

Отчет по лабораторной работе № 8

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

Алиса Алексеевна Скобеева

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	11
5	Контрольные вопросы	12

Список иллюстраций

3.1	Выполнение команд и просмотр результата	7
3.2	Выполнение команды	7
3.3	Два способа просмотра	8
3.4	Просмотр результата команды	8
3.5	Начало и завершение процесса	8
3.6	Выполнение команд	9
3.7	Выполнение команды	9
3.8	Выполнение команды	9
3.9	Просмотр названий директорий	10

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 Задание

Последовательно выполнить упражнения, указанные на платформе ТУИС.

3 Выполнение лабораторной работы

Записываем в file.txt названия файлов из /etc и из домашнего каталога:

```
[aaskobeeva@fedora ~]$ ls /etc > file.txt
[aaskobeeva@fedora ~]$ ls ~ >> file.txt
[aaskobeeva@fedora ~]$ ls
abc1      feathers  may      README.md  Документы  Общедоступные
australia file.txt  monthly  reports    Загрузки   'Рабочий стол'
Documents git-extended my_os    work       Изображения  Шаблоны
Downloads LICENSE  play     Видео      Музыка
[aaskobeeva@fedora ~]$ cat file.txt
abrt
adjtime
aliases
alsa
alternatives
anaconda
anacrontab
asound.conf
```

Рис. 3.1: Выполнение команд и просмотр результата

Выводим имена файлов, имеющих расширение .conf и записываем их в conf.txt:

```
[aaskobeeva@fedora ~]$ grep '\.conf$' file.txt > conf.txt
[aaskobeeva@fedora ~]$ ls
abc1      Downloads  LICENSE  play      Видео      Музыка
australia feathers   may      README.md  Документы  Общедоступные
conf.txt  file.txt  monthly  reports    Загрузки   'Рабочий стол'
Documents git-extended my_os    work       Изображения  Шаблоны
[aaskobeeva@fedora ~]$ cat conf.txt
asound.conf
chrony.conf
dnsmasq.conf
dracut.conf
fprintd.conf
fuse.conf
```

Рис. 3.2: Выполнение команды

Определение файлов в домашнем каталоге, начинающихся с “с”:

```
[aaskobeeva@fedora ~]$ ls ~ | grep ^c
conf.txt
[aaskobeeva@fedora ~]$ find ~ -maxdepth 1 -name "c*"
/home/aaskobeeva/conf.txt
```

Рис. 3.3: Два способа просмотра

Выводим на экран название файлов из /etc начинающихся с “h”:

```
host.conf
hostname
hosts
~
~
~
~
```

Рис. 3.4: Просмотр результата команды

Запускаем и завершаем процесс, который будет записывать определенные названия в файл:

```
[aaskobeeva@fedora ~]$ find ~ -name "log*" > ~/logfile &
[1] 4019
[aaskobeeva@fedora ~]$ rm ~/logfile
[1]+  Завершён      find ~ -name "log*" > ~/logfile
```

Рис. 3.5: Начало и завершение процесса

Запускаем в фоновом режиме gedit. Узнаем идентификатор процесса и завершаем его с помощью kill:


```
[aaskobeeva@fedora ~]$ gedit &
[1] 4025
[aaskobeeva@fedora ~]$ ps aux | grep gedit
aaskobe+  4025  5.7  0.6 774608 59672 pts/0    Sl   22:22   0:03 gedit
aaskobe+  4046 22.2  0.0 230364  2412 pts/0    S+   22:23   0:00 grep --colo
r=auto gedit
[aaskobeeva@fedora ~]$ pgrep gedit
4025
[aaskobeeva@fedora ~]$ man kill
[aaskobeeva@fedora ~]$ kill
(gedit:4025): Gtk-WARNING **: 22:24:21.212: Calling org.freedesktop.portal.Inh
ibit.Inhibit failed: GDBus.Error:org.freedesktop.DBus.Error.UnknownMethod: Не
существует интерфейса "org.freedesktop.portal.Inhibit" на объекте по пути /org
/freedesktop/portal/desktop
kill: использование: kill [-s назв_сигнала | -n номер_сигнала | -назв_сигнала]
ид_процесса | назв_задания] ... или kill -l [назв_сигнала]
[aaskobeeva@fedora ~]$ kill 4025
```

Рис. 3.6: Выполнение команд

Выполняем команду df:

```
[aaskobeeva@fedora ~]$ man df
[1]+  Завершено      gedit
[aaskobeeva@fedora ~]$ df -h
Файловая система  Размер  Использовано  Дост  Использовано%  Смонтир
/dev/sda3          80G         11G       69G          14% /
devtmpfs           4,0M           0       4,0M           0% /dev
tmpfs              4,8G         3,4M       4,8G           1% /dev/sh
tmpfs              1,9G         1,2M       1,9G           1% /run
tmpfs              1,0M           0       1,0M           0% /run/cr
emd-journald.service
tmpfs              1,0M           0       1,0M           0% /run/cr
emd-udev-udev-load-credentials.service
```

