

Отчёт по лабораторной работе №8

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

Хохлов Дмитрий Сергеевич

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	12
4	Контрольные вопросы	13

List of Figures

2.1	Запись в файл	5
2.2	Поиск расширения .conf	6
2.3	Поиск файлов	7
2.4	Поиск файлов	7
2.5	Фоновый запуск процесса	8
2.6	Фоновый запуск и завершение процесса	8
2.7	Справка по команде df	9
2.8	Запуск команды df	9
2.9	Справка по команде du	10
2.10	Запуск команды du	10
2.11	Поиск директорий	11

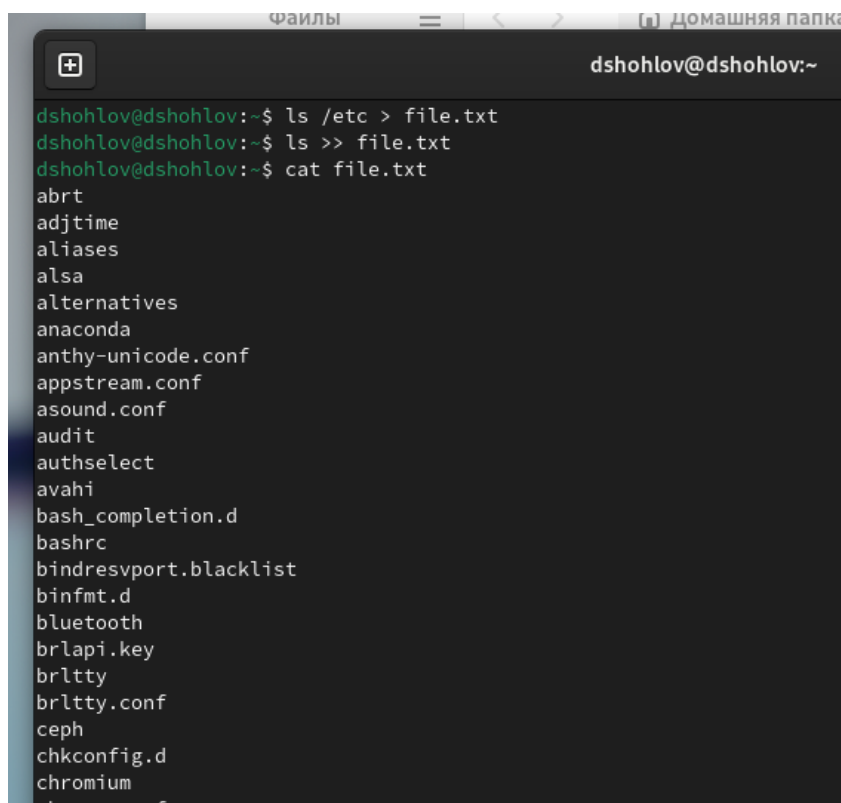
1 Цель работы

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

2 Выполнение лабораторной работы

1 Включаем компьютер, и заходим в учетную запись.

2 Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишем в этот же файл названия файлов, содержащихся в нашем домашнем каталоге.



```
dshohlov@dshohlov:~$ ls /etc > file.txt
dshohlov@dshohlov:~$ ls >> file.txt
dshohlov@dshohlov:~$ cat file.txt
abrt
adjtime
aliases
alsa
alternatives
anaconda
anthy-unicode.conf
appstream.conf
asound.conf
audit
authselect
avahi
bash_completion.d
bashrc
bindresvport.blacklist
binfmt.d
bluetooth
brlapi.key
brltty
brltty.conf
ceph
chkconfig.d
chromium
chromium.conf
```

Figure 2.1: Запись в файл

3 Выведем имена всех файлов из file.txt, имеющих расширение .conf, после чего запишем их в новый текстовый файл conf.txt.

```
dshohlov@dshohlov:~$  
dshohlov@dshohlov:~$ grep .conf file.txt > conf.txt  
dshohlov@dshohlov:~$ cat conf.txt  
anthy-unicode.conf  
appstream.conf  
asound.conf  
brltty.conf  
chkconfig.d  
chrony.conf  
dconf  
dleyna-server-service.conf  
dnsmasq.conf  
dracut.conf  
dracut.conf.d  
fprintd.conf  
fuse.conf  
host.conf  
idmapd.conf  
kdump.conf  
krb5.conf  
krb5.conf.d  
ld.so.conf  
ld.so.conf.d  
libaudit.conf  
libuser.conf  
locale.conf  
logrotate.conf  
makedumpfile.conf.sample  
man_db.conf  
mke2fs.conf  
mtools.conf  
ndctl.conf.d  
netconfig  
nfs.conf
```

Figure 2.2: Поиск расширения .conf

4 Определили, какие файлы в нашем домашнем каталоге имеют имена, начинающиеся с символа с?

