

Отчет по лабораторной работе №6

Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов

Кочкарев “sakochkarev” Станислав

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 3 |
| 2 | Задание | 4 |
| 3 | Выполнение лабораторной работы | 5 |
| 4 | Выводы | 13 |
| 5 | Ответы на контрольные вопросы | 14 |

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

- Выполнение последовательности действий
- Работа с файлами
- Поиск по файлам
- Перенаправление ввода-вывода
- Работа с конвейерами

3 Выполнение лабораторной работы

Сперва-наперво мы осуществили вход в систему под нашим пользователем (рис. 3.1).

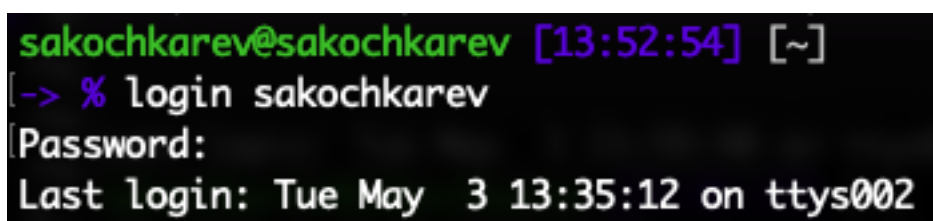
A terminal window with a black background and green text. The prompt is 'sakochkarev@sakochkarev [13:52:54] [~]'. The user enters '-> % login sakochkarev'. The prompt changes to 'Password:'. The user enters a password (not visible). The prompt changes to 'Last login: Tue May 3 13:35:12 on ttys002'.

Рис. 3.1: Осуществление входа в пользователя

После этого был выполнен первый шаг, коим являлась запись в файл `file.txt` названия файлов в директории `/etc` (рис. 3.2). Стоит отметить, что на компьютере, на котором выполнялась лабораторная работа директория по пути `/etc` на самом деле является ссылкой на `/private/etc`, поэтому именно этот путь был использован вместо `/etc`.

```
sakochkarev@sakochkarev [13:53:48] [~]
[-> % ls /etc > file.txt
sakochkarev@sakochkarev [13:54:03] [~]
[-> % cat file.txt
lrwxr-xr-x 1 root wheel 11B Mar 26 10:21 /etc@ -> private/etc
sakochkarev@sakochkarev [13:54:07] [~]
[-> % ls /private/etc > file.txt
sakochkarev@sakochkarev [13:54:19] [~]
[-> % cat file.txt
total 1640
drwxr-xr-x 119 root wheel 3.7K Apr 27 23:13 ./
drwxr-xr-x 6 root wheel 192B Apr 27 23:16 ../
-rw-r--r-- 1 root wheel 12K Dec 31 2018 .sudoers.tmp.swp
-rw-r--r-- 1 root wheel 24B Jul 17 2020 AFP.conf
-rw-r--r-- 1 root wheel 515B Mar 26 10:21 afpovertcp.cfg
-rw-r--r-- 1 root wheel 515B Feb 26 2019 afpovertcp.cfg-orig
lrwxr-xr-x 1 root wheel 15B Mar 26 10:21 aliases@ -> postfix/aliases
-rw-r--r-- 1 root wheel 16K Mar 26 10:21 aliases.db
drwxr-xr-x 11 root wheel 352B Apr 5 18:07 apache2/
drwxr-xr-x 16 root wheel 512B Mar 26 10:21 asl/
-rw-r--r-- 1 root wheel 1.0K Mar 26 10:21 asl.conf
-rw-r--r-- 1 root wheel 149B Mar 26 10:21 auto_home
-rw-r--r-- 1 root wheel 195B Mar 26 10:21 auto_master
-rw-r--r-- 1 root wheel 194B Aug 23 2018 auto_master-orig
```

