

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`, посмотрим содержимое корневой директории `/`.

```
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ ls -l /
total 2448
lrwxrwxrwx 1 root root 7 Apr 22 2024 bin -> usr/bin
drwxr-xr-x 2 root root 4096 Feb 26 2024 bin.usr-is-merged
drwxr-xr-x 2 root root 4096 Apr 22 2024 boot
drwxr-xr-x 16 root root 3580 Sep 10 17:41 dev
drwxr-xr-x 88 root root 4096 Sep 10 17:41 etc
drwxr-xr-x 3 root root 4096 Sep 10 17:41 home
-rwxrwxrwx 1 root root 2424984 Feb 12 2025 init
lrwxrwxrwx 1 root root 7 Apr 22 2024 lib -> usr/lib
drwxr-xr-x 2 root root 4096 Apr 8 2024 lib.usr-is-merged
lrwxrwxrwx 1 root root 9 Apr 22 2024 lib64 -> usr/lib64
drwx----- 2 root root 16384 Sep 10 17:40 lost+found
drwxr-xr-x 2 root root 4096 Aug 5 19:55 media
drwxr-xr-x 6 root root 4096 Sep 10 17:41 mnt
drwxr-xr-x 2 root root 4096 Aug 5 19:55 opt
dr-xr-xr-x 292 root root 0 Sep 10 17:41 proc
drwx----- 3 root root 4096 Aug 5 19:57 root
drwxr-xr-x 19 root root 580 Sep 10 18:32 run
lrwxrwxrwx 1 root root 8 Apr 22 2024 sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 Mar 31 2024 sbin.usr-is-merged
drwxr-xr-x 2 root root 4096 Sep 10 17:41 snap
drwxr-xr-x 2 root root 4096 Aug 5 19:55 srv
dr-xr-xr-x 11 root root 0 Sep 10 17:40 sys
drwxrwxrwt 12 root root 4096 Sep 10 18:33 tmp
drwxr-xr-x 12 root root 4096 Aug 5 19:55 usr
drwxr-xr-x 13 root root 4096 Sep 10 17:41 var
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$
```

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

```
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ ls -l /bin
lrwxrwxrwx 1 root root 7 Apr 22 2024 /bin -> usr/bin
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ ls -l /etc
total 796
drwxr-xr-x 2 root root 4096 Aug 5 19:57 PackageKit
drwxr-xr-x 7 root root 4096 Aug 5 19:57 X11
-rw-r--r-- 1 root root 3444 Jul 5 2023 adduser.conf
drwxr-xr-x 2 root root 4096 Aug 5 19:57 alternatives
drwxr-xr-x 2 root root 4096 Aug 5 19:57 apparmor
drwxr-xr-x 9 root root 4096 Aug 5 19:57 apparmor.d
drwxr-xr-x 3 root root 4096 Aug 5 19:57 appport
drwxr-xr-x 8 root root 4096 Aug 5 19:57 apt
-rw-r--r-- 1 root root 2319 Mar 31 2024 bash.bashrc
-rw-r--r-- 1 root root 45 Jan 25 2020 bash_completion
drwxr-xr-x 2 root root 4096 Aug 5 19:57 bash_completion.d
-rw-r--r-- 1 root root 367 Aug 2 2022 bindresvport.blacklist
drwxr-xr-x 2 root root 4096 Apr 19 2024 binfmt.d
drwxr-xr-x 2 root root 4096 Aug 5 19:57 byobu
drwxr-xr-x 3 root root 4096 Aug 5 19:55 ca-certificates
-rw-r--r-- 1 root root 6288 Aug 5 19:55 ca-certificates.conf
drwxr-xr-x 5 root root 4096 Sep 10 17:41 cloud
drwxr-xr-x 2 root root 4096 Aug 5 19:55 console-setup
drwx----- 2 root root 4096 Apr 19 2024 credstore
drwx----- 2 root root 4096 Apr 19 2024 credstore.encrypted
drwxr-xr-x 2 root root 4096 Aug 5 19:56 cron.d
drwxr-xr-x 2 root root 4096 Aug 5 19:57 cron.daily
drwxr-xr-x 2 root root 4096 Aug 5 19:55 cron.hourly
drwxr-xr-x 2 root root 4096 Aug 5 19:55 cron.monthly
drwxr-xr-x 2 root root 4096 Aug 5 19:57 cron.weekly
drwxr-xr-x 2 root root 4096 Aug 5 19:55 cron.yearly
-rw-r--r-- 1 root root 1136 Mar 31 2024 crontab
drwxr-xr-x 4 root root 4096 Aug 5 19:55 dbus-1
drwxr-xr-x 3 root root 4096 Aug 5 19:56 dconf
-rw-r--r-- 1 root root 2967 Apr 12 2024 debconf.conf
-rw-r--r-- 1 root root 11 Apr 22 2024 debian_version
```

```
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ ls -l /home
total 4
drwxr-xr-x 4 bogdanov18 bogdanov18 4096 Sep 10 18:32 bogdanov18
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ ls -l /var
total 44
drwxr-xr-x 2 root root 4096 Apr 22 2024 backups
drwxr-xr-x 13 root root 4096 Sep 10 17:41 cache
drwxrwsrwt 2 root root 4096 Aug 5 19:57 crash
drwxr-xr-x 32 root root 4096 Aug 5 19:57 lib
drwxrwsr-x 2 root staff 4096 Apr 22 2024 local
lrwxrwxrwx 1 root root 9 Aug 5 19:55 lock -> /run/lock
drwxrwxr-x 8 root syslog 4096 Sep 10 17:41 log
drwxrwsr-x 2 root mail 4096 Aug 5 19:55 mail
drwxr-xr-x 2 root root 4096 Aug 5 19:55 opt
lrwxrwxrwx 1 root root 4 Aug 5 19:55 run -> /run
drwxr-xr-x 2 root root 4096 May 21 18:46 snap
drwxr-xr-x 4 root root 4096 Aug 5 19:55 spool
drwxrwxrwt 7 root root 4096 Sep 10 18:33 tmp
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$
```

