

Отчет о выполнении лабораторной работы

Лабораторная работа №8

Филиппьева Ксения Дмитриевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	19
6	Ответы на контрольные вопросы	20

Список иллюстраций

4.1	ls >	8
4.2	ls »	8
4.3	grep .conf	9
4.4	grep >	10
4.5	find c	10
4.6	ls grep	11
4.7	find h	11
4.8	find log	12
4.9	rm logfile	12
4.10	gedit &	12
4.11	ps(kill)	13
4.12	man kill	13
4.13	man df	14
4.14	man du	15
4.15	df result	16
4.16	du result	17
4.17	find -d	18

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

Приобрести и отработать уже имеющиеся навыки по управлению процессами и заданиями, а также проверке использования диска и обслуживанию файловых систем.

3 Теоретическое введение

Инструменты поиска файлов, такие как команда `find`, позволяют эффективно находить нужные файлы в файловой системе по различным критериям - по имени, типу, размеру, времени создания и т.д.

Фильтрация текстовых данных осуществляется с помощью команд `grep`, `sed`, `awk`. Они позволяют находить в текстовых файлах нужную информацию по шаблонам, заменять и обрабатывать текст.

Для управления процессами используются команды `ps`, `top`, `kill`. С их помощью можно просматривать список запущенных процессов, отслеживать использование ресурсов, завершать зависшие процессы. Команды `jobs`, `fg`, `bg` предназначены для управления заданиями в текущей сессии терминала.

Для мониторинга использования дискового пространства и обслуживания файловых систем применяются команды `df`, `du`, `fsck`. Они позволяют проверить объем свободного места на разделах, размер каталогов, а также выполнить проверку и восстановление целостности файловой системы.

4 Выполнение лабораторной работы

Запишем названия файлов, хранящихся в каталоге /etc в файл file.txt (рис. 4.1).

```
foot
[kdfilipjeva@kdfilipjeva ~]$ ls /etc > file.txt
[kdfilipjeva@kdfilipjeva ~]$ cat file.txt
abrt
adjtime
aliases
alsa
alternatives
anaconda
anacrontab
appstream.conf
```

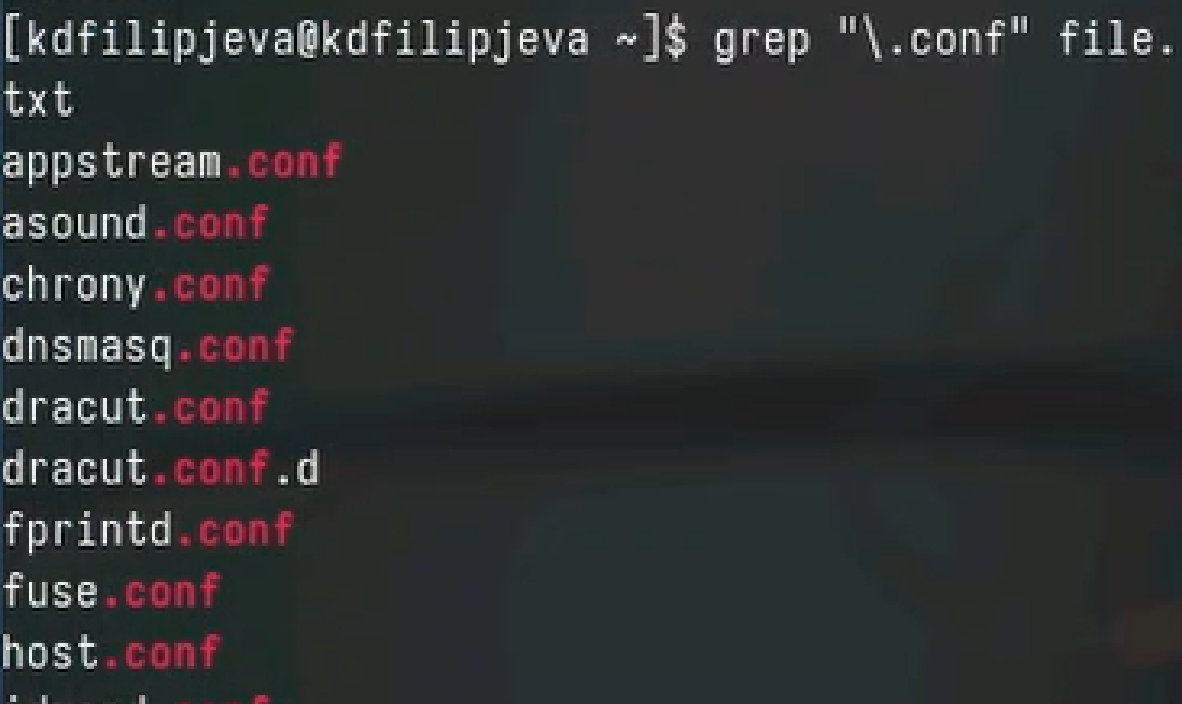
Рис. 4.1: ls >

Также запишем в этот файл названия файлов находящихся в нашем домашнем каталоге (рис. 4.2).

```
xml
yum.repos.d
[kdfilipjeva@kdfilipjeva ~]$ ls >> file.txt
```