Рис. 3.2: Запись содержимого каталога /etc (/private/etc) в файл file.txt

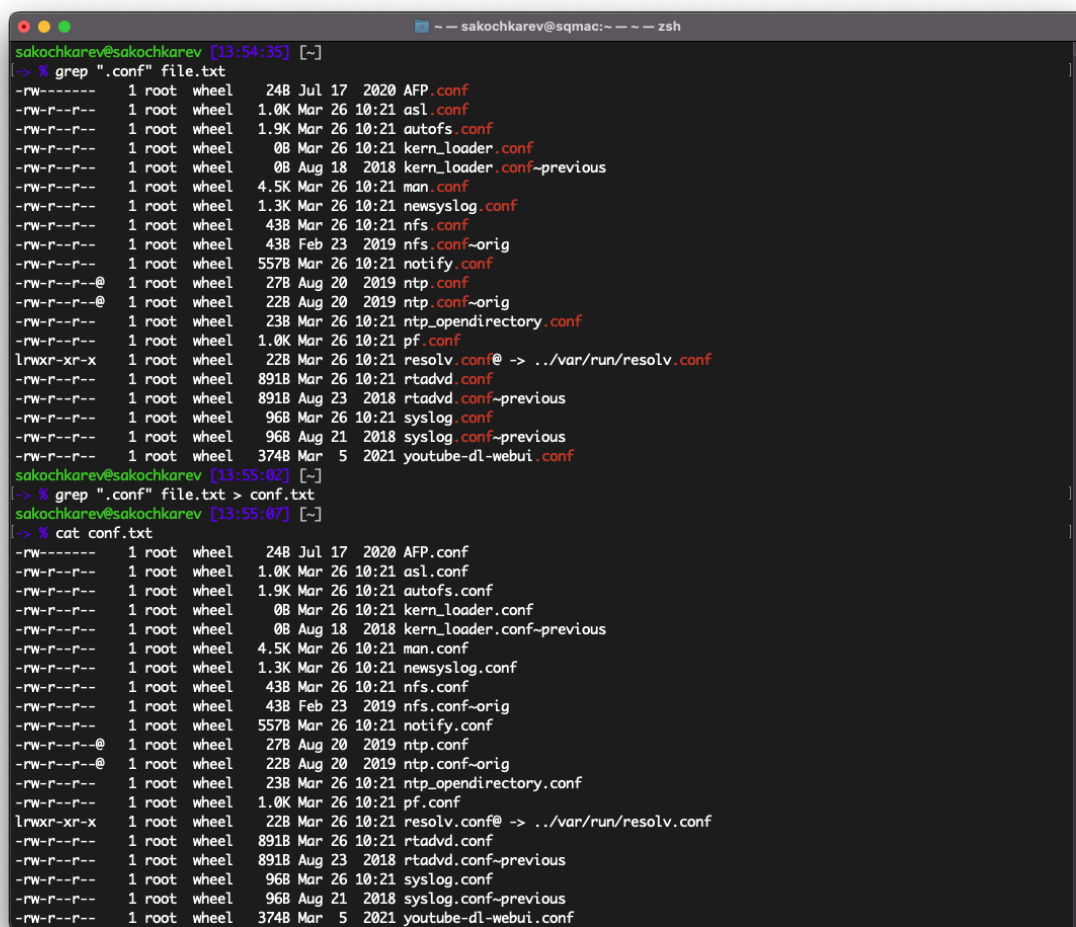
Дополнительно к этому также в file.txt было добавлено содержимое домашней директории (рис. 3.3).

```
sakochkarev@sakochkarev [13:54:21] [~]
[-> % ls >> file.txt
sakochkarev@sakochkarev [13:54:33] [~]
[-> % cat file.txt
total 1640
drwxr-xr-x 119 root wheel 3.7K Apr 27 23:13 ./
drwxr-xr-x 6 root wheel 192B Apr 27 23:16 ../
-rw-r--r-- 1 root wheel 12K Dec 31 2018 .sudoers.tmp.swp
-rw-r--r-- 1 root wheel 24B Jul 17 2020 AFP.conf
-rw-r--r-- 1 root wheel 515B Mar 26 10:21 afpovertcp.cfg
-rw-r--r-- 1 root wheel 515B Feb 26 2019 afpovertcp.cfg-orig
lrwxr-xr-x 1 root wheel 15B Mar 26 10:21 aliases@ -> postfix/aliases
-rw-r--r-- 1 root wheel 16K Mar 26 10:21 aliases.db
drwxr-xr-x 11 root wheel 352B Apr 5 18:07 apache2/
drwxr-xr-x 16 root wheel 512B Mar 26 10:21 asl/
-rw-r--r-- 1 root wheel 1.0K Mar 26 10:21 asl.conf
-rw-r--r-- 1 root wheel 149B Mar 26 10:21 auto_home
-rw-r--r-- 1 root wheel 195B Mar 26 10:21 auto_master
-rw-r--r-- 1 root wheel 194B Aug 23 2018 auto_master-orig
-rw-r--r-- 1 root wheel 1.9K Mar 26 10:21 autofs.conf
-rw-r--r-- 1 root wheel 265B Mar 26 10:21 bashrc
-rw-r--r-- 1 root wheel 9.1K Mar 26 10:21 bashrc_Apple_Terminal
-rw-r--r-- 1 root wheel 265B Aug 18 2018 bashrc-previous
-rw-r--r-- 1 root wheel 414B Jan 23 2020 bootpd.plist
```

