

Отчет по выполнению лабораторной работы №8

Операционные системы

Шихалиева Зурият Арсеновна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Выводы	17
5	Ответы на контрольные вопросы	18

Список иллюстраций

3.1	Утилита df	14
-----	----------------------	----

Список таблиц

1 Цель работы

Цель данной лабораторной работы - ознакомление с инструментами поиска файлов и фильтрации текстовых данных, а также приобретение практических навыков по управлению процессами (и заданиями), по проверке использования диска и по обслуживанию файловых систем.

2 Задание

1. Осуществите вход в систему, используя соответствующее имя пользователя.
2. Запишите в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.
3. Выведите имена всех файлов из `file.txt`, имеющих расширение `.conf`, после чего запишите их в новый текстовый файл `conf.txt`.
4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа `s`? Предложите несколько вариантов, как это сделать.
5. Выведите на экран (по странично) имена файлов из каталога `/etc`, начинающиеся с символа `h`.
6. Запустите в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`.
7. Удалите файл `~/logfile`.
8. Запустите из консоли в фоновом режиме редактор `gedit`.
9. Определите идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`. Как ещё можно определить идентификатор процесса?
10. Прочтите справку (`man`) команды `kill`, после чего используйте её для завершения процесса `gedit`.
11. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.
12. Воспользовавшись справкой команды `find`, выведите имена всех директо-

рий, имеющихся в вашем домашнем каталоге.

3 Выполнение лабораторной работы

Я вошла в систему под соответствующим именем пользователя, открыла терминал, записала в файл file.txt названия файлов из каталога /etc с помощью перенаправления “>” (и файл создала, и записала в него то, что могло быть выведено ls -lR /etc). В файл я добавила также все файлы из подкаталогов (рис. 1).

```
zashikhalieva@zashikhalieva:~$ sudo ls -lR /etc > file.txt
```

{#fig:001

width=70% рис. 1}

Проверила, что в файл записались нужные значения с помощью утилиты head, она выводит первые 10 строк файла на экран (рис. 2).

```
zashikhalieva@zashikhalieva:~$ head file.txt
/etc:
итого 1456
drwxr-xr-x. 1 root          root          126
  апр 15 01:59 abrt
-rw-r--r--. 1 root          root           16
  авг 24 09:25 adjtime
-rw-r--r--. 1 root          root        1529
  ноя 29  2023 aliases
drwxr-xr-x. 1 root          root           70
  июн 11 03:00 alsa
drwxr-xr-x. 1 root          root        1522
```

{#fig:002

width=70% рис. 2}

Добавила в созданный файл имена файлов из домашнего каталога, используя перенаправление “>” в режиме добавления (рис. 3).

```
zashikhalieva@zashikhalieva:~$ ls -lR ~/ >> file.txt
```

{#fig:003

width=70% рис. 3}

Вывела на экран имена всех файлов, имеющих расширение “.conf” с помощью утилиты grep (рис. 4).

```
zashikhalieva@zashikhalieva:~$ grep .conf file.txt
-rw-r--r--. 1 root          root          269
  мая  2 03:00 anthy-unicode.conf
-rw-r--r--. 1 root          root           55
  июн 10 03:00 asound.conf
-rw-r--r--. 1 root          root        30583
  янв 31  2024 brltty.conf
```

{#fig:004

width=70% рис. 4}

Добавила вывод прошлой команды в новый файл conf.txt с помощью перенаправления “>” (файл создается при выполнении этой команды) (рис. 5).

```
zashikhalieva@zashikhalieva:~$ 
zashikhalieva@zashikhalieva:~$ grep .conf file.txt > conf.txt
zashikhalieva@zashikhalieva:~$ head conf.txt
-rw-r--r--. 1 root          root          269
  мая  2 03:00 anthy-unicode.conf
-rw-r--r--. 1 root          root           55
  июн 10 03:00 asound.conf
-rw-r--r--. 1 root          root        30583
  янв 31  2024 brltty.conf
drwxr-xr-x. 1 root          root           0
  мая 14 03:00 chkconfig.d
-rw-r--r--. 1 root          root        1372
  дек  5  2023 chrony.conf
drwxr-xr-x. 1 root          root           18
  апр 15 01:57 dconf
-rw-r--r--. 1 root          root        1174
  мая  9 03:00 dleyna-server-service.conf
-rw-r--r--. 1 root          root       28602
  фев 13  2024 dnsmasq.conf
-rw-r--r--. 1 root          root          117
  июл 13 03:00 dracut.conf
drwxr-xr-x. 1 root          root           0
  июл 13 03:00 dracut.conf.d
```