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).

```
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ tree -L 1 /
/
├── bin -> usr/bin
├── bin.usr-is-merged
├── boot
├── dev
├── etc
├── home
├── init
├── lib -> usr/lib
├── lib.usr-is-merged
├── lib64 -> usr/lib64
├── lost+found
├── media
├── mnt
├── opt
├── proc
├── root
├── run
├── sbin -> usr/sbin
├── sbin.usr-is-merged
├── snap
├── srv
├── sys
├── tmp
├── usr
└── var

25 directories, 1 file
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ |
```

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.

```
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ mkdir test
\bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ echo "hello" > sample.txt
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ mv sample.txt test/
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ cp test/sample.txt test/sample_copy.txt
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ rm test/sample_copy.txt
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ pwd
/mnt/e/wsl
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ cd test
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl/test$ pwd
/mnt/e/wsl/test
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl/test$ ls -la
total 0
drwxrwxrwx 1 root root 512 Sep 10 18:50 .
drwxrwxrwx 1 root root 512 Sep 10 18:50 ..
-rwxrwxrwx 1 root root  6 Sep 10 18:50 sample.txt
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl/test$ cd ..
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ rm -r test
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ |
```

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.

```
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ sudo useradd -m labuser
```

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

```
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ id labuser
uid=1001(labuser) gid=1001(labuser) groups=1001(labuser)
bogdanov18@DESKTOP-R3SPK2V:/mnt/e/wsl$ |
```

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

Создание файла demo.txt и просмотр его начальных прав доступа с помощью **ls -**

1. По умолчанию владельцу доступны чтение и запись (rw-), группе и остальным — только чтение (r--).

```
bogdanov18@DESKTOP-R3SPK2V:~$ touch demo.txt
bogdanov18@DESKTOP-R3SPK2V:~$ ls -l demo.txt
-rw-r--r-- 1 bogdanov18 bogdanov18 0 Sep 10 19:22 demo.txt
```

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 (---).

```
bogdanov18@DESKTOP-R3SPK2V:~$ chmod 640 demo.txt
bogdanov18@DESKTOP-R3SPK2V:~$ ls -l demo.txt
-rw-r----- 1 bogdanov18 bogdanov18 0 Sep 10 19:22 demo.txt
```