Рис. 3.3: Дозапись содержимого домашнего каталога в file.txt

Далее из файла file.txt созданного ранее были выведены все файлы с расширением .conf. Этот вывод был также дополнительно записан в отдельный

файл `conf.txt` (рис. 3.4).



```
sakochkarev@sakochkarev [13:54:35] [~]
|-> % grep ".conf" file.txt
-rw----- 1 root wheel 248 Jul 17 2020 AFP.conf
-rw-r--r-- 1 root wheel 1.0K Mar 26 10:21 asl.conf
-rw-r--r-- 1 root wheel 1.9K Mar 26 10:21 autofs.conf
-rw-r--r-- 1 root wheel 0B Mar 26 10:21 kern_loader.conf
-rw-r--r-- 1 root wheel 0B Aug 18 2018 kern_loader.conf-previous
-rw-r--r-- 1 root wheel 4.5K Mar 26 10:21 man.conf
-rw-r--r-- 1 root wheel 1.3K Mar 26 10:21 newsyslog.conf
-rw-r--r-- 1 root wheel 43B Mar 26 10:21 nfs.conf
-rw-r--r-- 1 root wheel 43B Feb 23 2019 nfs.conf-orig
-rw-r--r-- 1 root wheel 557B Mar 26 10:21 notify.conf
-rw-r--r--@ 1 root wheel 27B Aug 20 2019 ntp.conf
-rw-r--r--@ 1 root wheel 22B Aug 20 2019 ntp.conf-orig
-rw-r--r-- 1 root wheel 23B Mar 26 10:21 ntp_opendirectory.conf
-rw-r--r-- 1 root wheel 1.0K Mar 26 10:21 pf.conf
lrwxr-xr-x 1 root wheel 22B Mar 26 10:21 resolv.conf@ -> ../var/run/resolv.conf
-rw-r--r-- 1 root wheel 891B Mar 26 10:21 rtadvd.conf
-rw-r--r-- 1 root wheel 891B Aug 23 2018 rtadvd.conf-previous
-rw-r--r-- 1 root wheel 96B Mar 26 10:21 syslog.conf
-rw-r--r-- 1 root wheel 96B Aug 21 2018 syslog.conf-previous
-rw-r--r-- 1 root wheel 374B Mar 5 2021 youtube-dl-webui.conf
sakochkarev@sakochkarev [13:55:02] [~]
|-> % grep ".conf" file.txt > conf.txt
sakochkarev@sakochkarev [13:55:07] [~]
|-> % cat conf.txt
-rw----- 1 root wheel 248 Jul 17 2020 AFP.conf
-rw-r--r-- 1 root wheel 1.0K Mar 26 10:21 asl.conf
-rw-r--r-- 1 root wheel 1.9K Mar 26 10:21 autofs.conf
-rw-r--r-- 1 root wheel 0B Mar 26 10:21 kern_loader.conf
-rw-r--r-- 1 root wheel 0B Aug 18 2018 kern_loader.conf-previous
-rw-r--r-- 1 root wheel 4.5K Mar 26 10:21 man.conf
-rw-r--r-- 1 root wheel 1.3K Mar 26 10:21 newsyslog.conf
-rw-r--r-- 1 root wheel 43B Mar 26 10:21 nfs.conf
-rw-r--r-- 1 root wheel 43B Feb 23 2019 nfs.conf-orig
-rw-r--r-- 1 root wheel 557B Mar 26 10:21 notify.conf
-rw-r--r--@ 1 root wheel 27B Aug 20 2019 ntp.conf
-rw-r--r--@ 1 root wheel 22B Aug 20 2019 ntp.conf-orig
-rw-r--r-- 1 root wheel 23B Mar 26 10:21 ntp_opendirectory.conf
-rw-r--r-- 1 root wheel 1.0K Mar 26 10:21 pf.conf
lrwxr-xr-x 1 root wheel 22B Mar 26 10:21 resolv.conf@ -> ../var/run/resolv.conf
-rw-r--r-- 1 root wheel 891B Mar 26 10:21 rtadvd.conf
-rw-r--r-- 1 root wheel 891B Aug 23 2018 rtadvd.conf-previous
-rw-r--r-- 1 root wheel 96B Mar 26 10:21 syslog.conf
-rw-r--r-- 1 root wheel 96B Aug 21 2018 syslog.conf-previous
-rw-r--r-- 1 root wheel 374B Mar 5 2021 youtube-dl-webui.conf
```

