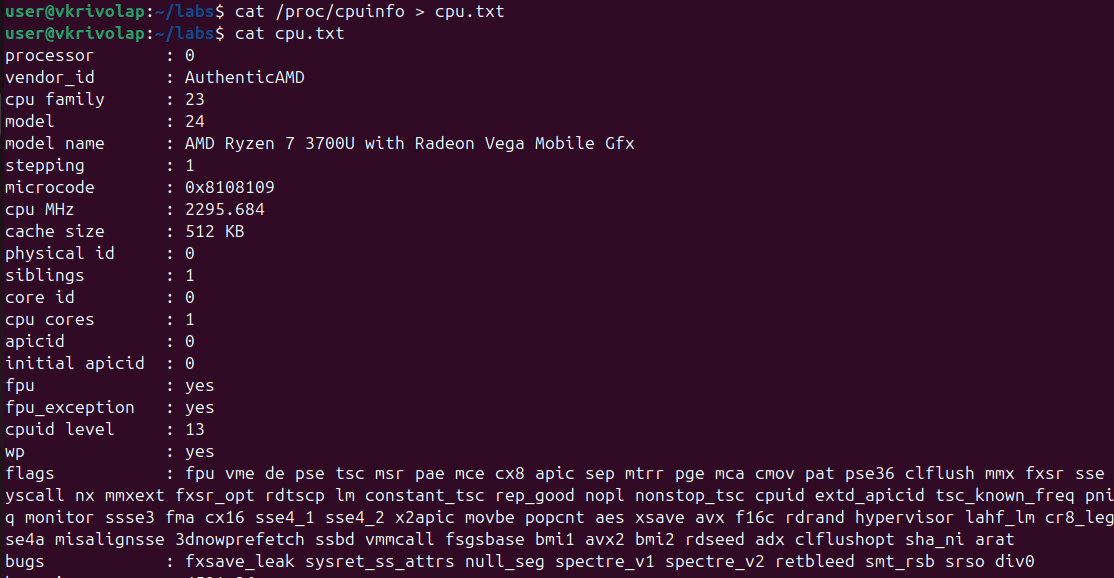
**6.4 Сохранить информацию о SWAP-е в файл swap.txt:**

****

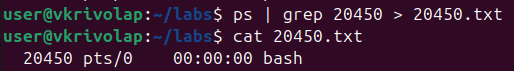
**6.5 Сохранить список всех SATA устройств в файл dev.txt:**

****

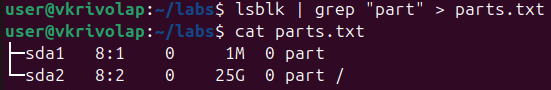
**6.6 Сохранить информацию о процессоре в файл cpu.txt:**

****

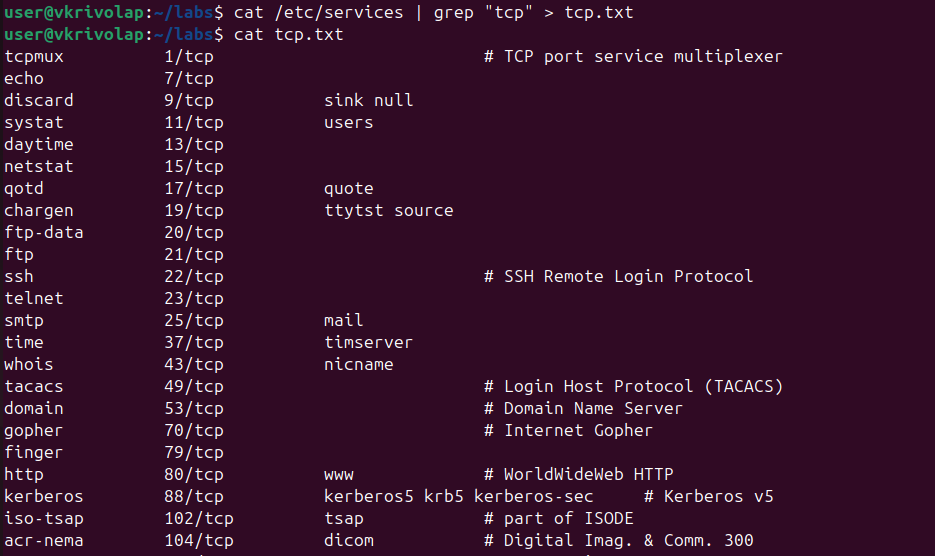
**6.7 Сохранить информацию о любом процессе в файл <id\_ процесса>.txt:**

