

Отчёт по лабораторной работе №8

Операционные системы

Бекауов Артур Тимурович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	11
5	Ответы на онтрольные вопросы	12

Список иллюстраций

3.1	Задание 1 и 2 - использование >	7
3.2	Задание 3 - использование grep и >	7
3.3	Задание 4 - поиск файлов	8
3.4	Задание 5 - поиск файлов в /etc	8
3.5	Задание 6 и 7 - фоновый процесс find	8
3.6	Задание 8 и 9 - идентификатор процесса	9
3.7	Задание 10 - Использование kill	9
3.8	Задание 11 - Использование df	9
3.9	Задание 11 - Использование du	10
3.10	Задание 12 - Использование find	10

Список таблиц

1 Цель работы

Цель данной лабораторной 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

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, начинающиеся 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 директорий, имеющихся в вашем домашнем каталоге.

3 Выполнение лабораторной работы

Осуществив вход в систему под именем atbekauov, приступаю к выполнению лабораторной работы. Сначала в файл file.txt записываю названия файлов, содержащихся в каталоге /etc с помощью перенаправления “>”. Также с помощью перенаправления в режиме добавления “>” дополняю file.txt выводом команды ls.(рис. 3.1).

```
[atbekauov@atbekauov ~]$ ls /etc > file.txt
[atbekauov@atbekauov ~]$ head file.txt
abrt
adjtime
aliases
alsa
alternatives
anaconda
anaconda
anaconda
asound.conf
at.deny
audit
[atbekauov@atbekauov ~]$ ls >> file.txt
```

Рис. 3.1: Задание 1 и 2 - использование >

Затем в файл conf.txt, ввожу имена всех файлов из file.txt имеющих расширение .conf. Делаю я это с помощью команды grep и перенаправления >. (рис. 3.2).

```
[atbekauov@atbekauov ~]$ grep .conf file.txt > conf.txt
[atbekauov@atbekauov ~]$ head conf.txt
asound.conf
chrony.conf
dconf
dnsmasq.conf
dracut.conf
dracut.conf.d
fprintd.conf
fuse.conf
host.conf
idmapd.conf
[atbekauov@atbekauov ~]$
```

Рис. 3.2: Задание 3 - использование grep и >

Вывожу файлы в домашнем каталоге начинающиеся с буквы “с” двумя вариан-

тами: 1. С помощью команд `ls` и `grep` с фильтром. 2. С помощью команды `find` с ключами `name` и `print` (рис. 3.3).

```
[atbekauov@atbekauov ~]$ ls | grep c*
conf.txt
[atbekauov@atbekauov ~]$ find -name "c*" - print
find: paths must precede expression: '-'
[atbekauov@atbekauov ~]$ find -name "c*" -print
./mozilla/firefox/tttvx33t.default-release/crashes
./mozilla/firefox/tttvx33t.default-release/compatibility.ini
./mozilla/firefox/tttvx33t.default-release/cookies.sqlite
./mozilla/firefox/tttvx33t.default-release/cert9.db
./mozilla/firefox/tttvx33t.default-release/storage/permanent/chrome
```

Рис. 3.3: Задание 4 - поиск файлов

После этого использую команду `find` как в прошлом задании, чтобы вывести имена файлов в `/etc`, начинающиеся на `"h"` (рис. 3.4).

```
[atbekauov@atbekauov ~]$ sudo find /etc -name "h*" -print
/etc/avahi/hosts
/etc/firewalld/helpers
/etc/libibverbs.d/hfi1verbs.driver
/etc/libibverbs.d/hns.driver
/etc/nvme/hostnqn
/etc/nvme/hostid
/etc/udev/hwdb.d
/etc/udev/hwdb.bin
/etc/host.conf
/etc/hosts
/etc/hostname
```

Рис. 3.4: Задание 5 - поиск файлов в `/etc`

Далее запускаю в фоновом режиме процесс, который с помощью `find` и `>` записывает в файл `logfile` имена файлов, которые начинаются с `log`. Дожидаюсь завершения процесса и удаляю `logfile`. (рис. 3.5).

```
[atbekauov@atbekauov ~]$ find ~ -name "log*" -print > logfile &
[1] 20390
[atbekauov@atbekauov ~]$ ls
australia  file.txt      layout.txt  may         play        work        Загрузки  Общедоступные
conf.txt   fun           LICENSE     monthly     reports     Видео       Изображения 'Рабочий стол'
feathers   git-extended logfile       my_os       ski.places  Документы  Музыка     Шаблоны
[1]+  Завершён      find ~ -name "log*" -print > logfile
[atbekauov@atbekauov ~]$ rm logfile
[atbekauov@atbekauov ~]$
```