Рис. 3.4: Вывод и запись файлов с расширением `.conf`

Для того, чтобы определить и вывести все файлы в директории, начинающиеся с буквы `s`, можно использовать команду `ls s*`, а также команду `find . -name "s*" (рис. 3.5).`

```

sakochkarev@sakochkarev [13:55:09] [~]
[-> % ls c*
-rw-r--r--  1 sakochkarev  staff   1.2K May  3 13:55 conf.txt
sakochkarev@sakochkarev [13:55:30] [~]
[-> % find . -name "c*" -depth 1
./conf.txt

```

Рис. 3.5: Вывод файлов начинающихся с буквы c

Продолжая направление предыдущего задания, было необходимо вывести имена всех файлов из каталога /etc (/private/etc), начинающихся с буквы h. Для этого была использована команда `ls h*` (рис. 3.6).

```

sakochkarev@sakochkarev [13:55:42] [~]
[-> % ls /private/etc h*
zsh: no matches found: h*
sakochkarev@sakochkarev [13:56:39] [~]
[-> % ls /private/etc/h*
-rw-r--r--  1 root  wheel   213B Mar 26 10:21 /private/etc/hosts
-rw-r--r--  1 root  wheel     0B Mar 26 10:21 /private/etc/hosts.equiv
-rw-r--r--  1 root  wheel   213B Feb 26 2019 /private/etc/hosts~orig

```

Рис. 3.6: Вывод файлов начинающихся с буквы h в директории /etc

Следующим заданием был запуск процесса в фоновом режиме, который должен был записывать в файл ~/logfile файлы, имена которых начинаются с log. Сделано это было командой `find . -name "log*" -print > logfile &` (рис. 3.7).


```
sakochkarev@sakochkarev [13:57:21] [~]
[-> % find . -name "log*" -print > logfile &
[1] 99868
sakochkarev@sakochkarev [13:57:48] [~]
[-> %
[1] + 99868 done      find . -name "log*" -print > logfile
sakochkarev@sakochkarev [13:57:48] [~]
[-> % cat logfile
./Library/Preferences/loginwindow.plist
./Library/Keychains/login.keychain-db
./Library/Group Containers/group.io.robby.homeassistant/logs
./Library/Caches/com.apple.proactive.eventtracker/log_stores
./log.txt
./logfile
./.oh-my-zsh/log
./.oh-my-zsh/.git/logs
```

Рис. 3.7: Фоновая работа по записи файлов с названием log*

После окончания работы, было необходимо удалить получившийся файл (рис. 3.8).

```
sakochkarev@sakochkarev [13:57:56] [~]
[-> % rm logfile
```

Рис. 3.8: Удаление ~/logfile

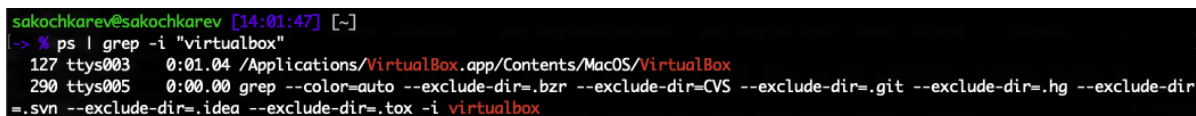
Далее шли задания с фоновыми процессами и работой с ними.

