

Отчёт по лабораторной работе №6

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

Александр Андриевский

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Вывод	12
4	Контрольные вопросы	13

Список иллюстраций

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

Список таблиц

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.

```
root@debian:~# ls /etc/ > file.txt
root@debian:~# ls >> file.txt
root@debian:~# cat file.txt
adduser.conf
alsa
alternatives
anacrontab
apache2
apg.conf
apparmor
apparmor.d
appstream.conf
apt
avahi
bash.bashrc
bash_completion
bash_completion.d
bindresvport.blacklist
binfmt.d
bluetooth
bogofilter.cf
ca-certificates
ca-certificates.conf
chatscripts
chromium
console-setup
```

Рис. 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.

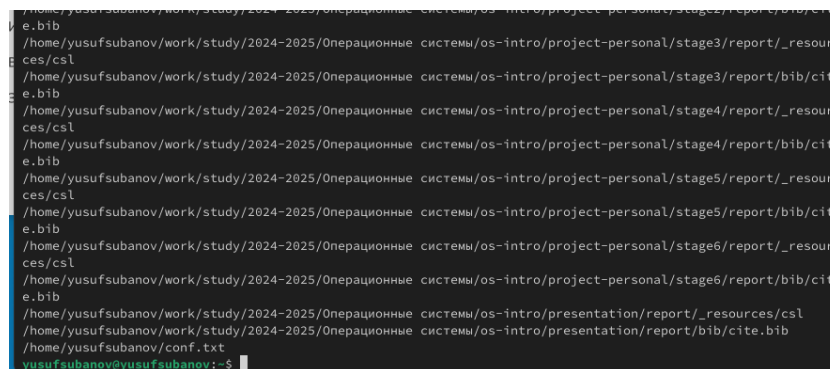
```

root@debian:~# grep .conf file.txt > conf.txt
root@debian:~# cat conf.txt
adduser.conf
apg.conf
appstream.conf
ca-certificates.conf
dconf
debconf.conf
deluser.conf

```

Рис. 2.2: Поиск расширения .conf

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



```

/home/yusufsubanov/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage3/report/_resources/csl
/home/yusufsubanov/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage3/report/bib/cite.bib
/home/yusufsubanov/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage4/report/_resources/csl
/home/yusufsubanov/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage4/report/bib/cite.bib
/home/yusufsubanov/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage5/report/_resources/csl
/home/yusufsubanov/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage5/report/bib/cite.bib
/home/yusufsubanov/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage6/report/_resources/csl
/home/yusufsubanov/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage6/report/bib/cite.bib
/home/yusufsubanov/work/study/2024-2025/Операционные системы/os-intro/presentation/report/_resources/csl
/home/yusufsubanov/work/study/2024-2025/Операционные системы/os-intro/presentation/report/bib/cite.bib
/home/yusufsubanov/conf.txt
yusufsubanov@yusufsubanov:~$

```

Рис. 2.3: Поиск файлов

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

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

```

root@debian:~# find /etc
/etc
/etc/mtab
/etc/hostname
/etc/networks
/etc/rcS.d
/etc/rcS.d/S01hwclock.sh
/etc/rcS.d/S01udev
/etc/rcS.d/S01alsa-utils
/etc/rcS.d/S01keyboard-setup.sh
/etc/rcS.d/S01apparmor
/etc/rcS.d/S01procps
/etc/rcS.d/S01networking
/etc/rcS.d/S01x11-common
/etc/rcS.d/S01kmod
/etc/rcS.d/S01plymouth-log
/etc/dpkg
/etc/dpkg/dpkg.cfg.d
/etc/dpkg/dpkg.cfg.d/force-unsafe-io
/etc/dpkg/dpkg.cfg
/etc/dpkg/origins
/etc/dpkg/origins/debian
/etc/dpkg/origins/default
/etc/shadow
/etc/passwd
/etc/logrotate.d
/etc/logrotate.d/dpkg
/etc/logrotate.d/apt
/etc/logrotate.d/speech-dispatcher
/etc/logrotate.d/btmp
/etc/logrotate.d/sane-utils
/etc/logrotate.d/ppp
/etc/logrotate.d/cups-daemon
/etc/logrotate.d/bootlog
/etc/logrotate.d/wtmp
/etc/logrotate.d/alternatives
/etc/emacs

```

Рис. 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.

```

root@debian:~# find /etc -name "hx*" -print | less

[5]+ Stopped find /etc -name "hx*" -print | less
root@debian:~# find /etc -name "log*" > logfile &
[6] 6752
root@debian:~# find /etc -name "hx*" -print | less
[6] Done find /etc -name "log*" > logfile

[7]+ Stopped find /etc -name "hx*" -print | less
root@debian:~# find ~ -name "log*" > logfile &
[8] 6764
root@debian:~# rm logfile
[8] Done find ~ -name "log*" > logfile
root@debian:~#