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.

```
bogdanov18@DESKTOP-R3SPK2V:~$ sudo groupadd labgrp
bogdanov18@DESKTOP-R3SPK2V:~$ sudo chown labuser:labgrp demo.txt
bogdanov18@DESKTOP-R3SPK2V:~$ ls -l demo.txt
-rw-r----- 1 labuser labgrp 0 Sep 10 19:22 demo.txt
```

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**.

```
bogdanov18@DESKTOP-R3SPK2V:~$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root             1  0.0  0.1 21684 13024 ?        Ss   17:58   0:00 /sbin/init
root             2  0.0  0.0   2776   1920 ?        Sl   17:58   0:00 /init
root             7  0.0  0.0   2776     68 ?        Sl   17:58   0:00 plan9 --control-socket 7 --log-level 4 --server-fd 8
root            52  0.0  0.2 50432 16544 ?        S<s  17:58   0:00 /usr/lib/systemd/systemd-journald
root           105  0.0  0.0 23988  6048 ?        Ss   17:58   0:00 /usr/lib/systemd/systemd-udevd
systemd+       111  0.0  0.1 21456 11964 ?        Ss   17:58   0:00 /usr/lib/systemd/systemd-resolved
systemd+       112  0.0  0.0 91024  6616 ?        Ssl  17:58   0:00 /usr/lib/systemd/systemd-timesyncd
root           151  0.0  0.0   4236   2644 ?        Ss   17:58   0:00 /usr/sbin/cron -f -p
message+       152  0.0  0.0   9644   5240 ?        Ss   17:58   0:00 @dbus-daemon --system --address=systemd: --nofork --n
root           159  0.0  0.1 17980   8228 ?        Ss   17:58   0:00 /usr/lib/systemd/systemd-logind
root           162  0.0  0.1 1756096 15872 ?       Ssl  17:58   0:00 /usr/libexec/wsl-pro-service -vv
root           167  0.0  0.0   3160   1064 hvc0     Ss+  17:58   0:00 /sbin/agetty -o -p -- \u --noclear --keep-baud - 1152
root           170  0.0  0.0   3116   1196 tty1     Ss+  17:58   0:00 /sbin/agetty -o -p -- \u --noclear - linux
syslog         173  0.0  0.0 222508  5288 ?        Ssl  17:58   0:00 /usr/sbin/rsyslogd -n -iNONE
root           193  0.0  0.2 107012 22268 ?        Ssl  17:58   0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unatt
root           407  0.0  0.0   6660   4652 pts/1    Ss   17:59   0:00 /bin/login -f
bogdano+       495  0.0  0.1 20264 11272 ?        Ss   17:59   0:00 /usr/lib/systemd/systemd --user
bogdano+       496  0.0  0.0 21152   1728 ?        S    17:59   0:00 (sd-pam)
bogdano+       509  0.0  0.0   6056   4916 pts/1    S+   17:59   0:00 -bash
polkitd        925  0.0  0.0 308164  7868 ?        Ssl  18:35   0:00 /usr/lib/polkit-1/polkitd --no-debug
root          1071  0.0  0.0   2776    208 ?        Ss   18:43   0:00 /init
root          1072  0.0  0.0   2776    208 ?        S    18:43   0:00 /init
bogdano+      1080  0.0  0.0   6072   5248 pts/2    Ss   18:43   0:00 -bash
bogdano+      1210  0.0  0.0   8280   4016 pts/2    R+   19:29   0:00 ps aux
```