Было необходимо запустить из консоли в фоновом режиме программу *gedit*. Так как ее не было на моем устройстве, она был заменена другой программой, а конкретно *VirtualBox*. Запуск производился командой `VirtualBox &` (рис. 3.9).

```
sakochkarev@sakochkarev [13:59:40] [~]
[-> % VirtualBox &
[1] 127
```

Рис. 3.9: Запуск VirtualBox в фоновом режиме

Далее нужно было определить идентификатор только что запущенного процесса. Для этого использовалась команда `ps` с конвейером и командой `grep`. В итоге получилась команда `ps | grep -i "VirtualBox"` (рис. 3.10). Дополнительными вариантами нахождения идентификатора процесса являются команды `top` (`htop`) и встроенные GUI-утилиты, как, например, `Activity Monitor` на `macOS`.



```
sakochkarev@sakochkarev [14:01:47] [~]
-> % ps | grep -i "virtualbox"
127 ttys003  0:01.04 /Applications/VirtualBox.app/Contents/MacOS/VirtualBox
290 ttys005  0:00.00 grep --color=auto --exclude-dir=.bzr --exclude-dir=CVS --exclude-dir=.git --exclude-dir=.hg --exclude-dir=.svn --exclude-dir=.idea --exclude-dir=.tox -i virtualbox
```

Рис. 3.10: Нахождение идентификатора процесса

Дальнейшим шагом было завершение запущенного процесса командой `kill`. Для этого был изучен соответствующий мануал (`man kill`). Результирующей командой с использованием полученного ранее идентификатора процесса было `kill -9 127` (рис. 3.11).



```
sakochkarev@sakochkarev [14:01:50] [~]
-> % kill -9 127
```

Рис. 3.11: Завершение процесса

Далее мы ознакомились с командами `df` и `du` изучив мануал по обеим командам и выполнив их (рис. 3.12).

```
sakochkarev@sakochkarev [14:01:28] [~]
-> % man df
sakochkarev@sakochkarev [14:02:33] [~]
-> % man du
sakochkarev@sakochkarev [14:02:40] [~]
-> % df -h
Filesystem      Size  Used Avail Capacity iused   ifree %used  Mounted on
/dev/disk1s5s1  182Gi  14Gi  24Gi    38% 500637 251687640    0% /
devfs           339Ki  339Ki   0Bi   100%  1172    0 100% /dev
/dev/disk1s4    182Gi   4.0Gi  24Gi    15%    4 251687640    0% /System/Volumes/VM
/dev/disk1s2    182Gi  729Mi  24Gi     3%  4976 251687640    0% /System/Volumes/Preboot
/dev/disk1s6    182Gi  12Mi  24Gi     1%   18 251687640    0% /System/Volumes/Update
/dev/disk1s1    182Gi  138Gi  24Gi    86% 1548282 251687640    1% /System/Volumes/Data
ntfs://disk0s3/BOOTCAMP 52Gi  41Gi  11Gi    79%    1    0 100% /Volumes/BOOTCAMP
map auto_home    0Bi    0Bi   0Bi   100%    0    0 100% /System/Volumes/Data/home
sakochkarev@sakochkarev [14:02:45] [~]
-> % ls Desktop
total 0
drwx-----+ 3 sakochkarev  staff   96B Apr 27 19:03 ./
drwxr-x----+ 28 sakochkarev  staff  896B May  3 14:02 ../
-rw-r--r--  1 sakochkarev  staff    0B Apr 27 19:03 .localized
sakochkarev@sakochkarev [14:02:59] [~]
-> % du -sh Downloads
0B    Downloads
sakochkarev@sakochkarev [14:03:15] [~]
-> % du -sh Movies
0B    Movies
sakochkarev@sakochkarev [14:03:20] [~]
-> % du -sh ski.places
8.0K  ski.places
sakochkarev@sakochkarev [14:03:29] [~]
-> % du -h ski.places
0B    ski.places/plans
8.0K  ski.places/equipment
8.0K  ski.places
```