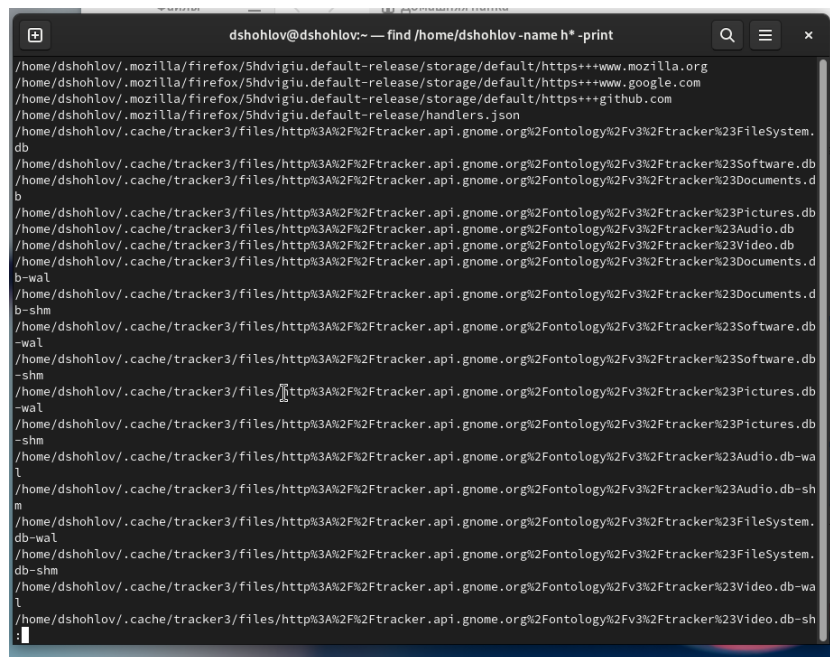


Figure 2.3: Поиск файлов

5 Выведем на экран (постранично) имена файлов из каталога /etc, начинающиеся с символа h.

```
find /etc -name "h*" -print | less
```

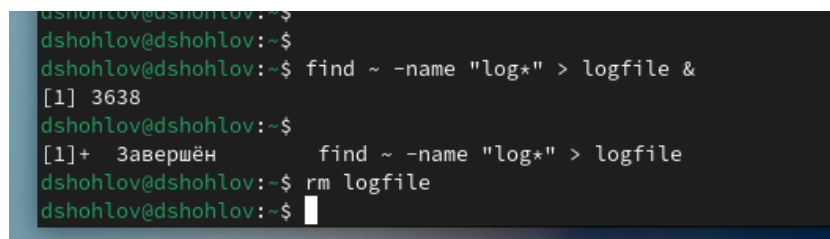


Figure 2.4: Поиск файлов

6 Запустили в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log. Процесс выполнен

7 Удалили файл ~/logfile. Но сначала убили процесс в нем.

```

dshohlov@dshohlov:~$
dshohlov@dshohlov:~$ find ~ -name "log*" > logfile &
[1] 3638
dshohlov@dshohlov:~$
[1]+  Завершён      find ~ -name "log*" > logfile
dshohlov@dshohlov:~$ rm logfile
dshohlov@dshohlov:~$
dshohlov@dshohlov:~$
dshohlov@dshohlov:~$ gedit &
[1] 3676
dshohlov@dshohlov:~$ ps | grep gedit
  3676 pts/0    00:00:00 gedit
dshohlov@dshohlov:~$ kill 3676
dshohlov@dshohlov:~$
[1]+  Завершено    gedit
dshohlov@dshohlov:~$

```

Figure 2.5: Фоновый запуск процесса

8 Запустили из консоли в фоновом режиме редактор gedit.