{#fig:005

width=70% рис. 5}

Определяю, какие файлы в домашнем каталоге начинаются с символа “с” с

помощью утилиты `find`, прописываю ей в аргументах домашнюю директорию (тогда вывод относительно корневого каталога, а не домашнего будет), выбираю опцию `-name` (ищем по имени), и пишу маску, по которой будем искать имя, где `*` - любое количество любых символов, добавляю опцию `-print`, чтобы мне вывелся результат (рис. 6). Но таким образом я получаю информацию даже о файлах из подкаталогов домашнего каталога.

```
zashikhalieva@zashikhalieva:~$ find ~ -name "c*"
/home/zashikhalieva/.mozilla/firefox/2jkwvxuv.default-release/
crashes
/home/zashikhalieva/.mozilla/firefox/2jkwvxuv.default-release/
compatibility.ini
/home/zashikhalieva/.mozilla/firefox/2jkwvxuv.default-release/
cookies.sqlite
/home/zashikhalieva/.mozilla/firefox/2jkwvxuv.default-release/
cert9.db
/home/zashikhalieva/.mozilla/firefox/2jkwvxuv.default-release/
storage/permanent/chrome
/home/zashikhalieva/.mozilla/firefox/2jkwvxuv.default-release/
storage/default/https+++web.telegram.org/cache
/home/zashikhalieva/.mozilla/firefox/2jkwvxuv.default-release/
storage/default/https+++web.telegram.org/cache/caches.sqlite
/home/zashikhalieva/.mozilla/firefox/2jkwvxuv.default-release/
storage/default/https+++web.telegram.org/cache/caches.sqlite-w
al
```

{#fig:006

width=70% рис. 6}

Второй способ использовать утилиту `ls -lR` и использовать `grep`, чтобы найти элементы с первым символом `c`. Однако этот способ не работает для поиска файлов из подкаталогов каталога (рис. 7).

```
zashikhalieva@zashikhalieva:~$ ls -lr | grep c*
-rw-r--r--. 1 zashikhalieva zashikhalieva 55898 сен  3 14:55
conf.txt
```

{#fig:007

width=70% рис. 7}

С помощью метода `find`, чьи опции я расписала ранее, ищу все файлы, начинающиеся с буквы `"h"` (рис. 8).

```

zashikhalieva@zashikhalieva:~$ sudo find /etc -name "h*" -prin
t
[sudo] пароль для zashikhalieva:
/etc/avahi/hosts
/etc/brltty/Contraction/ha.ctb
/etc/brltty/Input/bm/horizontal.kti
/etc/brltty/Input/hd
/etc/brltty/Input/hm
/etc/brltty/Input/ht

```

{#fig:008

width=70% рис. 8}

Запускаю в фоновом режиме (на это указывает символ &) процесс, который будет записывать в файл logfile (с помощью перенаправления >) файлы, имена которых начинаются с log (рис. 9).

```

zashikhalieva@zashikhalieva:~$ find ~ -name "log*" -print > lo
gfile &
[14] 181686
[13]   Завершён          find ~ -name "log*" -print > logfile

```

{#fig:009

width=70% рис. 9}

Проверяю, что файл создан, удаляю его, проверяю, что файл удален (рис. 10).

```

zashikhalieva@zashikhalieva:~$ ls
abcl          logfile      reports      Изображения
australia     logfile     repository   Музыка
conf.txt      may         ski.plases   Общедоступные
feathers      monthly    work         'Рабочий стол'
file.txt      my_os      Видео        Шаблоны
fun           play       Документы
git-extended  repo       Загрузки
[14]   Завершён          find ~ -name "log*" -print > logfile
zashikhalieva@zashikhalieva:~$ rm logfile

