МОЛДАВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ

ДЕПАРТАМЕНТ ИНФОРМАТИКИ

**Лабораторная работа № 1**

**по курсу**

**Sisteme de monitorizare și analiză a infrastructurii IT și a aplicațiilor**

**Тема:**

**“Основы Linux”**

Выполнил: Bogdanov Iurii,

студент группы I2302

Проверил: D. Borș

Кишинёв, 2025

**Цель лабораторной работы:**

Изучить основы операционной системы Linux, включая структуру файловой системы, базовые команды для работы с ней, создание и управление пользователями, а также основы управления процессами и правами доступа.

**Ход работы**

**Раздел 1: Структура файловой системы**

**Теоретическая часть:**

*Основные каталоги файловой системы Linux:*

• / — корневая директория, от которой начинается вся структура.

• /bin — бинарные исполнимые файлы (основные утилиты, доступные для всех

пользователей).

• /etc — системные конфигурационные файлы.

• /home — домашние директории пользователей.

• /lib — библиотеки, необходимые для работы программ.

• /tmp — временные файлы.

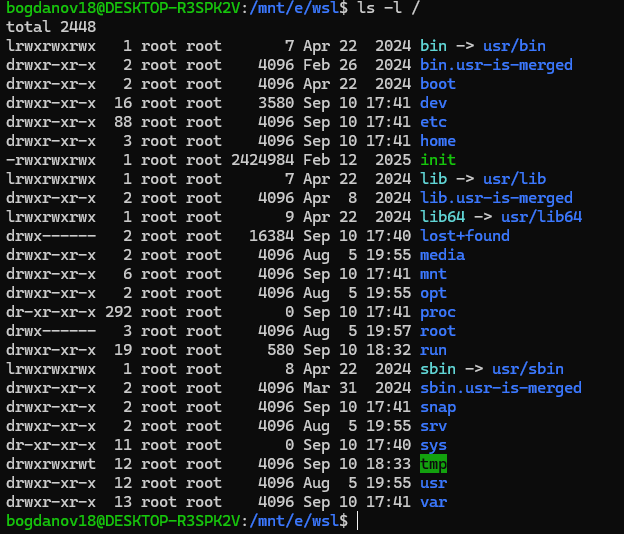
• /var — файлы, изменяющиеся во время работы системы (логи, базы данных и т.д.).

• /usr — утилиты и программы, установленные для использования пользователями.

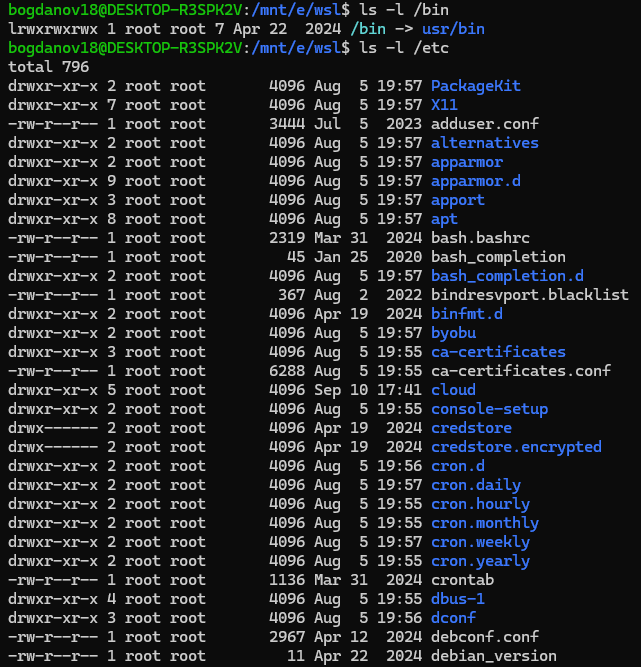
• /dev — устройства, доступные в системе.

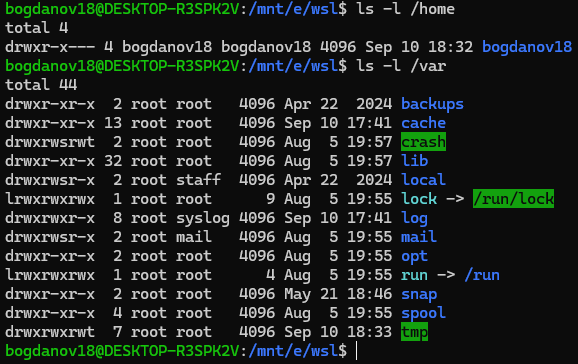
**Практическая часть:**

1. Используя команду **ls**, просмотрим содержимое корневой директории **/**.