****

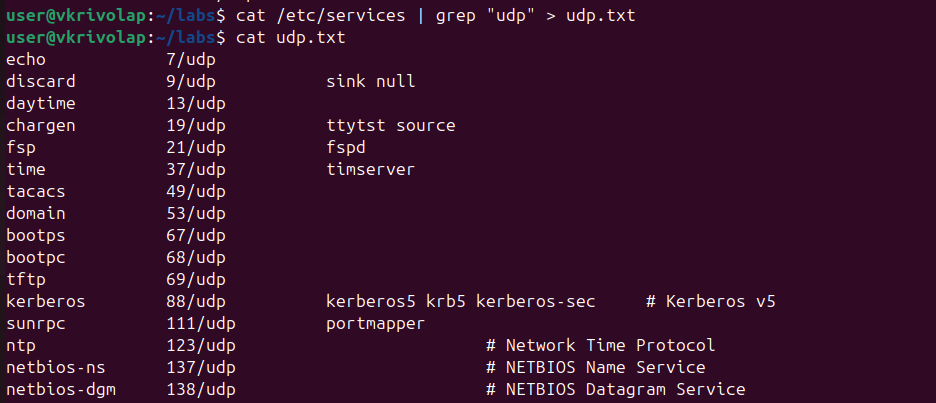
**6.8 Сохранить информацию о партициях в файл parts.txt:**

****

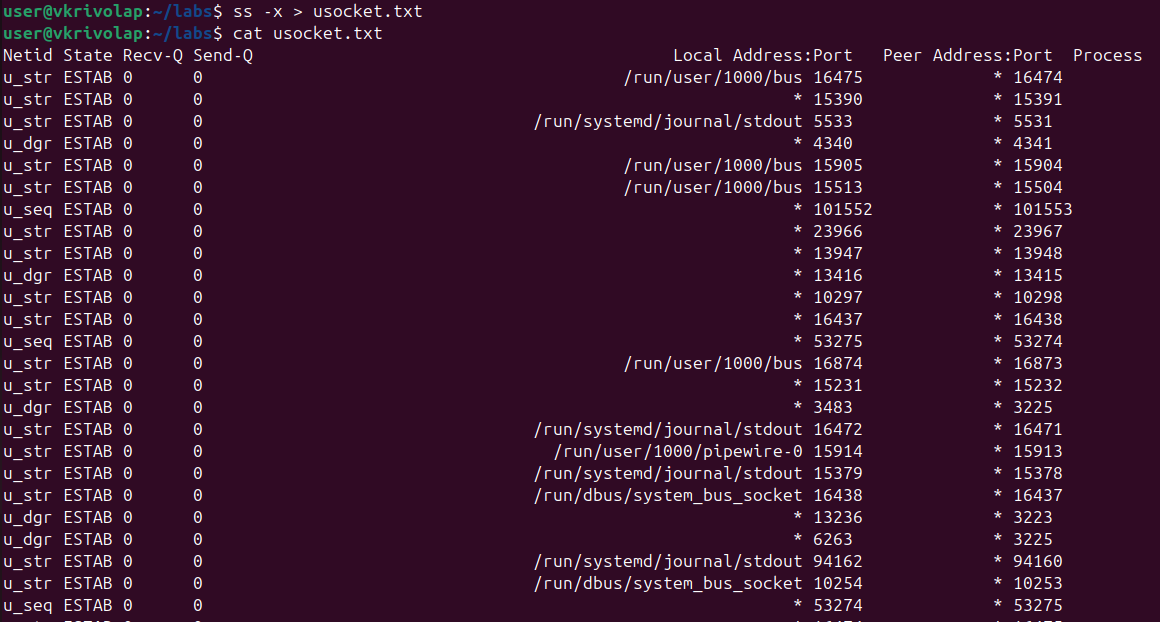
**6.9 Получить список всех TCP портов на машине и сохранить в файл tcp.txt:**

****

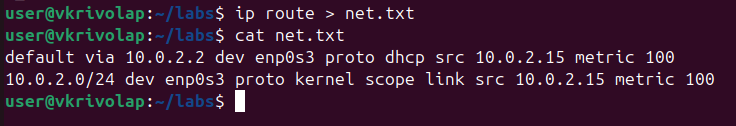
**6.10 Получить список всех UDP портов на машине и сохранить в файл udp.txt:**