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

```
bogdanov18@DESKTOP-R3SPK2V:~$ top
top - 19:30:15 up 1:31, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 24 total, 1 running, 23 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7756.4 total, 7094.3 free, 649.6 used, 246.3 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 7106.8 avail Mem

   PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
  1211 bogdano+  20   0   9424   5260  3128 R   0.4   0.1   0:00.02 top
      1 root      20   0  21684 13024  9644 S   0.0   0.2   0:00.70 systemd
      2 root      20   0   2776   1920  1796 S   0.0   0.0   0:00.00 init-systemd(Ub
      7 root      20   0   2776     68   68 S   0.0   0.0   0:00.00 init
     52 root     19  -1 50432 16544 15452 S   0.0   0.2   0:00.45 systemd-journal
    105 root     20   0 23988  6048  4884 S   0.0   0.1   0:00.36 systemd-udevd
    111 systemd+  20   0 21456 11964  9768 S   0.0   0.2   0:00.15 systemd-resolve
    112 systemd+  20   0 91024  6616  5768 S   0.0   0.1   0:00.26 systemd-timesyn
    151 root     20   0   4236   2644  2420 S   0.0   0.0   0:00.00 cron
    152 message+  20   0   9644   5240  4540 S   0.0   0.1   0:00.20 dbus-daemon
    159 root     20   0 17980   8228  7208 S   0.0   0.1   0:00.10 systemd-logind
    162 root     20   0 1756096 15872  9272 S   0.0   0.2   0:00.26 wsl-pro-service
    167 root     20   0   3160   1064   976 S   0.0   0.0   0:00.00 agetty
    170 root     20   0   3116   1196  1108 S   0.0   0.0   0:00.00 agetty
    173 syslog    20   0 222508  5288  4436 S   0.0   0.1   0:00.14 rsyslogd
    193 root     20   0 107012 22268 12928 S   0.0   0.3   0:00.06 unattended-upgr
    407 root     20   0   6660   4652  3872 S   0.0   0.1   0:00.00 login
    495 bogdano+  20   0 20264 11272  9188 S   0.0   0.1   0:00.14 systemd
    496 bogdano+  20   0 21152   1728    0 S   0.0   0.0   0:00.00 (sd-pam)
    509 bogdano+  20   0   6056   4916  3408 S   0.0   0.1   0:00.01 bash
    925 polkitd    20   0 308164  7868  7012 S   0.0   0.1   0:00.13 polkitd
   1071 root     20   0   2776    208    80 S   0.0   0.0   0:00.00 SessionLeader
   1072 root     20   0   2776    208    80 S   0.0   0.0   0:00.02 Relay(1080)
  1080 bogdano+  20   0   6072   5248  3532 S   0.0   0.1   0:00.05 bash
```


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>`.

```
bogdanov18@DESKTOP-R3SPK2V:~$ sleep 300 &
[2] 1216
bogdanov18@DESKTOP-R3SPK2V:~$ echo $!
1216
bogdanov18@DESKTOP-R3SPK2V:~$ kill -TERM <PID>
-bash: syntax error near unexpected token `newline'
bogdanov18@DESKTOP-R3SPK2V:~$ kill -TERM 1216
bogdanov18@DESKTOP-R3SPK2V:~$ ps -p 1216
  PID TTY          TIME CMD
[2]-  Terminated      sleep 300
bogdanov18@DESKTOP-R3SPK2V:~$ |
```

