

Презентация по лабораторной работе №6

Операционные системы

Скрипникова София Дмитриевна, НММбд-03-22

17 марта 2023

Российский университет дружбы народов, Москва, Россия

Цель работы

Ознакомиться с инструментами поиска файлов и фильтрации текстовых данных. Приобрести практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

1. Осуществили вход в систему, используя наше имя.
2. Далее запишем в файл *file.txt* названия файлов, содержащихся в каталоге */etc*. Для этого используем команду *ls -a /etc >file.txt*. С помощью команды *ls -a ~ - > file.txt* дописываем в этот же файл названия файлов, содержащихся в домашнем каталоге. Для проверки действий используем команду *cat file.txt*.

Запись в файл

3. Нужно вывести имена всех файлов из *file.txt*, которые имеют расширение *.conf* и записать их в новый текстовый файл *conf.txt*. Для этого используем команду *grep -e 'conf\$' file.txt > conf.txt*. Проверяем выполнение действий.

Вывод файлов

4. Затем найдём файлы в домашнем каталоге, которые начинаются на с. Это можно сделать несколькими командами, которые представлены на рисунке.

```
[sdskrpnikova@sdskrpnikova ~]$ find ~ -maxdepth 1 -name "c*" -print
/home/sdskrpnikova/cd ~
/home/sdskrpnikova/cd ~/.pub
/home/sdskrpnikova/conf.txt
[sdskrpnikova@sdskrpnikova ~]$ ls ~/c*
'/home/sdskrpnikova/cd ~'      /home/sdskrpnikova/conf.txt
'/home/sdskrpnikova/cd ~/.pub'
[sdskrpnikova@sdskrpnikova ~]$ ls | grep c*
```

Рис. 1: Нахождение файлов по символу

5. После этого выведем на экран (по странично) имена файлов из каталога */etc*, которые начинаются с символа *h*. Для этого я использовала команду `* find /etc -maxdepth1 -name "h" | less`.

```
/etc/hotplug.d  
/etc/hal  
/etc/hostname  
/etc/highlight  
/etc/harbour  
/etc/hosts  
/etc/hotplug  
/etc/htdig  
/etc/harbour.cfg  
/etc/hsqldb  
/etc/hosts.allow
```

6. Запустим в фоновом режиме процесс, который будет записывать в файл `~/logfile`, файлы, которые начинаются с `log` с помощью команды `find / -name "log" > logfile&»*`. Запустился непрерывный процесс записывания файла.

```
find: '/etc/nftables': Отказано в доступе
find: '/etc/openvpn/client': Отказано в доступе
find: '/etc/openvpn/server': Отказано в доступе
find: '/etc/polkit-1/localauthority': Отказано в доступе
find: '/etc/polkit-1/rules.d': Отказано в доступе
find: '/etc/sos/cleaner': Отказано в доступе
find: '/etc/ssh/ssh_config.d': Отказано в доступе
find: '/etc/ssh/sshd_config.d': Отказано в доступе
find: '/etc/sudoers.d': Отказано в доступе
find: '/lost+found': Отказано в доступе
find: '/usr/libexec/initscripts/legacy-actions/auditd': Отказано в доступе
```

Рис. 3: Нахождение файлов по символам

7. Проверим наличие файла *logfile*, а затем с помощью команды *rm logfile* удалим его.

```
[sdscripnikova@sdscripnikova ~]$ rm logfile  
[sdscripnikova@sdscripnikova ~]$ rm logfile  
rm: невозможно удалить 'logfile': Нет такого файла или каталога
```

Рис. 4: Удаление файла

8. Запускаем в консоли в фоновом режиме редактор *gedit*. После ввода команды *gedit &* появляется окно редактора.

```
[sdscripnikova@sdscripnikova ~]$ gedit &  
[1] 3465
```

Рис. 5: Редактор gedit

9. Для определения идентификатора процесса *gedit* используем команду *ps | grep -i "gedit"*.
Из рисунка видно, что наш процесс имеет PID 4507.

```
[sdskrpnikova@sdskrpnikova ~]$ gedit &
[1] 3465
[sdskrpnikova@sdskrpnikova ~]$ gedit &
[2] 3497
[1]   Завершён      gedit
[sdskrpnikova@sdskrpnikova ~]$ ps |grep -i "gedit"
[2]+   Завершён      gedit
[sdskrpnikova@sdskrpnikova ~]$ pgrep gedit
[sdskrpnikova@sdskrpnikova ~]$ pidof gedit
[sdskrpnikova@sdskrpnikova ~]$
```

Рис. 6: Определение идентификатора процесса

10. Далее ознакомимся со справкой команды *kill* и используем её для завершения процесса *gedit*.

```
KILL(1)                                User Commands                                KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds
    signal] [--] pid|name...

    kill -l [number] | -L

DESCRIPTION
    The command kill sends the specified signal to the specified processes
    or process groups.

    If no signal is specified, the TERM signal is sent. The default action
    for this signal is to terminate the process. This signal should be used
    in preference to the KILL signal (number 9), since a process may install
    a handler for the TERM signal in order to perform clean-up steps before
    terminating in an orderly fashion. If a process does not terminate after
    a TERM signal has been sent, then the KILL signal may be used; be aware
    that the latter signal cannot be caught, and so does not give the target
    process the opportunity to perform any clean-up before terminating.

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

Рис. 7: Опции команды kill