2. Ознакомимся с содержимым других системных директорий с помощью **ls** **/**директория.





**Назначение:**

- /bin — базовые исполнимые файлы (утилиты оболочки).

- /etc — конфигурационные файлы системы.

- /home — домашние каталоги пользователей.

- /var — изменяющиеся во времени данные (логи, очереди, БД).

3. Используем команду **tree** для визуализации структуры каталогов (на первом уровне).



**Раздел 2: Базовые команды Linux**

**Теоретическая часть:**

*Основные команды для работы с файловой системой:*

• ls — выводит список файлов и каталогов в текущей директории.

• cd — смена текущей директории.

• pwd — выводит путь к текущей директории.

• cp — копирование файлов и каталогов.

• mv — перемещение или переименование файлов.

• rm — удаление файлов и каталогов.

• mkdir — создание каталога.

• rmdir — удаление пустого каталога.

**Практическая часть:**

1. Создаём каталог с помощью **mkdir**.

Создан каталог test и файл sample.txt.

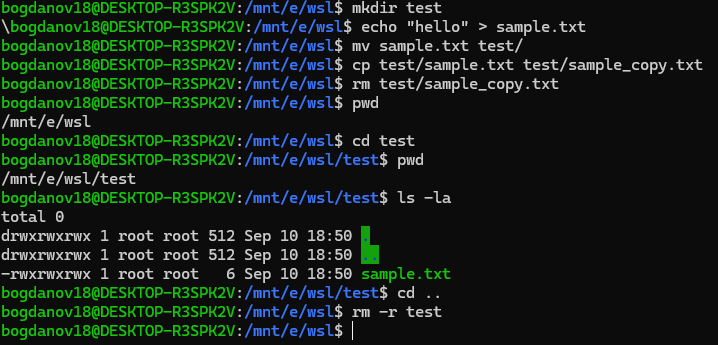
2. Перемещаем в него файл с помощью **mv**.

Файл sample.txt перенесён в test, а затем скопирован в sample\_copy.txt.

3. Скопируем файл с помощью **cp**.

Удалена копия файла и сам каталог test.

4. Удаляем файл и каталог с помощью **rm**.  
pwd показывает путь к текущему каталогу, ls выводит список файлов с правами и владельцами.



**Раздел 3: Управление пользователями и правами**

**Теоретическая часть:**

*Каждый файл и каталог в Linux имеет три типа прав доступа:*

• r — право на чтение.

• w — право на запись.

• x — право на выполнение.

Права могут быть назначены владельцу файла, группе и всем остальным пользователям.

*Команды для управления пользователями:*

• useradd — создание нового пользователя.

• passwd — изменение пароля пользователя.

• usermod — изменение свойств пользователя.

• groupadd — создание новой группы.

Команды для работы с правами доступа:

• chmod — изменение прав доступа.

• chown — изменение владельца и группы файла.

• chgrp — изменение группы файла.

**Практическая часть:**

1. Создаём нового пользователя с помощью **useradd** и устанавливаем для него пароль с помощью **passwd**.

Создан новый пользователь labuser с домашней директорией.

C:\Users\Asus\OneDrive\Изображения\Снимки экрана\Снимок экрана 2025-09-10 191928.png

Результат выполнения команды id labuser. Отображаются UID, GID и группа пользователя labuser, что подтверждает его успешное создание.

C:\Users\Asus\OneDrive\Изображения\Снимки экрана\Снимок экрана 2025-09-10 191937.png

2. Ознакамливаемся с текущими правами доступа к файлам с помощью команды **ls -**

Создание файла demo.txt и просмотр его начальных прав доступа с помощью ls -l. По умолчанию владельцу доступны чтение и запись (rw-), группе и остальным — только чтение (r--).

C:\Users\Asus\OneDrive\Изображения\Снимки экрана\Снимок экрана 2025-09-10 192427.png

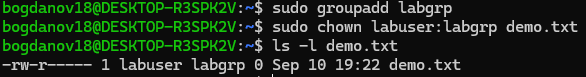
3. Изменяем права доступа с помощью **chmod** и проверяем результат.

Изменение прав доступа к файлу demo.txt с помощью числовой формы команды chmod 640. Теперь владельцу доступны чтение и запись (rw-), группе — только чтение (r--), остальные пользователи доступа не имеют (---).