****

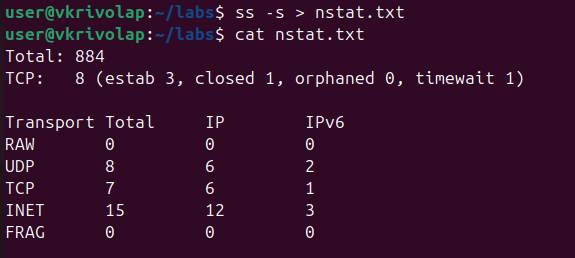
**6.11 Получить список всех UNIX Socket портов на машине и сохранить в файл usocket.txt:**

****

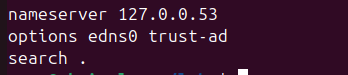
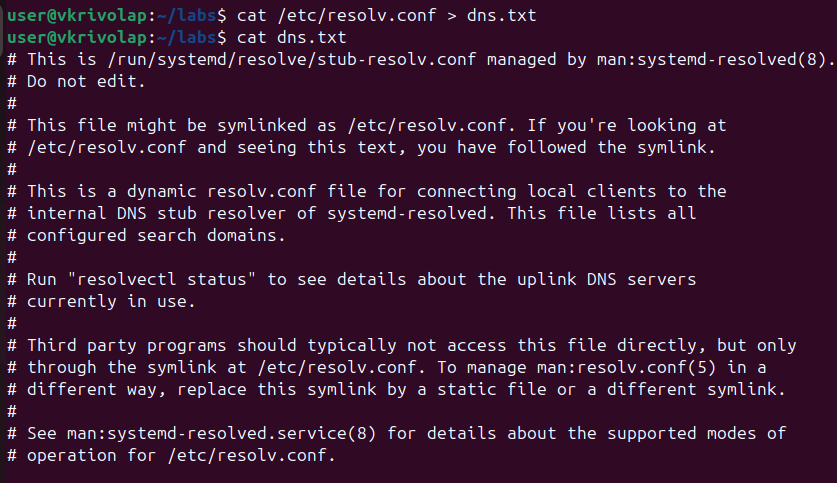
**6.12 Получить таблицу маршрутизации на машине и сохранить в файл net.txt:**

****

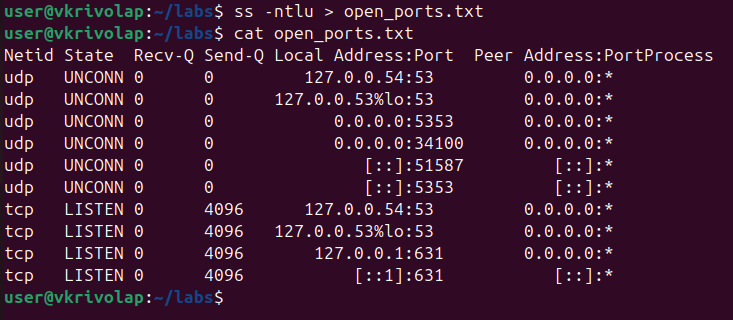
**6.13 Получить статистику для каждого протокола на машине и сохранить в файл nstat.txt:**

