

Лабораторная работа №8

**Поиск файлов. Перенаправление ввода-вывода. Просмотр
запущенных процессов**

Юсупова Ксения Равилевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Ответы на контрольные вопросы	15
4	Выводы	18

Список иллюстраций

2.1	Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc.	6
2.2	Допишем в этот же файл названия файлов, содержащихся в моём домашнем каталоге.	7
2.3	Вывели имена всех файлов из file.txt, имеющих расширение .conf .	8
2.4	названия все файлов из file.txt, имеющие расширение .conf в новый текстовой файл conf.txt	8
2.5	первый вариант нахождения файлов с символом с	8
2.6	анализируем вывод	9
2.7	Второй и третий варианты нахождения файлов с символом с . . .	10
2.8	выводим имена файлов из каталога /etc, начинающиеся с символа h	11
2.9	Смотрим на верность вывода	11
2.10	процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log	11
2.11	Удаляем файл ~/logfile	12
2.12	Запустили из консоли в фоновом режиме редактор gedit	12
2.13	Определим идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep.	12
2.14	Завершили процесс	13
2.15	используем df	13
2.16	используем du	13
2.17	выведем имена всех директорий, имеющих в вашем домашнем каталоге	14
2.18	получаем такой вывод	14

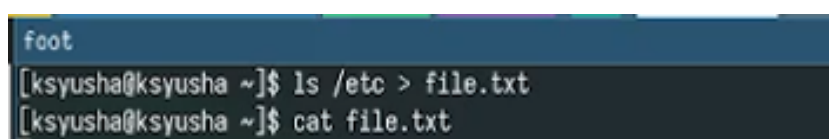
Список таблиц

1 Цель работы

Ознакомление с инструментами поиска файлов и фильтрации текстовых данных. Приобретение практических навыков: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

2 Выполнение лабораторной работы

Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc. (рис. 2.1).

A terminal window with a dark background and a blue title bar labeled 'foot'. The terminal shows two commands being executed: '[ksyusha@ksyusha ~]\$ ls /etc > file.txt' and '[ksyusha@ksyusha ~]\$ cat file.txt'.

```
foot
[ksyusha@ksyusha ~]$ ls /etc > file.txt
[ksyusha@ksyusha ~]$ cat file.txt
```

Рис. 2.1: Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc.

Допишем в этот же файл названия файлов, содержащихся в моём домашнем каталоге.(рис. 2.2).

```
[ksyusha@ksyusha ~]$ ls >> file.txt
[ksyusha@ksyusha ~]$ tac file.txt
Шаблоны
Рабочий стол
Общедоступные
Музыка
Изображения
Загрузки
Документы
Видео
work
ski.places
reports
play
pass.txt
my_os
monthly
may
LICENSE
git-extended
file.txt
feathers
Downloads
Documents
blog
```

Рис. 2.2: Допишем в этот же файл названия файлов, содержащихся в моём домашнем каталоге.

Выведем имена всех файлов из file.txt, имеющих расширение .conf(рис. 2.3).

```
[ksyusha@ksyusha ~]$ grep "\.conf" file.txt
asound.conf
chrony.conf
dnsmasq.conf
dracut.conf
dracut.conf.d
fprintd.conf
fuse.conf
host.conf
idmapd.conf
ipsec.conf
kdump.conf
krb5.conf
krb5.conf.d
ld.so.conf
ld.so.conf.d
libaudit.conf
locale.conf
logrotate.conf
makedumpfile.conf.sample
man_db.conf
mke2fs.conf
mtools.conf
ndctl.conf.d
nfs.conf
```

Рис. 2.3: Вывели имена всех файлов из file.txt, имеющих расширение .conf

Запишем названия все файлов из file.txt, имеющие расширение .conf в новый текстовый файл conf.txt.(рис. 2.4).

```
[ksyusha@ksyusha ~]$ grep "\.conf" file.txt > conf.txt
[ksyusha@ksyusha ~]$
```

Рис. 2.4: названия все файлов из file.txt, имеющие расширение .conf в новый текстовый файл conf.txt