C:\Users\Asus\OneDrive\Изображения\Снимки экрана\Снимок экрана 2025-09-10 192435.png

4. Изменяем владельца файла с помощью **chown**.

Создание новой группы labgrp и смена владельца и группы файла demo.txt с помощью команды chown. В результате владельцем стал пользователь labuser, а группой — labgrp.

****

**Раздел 4: Процессы в Linux**

**Теоретическая часть:**

*Основные команды для работы с процессами:*

• ps — отображение списка текущих процессов.

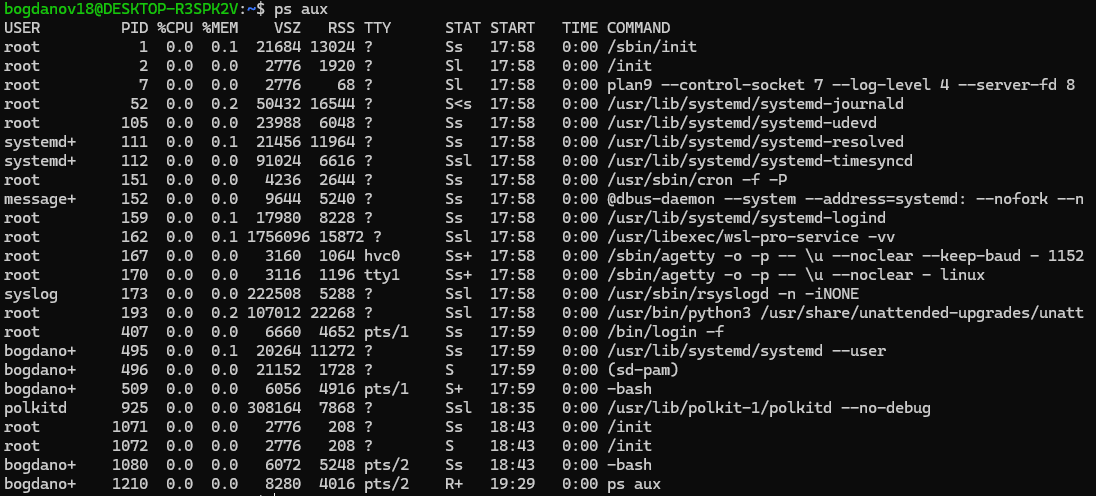
• top — отображение динамического списка процессов с подробной информацией.

• kill — завершение процесса по PID.

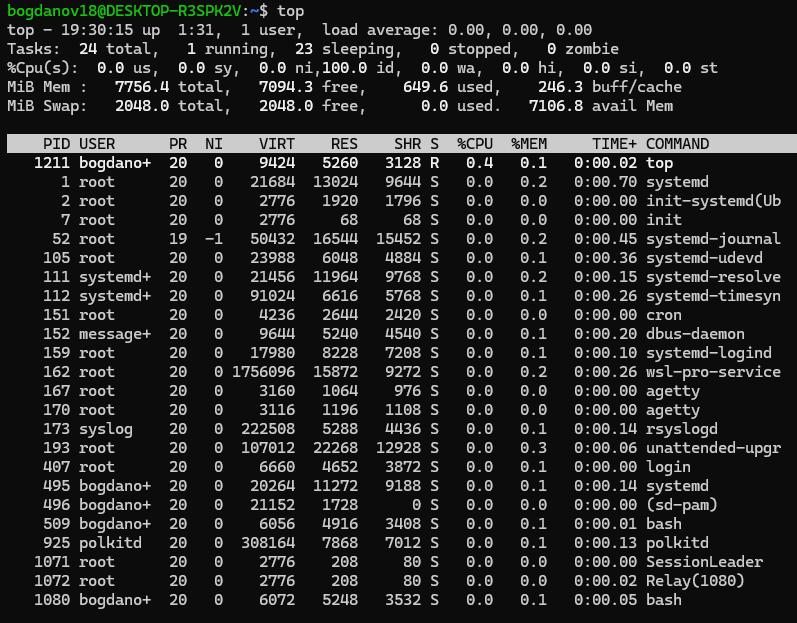
• bg и fg — запуск процесса в фоне и возвращение в передний план.

**Практическая часть:**

1. Ознакомимся с процессами в системе с помощью **ps aux**.

