

Отчёт по лабораторной работе №8

Поиск файлов, перенаправление ввода-вывода, просмотр запущенных процессов

Зоригоо Номун

Содержание

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

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

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 c? 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

Вошла в систему под моим именем, открыла терминал и записала в файле file.txt названия файлов, содержащихся в каталоге /etc с помощью `ls -lR /etc > file.txt` :

```
(nomun@Zorigoo)-[~]  
$ sudo ls -lR /etc > file.txt  
[sudo] password for nomun:
```

Рис. 1: Запись в файл

С помощью head я проверяю, что в файл записались названия файлов, содержащихся в каталоге /etc:

```
(nomun@Zorigoo)-[~]  
$ head file.txt  
/etc:  
adduser.conf  
alsa  
alternatives  
apache2  
apg.conf  
apparmor  
apparmor.d  
apt  
arp-scan
```

Рис. 2: Первые 8 файлов в file.txt

В file.txt добавляю названия файлов, из домашнего каталога используя `ls -lR ~/ >> file.txt`:

```
(nomun@Zorigoo)-[~]  
$ sudo ls -lR ~/ >> file.txt
```

Рис. 3: Добавление файлов из домашнего каталога

Вывожу имена всех файлов из file.txt, имеющих расширение .conf с помощью grep:

```

(nomun@Zorigoo)-[~]
$ grep .config file.txt
manpath.config
netconfig
openblas_config.h-x86_64-linux-gnu
im-config
im-config_wayland.sh
debian_config
ssh_config
ssh_config.d
sshd_config
sshd_config.d
/etc/ssh/ssh_config.d:
/etc/ssh/sshd_config.d:
/etc/texmf/dvips/config:
/etc/texmf/tex/generic/config:
Xwrapper.config
70im-config_launch
/home/nomun/work/blog/config:
/home/nomun/work/blog/config/_default:

```

Рис. 4: файлы с расширением .conf

Затем запишу их в новый текстовый файл conf.txt (grep .conf file.txt > conf.txt) и проверяю с помощью head:

```

(nomun@Zorigoo)-[~]
$ grep .config file.txt > conf.txt

(nomun@Zorigoo)-[~]
$ head conf.txt
manpath.config
netconfig
openblas_config.h-x86_64-linux-gnu
im-config
im-config_wayland.sh
debian_config
ssh_config
ssh_config.d
sshd_config
sshd_config.d

```

Рис. 5: добавление файлов с расширением .conf

Чтобы определить, какие файлы в домашнем каталоге имеют имена, начинавшиеся с символа "с", использую find ~ -name "с" print ; ~ обозначается домашний каталог, -

name (имя файлов) "с" строка символов, определяющая имя файла и `print` выводит результаты на экране:

```
(nomun@Zorigoo)-[~]  
$ find ~ -name "с*" -print  
/home/nomun/.mozilla/firefox/ihklkckx.default-esr/storage/permanent/chrome  
/home/nomun/.mozilla/firefox/ihklkckx.default-esr/storage/default/https+++www.yo  
utube.com/cache  
/home/nomun/.mozilla/firefox/ihklkckx.default-esr/storage/default/https+++www.yo  
utube.com/cache/caches.sqlite  
/home/nomun/.mozilla/firefox/ihklkckx.default-esr/storage/default/https+++web.te  
legram.org/cache
```

Рис. 6: файлы в домашнем каталоге начинающихся с "с"

Также можно это действие выполнить используя `ls -lR | grep "с"`

```
(nomun@Zorigoo)-[~]  
$ ls -lR | grep с*  
conf.txt
```

Рис. 7: поиск файла используя `grep`

с помощью `find /etc -name "h*" -print`, вывожу файлы из каталога `/etc`, начинающиеся с символа `h`:

```
(nomun@Zorigoo)-[~]  
$ sudo find /etc -name "h*" -print  
/etc/systemd/system/sysinit.target.wants/haveged.service  
/etc/init.d/haveged  
/etc/udev/hwdb.d  
/etc/xdg/xfce4/helpers.rc  
/etc/xdg/xfce4/helpers.rc.dpkg-new  
/etc/host.conf  
/etc/strongswan.d/charon/hmac.conf  
/etc/hosts.deny  
/etc/default/haveged  
/etc/hostname  
/etc/libibverbs.d/hns.driver  
/etc/libibverbs.d/hfi1verbs.driver  
/etc/apparmor.d/abstractions/hosts_access  
/etc/apparmor.d/tunables/home.d  
/etc/apparmor.d/tunables/home  
/etc/initramfs-tools/hooks  
/etc/hdparm.conf
```

Рис. 8: файлы в `etc` начинающихся с "h"

В фоновом режиме запускаю процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log:

```
(nomun@Zorigoo)-[~]  
$ sudo find ~ -name "log*" -print > logfile &  
[1] 3032  
  
(nomun@Zorigoo)-[~]  
$  
[1] + done      sudo find ~ -name "log*" -print > logfile
```