```

Рис. 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.

```
update-alternatives: using /usr/bin/gedit to provide /usr/bin/gnome-text-editor (gnome-text-editor) in auto mode
update-alternatives: warning: not replacing /usr/bin/gnome-text-editor with a link
[8] Exit 127      gedit
root@debian:~# gedit &
[8] 6925
root@debian:~# Error creating proxy: The connection is closed (g-io-error-quark, 18)
:Error creating proxy: The connection is closed (g-io-error-quark, 18)
:Error creating proxy: The connection is closed (g-io-error-quark, 18)
:Error creating proxy: The connection is closed (g-io-error-quark, 18)
:Error creating proxy: The connection is closed (g-io-error-quark, 18)

(gedit:6925): dconf-WARNING **: 13:57:48.386: failed to commit changes to dconf: The connection is closed
^C
[8] Done      gedit
root@debian:~# ps | grep gedit
      █
```

Рис. 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.

```
DF(1)                                                    User Commands                                                    DF(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.

  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

  -h, --human-readable
      print sizes in powers of 1024 (e.g., 1023M)

  -H, --si
      print sizes in powers of 1000 (e.g., 1.1G)

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

Рис. 2.7: Справка по команде df

```

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

    -D, --dereference-args
        dereference only symlinks that are listed on the command line

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

```

Рис. 2.8: Запуск команды df

```

root@debian:~# df
Filesystem      1K-blocks      Used Available Use% Mounted on
udev             978708          0      978708   0% /dev
tmpfs            201424        1208      200216   1% /run
/dev/sda1       19480400    5732132    12733384  32% /
tmpfs           1007116          0      1007116   0% /dev/shm
tmpfs             5120           8         5112   1% /run/lock
ucheba          486660196 483815444    2844752 100% /media/sf_ucheba
tmpfs            201420          96      201324   1% /run/user/1000
root@debian:~#

```

Рис. 2.9: Справка по команде du

```

312  ./work/study/2024-2025/Операционные системы/os-intro/presentation/presentation
764  ./work/study/2024-2025/Операционные системы/os-intro/presentation
61792 ./work/study/2024-2025/Операционные системы/os-intro
61792 ./work/study/2024-2025/Операционные системы
61792 ./work/study/2024-2025
61792 ./work/study
61792 ./work
540  ./texlive2023/texmf-var/luatex-cache/generic/names
35824 ./texlive2023/texmf-var/luatex-cache/generic/fonts/otl
35824 ./texlive2023/texmf-var/luatex-cache/generic/fonts
36364 ./texlive2023/texmf-var/luatex-cache/generic
36364 ./texlive2023/texmf-var/luatex-cache
36364 ./texlive2023/texmf-var
36364 ./texlive2023
0    ./monthly
0    ./reports/monthly/monthly
0    ./reports/monthly
0    ./reports
4    ./ski.plases/equipment
0    ./ski.plases/plans
4    ./ski.plases
0    ./australia
0    ./play/games/play
0    ./play/games
0    ./play
584224 .

```

Рис. 2.10: Запуск команды du

12 Воспользовавшись справкой команды `find`, вывести имена всех директорий, имеющих в нашем домашнем каталоге.

```
find ~ -type d
```

```
/root/reports/monthly/monthly/june
/root/reports/monthly/monthly/april
/root/reports/monthly/monthly/may
/root/vboxpostinstall.sh
/root/ski.plases
/root/ski.plases/plans
/root/ski.plases/equipment
/root/ski.plases/equipment/equiplist
/root/ski.plases/equipment/equiplist2
/root/may
/root/.profile
/root/feathers2
/root/my_os
/root/.gnupg
/root/.gnupg/.#lk0x0000562b92d0a3c0.debian.2632
/root/.gnupg/pubring.kbx~
/root/.gnupg/private-keys-v1.d
/root/.gnupg/private-keys-v1.d/809F4270623A4F638BD3644B8BD1C5A5622365B1.key
/root/.gnupg/private-keys-v1.d/2FA2378C60E5FBF64B2FB771BB2875322ECD026B.key
/root/.gnupg/S.gpg-agent.extra
/root/.gnupg/openpgp-revocs.d
/root/.gnupg/openpgp-revocs.d/066B56D1B6A6287F6AEA1750CC745529D6E23236.rev
/root/.gnupg/S.gpg-agent
/root/.gnupg/S.gpg-agent.browser
/root/.gnupg/.#lk0x000055da5e9e1760.debian.2631
/root/.gnupg/S.gpg-agent.ssh
/root/.gnupg/trustdb.gpg
/root/.gnupg/pubring.kbx
/root/.bash_history
/root/monthly
/root/monthly/june
/root/monthly/april
/root/monthly/may
/root/.bashrc
/root/.cache
/root/.gitconfig
root@debian:~#
```

Рис. 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 фоном 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`