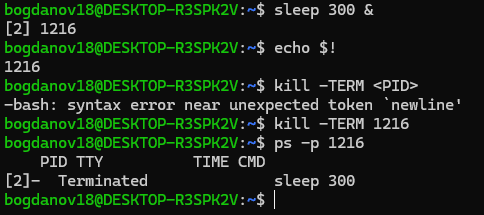


2. Используем команду **top** для отображения списка процессов.



3. Завершим процесс с помощью **kill**.

Запуск фонового процесса sleep 300 &, получение его PID через echo $!, завершение процесса с помощью kill -TERM <PID> и подтверждение завершения командой ps -p <PID>.



**Раздел 5: Обработка текста с помощью grep, sed и awk**

**Теоретическая часть:**

**grep** — Поиск текста в файлах

**grep** — это команда, которая используется для поиска строк, содержащих заданный текст (или

регулярное выражение) в одном или нескольких файлах.

*Основные синтаксисы:*

• grep <шаблон> <файл> — поиск строк, содержащих шаблон.

• grep -r <шаблон> <директория> — рекурсивный поиск по всей директории.

• grep -i <шаблон> <файл> — поиск без учета регистра.

• grep -v <шаблон> <файл> — вывод строк, которые не содержат шаблон.

**sed** — Потоковый редактор

**sed** (stream editor) используется для обработки и преобразования текста в потоке (например, в

файлах или выводе других команд). Это один из самых мощных инструментов для

редактирования текстовых данных.

*Основные синтаксисы:*

• sed 's/<старый\_шаблон>/<новый\_шаблон>/' <файл> — заменяет первый

найденный экземпляр шаблона на новый в каждой строке.

• sed 's/<старый\_шаблон>/<новый\_шаблон>/g' <файл> — заменяет все

экземпляры шаблона в строках.

• sed -i 's/<старый\_шаблон>/<новый\_шаблон>/' <файл> — делает замену

непосредственно в файле (инлайн).

**awk** — Обработка и анализ текстовых данных

**awk** — это язык программирования для обработки и анализа текстовых данных, в частности,

предназначен для работы с данными, организованными в виде таблиц (например, CSV).

*Основные синтаксисы:*

• awk '{print $1}' <файл> — выводит первый столбец данных.

• awk '{print $1, $3}' <файл> — выводит первый и третий столбцы.

• awk '/<шаблон>/{print $0}' <файл> — выводит все строки, содержащие

шаблон.

• awk '{if ($3 > 50) print $1}' <файл> — выводит первую колонку, если

значение в третьей колонке больше 50.

**Практическая часть:**

1. **grep:**

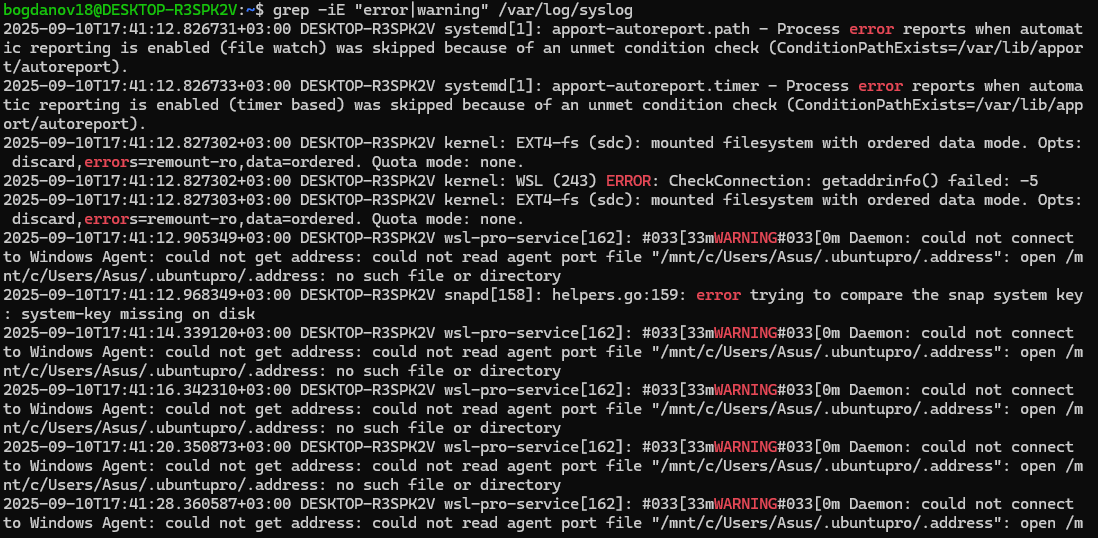
• Используем команду grep для поиска определенных строк в системных

журналах, например, в файле /var/log/syslog.