```

{#fig:010

width=70% рис. 11}

Запускаю в консоли в фоновом режиме (с помощью символа &) редактор gedit(рис. 11).

```
zashikhalieva@zashikhalieva:~$ gedit &  
[13] 182055
```

{#fig:011 width=70%

рис. 11}

С помощью утилиты ps определяю идентификатор процесса mousepad, его значение 3913 (рис. 12). Также мы можем определить идентификатор с помощью pgrep.

```
zashikhalieva@zashikhalieva:~$ ps aux | grep gedit  
zashikh+ 181895  0.0  0.1 528020  6168 pts/0    Tl   15:09  
0:00 /usr/libexec/pk-command-not-found gedit  
zashikh+ 181936  0.0  0.1 601752  6480 pts/0    Tl   15:09  
0:00 /usr/libexec/pk-command-not-found gedit  
zashikh+ 182307  3.6  1.7 857968 71316 pts/0    Sl   15:15  
0:00 gedit  
zashikh+ 182326  0.0  0.0 227812  2428 pts/0    S+   15:16  
0:00 grep --color=auto gedit  
zashikhalieva@zashikhalieva:~$ pgrep gedit  
182307
```

{#fig:012

width=70% рис. 12}

Прочитала справку команды kill (рис. 13).

NAME

kill - terminate a process

SYNOPSIS

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

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

DESCRIPTION

The command **kill** sends the specified signal to the specified processes or process groups.

If no signal is specified, the **TERM** signal is sent. The default action for this signal is to terminate the process. This signal should be used in preference to the **KILL** signal (number 9), since a process may install a handler for the TERM signal in order to perform clean-up steps before terminating in an orderly fashion. If a process does not terminate after a **TERM** signal has been sent, then the **KILL** signal may be used; be aware that the latter signal cannot be caught, and so does not give the target process the opportunity to perform any clean-up before terminating.

{#fig:013

width=70% рис. 13}

Использую команду **kill** и идентификатор процесса, чтобы его удалить (рис. 14). Заметила, как у меня закрылась программа gedit.

```
zashikhalieva@zashikhalieva:~$ man kill  
zashikhalieva@zashikhalieva:~$ kill 182307
```

{#fig:014

width=70% рис. 14}

Прочитала документацию про функции **df** и **du** (рис. 15).

```
zashikhalieva@zashikhalieva:~$ man df  
[14]   Завершено      gedit  
zashikhalieva@zashikhalieva:~$ man du
```

{#fig:015 width=70%

рис. 15}

Используя утилиту `df` опции `-iv` позволяют увидеть информацию об инодах и сделать вывод читаемым, игнорируя сообщения системы о нем (рис. fig. 3.1). Эта утилита нам нужна, чтобы выяснить, сколько свободного места есть у нашей системы.

```
zashikhalieva@zashikhalieva:~$ df -iv
Файловая система  Инодов  IИспользовано  ISвободно  IИспользовано
% Смонтировано в
/dev/sda3          0          0          0
- /
devtmpfs           495395        534      494861         1
% /dev
tmpfs              500210         2      500208         1
% /dev/shm
tmpfs              819200       1024      818176         1
% /run
tmpfs             1048576       132     1048444         1
% /tmp
/dev/sda3          0          0          0
- /home
/dev/sda2          65536        395      65141         1
% /boot
tmpfs             100042        216      99826         1
% /run/user/1000
/dev/sr0           0          0          0
- /run/media/zashikhalieva/VBox_GAs_7.0.14
```

Рис. 3.1: Утилита `df`

Используя утилиту `du`. Она нужна чтобы просмотреть, сколько места занимают файлы в определенной директории и найти самые большие из них (рис. 17).

```

zashikhalieva@zashikhalieva:~$ du ~/work
0      /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/branches
64     /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/hooks
4      /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/info
28     /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/pack
0      /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/info
32     /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/8c
20     /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/f9
24     /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/e2
8      /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/19
88     /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/65
4      /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/02
28     /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/1e
40     /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/52
272    /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/59
36     /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/9b
12     /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/93
8      /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/d8
20     /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/fb
8      /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/aa
12     /home/zashikhalieva/work/study/2024-2025/Операционные системы/os-intro/.git/objects/8f