****

**6.14 Получить список DNS и сохранить в файл dns.txt:**

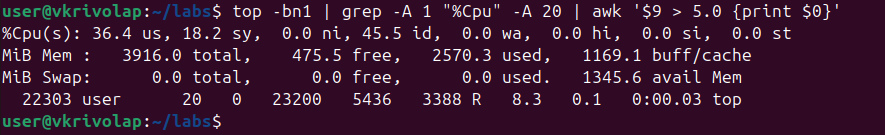
****

**6.15 Получить список всех открытых портов и сохранить в файл open\_ports.txt**

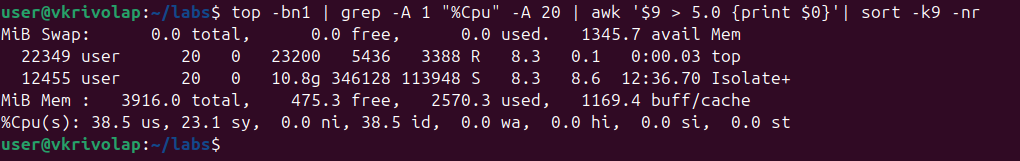
****

**7. Управление процессами и службами:**

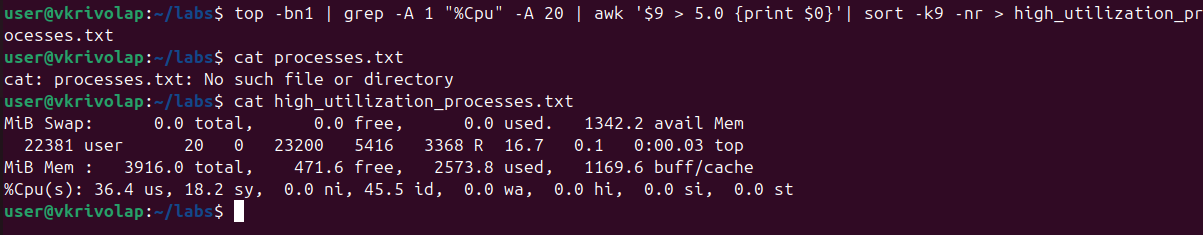
**7.1 Используя команды для получения информации о процесса в ОС получите список процессов, использующие более 5% CPU**

****

**7.2 Отсортируйте список в порядке убывания**

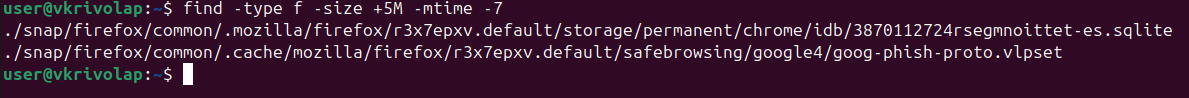
****

**7.3 Запишите информацию о процессах в файл high\_utilization\_processes.txt.**

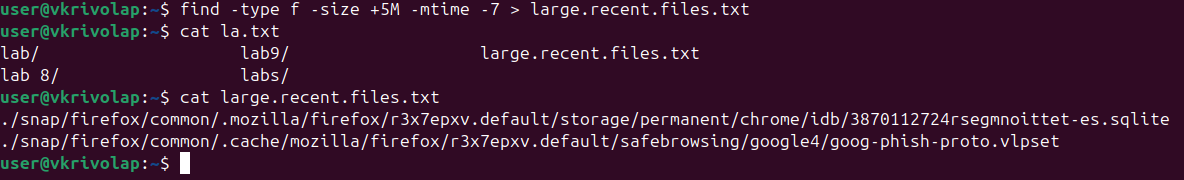
****

**8. Поиск файлов по размеру и дате изменения:**

**8.1 Используйте команду find, чтобы найти все файлы в каталоге /home, размер которых превышает 5MB и которые были изменены за последние 7 дней.**