Определим, какие файлы в нашем домашнем каталоге имеют имена, начинавшиеся с символа с. Первый вариант, как это сделать: (рис. 2.5).

```
[ksyusha@ksyusha ~]$ find ~ -name "с*" -print
```

Рис. 2.5: первый вариант нахождения файлов с символом с

Анализируем вывод (рис. 2.6).

```
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab08/report/pandoc/csl
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab08/report/pandoc/filters/pandocxnos/core.py
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab09/report/bib/cite.bib
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab09/report/pandoc/csl
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab09/report/pandoc/filters/pandocxnos/core.py
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab10/report/bib/cite.bib
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab10/report/pandoc/csl
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab10/report/pandoc/filters/pandocxnos/core.py
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab11/report/bib/cite.bib
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab11/report/pandoc/csl
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab11/report/pandoc/filters/pandocxnos/core.py
```

Рис. 2.6: анализируем вывод

Второй и третий варианты нахождения файлов с символом с (рис. 2.7).

```
[ksyusha@ksyusha ~]$ ls | grep "c+"
abc1
australia
bin
blog
conf.txt
Documents
Downloads
feathers
file.old
file.txt
git-extended
LICENSE
may
monthly
my_os
pass.txt
play
reports
ski.places
work
Видео
Документы
Загрузки
Изображения
Музыка
Общедоступные
Рабочий стол
Шаблоны
[ksyusha@ksyusha ~]$ ls -Ra | grep ^c
```

Рис. 2.7: Второй и третий варианты нахождения файлов с символом c

Выведем на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h.(рис. 2.8).

```
[ksyusha@ksyusha ~]$ find ~ -name "h*" -print | less
```

Рис. 2.8: выводим имена файлов из каталога /etc, начинающиеся с символа h

Смотрим на верность вывода(рис. 2.9).

```
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab08/report/pandoc/csl
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab08/report/pandoc/filters/pandocxnos/core.py
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab09/report/bib/cite.bib
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab09/report/pandoc/csl
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab09/report/pandoc/filters/pandocxnos/core.py
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab10/report/bib/cite.bib
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab10/report/pandoc/csl
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab10/report/pandoc/filters/pandocxnos/core.py
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab11/report/bib/cite.bib
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab11/report/pandoc/csl
/home/ksyusha/work/study/2024-2025/Операционные системы/os-intro/labs/lab11/report/pandoc/filters/pandocxnos/core.py
```

Рис. 2.9: Смотрим на верность вывода

Запустите в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log(рис. 2.10).

```
[ksyusha@ksyusha ~]$ find ~ -name "log" -print >> logfile &
[1] 2648
```

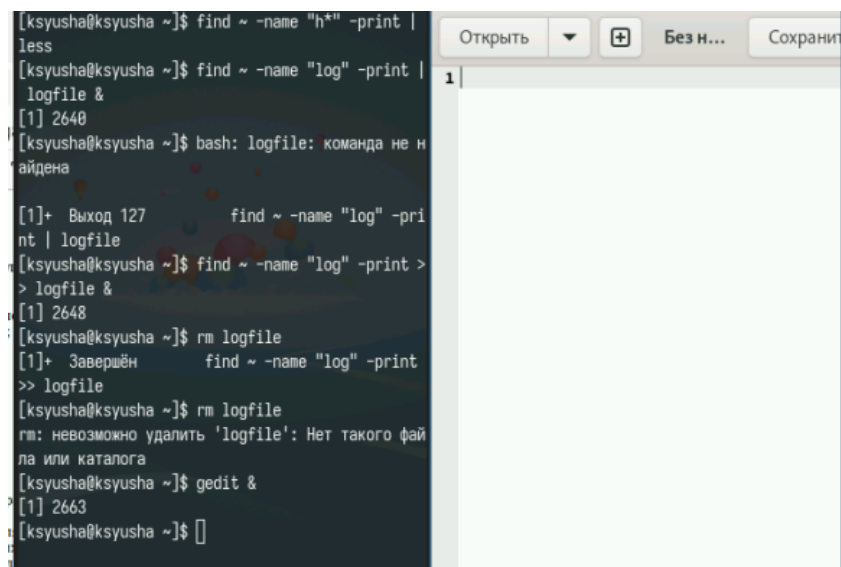
Рис. 2.10: процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log