Рис. 4.2: ls »

Найдем все названия файлов в нашем файле, которые оканчиваются на .conf (рис. 4.3).

A terminal window with a dark background. The prompt is [kdfilipjeva@kdfilipjeva ~]\$. The command entered is grep "\.conf" file.txt. The output lists several files with their .conf extensions in red text: appstream.conf, asound.conf, chrony.conf, dnsmasq.conf, dracut.conf, dracut.conf.d, fprintd.conf, fuse.conf, host.conf, and systemd.conf.

```
[kdfilipjeva@kdfilipjeva ~]$ grep "\.conf" file.txt
appstream.conf
asound.conf
chrony.conf
dnsmasq.conf
dracut.conf
dracut.conf.d
fprintd.conf
fuse.conf
host.conf
systemd.conf
```

Рис. 4.3: grep .conf

Переместим найденные названия в новый файл conf.txt (рис. 4.4).

```
[kdfilipjeva@kdfilipjeva ~]$ grep "\.conf" file.txt > conf.txt
[kdfilipjeva@kdfilipjeva ~]$ cat conf.txt
appstream.conf
asound.conf
chrony.conf
dnsmasq.conf
dracut.conf
dracut.conf.d
fprintd.conf
fuse.conf
```

Рис. 4.4: grep >

Найдем и выведем в консоль все файлы из домашнего каталога, которые начинаются с английской с (рис. 4.5).

```
xvcc.conf
[kdfilipjeva@kdfilipjeva ~]$ find ~ -name "c*" -
print
```

Рис. 4.5: find c

Сделаем то же самое, только другим способом (рис. 4.6).

```
[kdfilipjeva@kdfilipjeva ~]# ls -R | grep c*
conf.txt
[kdfilipjeva@kdfilipjeva ~]$ ls -R | grep ^c
conf.txt
chezmoi
config
content
chris-montgomery-smgTvepind4-unsplash.jpg
conference-paper
```

Рис. 4.6: ls grep

Выведем в консоль все файлы из каталога /etc , которые начинаются на h(рис. 4.7).

```
foot
/etc/avahi/hosts
/etc/libibverbs.d/hfi1verbs.driver
/etc/libibverbs.d/hns.driver
/etc/systemd/homed.conf
/etc/udev/hwdb.d
/etc/udev/hwdb.bin
/etc/host.conf
/etc/hosts
/etc/hostname
/etc/mercurial/hgrc.d
```

Рис. 4.7: find h

Найдем все файлы в домашнем каталоге которые начинаются на log и переместим их в файл logfile в фоновом режиме (рис. 4.8).

```

[kdfilipjeva@kdfilipjeva ~]$ find ~ -name "log*"
-print >> logfile &
[1] 5026
[kdfilipjeva@kdfilipjeva ~]$ cat logfile
/home/kdfilipjeva/.mozilla/firefox/pybt7zrm.default-release/logins-backup.json
/home/kdfilipjeva/.mozilla/firefox/pybt7zrm.default-release/logins.json

```

Рис. 4.8: find log

Удалим получившийся файл logfile (рис. 4.9).

```

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

```

Рис. 4.9: rm logfile

Запустим gedit в фоновом режиме и узнаем его идентификатор (рис. 4.10).

```

[kdfilipjeva@kdfilipjeva ~]$ gedit &
[1] 5048
[kdfilipjeva@kdfilipjeva ~]$ █

```

Рис. 4.10: gedit &

Поскольку при вводе команды ps у нас завершился фоновый процесс с gedit'ом, то команду kill использовать не имеет смысла (рис. 4.11).

```
[kdfilipjeva@kdfilipjeva ~]$ ps | grep gedit
[1]+  Завершён          gedit
[kdfilipjeva@kdfilipjeva ~]$ m
```

Рис. 4.11: ps(kill)

Описание команды kill (рис. 4.12).

```
KILL(1)                                User Commands                                KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q
value] [-a] [--timeout milliseconds
signal] [--] pid|name...

    kill -l [number] | -L
```

Рис. 4.12: man kill

Описание команды df(рис. 4.13).

```
DE(1)                                User Commands                                DE(1)

NAME
    df - report file system space usage

SYNOPSIS
    df [OPTION]... [FILE]...

DESCRIPTION
    This manual page documents the GNU version of df. df displays the amount of space available on the file system containing each file name argument. If no file name is given, the space available on all currently mounted file systems is shown. Space is shown in 1K blocks by default, unless the environment variable POSIXLY_CORRECT is set, in which case 512-byte blocks are used.
```

Рис. 4.13: man df

Описание команды du (рис. 4.14).