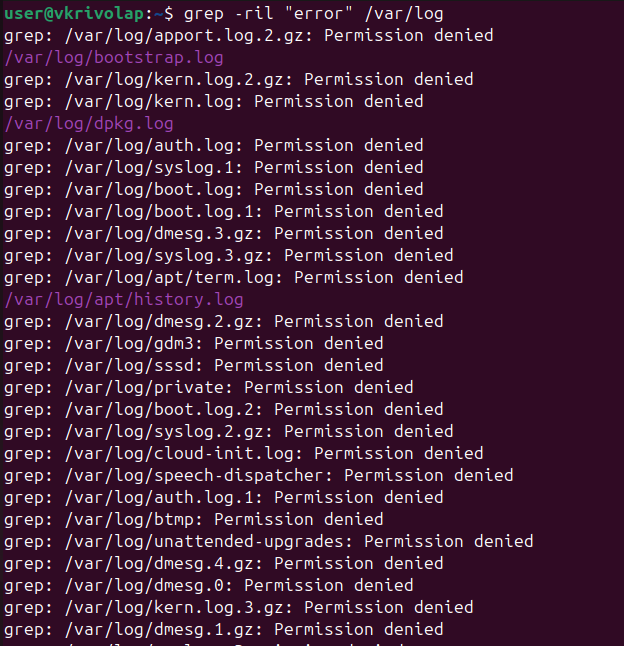
****

**8.2 Запишите список найденных файлов в файл large\_recent\_files.txt.**

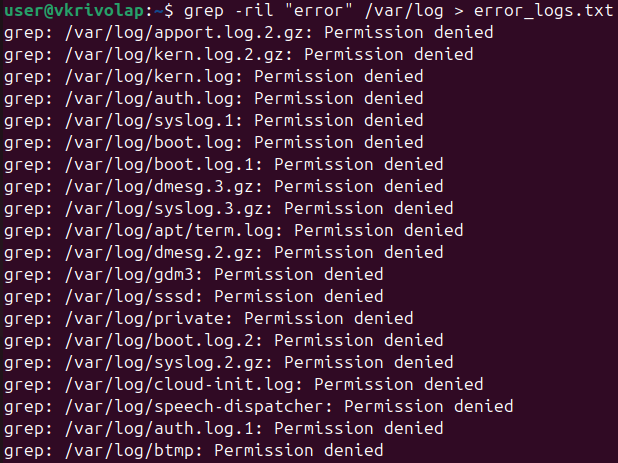
****

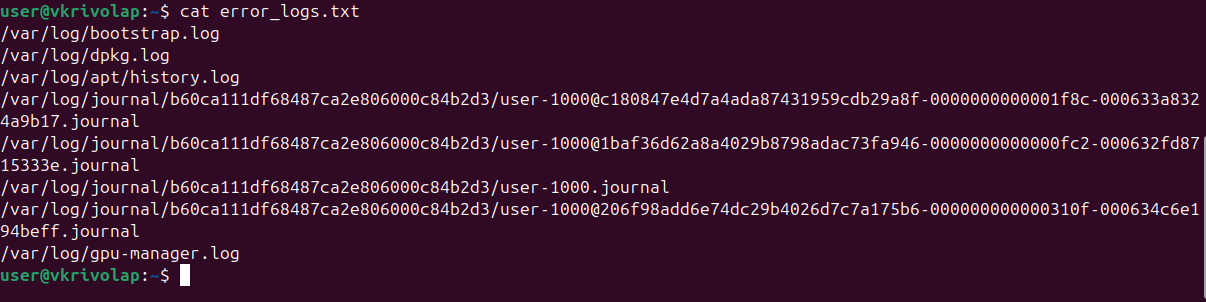
**9. Поиск файлов с определенным содержимым:**

**9.1 Используйте команду grep, чтобы найти все файлы в каталоге /var/log, содержащие слово "error".**

****

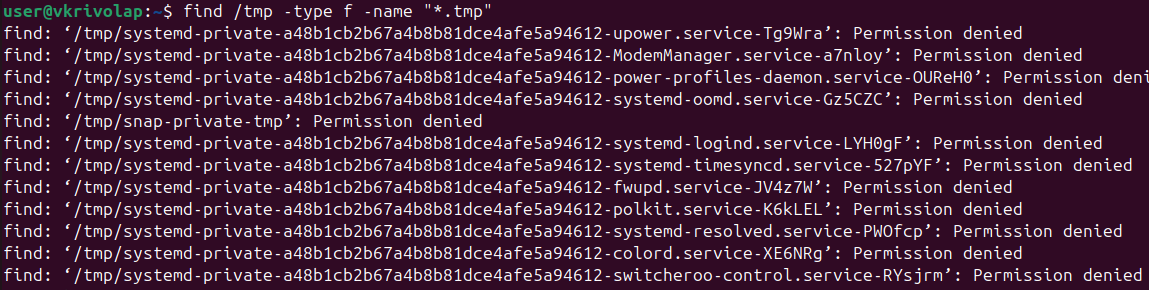
**9.2 Запишите список найденных файлов и строки, содержащие "error", в файл error\_logs.txt.**