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 — Поточковый 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).

```
bogdanov18@DESKTOP-R3SPK2V:~$ grep -iE "error|warning" /var/log/syslog
2025-09-10T17:41:12.826731+03:00 DESKTOP-R3SPK2V systemd[1]: appport-autoreport.path - Process error reports when automatic reporting is enabled (file watch) was skipped because of an unmet condition check (ConditionPathExists=/var/lib/apport/autoreport).
2025-09-10T17:41:12.826733+03:00 DESKTOP-R3SPK2V systemd[1]: appport-autoreport.timer - Process error reports when automatic reporting is enabled (timer based) was skipped because of an unmet condition check (ConditionPathExists=/var/lib/apport/autoreport).
2025-09-10T17:41:12.827302+03:00 DESKTOP-R3SPK2V kernel: EXT4-fs (sdc): mounted filesystem with ordered data mode. Opts: discard,errors=remount-ro,data=ordered. Quota mode: none.
2025-09-10T17:41:12.827302+03:00 DESKTOP-R3SPK2V kernel: WSL (243) ERROR: CheckConnection: getaddrinfo() failed: -5
2025-09-10T17:41:12.827303+03:00 DESKTOP-R3SPK2V kernel: EXT4-fs (sdc): mounted filesystem with ordered data mode. Opts: discard,errors=remount-ro,data=ordered. Quota mode: none.
2025-09-10T17:41:12.905349+03:00 DESKTOP-R3SPK2V wsl-pro-service[162]: #033[33mWARNING#033[0m Daemon: could not connect to Windows Agent: could not get address: could not read agent port file "/mnt/c/Users/Asus/.ubuntupro/.address": open /mnt/c/Users/Asus/.ubuntupro/.address: no such file or directory
2025-09-10T17:41:12.968349+03:00 DESKTOP-R3SPK2V snapd[158]: helpers.go:159: error trying to compare the snap system key : system-key missing on disk
2025-09-10T17:41:14.339120+03:00 DESKTOP-R3SPK2V wsl-pro-service[162]: #033[33mWARNING#033[0m Daemon: could not connect to Windows Agent: could not get address: could not read agent port file "/mnt/c/Users/Asus/.ubuntupro/.address": open /mnt/c/Users/Asus/.ubuntupro/.address: no such file or directory
2025-09-10T17:41:16.342310+03:00 DESKTOP-R3SPK2V wsl-pro-service[162]: #033[33mWARNING#033[0m Daemon: could not connect to Windows Agent: could not get address: could not read agent port file "/mnt/c/Users/Asus/.ubuntupro/.address": open /mnt/c/Users/Asus/.ubuntupro/.address: no such file or directory
2025-09-10T17:41:20.350873+03:00 DESKTOP-R3SPK2V wsl-pro-service[162]: #033[33mWARNING#033[0m Daemon: could not connect to Windows Agent: could not get address: could not read agent port file "/mnt/c/Users/Asus/.ubuntupro/.address": open /mnt/c/Users/Asus/.ubuntupro/.address: no such file or directory
2025-09-10T17:41:28.360587+03:00 DESKTOP-R3SPK2V wsl-pro-service[162]: #033[33mWARNING#033[0m Daemon: could not connect to Windows Agent: could not get address: could not read agent port file "/mnt/c/Users/Asus/.ubuntupro/.address": open /mnt/c/Users/Asus/.ubuntupro/.address: no such file or directory
```