9 Определили идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep

10 Прочитали справку (man) команды kill, после чего используйте её для завершения процесса gedit.

```

dshohlov@dshohlov:~$
dshohlov@dshohlov:~$
dshohlov@dshohlov:~$ find ~ -name "log*" > logfile &
[1] 3638
dshohlov@dshohlov:~$
[1]+  Завершён      find ~ -name "log*" > logfile
dshohlov@dshohlov:~$ rm logfile
dshohlov@dshohlov:~$
dshohlov@dshohlov:~$
dshohlov@dshohlov:~$ gedit &
[1] 3676
dshohlov@dshohlov:~$ ps | grep gedit
  3676 pts/0    00:00:00 gedit
dshohlov@dshohlov:~$ kill 3676
dshohlov@dshohlov:~$
[1]+  Завершено    gedit

```

Figure 2.6: Фоновый запуск и завершение процесса

11 Выполним команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man.


```
Файлы | Домашняя папка
dshohlov@dshohlov:~ -- man df
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 environ-
  ment variable POSIXLY_CORRECT is set, in which case 512-byte blocks are used.

  If an argument is the absolute file name of a device node containing a mounted file system, df shows
  the space available on that file system rather than on the file system containing the device node.
  This version of df cannot show the space available on unmounted file systems, because on most kinds
  of systems doing so requires very nonportable intimate knowledge of file system structures.

OPTIONS
  Show information about the file system on which each FILE resides, or all file systems by default.

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

  -a, --all
      include pseudo, duplicate, inaccessible file systems

  -B, --block-size=SIZE
      scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576
      bytes; see SIZE format below

  --direct
      show statistics for a file instead of mount point

  -h, --human-readable
      Manual page df(1) line 1 (press h for help or q to quit)
```

Figure 2.7: Справка по команде df

```
Файлы | Домашняя папка
dshohlov@dshohlov:~ -- man du
DU(1) User Commands DU(1)

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.

  -0, --null
      end each output line with NUL, not newline

  -a, --all
      write counts for all files, not just directories

  --apparent-size
      print apparent sizes rather than device usage; although the apparent size is usually smaller,
      it may be larger due to holes in ('sparse') files, internal fragmentation, indirect blocks,
      and the like

  -B, --block-size=SIZE
      scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576
      bytes; see SIZE format below

  -b, --bytes
      equivalent to '--apparent-size --block-size=1'

  -c, --total
      produce a grand total

  Manual page du(1) line 1 (press h for help or q to quit)
```

Figure 2.8: Запуск команды df

```

dshohlov@dshohlov:~$ 
dshohlov@dshohlov:~$ df
Файловая система 1К-блоков  Использовано  Доступно  Использовано%  Смонтировано в
/dev/sda3          103805952    23329444  76309660        24% /
devtmpfs           4096          0      4096           0% /dev
tmpfs              4044844       0    4044844         0% /dev/shm
tmpfs              1617940      1884    1616056         1% /run
tmpfs              4044848       16    4044832         1% /tmp
/dev/sda3          103805952    23329444  76309660        24% /home
/dev/sda2           996780      271404    656564         30% /boot
tmpfs              808968       168    808800          1% /run/user/1089
dshohlov@dshohlov:~$ 

```

Figure 2.9: Справка по команде du

```

10  ./git-extended/.git/objects
8   ./git-extended/.git/logs/refs/heads
8   ./git-extended/.git/logs/refs/remotes/origin
8   ./git-extended/.git/logs/refs/remotes
16  ./git-extended/.git/logs/refs
20  ./git-extended/.git/logs
212 ./git-extended/.git
220 ./git-extended
0   ./emacs.d/eln-cache/29.3-bf1c73ce
0   ./emacs.d/eln-cache
0   ./emacs.d
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   ./paly/games/paly
0   ./paly/games
0   ./paly
396872 .
dshohlov@dshohlov:~$ 