****

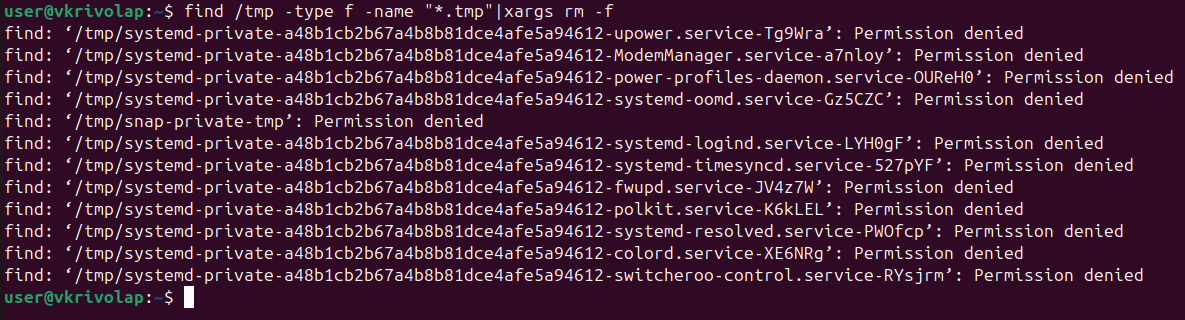
****

**10. Поиск и удаление временных файлов:**

**10.1 Выполните поиск файлов с расширением .tmp в каталоге /tmp.**

****

**10.2 Используйте команду xargs, чтобы удалить найденные файлы.**

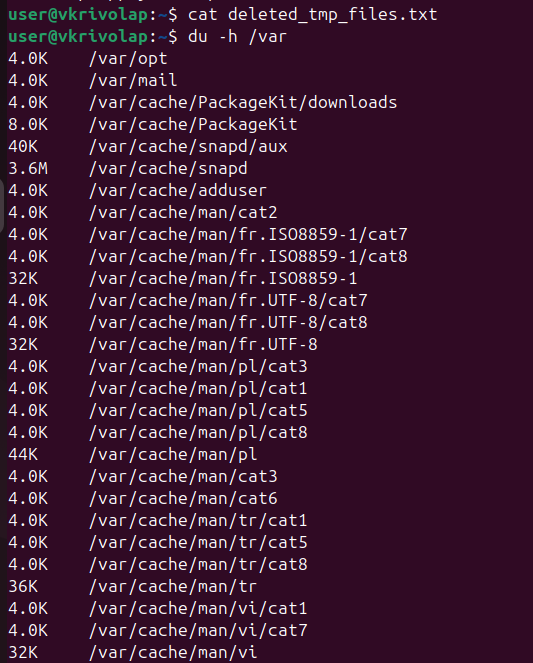
****

**10.3 Запишите список удаленных файлов в файл deleted\_tmp\_files.txt.**

****

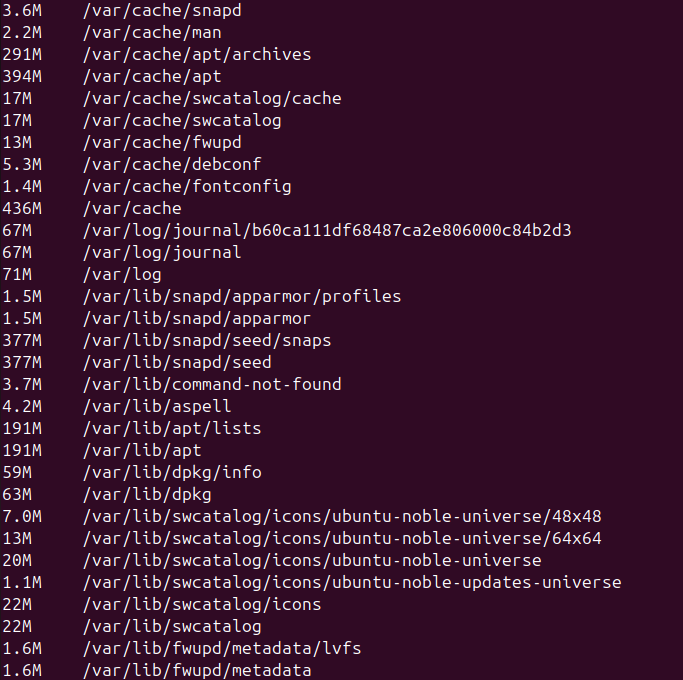
**11. Анализ использования дискового пространства:**

