

Отчёт по лабораторной работе №8

Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов

Даваасурэн Цэгцтур

Содержание

1	Цель работы	1
2	Задание.....	1
3	Выполнение лабораторной работы	2
4	Выводы.....	9
5	Ответы на контрольные вопросы	10

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` :

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

Рис. 1: Запись в файл

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

```
(tsegtstur@Zorigoo)-[~]  
$ head file.txt  
/etc:  
total 1636  
-rw-r--r-- 1 root root 3386 Jun 27 2023 adduser.conf  
drwxr-xr-x 3 root root 4096 Mar 23 14:26 alsa  
drwxr-xr-x 2 root root 20480 Mar 23 18:23 alternatives  
drwxr-xr-x 8 root root 4096 Mar 23 14:35 apache2  
-rw-r--r-- 1 root root 433 Aug 23 2020 apg.conf  
drwxr-xr-x 2 root root 4096 Mar 23 14:19 apparmor  
drwxr-xr-x 9 root root 4096 Mar 23 14:36 apparmor.d  
drwxr-xr-x 8 root root 4096 Mar 23 14:40 apt  
  
(tsegtstur@Zorigoo)-[~]  
$
```

Рис. 2: Первые 8 файлов в file.txt

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

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

(tsegtstur@Zorigoo)-[~]
$
```

Рис. 3: Добавление файлов из домашнего каталога

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

```
(tsegtstur@Zorigoo)-[~]
$ grep .config file.txt
-rw-r--r-- 1 root root 5230 Jan 11 13:59 manpath.config
-rw-r--r-- 1 root root 767 Nov 13 02:45 netconfig
lrwxrwxrwx 1 root root 69 Jan 31 13:58 blas.pc-x86_64-linux-gnu -> /usr/lib/x86_64-linux-gnu/openblas-
pthread/pkgconfig/blas-openblas.pc
lrwxrwxrwx 1 root root 71 Jan 31 13:58 lapack.pc-x86_64-linux-gnu -> /usr/lib/x86_64-linux-gnu/openbla
s-pthread/pkgconfig/lapack-openblas.pc
lrwxrwxrwx 1 root root 64 Jan 31 13:58 openblas_config.h-x86_64-linux-gnu -> /usr/include/x86_64-linux
-gnu/openblas-pthread/openblas_config.h
lrwxrwxrwx 1 root root 64 Jan 31 13:58 openblas.pc-x86_64-linux-gnu -> /usr/lib/x86_64-linux-gnu/openb
las-pthread/pkgconfig/openblas.pc
-rw-r--r-- 1 root root 1752 Jul 20 2023 im-config
lrwxrwxrwx 1 root root 55 Mar 23 14:32 10-hinting-slight.conf -> /usr/share/fontsconfig/conf.avail/10-h
inting-slight.conf
lrwxrwxrwx 1 root root 59 Jan 6 03:33 10-scale-bitmap-fonts.conf -> /usr/share/fontsconfig/conf.avail/
10-scale-bitmap-fonts.conf
lrwxrwxrwx 1 root root 55 Mar 23 14:32 10-sub-pixel-none.conf -> /usr/share/fontsconfig/conf.avail/10-s
ub-pixel-none.conf
lrwxrwxrwx 1 root root 54 Jan 6 03:33 10-yes-antialias.conf -> /usr/share/fontsconfig/conf.avail/10-ye
s-antialias.conf
```

Рис. 4: файлы с расширением .conf

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

```
(tsegtstur@Zorigoo)-[~]
$ grep .config file.txt
-rw-r--r-- 1 root root 5230 Jan 11 13:59 manpath.config
-rw-r--r-- 1 root root 767 Nov 13 02:45 netconfig
lrwxrwxrwx 1 root root 69 Jan 31 13:58 blas.pc-x86_64-linux-gnu -> /usr/lib/x86_64-linux-gnu/openblas-
pthread/pkgconfig/blas-openblas.pc
lrwxrwxrwx 1 root root 71 Jan 31 13:58 lapack.pc-x86_64-linux-gnu -> /usr/lib/x86_64-linux-gnu/openbla
s-pthread/pkgconfig/lapack-openblas.pc
lrwxrwxrwx 1 root root 64 Jan 31 13:58 openblas_config.h-x86_64-linux-gnu -> /usr/include/x86_64-linux
-gnu/openblas-pthread/openblas_config.h
lrwxrwxrwx 1 root root 64 Jan 31 13:58 openblas.pc-x86_64-linux-gnu -> /usr/lib/x86_64-linux-gnu/openb
las-pthread/pkgconfig/openblas.pc
-rw-r--r-- 1 root root 1752 Jul 20 2023 im-config
lrwxrwxrwx 1 root root 55 Mar 23 14:32 10-hinting-slight.conf -> /usr/share/fontconfig/conf.avail/10-h
inting-slight.conf
lrwxrwxrwx 1 root root 59 Jan 6 03:33 10-scale-bitmap-fonts.conf -> /usr/share/fontconfig/conf.avail/
10-scale-bitmap-fonts.conf
lrwxrwxrwx 1 root root 55 Mar 23 14:32 10-sub-pixel-none.conf -> /usr/share/fontconfig/conf.avail/10-s
ub-pixel-none.conf
lrwxrwxrwx 1 root root 54 Jan 6 03:33 10-yes-antialias.conf -> /usr/share/fontconfig/conf.avail/10-ye
s-antialias.conf
```

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:

```
$ head conf.txt
-rw-r--r-- 1 root root 5230 Jan 11 13:59 manpath.config
-rw-r--r-- 1 root root 767 Nov 13 02:45 netconfig
lrwxrwxrwx 1 root root 69 Jan 31 13:58 blas.pc-x86_64-linux-gnu -> /usr/lib/x86_64-linux-gnu/openblas-
pthread/pkgconfig/blas-openblas.pc
lrwxrwxrwx 1 root root 71 Jan 31 13:58 lapack.pc-x86_64-linux-gnu -> /usr/lib/x86_64-linux-gnu/openbla
s-pthread/pkgconfig/lapack-openblas.pc
lrwxrwxrwx 1 root root 64 Jan 31 13:58 openblas_config.h-x86_64-linux-gnu -> /usr/include/x86_64-linux
-gnu/openblas-pthread/openblas_config.h
lrwxrwxrwx 1 root root 64 Jan 31 13:58 openblas.pc-x86_64-linux-gnu -> /usr/lib/x86_64-linux-gnu/openb
las-pthread/pkgconfig/openblas.pc
-rw-r--r-- 1 root root 1752 Jul 20 2023 im-config
lrwxrwxrwx 1 root root 55 Mar 23 14:32 10-hinting-slight.conf -> /usr/share/fontconfig/conf.avail/10-h
inting-slight.conf
lrwxrwxrwx 1 root root 59 Jan 6 03:33 10-scale-bitmap-fonts.conf -> /usr/share/fontconfig/conf.avail/
10-scale-bitmap-fonts.conf
lrwxrwxrwx 1 root root 55 Mar 23 14:32 10-sub-pixel-none.conf -> /usr/share/fontconfig/conf.avail/10-s
ub-pixel-none.conf

(tsegtstur@Zorigoo)-[~]
$
```

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

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

```
(tsegtstur@Zorigoo)-[~]
$ ls -lR | grep c*
-rw-r--r-- 1 tsegtstur tsegtstur 8378 Mar 28 19:22 conf.txt

(tsegtstur@Zorigoo)-[~]
$
```

Рис. 7: поиск файла используя grep

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

```
(tsegtstur@Zorigoo)-[~]
$ sudo find /etc -name "h*" -print
/etc/alternatives/html2markdown
/etc/alternatives/html2markdown.1.gz
/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
/etc/apache2/mods-available/heartmonitor.load
/etc/apache2/mods-available/heartmonitor.load
```

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:

```
(tsegtstur@Zorigoo)-[~]
$ sudo find ~ -name "log*" -print > logfile &
[1] 3840

(tsegtstur@Zorigoo)-[~]
$

[1]+  Done                  sudo find ~ -name "log*" -print > logfile
(tsegtstur@Zorigoo)-[~]
$
```

Рис. 9: Создание фонового режима

Удаляю созданный logfile и проверяю:

```
(tsegtstur@Zorigoo)-[~]
$ ls
abc1      Desktop  Downloads  file.txt  may      Music    Public  ski.places  Videos
conf.txt  Documents  equipment  logfile  monthly  Pictures  reports  Templates

(tsegtstur@Zorigoo)-[~]
$ cm logfile
cm: command not found

(tsegtstur@Zorigoo)-[~]
$ rm logfile

(tsegtstur@Zorigoo)-[~]
$ ls
abc1      Desktop  Downloads  file.txt  monthly  Pictures  reports  Templates
conf.txt  Documents  equipment  may      Music    Public    ski.places  Videos