NAME

du - estimate file space usage

SYNOPSIS

du [OPTION]... [FILE]...

du [OPTION]... --files0-from=F

DESCRIPTION

Summarize device usage of the set of FILES, recursively for directories.

Mandatory arguments to long options are mandatory for short options too.

Рис. 4.14: man du

Результат выполнения команды df (рис. 4.15).

```
[kdfilipjeva@kdfilipjeva ~]$ df
Файловая система 1К-блоков Использовано Доступно
Использовано% Смонтировано в
/dev/sda3          82834432      10572084 71214380
          13% /
devtmpfs           4096          0      4096
          0% /dev
tmpfs              4062512          3724 4058788
          1% /dev/shm
tmpfs              1625008          1216 1623792
          1% /run
/dev/sda3          82834432      10572084 71214380
          13% /home
tmpfs              4062512          32 4062480
          1% /tmp
/dev/sda2          996780          272920 655048
          30% /boot
tmpfs              812500          104 812396
          1% /run/user/1000
```

Рис. 4.15: df result

Результат выполнения команды du (рис. 4.16).


```
176      ./password-store
35184    ./bin
32       ./bashrc.d
48       ./Downloads/academic-presentation-markdo
wn-template-master/presentation/image
1028     ./Downloads/academic-presentation-markdo
wn-template-master/presentation
1072     ./Downloads/academic-presentation-markdo
wn-template-master
1568     ./Downloads
0        ./monthly
0        ./reports/monthly/monthly
0        ./reports/monthly
0        ./reports
8        ./ski.plases/equipment
0        ./ski.plases/plans
8        ./ski.plases
0        ./australia
0        ./play/games
0        ./play
933044  .
```

Рис. 4.16: du result

Используя команду `find` выведем все дирректории в нашем домашнем каталоге(рис. 4.17).

```
[kdfilipjeva@kdfilipjeva ~]$ find ~ -name "*" -t  
ype d -print
```

Рис. 4.17: find -d

5 Выводы

Мы получили новые и отработали уже имеющиеся навыки работы с инструментами поиска файлов.

6 Ответы на контрольные вопросы

1. Основные потоки ввода-вывода в Linux:

- `stdin` (стандартный ввод)
- `stdout` (стандартный вывод)
- `stderr` (стандартный поток ошибок)

2. Операторы `>` и `>>`:

- `>` перенаправляет вывод команды в файл, перезаписывая его содержимое
- `>>` перенаправляет вывод команды в файл, добавляя его в конец файла

3. Конвейер (`pipe`) позволяет перенаправить вывод одной команды на ввод другой. Обозначается символом `|`. Например: `cat file.txt | grep "hello"`

4. Процесс - это экземпляр выполняющейся программы. Программа - это набор инструкций, а процесс - это работающая программа, имеющая выделенные ресурсы (память, файлы и т.д.)

5. PID (Process ID) - уникальный номер процесса. GID (Group ID) - идентификатор группы, к которой принадлежит процесс.

6. Задачи (`jobs`) - это процессы, запущенные в текущей сессии терминала. Ими можно управлять с помощью команд `jobs`, `fg`, `bg`.

7. `top` и `htop` - это утилиты для мониторинга запущенных процессов и использования ресурсов системы в реальном времени. `htop` имеет более удобный текстовый интерфейс.

8. Команда `find` используется для поиска файлов. Примеры:
- `find /home -name "*.txt"` - найти все файлы с расширением `.txt` в каталоге `/home`
 - `find . -type d` - найти все подкаталоги в текущем каталоге
9. Да, для поиска по содержимому файлов можно использовать команду `grep`.
Например:
`grep -r "hello world" /home` - найти все вхождения фразы `"hello world"` в файлах каталога `/home`
10. Для определения свободного места на диске используется команда `df -h`
11. Размер домашнего каталога можно узнать командой `du -sh ~`
12. Зависший процесс можно завершить командой `kill`. Например: `kill -9 1234`
- принудительно завершить процесс с PID 1234.