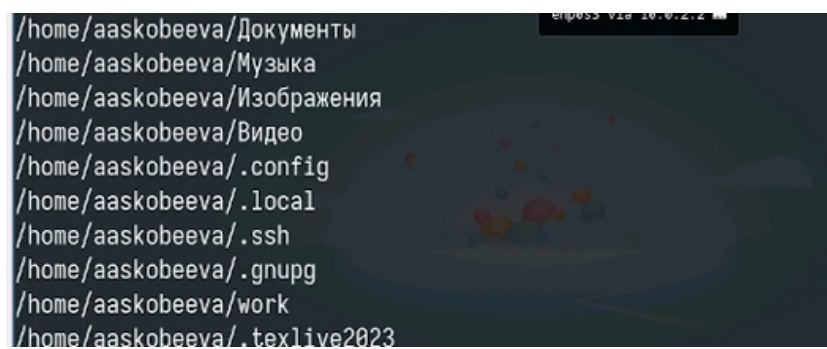
Рис. 3.7: Выполнение команды

Выполняем команду du:

```
[aaskobeeva@fedora ~]$ man du
[aaskobeeva@fedora ~]$ du -sh ~
840M    /home/aaskobeeva
```

Рис. 3.8: Выполнение команды

С помощью find выводим имена всех директорий, имеющихя в домашнем каталоге:

A terminal window with a dark background and light-colored text. The text lists various directories under the path /home/aaskobeeva/. The directories are: Документы, Музыка, Изображения, Видео, .config, .local, .ssh, .gnupg, work, and .texlive2023. The text is aligned to the left. In the top right corner, there is a small black box with white text that reads "eflps5 via 10.0.2.2".

```
/home/aaskobeeva/Документы
/home/aaskobeeva/Музыка
/home/aaskobeeva/Изображения
/home/aaskobeeva/Видео
/home/aaskobeeva/.config
/home/aaskobeeva/.local
/home/aaskobeeva/.ssh
/home/aaskobeeva/.gnupg
/home/aaskobeeva/work
/home/aaskobeeva/.texlive2023
```

Рис. 3.9: Просмотр названий директорий

4 Выводы

Мы успешно выполнили все задания лабораторной работы и приобрели навыки по управлению процессами, по проверке использования диска и обслуживанию файловых систем.

5 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?

В Linux существует три стандартных потока ввода-вывода:

- Стандартный ввод (stdin, file descriptor 0): Поток, из которого программа получает ввод. Обычно это клавиатура.
- Стандартный вывод (stdout, file descriptor 1): Поток, в который программа отправляет обычный вывод. Обычно это экран.
- Стандартный поток ошибок (stderr, file descriptor 2): Поток, в который программа отправляет сообщения об ошибках. Обычно это экран.

2. Объясните разницу между операцией > и ».

Обе операции используются для перенаправления стандартного вывода в файл, но они делают это по-разному:

- > (перезапись): Перенаправляет стандартный вывод в файл, перезаписывая его содержимое, если файл уже существует. Если файл не существует, он будет создан.

Пример: `ls -l > file.txt` (результат команды `ls -l` будет записан в файл `file.txt`, перезаписав его содержимое).

- » (добавление): Перенаправляет стандартный вывод в файл, добавляя его в конец файла, если он уже существует. Если файл не существует, он будет создан.

Пример: `ls -l >> file.txt` (результат команды `ls -l` будет добавлен в конец файла `file.txt`).

3. Что такое конвейер?

Конвейер (pipeline) - это последовательность команд, соединенных символом | (вертикальная черта). Вывод одной команды (stdout) становится вводом для следующей команды (stdin). Конвейеры позволяют создавать сложные операции, комбинируя простые команды.

Пример: `ls -l /etc | grep "conf"` (вывести список файлов в каталоге /etc и найти строки, содержащие "conf").

4. Что такое процесс? Чем это понятие отличается от программы?

• Программа: Это набор инструкций, хранящихся на диске в виде исполняемого файла. Это пассивный объект. • Процесс: Это экземпляр программы, выполняющийся в оперативной памяти. Это активный объект, имеющий свой PID, ресурсы (память, открытые файлы и т. д.) и состояние.

Простыми словами, программа - это рецепт, а процесс - это приготовленное блюдо по этому рецепту. Одна и та же программа может быть запущена несколько раз, создавая несколько независимых процессов.

5. Что такое PID и GID?