Удаляем файл ~/logfile(рис. 2.11).

```
[ksyusha@ksyusha ~]$ rm logfile
[1]+  Завершён      find ~ -name "log" -print
>> logfile
```

Рис. 2.11: Удаляем файл ~/logfile

Запустим из консоли в фоновом режиме редактор gedit с помощью gedit & (рис. 2.12).



```
[ksyusha@ksyusha ~]$ find ~ -name "h*" -print | less
[ksyusha@ksyusha ~]$ find ~ -name "log" -print | logfile &
[1] 2648
[ksyusha@ksyusha ~]$ bash: logfile: команда не найдена
[1]+  Выход 127      find ~ -name "log" -print | logfile
[ksyusha@ksyusha ~]$ find ~ -name "log" -print > logfile &
[1] 2648
[ksyusha@ksyusha ~]$ rm logfile
[1]+  Завершён      find ~ -name "log" -print
>> logfile
[ksyusha@ksyusha ~]$ rm logfile
rm: невозможно удалить 'logfile': Нет такого файла или каталога
[ksyusha@ksyusha ~]$ gedit &
[1] 2663
[ksyusha@ksyusha ~]$
```

Рис. 2.12: Запустили из консоли в фоновом режиме редактор gedit

Определим идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep.(рис. 2.13).

```
[ksyusha@ksyusha ~]$ ps | grep gedit
2663 pts/1    00:00:03 gedit
```

Рис. 2.13: Определим идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep.

Убьём процесс по его pid(рис. 2.14).

```
[ksyusha@ksyusha ~]$ kill 2663
```

Рис. 2.14: Завершили процесс

С помощью df посмотрим на размер допустимого места в системе (рис. 2.15).

```
[ksyusha@ksyusha ~]$ df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтиро-
ано в
/dev/sda3          82834432    13752080  68055936         17% /
devtmpfs           4096          0      4096          0% /dev
tmpfs              4570376      3572    4566804         1% /dev/shm
tmpfs              1828152      1196    1826956         1% /run
tmpfs              1024          0      1024          0% /run/cred
entials/systemd-journald.service
tmpfs              1024          0      1024          0% /run/cred
entials/systemd-network-generator.service
tmpfs              1024          0      1024          0% /run/cred
entials/systemd-udev-load-credentials.service
tmpfs              1024          0      1024          0% /run/cred
entials/systemd-tmpfiles-setup-dev-early.service
tmpfs              1024          0      1024          0% /run/cred
entials/systemd-sysctl.service
tmpfs              1024          0      1024          0% /run/cred
entials/systemd-tmpfiles-setup-dev.service
tmpfs              4570380          4    4570376         1% /tmp
tmpfs              1024          0      1024          0% /run/cred
entials/systemd-vconsole-setup.service
/dev/sda3          82834432    13752080  68055936         17% /home
/dev/sda2          996780      328740    599228         36% /boot
tmpfs              1024          0      1024          0% /run/cred
entials/systemd-tmpfiles-setup.service
tmpfs              1024          0      1024          0% /run/cred
entials/systemd-resolved.service
tmpfs              914072       104     913968         1% /run/user
/1000
```

Рис. 2.15: используем df

И смотрим на место с помощью du(рис. 2.16).

```
[ksyusha@ksyusha ~]$ du
```

Рис. 2.16: используем du

Воспользовавшись справкой команды find, выведем имена всех директорий, имеющих в вашем домашнем каталоге.(рис. 2.17).

```
72  ./password-store/.git/objects
4   ./password-store/.git/logs/refs/heads
8   ./password-store/.git/logs/refs/remotes/origin
8   ./password-store/.git/logs/refs/remotes
12  ./password-store/.git/logs/refs
16  ./password-store/.git/logs
196 ./password-store/.git
208 ./password-store
56840 ./bin
32  ./bashrc.d
928 ./Downloads
0   ./Documents
0   ./monthly
0   ./reports/monthly/monthly
0   ./reports/monthly
0   ./reports
8   ./ski.places/equipment
0   ./ski.places/plans
```