- 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.

```
bogdanov18@DESKTOP-R3SPK2V:~$ grep -v debug /var/log/syslog
2025-09-10T17:41:12.826486+03:00 DESKTOP-R3SPK2V systemd[1]: Finished systemd-tmpfiles-setup-dev-early.service - Create Static Device Nodes in /dev gracefully.
2025-09-10T17:41:12.826526+03:00 DESKTOP-R3SPK2V kernel: Linux version 5.15.167.4-microsoft-standard-WSL2 (root@f9c826d3017f) (gcc (GCC) 11.2.0, GNU ld (GNU Binutils) 2.37) #1 SMP Tue Nov 5 00:21:55 UTC 2024
2025-09-10T17:41:12.826555+03:00 DESKTOP-R3SPK2V systemd[1]: Mounted sys-fs-fuse-connections.mount - FUSE Control File System.
2025-09-10T17:41:12.826563+03:00 DESKTOP-R3SPK2V systemd[1]: Finished keyboard-setup.service - Set the console keyboard layout.
2025-09-10T17:41:12.826565+03:00 DESKTOP-R3SPK2V systemd[1]: Starting systemd-journal-flush.service - Flush Journal to Persistent Storage...
2025-09-10T17:41:12.826566+03:00 DESKTOP-R3SPK2V systemd-sysctl[71]: Couldn't write '1' to 'kernel/apparmor_restrict_unprivileged_userns', ignoring: No such file or directory
2025-09-10T17:41:12.826568+03:00 DESKTOP-R3SPK2V systemd-sysctl[71]: Couldn't write '176' to 'kernel/sysrq', ignoring: No such file or directory
2025-09-10T17:41:12.826570+03:00 DESKTOP-R3SPK2V systemd-sysctl[71]: Couldn't write '1' to 'kernel/yama_ptrace_scope', ignoring: No such file or directory
2025-09-10T17:41:12.826574+03:00 DESKTOP-R3SPK2V systemd[1]: Starting systemd-sysusers.service - Create System Users...
2025-09-10T17:41:12.826621+03:00 DESKTOP-R3SPK2V kernel: KERNEL supported cpus:
2025-09-10T17:41:12.826622+03:00 DESKTOP-R3SPK2V kernel: Intel GenuineIntel
2025-09-10T17:41:12.826622+03:00 DESKTOP-R3SPK2V kernel: AMD AuthenticAMD
2025-09-10T17:41:12.826619+03:00 DESKTOP-R3SPK2V systemd[1]: Finished systemd-sysctl.service - Apply Kernel Variables.
2025-09-10T17:41:12.826623+03:00 DESKTOP-R3SPK2V kernel: Contaurus ContaurusHails
2025-09-10T17:41:12.826624+03:00 DESKTOP-R3SPK2V kernel: BIOS-provided physical RAM map:
2025-09-10T17:41:12.826624+03:00 DESKTOP-R3SPK2V systemd[1]: Finished systemd-sysusers.service - Create System Users.
2025-09-10T17:41:12.826625+03:00 DESKTOP-R3SPK2V kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009ffff] usable
2025-09-10T17:41:12.826626+03:00 DESKTOP-R3SPK2V kernel: BIOS-e820: [mem 0x00000000000e0000-0x00000000000e0fff] reserved
2025-09-10T17:41:12.826627+03:00 DESKTOP-R3SPK2V kernel: BIOS-e820: [mem 0x0000000000100000-0x00000000001fffff] ACPI data
2025-09-10T17:41:12.826627+03:00 DESKTOP-R3SPK2V kernel: BIOS-e820: [mem 0x0000000000200000-0x00000000007fffff] usable
2025-09-10T17:41:12.826627+03:00 DESKTOP-R3SPK2V systemd[1]: Starting systemd-tmpfiles-setup-dev.service - Create Static Device Nodes in /dev...
2025-09-10T17:41:12.826628+03:00 DESKTOP-R3SPK2V kernel: BIOS-e820: [mem 0x0000000010000000-0x000000001fc3ffff] usable
2025-09-10T17:41:12.826628+03:00 DESKTOP-R3SPK2V kernel: NX (Execute Disable) protection: active
2025-09-10T17:41:12.826629+03:00 DESKTOP-R3SPK2V kernel: DMI not present or invalid.
2025-09-10T17:41:12.826629+03:00 DESKTOP-R3SPK2V systemd[1]: Finished systemd-udev-trigger.service - Coldplug All udev Devices.
2025-09-10T17:41:12.826630+03:00 DESKTOP-R3SPK2V kernel: Hypervisor detected: Microsoft Hyper-V
2025-09-10T17:41:12.826630+03:00 DESKTOP-R3SPK2V kernel: Hyper-V: privilege flags 0xae7f, high 0x3b0830, hints 0x9a0e20, misc 0xe0bed7b2
2025-09-10T17:41:12.826631+03:00 DESKTOP-R3SPK2V kernel: Hyper-V Host Build:26100-10.0-4-0.4946
2025-09-10T17:41:12.826631+03:00 DESKTOP-R3SPK2V kernel: Hyper-V: Nested features: 0x3e0101
2025-09-10T17:41:12.826632+03:00 DESKTOP-R3SPK2V kernel: Hyper-V: LAPIC Timer Frequency: 0x1e0800
2025-09-10T17:41:12.826632+03:00 DESKTOP-R3SPK2V kernel: Hyper-V: Using hypercall for remote TLB flush
2025-09-10T17:41:12.826635+03:00 DESKTOP-R3SPK2V kernel: clocksource: hyperv_clocksource_tsc_page: mask: 0xffffffffffffff max_cycles: 0x24e6a1710, max_idle_ns: 440795202120 ns
2025-09-10T17:41:12.826635+03:00 DESKTOP-R3SPK2V kernel: tsc: Detected 2688.000 MHz processor
2025-09-10T17:41:12.826635+03:00 DESKTOP-R3SPK2V kernel: e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
2025-09-10T17:41:12.826636+03:00 DESKTOP-R3SPK2V kernel: e820: remove [mem 0x000a0000-0x000fffff] usable
2025-09-10T17:41:12.826636+03:00 DESKTOP-R3SPK2V kernel: last_pfn = 0x1fc000 max_arch_pfn = 0x400000000
2025-09-10T17:41:12.826636+03:00 DESKTOP-R3SPK2V kernel: x86/PAT: Configuration [0-7]: WB UC WC UC- WB WP UC- WT
2025-09-10T17:41:12.826638+03:00 DESKTOP-R3SPK2V kernel: last_pfn = 0x1fc000 max_arch_pfn = 0x400000000
2025-09-10T17:41:12.826639+03:00 DESKTOP-R3SPK2V kernel: Using GB pages for direct mapping
2025-09-10T17:41:12.826639+03:00 DESKTOP-R3SPK2V kernel: RAMDISK: [mem 0x83de4000-0x04034fff]
2025-09-10T17:41:12.826640+03:00 DESKTOP-R3SPK2V kernel: ACPI: Early table checksum verification disabled
2025-09-10T17:41:12.826640+03:00 DESKTOP-R3SPK2V kernel: ACPI: RSDP 0x0000000000000000 000024 (v02 VIRTUAL)
2025-09-10T17:41:12.826640+03:00 DESKTOP-R3SPK2V kernel: ACPI: XSDT 0x0000000000010000 000044 (v01 VIRTUAL MICROSOFT 00000001 MSFT 00000001)
2025-09-10T17:41:12.826640+03:00 DESKTOP-R3SPK2V kernel: ACPI: FACP 0x0000000000010000 000114 (v06 VIRTUAL MICROSOFT 00000001 MSFT 00000001)
2025-09-10T17:41:12.826642+03:00 DESKTOP-R3SPK2V kernel: ACPI: DSDT 0x0000000000010118 01E11C (v02 MSFTVM DSDT01 00000001 INTL 20230628)
2025-09-10T17:41:12.826642+03:00 DESKTOP-R3SPK2V kernel: ACPI: FACS 0x0000000000010114 000040
2025-09-10T17:41:12.826642+03:00 DESKTOP-R3SPK2V kernel: ACPI: OEM0 0x0000000000010115 000064 (v01 VIRTUAL MICROSOFT 00000001 MSFT 00000001)
2025-09-10T17:41:12.826643+03:00 DESKTOP-R3SPK2V kernel: ACPI: SRAT 0x0000000000011f20 0003f0 (v02 VIRTUAL MICROSOFT 00000001 MSFT 00000001)
2025-09-10T17:41:12.826643+03:00 DESKTOP-R3SPK2V kernel: ACPI: APIC 0x0000000000011f6c 0000e8 (v04 VIRTUAL MICROSOFT 00000001 MSFT 00000001)
2025-09-10T17:41:12.826643+03:00 DESKTOP-R3SPK2V kernel: ACPI: Reserving FACP table memory at [mem 0x101000-0x101113]
```