```

Figure 2.10: Запуск команды du

12 Воспользовавшись справкой команды find, вывести имена всех директо-
рий, имеющихсся в нашем домашнем каталоге.

```
find ~ -type d
```

```
/home/dshohlov/git-extended/.git/logs/refs
/home/dshohlov/git-extended/.git/logs/refs/heads
/home/dshohlov/git-extended/.git/logs/refs/remotes
/home/dshohlov/git-extended/.git/logs/refs/remotes/origin
/home/dshohlov/.emacs.d
/home/dshohlov/.emacs.d/eln-cache
/home/dshohlov/.emacs.d/eln-cache/29.3-bf1c73ce
/home/dshohlov/monthly
/home/dshohlov/reports
/home/dshohlov/reports/monthly
/home/dshohlov/reports/monthly/monthly
/home/dshohlov/ski.places
/home/dshohlov/ski.places/equipment
/home/dshohlov/ski.places/plans
/home/dshohlov/australia
/home/dshohlov/paly
/home/dshohlov/paly/games
/home/dshohlov/paly/games/paly
dshohlov@dshohlov:~$
```

Figure 2.11: Поиск директорий

3 Вывод

В данной работе мы ознакомились с инструментами поиска файлов и фильтрации текстовых данных. А также приобрели практические навыки по управлению процессами.

4 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете? Ответ:

- a) `stdin` — стандартный поток ввода (клавиатура),
- b) `stdout` — стандартный поток вывода (консоль),
- c) `stderr` — стандартный поток вывод сообщений об ошибках на экран

2. Объясните разницу между операцией `>` и `>>` Ответ: Разница заключается в том, что Символ `>` используется для переназначения стандартного ввода команды, а символ `>>` используется для присоединения данных в конец файла стандартного вывода команды.

3. Что такое конвейер? Ответ: Конвейер – это способ связи между двумя программами. Например: конвейер `pipe` служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передается последующей. Синтаксис у конвейера следующий: команда1 | команда 2

4. Что такое процесс? Чем это понятие отличается от программы? Ответ: Процесс - это программа, которая выполняется в отдельном виртуальном адресном пространстве независимо от других программ или их пользованию по необходимости.

5. Что такое PID и GID? Ответ: Во первых id — UNIX-утилита, выводящая информацию об указанном пользователе USERNAME или текущем пользователе, который запустил данную команду и не указал явно имя пользователя.
- 1) GID – (Group ID) - идентификатор группы
- 2) UID – (User ID) - идентификатор группы Обычно UID является — положительным целым числом в диапазоне от 0 до 65535, по которому в системе однозначно отслеживаются действия пользователя
6. Что такое задачи и какая команда позволяет ими управлять? Ответ: Запущенные фонов программы называются задачами(процессами) (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент процессов. Для завершения процесса необходимо выполнить команду : kill % номер задачи
7. Найдите информацию об утилитах top и htop. Каковы их функции? Ответ: Top это консольная команда, которая выводит список работающих в системе процессов и информации о них. По умолчанию она в реальном времени сортирует их по нагрузке на процессор. Htop же является альтернативой программы top она предназначена для вывода на терминал списка запущенных процессов и информации о них.
8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды. Ответ: Команда find используется для поиска и отображения имен файлов, соответствующих заданной строке символов. Синтаксис: find trek [-options] Пример: Задача - Вывести на экран имена файлов из каталога /etc и его подкаталогов, Заканчивающихся на k: find ~ -name “*k” -print
9. Можно ли по контексту (содержанию) найти файл? Если да, то как? Ответ: Можно, команда grep способна обрабатывать вывод других файлов. Для

этого надо использовать конвейер, связав вывод команды с вводом `grep`.
Пример: Задача - показать строки в каталоге `/dreams` с именами начинающимися на `t`, в которых есть фраза: `I like of Operating systems`
`grep I like of Operating systems t*`

10. Как определить объем свободной памяти на жёстком диске? Ответ: Команда `df` показывает размер каждого смонтированного раздела диска. Например команда: `df -h`
11. Как определить объем вашего домашнего каталога? Ответ: Команда `du` показывает число килобайт, используемое каждым файлом или каталогом. Например команда: `du -sh`
12. Как удалить зависший процесс? Ответ: Перед тем, как выполнить остановку процесса, нужно определить его PID. Когда известен PID, мы можем убить его командой `kill`. Команда `kill` принимает в качестве параметра PID процесса. PID можно узнать с помощью команд `ps`, `grep`, `top` или `htop`