Рис. 9: Создание фонового режима

Удаляю созданный logfile и проверяю:

```
(nomun@Zorigoo)-[~]  
$ ls  
ab      bon      Downloads logfile my_os  reports work  
abc1    conf.txt feathers may    Pictures ski.places  
australia Desktop file.txt monthly play  Templates  
bin     Documents fun     Music  Public  Videos  
  
(nomun@Zorigoo)-[~]  
$ rm logfile  
  
(nomun@Zorigoo)-[~]  
$ ls  
ab      bon      Downloads may    Pictures ski.places  
abc1    conf.txt feathers monthly play  Templates  
australia Desktop file.txt Music  Public  Videos  
bin     Documents fun     my_os  reports work
```

Рис. 10: удаление logfile

Запускаю из консоли в фоновом режиме редактор gedit указывая &:

```
(nomun@Zorigoo)-[~]  
$ gedit &  
[2] 3356
```

Рис. 11: запуск gedit в фоновом режиме

Используя команду ps, конвейер и фильтр grep, определяю идентификатор процесса gedit (3576):

```

(nomun@Zorigoo)-[~]
$ ps aux | grep gedit
nomun      3135  0.0  0.1 32648 20224 pts/0    TN   21:23   0:00 /usr/bin/pyth
on3 /usr/lib/command-not-found -- gedit
nomun      3356  1.6  0.5 705184 60964 pts/0    SNl  21:23   0:00 gedit
nomun      3421  0.0  0.0  6344  2048 pts/0    S+   21:24   0:00 grep --color=
auto gedit

```

Рис. 12: идентификатор процесса gedit

```

(nomun@Zorigoo)-[~]
$ ps aux | grep gedit | grep -v grep
nomun      3135  0.0  0.1 32648 20224 pts/0    TN   21:23   0:00 /usr/bin/pyth
on3 /usr/lib/command-not-found -- gedit
nomun      3356  0.9  0.5 705184 60964 pts/0    SNl  21:23   0:00 gedit

```

Рис. 13: Другой способ нахождения идентификатора процесса

С помощью man прочитала справку команды kill и использую её для завершения процесса gedit:

```

(nomun@Zorigoo)-[~]
$ man kill

(nomun@Zorigoo)-[~]
$ kill 3356

```

Рис. 14: завершения процесса gedit

С помощью man прочитала справку команд df и du:

```

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 cur-
    rently 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.

```

Рис. 15: справка команды 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.
```

Рис. 16: справка команды du

Используя df -vi я вывожу информацию об инодах и вижу сколько свободного места у моей системы:

```
(nomun@Zorigoo)-[~]
$ df -vi
Filesystem      Inodes   IUsed   IFree IUse% Mounted on
udev            1327758    427 1327331    1% /dev
tmpfs           1338299    751 1337548    1% /run
/dev/sda1       2965504 739714 2225790   25% /
tmpfs           1338299      1 1338298    1% /dev/shm
tmpfs           1338299      3 1338296    1% /run/lock
tmpfs           267659    163 267496    1% /run/user/1000
```

Рис. 17: df -vi

Используя du -a вижу сколько места занимают файлы в директории Загрузки:

```
(nomun@Zorigoo)-[~]
$ du -a Downloads
33992 Downloads/pandoc-3.1.12.3-linux-arm64.tar.gz
```

Рис. 18: du -a

Воспользовавшись справкой команды find и аргумент d, вывожу всех директорий, имеющих в домашнем каталоге:

```
(nomun@Zorigoo)-[~]  
$ find ~ -type d  
/home/nomun
```

Рис. 19: Поиск директорий

```
(nomun@Zorigoo)-[~]  
$ find ~ -type d  
/home/nomun  
/home/nomun/Pictures  
/home/nomun/Pictures/Screenshots  
/home/nomun/reports  
/home/nomun/reports/monthly  
/home/nomun/reports/monthly/monthly  
/home/nomun/Desktop  
/home/nomun/bin  
/home/nomun/bin/pandoc-3.1.12.3-linux-arm64  
/home/nomun/bin/pandoc-3.1.12.3-linux-arm64/pandoc-3.1.12.3  
/home/nomun/bin/pandoc-3.1.12.3-linux-arm64/pandoc-3.1.12.3/share
```

Рис. 20: результаты `find ~ -type d`

4 Выводы

При выполнении данной работы я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных. Также приобрела практические навыки по управлению процессами и по проверке использования диска по обслуживанию файловых систем.

5 Ответы на контрольные вопросы

1. `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
 - Перенаправление вывода (`stdout`) в файл “filename”, `>>` файл открывается в режиме добавления.
2. Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передается последующей.
3. Программа - это набор инструкций, который позволяет ЦПУ выполнять определенную задачу, в то время как процесс - это исполняемая программа.

4. PPID - (parent process ID) идентификатор родительского процесса. Процесс может порождать и другие процессы. UID, GID - реальные идентификаторы пользователя и его группы, запустившего данный процесс.
5. Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач.
6. 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.
7. Команда find - это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, группам, типу, размеру и другим подобным критериям. Утилита find предустановлена по умолчанию во всех Linux дистрибутивах. Команда find имеет такой синтаксис: find [папка] [параметры] критерий шаблон [действие] Пример: find /etc -name "p*" -print
8. find / -type f -exec grep -H 'текстДляПоиска' {} ;
9. df -h.
10. du -s.
11. kill% номер задачи.