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.

```
bogdanov18@DESKTOP-R3SPK2V:~$ sed 's/Linux/Unix/g' demo.txt
alpha 10
beta 200
gamma 50
debug line
Error: disk full
warning: low memory
Unix is great
bogdanov18@DESKTOP-R3SPK2V:~$ |
```

- 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.

```
bogdanov18@DESKTOP-R3SPK2V:~$ sed '/^$/d' demo.txt
alpha 10
beta 200
gamma 50
debug line
Error: disk full
warning: low memory
Linux is great
```

- Сделаем замену в файле с помощью опции **-i**, чтобы изменения были применены непосредственно 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`.

```
bogdanov18@DESKTOP-R3SPK2V:~$ sed -i 's/gamma/GAMMA/g' demo.txt
bogdanov18@DESKTOP-R3SPK2V:~$ cat demo.txt
alpha 10
beta 200
GAMMA 50
debug line
Error: disk full
warning: low memory
Linux is great
```

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.

```
bogdanov18@DESKTOP-R3SPK2V:~$ awk '{print $2, $3}' demo.txt
10
200
50
line
disk full
low memory
is great
```

- 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.

```
bogdanov18@DESKTOP-R3SPK2V:~$ awk '$2 > 100 {print $0}' demo.txt
beta 200
debug line
Error: disk full
warning: low memory
Linux is great
```

- 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.

```
bogdanov18@DESKTOP-R3SPK2V:~$ awk -F',' '$2 >= 50 {print $0}' demo.csv
beta,200
GAMMA,50
debug,line
Error:,disk,full
warning:,low,memory
Linux,is,great
```

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.