```
[sds-kripnikova@sds-kripnikova ~]$ man kill
[sds-kripnikova@sds-kripnikova ~]$ kill 3497
```

Рис. 8: Завершение процесса

11. Далее получим более подробную информацию о командах *df* и *du*.

- *df*– утилита, показывающая список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования.
- *du* – утилита, предназначенная для вывода информации об объеме дискового пространства, занятого файлами и директориями. Она принимает путь к элементу файловой системы и выводит информацию о количестве байт дискового пространства или блоков диска, задействованных для его хранения

```
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
    [Manual page df(1) 1400 1 (press h for help or q to quit)]
```

```

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

```

Рис. 10: Опции команды du

```

[sdskripnikova@sdskripnikova ~]$ df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs          4096            0      4096            0% /dev
tmpfs             1008740        13000    995740            2% /dev/shm
tmpfs             403500        1392    402108            1% /run
/dev/sda2         82835456    22776652  57623668           29% /
tmpfs             1008744         24    1008720            1% /tmp
/dev/sda2         82835456    22776652  57623668           29% /home
/dev/sda1         996780        230280    697688           25% /boot
tmpfs             201748         252    201496            1% /run/user/1000
/dev/sr0          62390        62390         0          100% /run/media/sdskripnikova/VBox_GAs_6.1.38
sdskripnikova@sdskripnikova ~$

```

Рис. 11: Команда df

```
12 ./sds-kripnikova.github.io/.git/objects
4 ./sds-kripnikova.github.io/.git/logs/refs/heads
4 ./sds-kripnikova.github.io/.git/logs/refs/remotes/origin
4 ./sds-kripnikova.github.io/.git/logs/refs/remotes
8 ./sds-kripnikova.github.io/.git/logs/refs
12 ./sds-kripnikova.github.io/.git/logs
116 ./sds-kripnikova.github.io/.git
116 ./sds-kripnikova.github.io
0 ./letters
0 ./memos
0 ./misk
8 ./ski.places/equipment
0 ./ski.places/plans
8 ./ski.places
du: невозможно получить доступ к './play/file.old': Отказано в доступе
0 ./play
```

Рис. 12: Команда du

12. Выведем имена всех директорий, которые имеются в домашнем каталоге, предварительно узнаем опции команды *find*.

```
FIND(1)                                General Commands Manual          FIND(1)

NAME

    find - search for files in a directory hierarchy

SYNOPSIS

    find [-H] [-L] [-P] [-D debugopts] [-O level] [starting-point...] [ex-
    pression]

DESCRIPTION

    This manual page documents the GNU version of find. GNU find searches
    the directory tree rooted at each given starting-point by evaluating the
    given expression from left to right, according to the rules of prece-
    dence (see section OPERATORS), until the outcome is known (the left hand
    side is false for and operations, true for or), at which point find
    moves on to the next file name. If no starting-point is specified, .'
    is assumed.

    If you are using find in an environment where security is important (for
    example if you are using it to search directories that are writable by
    other users), you should read the 'Security Considerations' chapter of
    the findutils documentation, which is called Finding Files and comes
    with findutils. That document also includes a lot more detail and dis-
    cussion than this manual page, so you may find it a more useful source
    of information.

Manual page find(1) 14m 1 (press h for help or q to quit)
```

Рис. 13: Опции команды *find*

```
./sdskrpnikova.github.io/.git/description
./sdskrpnikova.github.io/.git/refs
./sdskrpnikova.github.io/.git/refs/heads
./sdskrpnikova.github.io/.git/refs/heads/main
./sdskrpnikova.github.io/.git/refs/tags
./sdskrpnikova.github.io/.git/refs/remotes
./sdskrpnikova.github.io/.git/refs/remotes/origin
./sdskrpnikova.github.io/.git/refs/remotes/origin/main
./sdskrpnikova.github.io/.git/objects
./sdskrpnikova.github.io/.git/objects/pack
./sdskrpnikova.github.io/.git/objects/info
./sdskrpnikova.github.io/.git/objects/e6
./sdskrpnikova.github.io/.git/objects/e6/9de29bb2d1d6434b8b29ae775ad8c2e48c5391
./sdskrpnikova.github.io/.git/objects/f9
./sdskrpnikova.github.io/.git/objects/f9/3e3a1a1525fb5b91020da86e44810c87a2d7bc
./sdskrpnikova.github.io/.git/objects/f9
```


Выводы

В ходе выполнения данной лабораторной работы я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных, а также приобрела практические навыки по управлению процессами, по проверке использования диска и обслуживанию файловых систем.