Рис. 3.12: Изучение мануала и использование команд df, du

Последним заданием было изучение команды find (man find) и использование ее параметров для вывода всех директорий в домашнем каталоге (рис. 3.13).

```
sakochkarev@sakochkarev [14:05:38] [~]  
[-> % man find  
sakochkarev@sakochkarev [14:07:50] [~]  
[-> % find . -depth 1 -type d  
./Music  
./Pictures  
./Hello world  
./Desktop  
./Library  
./Public  
./Movies  
./ski.places  
./Trash  
./Documents  
./play  
./oh-my-zsh  
./Downloads  
./zsh_sessions
```

Рис. 3.13: Изучение мануала и использование команды `find` для вывода всех директорий в домашнем каталоге

В конце стоит подвести, что в работе было использовано понятие файла, как обобщенное понятие, поэтому во всех заданиях с требованиями нахождения файлов, они не были классифицированы отдельно. При необходимости соблюдения условий строгой классификации, достаточно добавить в каждую команду `find` флаг `-type f`.

4 Выводы

По выполнении лабораторной работы, мы познакомились с инструментами поиска файлов и фильтрации текстовых данных. Также приобрели практические навыки по управлению процессами, проверке использования диска и обслуживанию файловых систем.

5 Ответы на контрольные вопросы

1. По умолчанию в системе открыто три потока:
 - `stdin` – стандартный поток ввода
 - `stdout` – стандартный поток вывода
 - `stderr` – стандартный поток вывода сообщений об ошибках
2. `>` – открывает файл в режиме перезаписывания данных (новые данные перезапишут старые данные содержащиеся в файле)
`>>` – открывает файл в режиме добавления данных (новые данные будут добавлены в конце файла)
3. Конвейер – это инструмент “связывания” команд и объединения их в цепочки, в которых вывод одной команды перенаправляется в следующую и т.д.
4. Процесс – это абстракция совокупности взаимосвязанных системных ресурсов. Отличие процесса от программы в том, что процесс – это то, что запускает программа, которая чаще всего является исполняемым файлом.
5. PID – Process ID (идентификатор процесса)
GID – Group ID (идентификатор группа, под которой запущен процесс)
6. Задачи – это запущенные фоном программы. Ими можно управлять командой `jobs`.

7. Функции их идентичны и заключаются в менеджменте запущенных в системе процессов. Разница двух утилит заключается в “новизне” и поддерживаемости. `htop` являются более новой и улучшенной версией `top`.
8. Команда `find` являются командой по поиску файлов, директорий и других типов вхождений. Она позволяет искать вхождения по шаблону в указанных директориях. Также имеет дополнительные параметры для более тонкой настройки. Примеры:
- `find . -name "*.txt"` – ищет все файлы (**и другие типы вхождений**) с `.txt` в конце названия
 - `find . -name "*.txt" -type f` – ищет все файлы (**и только файлы**) с `.txt` в конце названия
9. Одна из команд, которую можно использовать для нахождения файлов по содержанию является команда `grep`. Пример: `grep -rnw '/home/sakochkarev' -e 'hello world'`.
10. Можно использовать команду `df -h` для определения дисков и свободного места на них. Если диск разделен на несколько разделов, то можно суммировать свободное место в каждом из них и получить общее свободное место на диске. А можно просто использовать встроенный GUI дисковый менеджер.
11. Находясь в домашнем каталоге использовать команду `du -sh`, либо находясь вне домашнего каталога использовать `du -sh ~`. Флаг `-s` суммирует объем всех вложенных файлов и директорий, получая общий объем, а `-h` просто делает объем более понятным благодаря преобразованию байтов в ближайшее деление (КБ, МБ, ГБ и т.д.).
12. Самый простой способ – используя встроенный менеджер процессов. Например `htop`. Найдя зависший процесс в списке отправить сигнал `SIGTERM`.