**11.1 Получите информацию о размере всех файлов и папок в каталоге /var.**

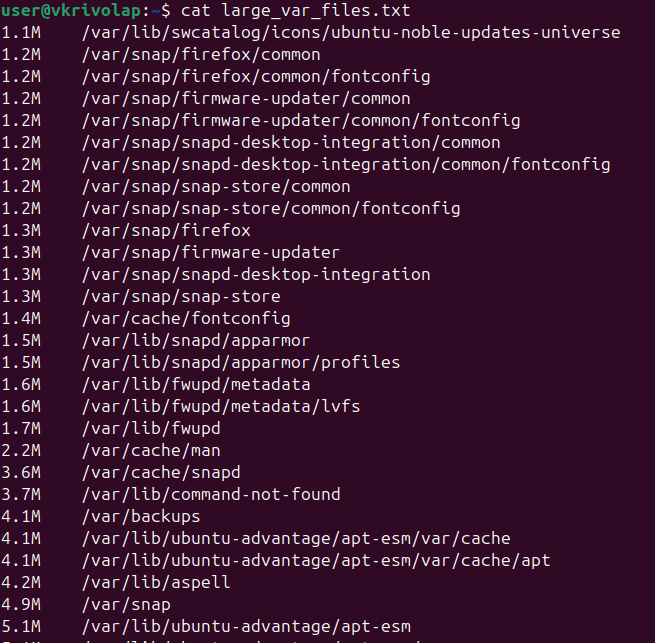
****

**11.2 Отфильтруйте вывод, оставив только только информацию о файлах, размер которых больше 1MB.**

****

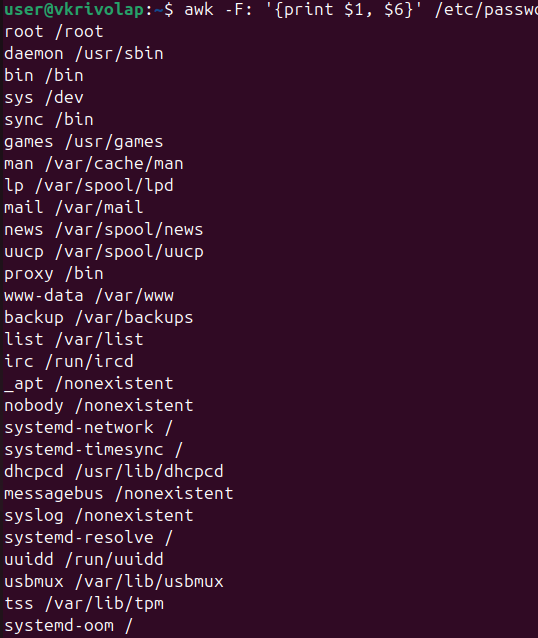
****

**11.3 Отсортируйте вывод по размеру и запишите результат в файл large\_var\_files.txt.**

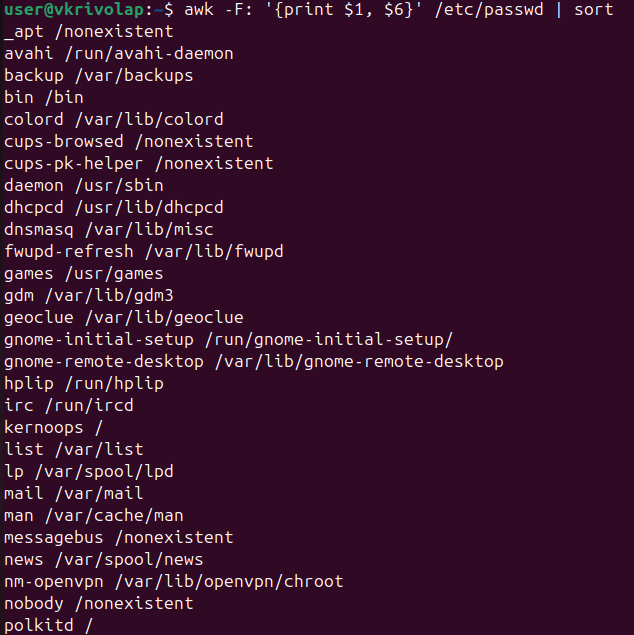
****

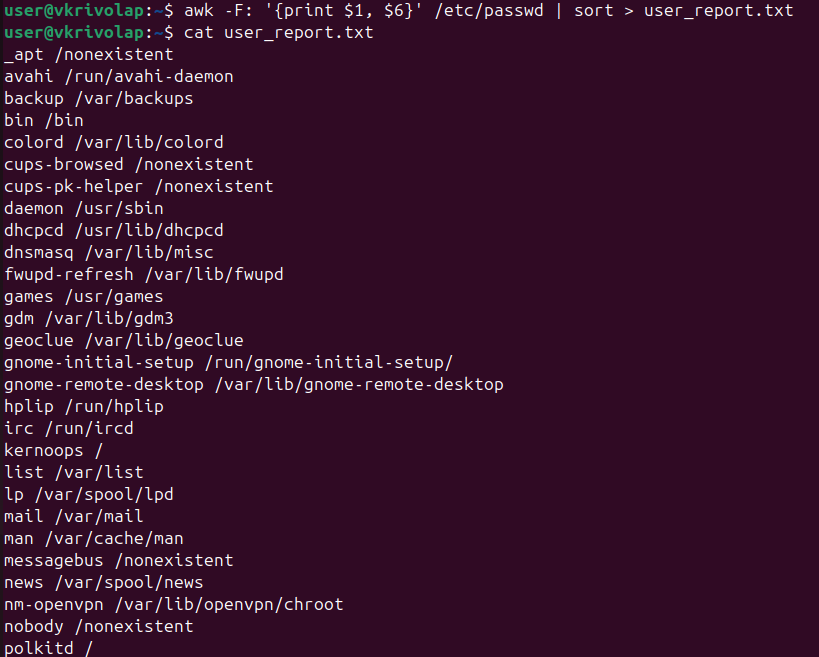
**12. Создание отчета о пользователях системы:**