Рис. 3.5: Задание 6 и 7 - фоновый процесс `find`

Запускаю в фоновом режиме текстовый редактор gedit, с помощью команды ps определяю идентификатор процесса gedit (можно также с помощью pgrep) (рис. 3.6).

```
[atbekauov@atbekauov ~]$ gedit &
[1] 20416
[atbekauov@atbekauov ~]$ ps aux | grep gedit
atbekau+  20416  1.5  0.7 771828 59324 pts/0    Sl   12:47   0:01 gedi
atbekau+  20440  0.0  0.0 227776  2304 pts/0    S+   12:48   0:00 grep
--color=auto gedit
[atbekauov@atbekauov ~]$ pgrep gedit
20416
[atbekauov@atbekauov ~]$
```

Рис. 3.6: Задание 8 и 9 - идентификатор процесса

Прочитав справку по команде kill, использую её в паре с идентификатором процесса gedit, чтобы этот процесс завершить. (рис. 3.7).

```
[atbekauov@atbekauov ~]$ man kill
[atbekauov@atbekauov ~]$ kill 20416
[atbekauov@atbekauov ~]$
[1]+  Завершено      gedit
[atbekauov@atbekauov ~]$
```

Рис. 3.7: Задание 10 - Использование kill

Прочитав справку по df и du, использую команду df с ключами i и v, чтобы вывести информацию об использовании памяти в моей файловой системе (рис. 3.8).

```
[atbekauov@atbekauov ~]$ df -iv
Файловая система  Иноды  Использовано  Свободно  Использовано%  Смонтировано в
/dev/sda3          0         0         0          1% /dev
devtmpfs          1011564      512  1011052          1% /dev
tmpfs             1015561         9  1015552          1% /dev/shm
tmpfs             819200      835   818365          1% /run
tmpfs             1048576        44  1048532          1% /tmp
/dev/sda3          0         0         0          - /home
/dev/sda2          65536      395   65141          1% /boot
tmpfs             203112      125   202987          1% /run/user/1000
```

Рис. 3.8: Задание 11 - Использование df

Также использую команду du, чтобы вывести информацию о том сколько места занимают файлы в директории ~/mounthly. (рис. 3.9).

```
[atbekauov@atbekauov ~]$ du -a ~/monthly
0      /home/atbekauov/monthly/april
0      /home/atbekauov/monthly/may
0      /home/atbekauov/monthly/june
0      /home/atbekauov/monthly
[atbekauov@atbekauov ~]$
```

Рис. 3.9: Задание 11 - Использование du

Затем с помощью команды `find` нахожу имена всех директорий в моём домашнем каталоге (рис. 3.10).

```
[atbekauov@atbekauov ~]$ find -type d
.
./mozilla
./mozilla/extensions
./mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
./mozilla/plugins
./mozilla/firefox
./mozilla/firefox/Crash Reports
./mozilla/firefox/Crash Reports/events
./mozilla/firefox/Pending Pings
./mozilla/firefox/tttx33t.default-release
./mozilla/firefox/tttx33t.default-release/minidumps
./mozilla/firefox/tttx33t.default-release/crashes
```

Рис. 3.10: Задание 12 - Использование find

4 Выводы

В ходе данной лабораторной работы я ознакомился с инструментами поиска файлов и фильтрации текстовых данных. Приобрёл практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

5 Ответы на онтрольные вопросы

1. Какие потоки ввода вывода вы знаете? 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. Объясните разницу между операцией `>` и `»`. Этот знак `>` - перенаправление ввода/вывода, а `»` - перенаправление в режиме добавления.
3. Что такое конвейер? Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.
4. Что такое процесс? Чем это понятие отличается от программы? 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. Что такое PID и GID? 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. Что такое задачи и какая команда позволяет ими управлять? Запущенные

фоном программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач.

7. Найдите информацию об утилитах top и htop. Каковы их функции?

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. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.

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 ли по контексту (содержанию) найти файл? Если да, то как? `find / -type f -exec grep -H 'текстДляПоиска' {} ;`
10. Как определить объем свободной памяти на жёстком диске? С помощью команды `df -h`.
11. Как определить объем вашего домашнего каталога? С помощью команды `du -s`.
12. Как удалить зависший процесс? С помощью команды `kill%` номер задачи.