• PID (Process ID): Уникальный числовой идентификатор, присваиваемый каждому процессу в системе. Он используется для управления процессами (например, для завершения процесса). • GID (Group ID): Уникальный числовой идентификатор, присваиваемый каждой группе пользователей в системе. Он используется для определения прав доступа к файлам и каталогам для группы пользователей.

6. Что такое задачи и какая команда позволяет ими управлять?

В контексте командной строки Linux, "задачи" обычно относятся к процессам, запущенным в фоновом режиме. Команда `jobs` позволяет просматривать список задач, выполняющихся в текущей оболочке. Другие полезные команды:

• `&`: Запустить команду в фоновом режиме. * Пример: `sleep 60 &` (запустить команду `sleep 60` в фоновом режиме). • `fg`: Переместить задачу из фонового режима

в foreground (передний план). * Пример: fg %1 (переместить задачу с номером 1 в foreground). • bg: Переместить остановленную задачу в фоновый режим. * Пример: bg %1 (переместить задачу с номером 1 в background). • kill: Отправить сигнал процессу (обычно для завершения процесса). * Пример: kill %1 (завершить задачу с номером 1).

7. Найдите информацию об утилитах top и htop. Каковы их функции?

top и htop - это интерактивные утилиты для мониторинга процессов в Linux. Они отображают список процессов, отсортированных по использованию ресурсов (например, CPU, памяти).

- top: Стандартная утилита мониторинга, доступная в большинстве дистрибутивов Linux. Она показывает список процессов, использование ресурсов, время работы системы и другую полезную информацию.
- htop: Более продвинутая версия top. Она имеет более удобный интерфейс, поддерживает подсветку синтаксиса, позволяет управлять процессами (например, завершать их) с помощью клавиш, и показывает информацию о процессах в виде дерева. htop обычно не установлен по умолчанию и требует установки.

Обе утилиты полезны для выявления процессов, потребляющих много ресурсов, и для мониторинга общей производительности системы.

8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.

Команда для поиска файлов в Linux - find. Она позволяет искать файлы по имени, типу, размеру, времени изменения, правам доступа и другим критериям.

Примеры:

- find . -name "file.txt": Найти файл с именем "file.txt" в текущем каталоге и его подкаталогах.
- find / -name "file.txt": Найти файл с именем "file.txt" во всей файловой системе (начиная с корневого каталога).
- find . -type f -name ".txt": *Найти все файлы с расширением ".txt" в текущем каталоге и его подкаталогах.*
- find . -type d

-name "dir": Найти все каталоги с именем "dir" в текущем каталоге и его подкаталогах.
• *find . -size +10M: Найти все файлы размером больше 10 МБ в текущем каталоге и его подкаталогах.* • *find . -mtime -7: Найти все файлы, измененные за последние 7 дней.* *find . -user username: Найти все файлы, принадлежащие пользователю username.* *
find / -perm 777: Найти все файлы с правами 777.

9. Можно ли по контексту (содержанию) найти файл? Если да, то как?

Да, для поиска файлов по содержимому используется команда `grep`. Она ищет строки, соответствующие заданному шаблону, в указанных файлах.

Пример:

- `grep "keyword" file.txt`: Найти все строки, содержащие "keyword", в файле `file.txt`.
- `grep -r "keyword" .`: Найти все файлы в текущем каталоге и его подкаталогах, содержащие "keyword". Опция `-r` означает рекурсивный поиск.
- `grep -l "keyword" /*`: Найти только имена файлов, которые содержат "keyword".
- `grep -i "keyword" file.txt`: Искать "keyword" без учета регистра.

Для более сложных поисков можно использовать регулярные выражения с `grep` -E или `egrep`.

10. Как определить объем свободной памяти на жёстком диске?

Используйте команду `df` (disk free).

- `df -h`: Отображает информацию о свободном и занятом месте на дисках в удобочитаемом формате (например, КБ, МБ, ГБ).
- `df -Th`: То же самое, что и `df -h`, но добавляет информацию о типе файловой системы.

11. Как определить объем вашего домашнего каталога?

Используйте команду `du` (disk usage).

- `du -sh ~`: Отображает общий размер домашнего каталога в удобочитаемом формате.
- `du -sh /home/username`: Отображает общий размер домашнего каталога пользователя `username`.

12. Как удалить зависший процесс?
13. Найдите PID зависшего процесса с помощью команд `top`, `htop` или `ps aux`.
14. Попробуйте отправить процессу сигнал `SIGTERM` (15), который просит процесс завершиться корректно: `kill PID`.
15. Если процесс не завершается после `SIGTERM`, отправьте сигнал `SIGKILL` (9), который принудительно завершает процесс: `kill -9 PID`.

Пример: `top` # Найти PID зависшего процесса (например, 1234) `kill 1234` # Попробуйте завершить процесс корректно `kill -9 1234` # Принудительное завершение процесса, если предыдущая команда не сработала