**12.1 Используйте команду awk, чтобы извлечь имена пользователей и их домашние каталоги из файла /etc/passwd.**

****

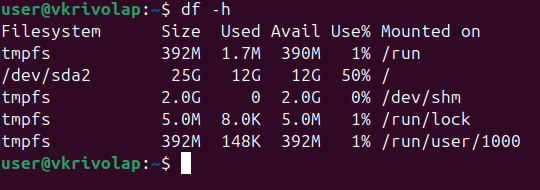
**12.2 Отсортируйте пользователей по имени.**

****

**12.3 Запишите результат в файл user\_report.txt.  
**

**13. Создание отчета о дисковом пространстве:**

**13.1 Получите информацию о дисковом пространстве.**

****

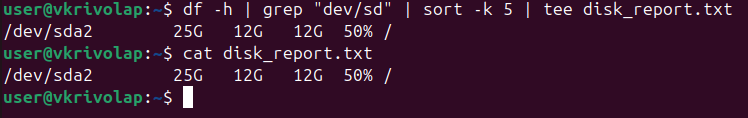
**13.2 Отфильтруйте вывод, чтобы оставить только строки, содержащие /dev/sd.**

****

**13.3 Сортируйте вывод по использованию диска**

****

**13.4 Передайте вывод в команду tee, чтобы записать его в файл disk\_report.txt и одновременно отобразить на экране.**

****

**Контрольные вопросы**

1. Для чего используются текстовые файлы в ОС?

В ОС текстовые файлы используются для для хранения конфигураций, логов, отчётов об ошибках.

1. Что такое стандартные файлы ОС?

Существует как минимум 3 стандартных файла:

● стандартный файл ввода (stdio); ● стандартный файл вывода (stdout); ● стандартный файл ошибок (stderr).

Эти три стандартных файла доступны любому пользовательскому процессу с момента его возникновения. При работе пользовательского процесса можно организовать вывод в файл stdout (standard output), а сообщения о состоянии – в специальный файл стандартный вывод ошибок stderr (standard error). Многие программы принимают ввод из специального файла с названием стандартный ввод stdin (standard input), который по умолчанию связан с клавиатурой.

1. Как осуществляется перенаправление работы стандартных файлов?

Чтобы перенаправить стандартный вывод в другой файл вместо экрана, нужно добавить в команду оператор перенаправления > и имя файла.

Для добавления вывода в конец существующего файла используем оператор перенаправления >>.

Чтобы перенаправить стандартный вывод ошибок, нужно указать его дескриптор файла. Программа может производить вывод в любой из нескольких нумерованных файловых потоков. Первые три из них мы упомянули как стандартный ввод, вывод и вывод ошибок. Командная оболочка ссылается на них как на файловые дескрипторы 0, 1 и 2 соответственно.

1. Какие утилиты имеются в Linux для работы с текстом?
   1. echo – выводит строку текста в стандартный вывод;
   2. cat – объединяет и выводит их в стандартный вывод;
   3. less и more – осуществляют постраничный вывод информации на экран;
   4. sort – сортирует строки текста;
   5. uniq – сообщает о повторяющихся строках или удаляет их;
   6. head – выводит первые строки из файла;
   7. tail – выводит последние строки из файла;
   8. tee – читает данные со стандартного ввода и записывает в стандартный вывод и в файлы;
   9. cut – удаляет фрагменты из каждой строки в файлах;
   10. paste – выполняет слияние строк из файлов;
   11. join – объединяет строки из двух файлов по общему полю;
   12. comm – выполняет построчное сравнение двух сортированных файлов;
   13. diff – выполняет построчное сравнение файлов;
   14. patch – применяет diff-файл (файл с результатами сравнения командой к оригиналу;
   15. tr – перекодирует или удаляет символы;
   16. sed – потоковый редактор для фильтрации и преобразования текста,
   17. aspell – интерактивная программа проверки орфографии.

1. Как произвести сохранение вывода любой команды в файл?

Сохранение производится с помощью направления вывода команды в файл. Данное действие производится при помощи “>” (команда > файл).

1. Как сделать замену символов в текстовом файле?

С помощью команды sed, указав необходимые символы для замены.