(tsegtstur@Zorigoo)-[~]
$
```

Рис. 10: удаление logfile

Запускаю из консоли в фоновом режиме редактор gedit указывая &:

```
(tsegtstur@Zorigoo)-[~]
$ gedit &
[1] 3886
```

Рис. 11: запуск gedit в фоновом режиме

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


```
(tsegtstur@Zorigoo)-[~]
$ ps aux | grep gedit
tsegtst+  3887  0.1  0.1 32568 20096 pts/0    T   19:26   0:00 /usr/bin/python3 /usr/lib/command-no
t-found -- gedit
tsegtst+  3911  0.0  0.0  6344  2176 pts/0    S+  19:27   0:00 grep --color=auto gedit

(tsegtstur@Zorigoo)-[~]
$
```

Рис. 12: идентификатор процесса gedit

```
(tsegtstur@Zorigoo)-[~]
$ ps aux | grep gedit | grep -v grep
tsegtst+  3887  0.0  0.1 32568 20096 pts/0    T   19:26   0:00 /usr/bin/python3 /usr/lib/command-no
t-found -- gedit

(tsegtstur@Zorigoo)-[~]
$
```

Рис. 13: Другой способ нахождения идентификатора процесса

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

```
(tsegtstur@Zorigoo)-[~]
$ man kill

(tsegtstur@Zorigoo)-[~]
$ kill 3576
```

Рис. 14: завершения процесса gedit

С помощью man прочитала справку команд df и du:

```
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 non-portable intimate knowledge of file system structures.

OPTIONS
    Show information about the file system on which each FILE resides, or all file systems by default.
    Manual page df(1) line 1 (press h for help or q to quit)
```

Рис. 15: справка команды df

```

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

```

Рис. 16: справка команды `du`

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

```

(tsegtstur@Zorigoo)-[~]
$ df -vi
Filesystem      Inodes    IUsed    IFree IUse% Mounted on
udev            1327851    427    1327424    1% /dev
tmpfs           1338297    755    1337542    1% /run
/dev/sda1       2965504  732248  2233256   25% /
tmpfs           1338297     1    1338296    1% /dev/shm
tmpfs           1338297     2    1338295    1% /run/lock
tmpfs           267659    162    267497    1% /run/user/1001

(tsegtstur@Zorigoo)-[~]
$

```

Рис. 17: `df -vi`

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

```

(tsegtstur@Zorigoo)-[~]
$ du -a Downloads
4      Downloads

(tsegtstur@Zorigoo)-[~]
$

```

Рис. 18: `du -a`

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


```
(tsegtstur@Zorigoo)-[~]
$ find ~ -type d
/home/tsegtstur
/home/tsegtstur/Pictures
/home/tsegtstur/Pictures/Screenshots
/home/tsegtstur/reports
/home/tsegtstur/reports/monthly
/home/tsegtstur/reports/monthly/monthly
/home/tsegtstur/Desktop
/home/tsegtstur/Public
/home/tsegtstur/Downloads
/home/tsegtstur/equipment
/home/tsegtstur/Videos
/home/tsegtstur/.local
/home/tsegtstur/.local/share
/home/tsegtstur/.local/share/evolution
/home/tsegtstur/.local/share/evolution/calendar
/home/tsegtstur/.local/share/evolution/calendar/system
```

Рис. 19: Поиск директорий

```
/home/tsegtstur/.config/evolution
/home/tsegtstur/.config/evolution/sources
/home/tsegtstur/.config/gnome-session
/home/tsegtstur/.config/gnome-session/saved-session
/home/tsegtstur/.config/gtk-3.0
/home/tsegtstur/.config/goa-1.0
/home/tsegtstur/.config/dconf
/home/tsegtstur/.config/nautilus
/home/tsegtstur/.config/Thunar
/home/tsegtstur/.config/pulse
/home/tsegtstur/.config/powershell
/home/tsegtstur/.config/cherrytree
/home/tsegtstur/.config/ibus
/home/tsegtstur/.config/ibus/bus
/home/tsegtstur/.config/enchant
/home/tsegtstur/.config/xfce4
/home/tsegtstur/.config/xfce4/xfconf
/home/tsegtstur/.config/xfce4/xfconf/xfce-perchannel-xml
/home/tsegtstur/.config/xfce4/panel
/home/tsegtstur/ski.places
/home/tsegtstur/Templates

(tsegtstur@Zorigoo)-[~]
$
```

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.