Рис. 2.17: выведем имена всех директорий, имеющих в вашем домашнем каталоге

Получим следующий вывод(рис. 2.18).

```
[ksyusha@ksyusha ~]$ find ~ -name "*" -type d -print
```

Рис. 2.18: получаем такой вывод

3 Ответы на контрольные вопросы

1. В системе по умолчанию открыты три особых потока: `stdin` — это стандартный поток ввода (по умолчанию это клавиатура), его файловый дескриптор равен 0. `stdout` — это стандартный поток вывода (по умолчанию это консоль), его файловый дескриптор равен 1. `stderr` — это стандартный поток вывода сообщений об ошибках (по умолчанию это консоль), его файловый дескриптор равен 2.
2. Символ `>` используется для перенаправления ввода/вывода, а символ `>>` используется для перенаправления в режиме добавления.
3. Конвейер (`pipe`) используется для объединения отдельных команд или утилит в цепочку, в которой вывод одной команды передается на вход следующей команды.
4. Основное различие между программой и процессом заключается в том, что программа представляет собой набор инструкций, предназначенных для выполнения определенной задачи центральным процессором (ЦПУ), в то время как процесс - это экземпляр исполняемой программы, который активно выполняется в операционной системе.
5. PID (Process ID) - это идентификатор процесса, который уникально идентифицирует каждый запущенный процесс в операционной системе. GID (Group ID) - это идентификатор группы, который определяет принадлежность процесса к определенной группе пользователей в операционной системе.

6. Программы, запущенные в фоновом режиме, действительно называются задачами (jobs). Управлять ими можно с помощью команды jobs, которая выводит список запущенных в данный момент задач.
7. Команда htop и команда top выполняют аналогичные функции, показывая информацию о процессах в реальном времени и отображая данные о потреблении системных ресурсов. Обе команды также предоставляют возможность поиска, остановки и управления процессами. Однако у них есть различия и преимущества. Например, в htop реализован более удобный поиск и фильтрация процессов, что делает его использование более интуитивно понятным по сравнению с top, где для активации функции поиска требуется знать соответствующую комбинацию клавиш. С другой стороны, в top можно разделить область окна и настроить отображение информации о процессах согласно различным настройкам, что делает его более гибким в настройке отображения.
8. Команда find является одной из наиболее важных и часто используемых утилит в системе Linux. Она предназначена для поиска файлов и каталогов на основе определенных условий. find можно применять в различных сценариях, таких как поиск файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям. Утилита find по умолчанию предустановлена во всех дистрибутивах Linux, что обеспечивает готовность к использованию без необходимости установки дополнительных пакетов. Это делает find важным инструментом для эффективной работы в командной строке Linux. Синтаксис команды find следующий: find путь параметры критерий действие. Например: find /etc -name "p*" -print - это команда, которая ищет файлы, начинающиеся с символа "p" в каталоге /etc и выводит результаты поиска.
9. Да, можно использовать команду find в сочетании с grep для поиска текста в файлах. Пример использования: find / -type f -exec grep -H 'ТЕКСТ' {}; Эта

команда будет рекурсивно искать файлы в корневом каталоге / и его подкаталогах. Затем она передаст каждый найденный файл в качестве аргумента команде `grep`, которая выполнит поиск строки 'ТЕКСТ' в каждом файле. Результатом будут строки с соответствующим текстом и именами файлов, в которых он найден.

10. С помощью `df -h`
11. С помощью команды `du -s`
12. С помощью команды `kill PID`

4 Выводы

В ходе лабораторной работы мы ознакомились с инструментами поиска файлов и фильтрации текстовых данных, приобрели практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.