```

{#fig:017

width=70% рис. 17}

Прочитала документацию о команде find (рис. 18).

```

FIND(1)                                General Commands Manual      FIND(1)

NAME

    find - search for files in a directory hierarchy

SYNOPSIS

    find [-H] [-L] [-P] [-D debugopts] [-Olevel] [start-
    ing-point...] [expression]

DESCRIPTION

    This manual page documents the GNU version of find.
    GNU find searches the directory tree rooted at each
    given starting-point by evaluating the given expres-
    sion from left to right, according to the rules of
    precedence (see section OPERATORS), until the outcome
    is known (the left hand side is false for and opera-
    tions, true for or), at which point find moves on to
    the next file name. If no starting-point is speci-
    fied, `.` is assumed.

```

{#fig:018

width=70% рис. 18}

Вывела имена всех директорий, имеющихся в моем домашнем каталоге, используя аргумент d у утилиты find опции -type, то есть указываю тип файлов, который мне нужен и этот тип Директория (рис. 19). Утилита -a позволит увидеть размер всех файлов, а не только директорий.

```
./.texlive2023/texmf-var
./.texlive2023/texmf-var/fonts
./.texlive2023/texmf-var/fonts/tfm
./.texlive2023/texmf-var/fonts/tfm/unknown
./.texlive2023/texmf-var/fonts/tfm/unknown/unknown
./monthly
./reports
./reports/monthly
./reports/monthly/monthly
./ski.places
./ski.places/equipment
./ski.places/plans
./australia
./play
./play/games
./fun
zashikhalieva@zashikhalieva:~$
zashikhalieva@zashikhalieva:~$ find -type d
```

{#fig:019

width=70% рис. 19}

4 Выводы

В результате данной лабораторной работы я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных, а также приобрела практические навыки по управлению процессами (и заданиями), по проверке использования диска и по обслуживанию файловых систем.

5 Ответы на контрольные вопросы

1. В системе по умолчанию открыто три специальных потока: – `stdin` — стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0; – `stdout` — стандартный поток вывода (по умолчанию: консоль), файловый дескриптор 1; – `stderr` — стандартный поток вывод сообщений об ошибках (по умолчанию: консоль), файловый дескриптор 2.
2. Этот знак `>` - перенаправление ввода/вывода, а `>>` - перенаправление в режиме добавления.
3. Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.
4. Главное отличие между программой и процессом заключается в том, что программа - это набор инструкций, который позволяет ЦПУ выполнять определенную задачу, в то время как процесс - это исполняемая программа.
5. PPID - (`parent process ID`) идентификатор родительского процесса. Процесс может порождать и другие процессы. UID, GID - реальные идентификаторы пользователя и его группы, запустившего данный процесс.
6. Запущенные фоном программы называются задачами (`jobs`). Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач.

7. Команда `htop` похожа на команду `top` по выполняемой функции: они обе показывают информацию о процессах в реальном времени, выводят данные о потреблении системных ресурсов и позволяют искать, останавливать и управлять процессами. У обеих команд есть свои преимущества. Например, в программе `htop` реализован очень удобный поиск по процессам, а также их фильтрация. В команде `top` это не так удобно — нужно знать кнопку для вывода функции поиска. Зато в `top` можно разделять область окна и выводить информацию о процессах в соответствии с разными настройками. В целом `top` намного более гибкая в настройке отображения процессов.
8. Команда `find` - это одна из наиболее важных и часто используемых утилит системы Linux. Это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям. Утилита `find` предустановлена по умолчанию во всех Linux дистрибутивах, поэтому вам не нужно будет устанавливать никаких дополнительных пакетов. Это очень важная находка для тех, кто хочет использовать командную строку наиболее эффективно.

Команда `find` имеет такой синтаксис: `find [папка] [параметры] критерий шаблон [действие]` Пример: `find /etc -name "p*" -print`

9. Можно. Через следующую команду: `find / -type f -exec grep -H 'текстДляПоиска' {} ;`
10. Это можно определить с помощью команды `df -h`.
11. Это можно определить с помощью команды `du -s`.
12. Это можно сделать с помощью команды `kill% номер задачи`.