• Выполняем поиск всех строк, содержащих ошибку **error** или предупреждение

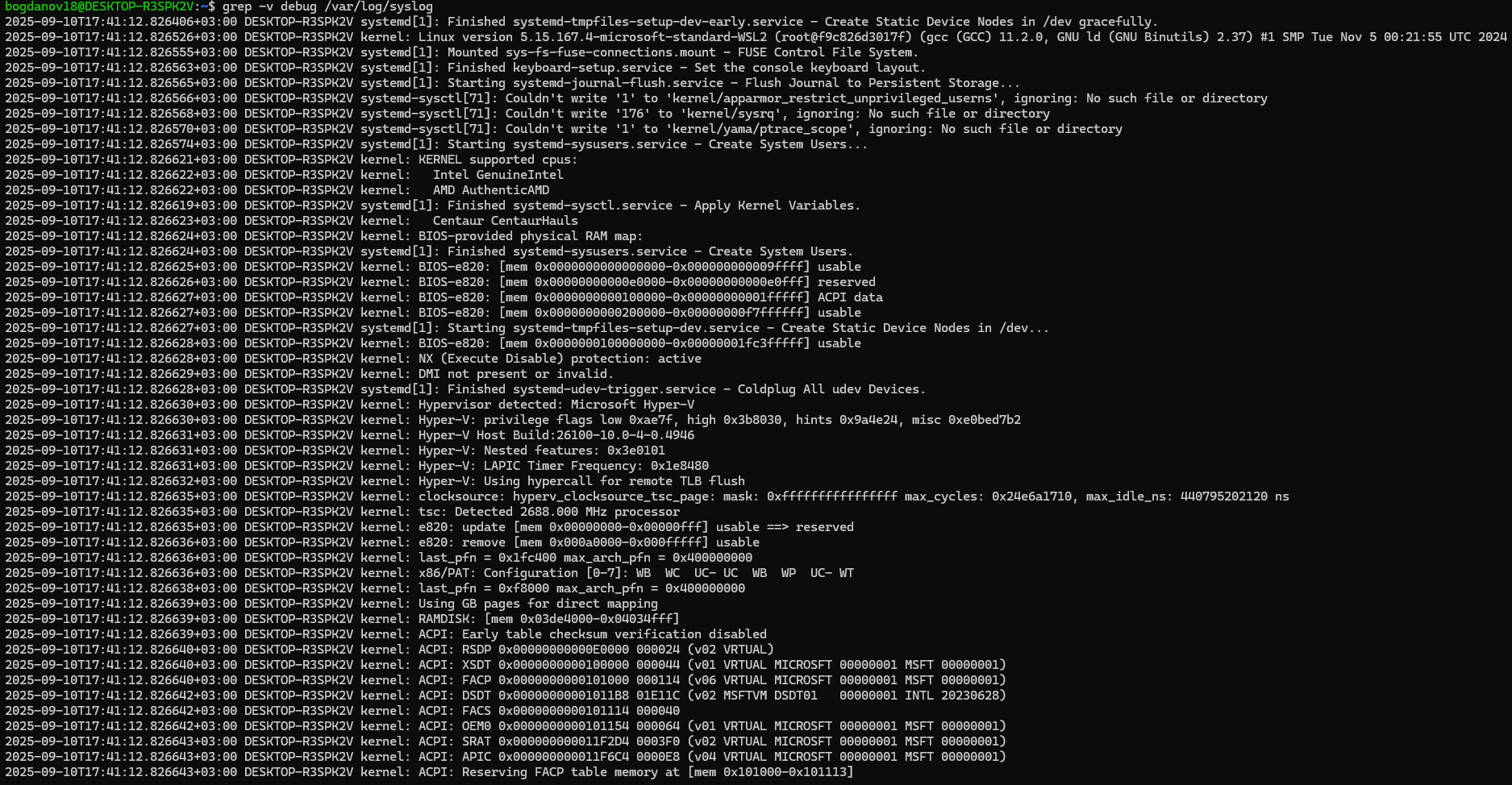
**warning**.

Поиск строк с ключевыми словами error и warning в системном журнале /var/log/syslog с помощью команды grep -iE "error|warning". На выводе видны строки с ошибками (ERROR) и предупреждениями (WARNING).



• Находим все строки, не содержащие слово **debug**.

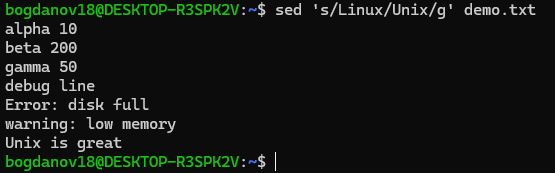
Использование команды dmesg | grep для поиска сообщений ядра. На выводе видны строки с системными событиями, включая загрузку драйверов и параметры ядра.



2. **sed:**

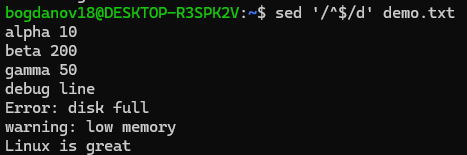
• Заменяем все вхождения слова **Linux** на **Unix** в тексте.

Использование команды sed 's/Linux/Unix/g' demo.txt для замены всех вхождений слова Linux на Unix в тексте.



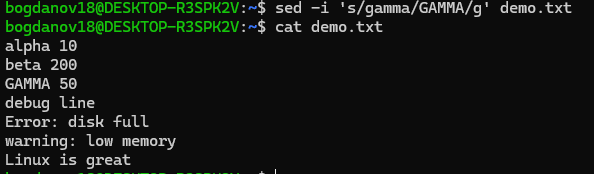
• Используя **sed**, удалим все пустые строки из файла (заменяем их на что-то другое или просто удаляем).

Применение команды sed '/^$/d' demo.txt для удаления пустых строк из файла. В результате вывод содержит только строки с данными.



• Сделаем замену в файле с помощью опции **-i**, чтобы изменения были примененынепосредственно в файле.

Применение команды sed -i 's/gamma/GAMMA/g' demo.txt с опцией -i. Изменения внесены прямо в файл: строка с gamma преобразована в GAMMA.

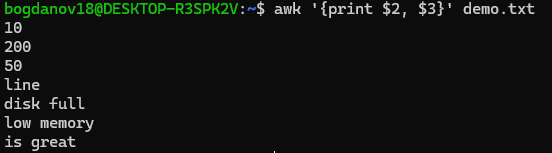


3**. awk:**

• Используем команду **awk**, чтобы вывести второй и третий столбцы из файла с

данными, разделенными пробелами или табуляциями.

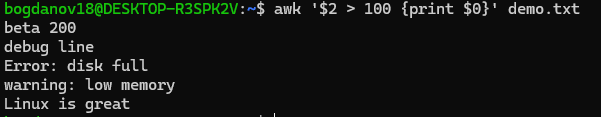
Использование команды awk '{print $2, $3}' demo.txt для вывода второго и третьего столбцов из файла. В результате отображаются числовые значения и слова, находящиеся во 2-м и 3-м столбцах строк.



• Применяем команду **awk** для выборки строк, где значение в одном из столбцов

больше заданного числа.

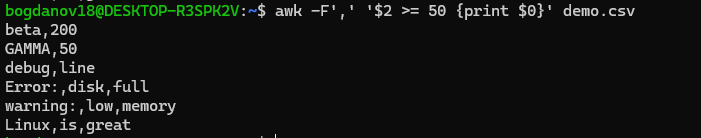
Использование команды awk '$2 > 100 {print $0}' demo.txt для фильтрации строк, где значение второго столбца превышает 100. В результате отображаются только строки с числом 200 и другими подходящими данными.



• Используя **awk**, обработаем файл **CSV** и выведем только те строки, которые

соответствуют определенному условию.

Использование команды awk -F',' '$2 >= 50 {print $0}' demo.csv для обработки CSV-файла. В результате отображаются строки, где значение во втором столбце больше или равно 50.



**Заключение**

В ходе лабораторной работы я освоил базовые команды Linux, познакомился со структурой файловой системы, научился создавать пользователей и управлять правами доступа. Также я попробовал работать с процессами и использовать утилиты grep, sed и awk для обработки текстовых данных.

Работа показалась полезной и наглядной: особенно интересной оказалась часть с правами доступа и управление процессами, так как это основа администрирования в Linux. Полученный опыт помог лучше понять принципы работы системы и дал практические навыки, которые пригодятся в дальнейшем обучении и реальной работе.