

Лабораторная работа №6: Презентация.

Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов.

Евдокимов Максим Михайлович. Группа - НФИбд-01-20.¹

25 декабря, 2023, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи 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.

1. Изучить основы работы с процессами на базовом уровне.
2. Научиться работать с методами поиска и фильтрации данных.
3. Научиться работать с записью данных и проверкой диска и обслуживанием файловой системы.

Указание к работе

Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей (action1 | action2).

Конвейеры можно группировать в цепочки и выводить с помощью перенаправления в файл, например `"ls -la | sort > sortilg_list"`.

Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода stdout. Например, команда `ls` выводит в стандартный поток вывода (консоль) список файлов в текущей директории. Потоки вывода и ввода можно перенаправлять на другие файлы или устройства. Проще всего это делается с помощью символов `>`, `»`, `<`, `«`. Рассмотрим пример.

Команда `find` используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов. Путь определяет каталог, начиная с которого по всем подкаталогам будет вестись поиск.

Процесс выполнения лабораторной работы

1. Осуществите вход в систему, используя соответствующее имя пользователя.

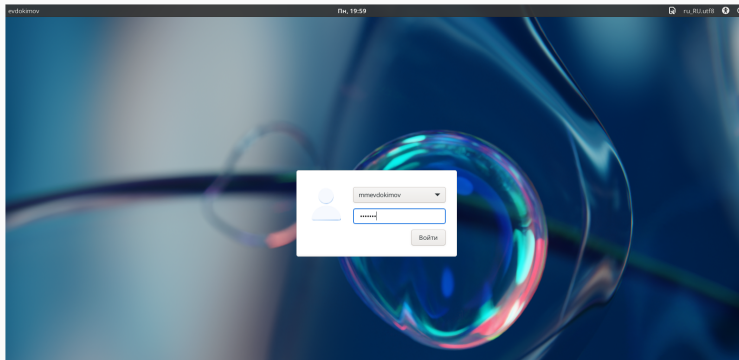


Рис. 1: Вход в систему

2. Запишите в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.

```
van@evdokiiev:~$ touch file.txt; ls /etc > file.txt; cat file.txt
abrt
adjtime
aliases
alsa
alternatives
anaconda
anacrontab
anthy-unicode.conf
appstream.conf
asound.conf
at.deny
audit
authselect
avahi
bash_completion.d
bashrc
bindresvport.blacklist
binfmt.d
bluetooth
chromium
chrony.conf
cifs-utils
credstore
credstore.encrypted
cron.d
cron.daily
cron.deny
cron.hourly
cron.monthly
crontab
cron.weekly
crypto-policies
crypttab
csh.cshrc
csh.login
```

Рис. 2: Создание и заполнение файла названиями из /etc

```
Irolltech.conf
trusted-key.key
ts.conf
udev
udisks2
unbound
updatedb.conf
UPower
usb_modeswitch.conf
vconsole.conf
vdpau_wrapper.cfg
virc
vpl
vulkan
wgetrc
whois.conf
wireplumber
wpa_supplicant
X11
xattr.conf
xdg
xml
yum.repos.d
sax@evdokiMOV:~$ ls >> file.txt; tail file.txt
ski_places
times
Видео
Документы
Загрузки
Изображения
Музыка
Общедоступные
Рабочий стол
Шаблоны
sax@evdokiMOV:~$
```

Рис. 3: Результаты добавления новых данных в файл

3. Выведите имена всех файлов из file.txt, имеющих расширение .conf, после чего запишите их в новый текстовый файл conf.txt.

```
max@evdokimov:~$ grep .conf file.txt
anthy-unicode.conf
appstream.conf
asound.conf
chrony.conf
dconf
dnsmasq.conf
dracut.conf
dracut.conf.d
fprintd.conf
fuse.conf
host.conf
i3status.conf
idmapd.conf
kdump.conf
krb5.conf
krb5.conf.d
ld.so.conf
ld.so.conf.d
libaudit.conf
libuser.conf
locale.conf
logrotate.conf
makedumpfile.conf.sample
man_db.conf
mke2fs.conf
mtools.conf
```

Рис. 4: Поиск и определение файлов с текстом conf

```
max@evdokimov:~$ cat file.txt | grep ".conf" > conf.txt
max@evdokimov:~$ head conf.txt
anthy-unicode.conf
appstream.conf
asound.conf
chrony.conf
dconf
dnsmasq.conf
dracut.conf
dracut.conf.d
fprintd.conf
fuse.conf
max@evdokimov:~$
```

Рис. 5: Создание, заполнение и вывод первых элементов файла conf.txt

4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа с? Предложите несколько вариантов, как это сделать.

```
max@evdokimov:~$ find ~ -name "c*" -print
/home/max/.config/i3/config
/home/max/conf.txt
max@evdokimov:~$ ls ~ | grep c*
conf.txt
max@evdokimov:~$
```

Рис. 6: Результат поиска файлов с символом “с”

5. Выведите на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h.

```
max@evdokimov:~$ sudo find /etc -name "h*" -print
[sudo] пароль для max:
/etc/avahi/hosts
/etc/firewalld/helpers
/etc/libibverbs.d/hfi1verbs.driver
/etc/libibverbs.d/hns.driver
/etc/systemd/homed.conf
/etc/udev/hwdb.d
/etc/udev/hwdb.bin
/etc/host.conf
/etc/hosts
/etc/hostname
max@evdokimov:~$
```

Рис. 7: Результат поиска файлов с символом "h" в каталоге /etc

6. Запустите в фоновом режиме процесс, который будет записывать в файл ~/logfile.txt файлы, имена которых начинаются с log.

```
max@evdokimov:~$ find /etc -name "log*" -print > ~/logfile.txt &
[1] 2568
find: '/etc/audit': Отказано в доступе
find: '/etc/credstore.encrypted': Отказано в доступе
find: '/etc/credstore': Отказано в доступе
find: '/etc/cups/ssl': Отказано в доступе
find: '/etc/dhcp': Отказано в доступе
find: max@evdokimov:~$ '/etc/firewalld': Отказано в доступе
find: '/etc/grub.d': Отказано в доступе
find: '/etc/lvm/archive': Отказано в доступе
find: '/etc/lvm/backup': Отказано в доступе
find: '/etc/lvm/cache': Отказано в доступе
find: '/etc/lvm/devices': Отказано в доступе
find: '/etc/nftables': Отказано в доступе
find: '/etc/pki/rsyslog': 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/sshd_config.d': Отказано в доступе
find: '/etc/sssd': Отказано в доступе
find: '/etc/sudoers.d': Отказано в доступе
find /etc -name "q
bash: q: команда не найдена
[1]+  Выход 1      find /etc -name "log*" -print > ~/logfile.txt
max@evdokimov:~$ sudo find /etc -name "log*" -print > ~/logfile.txt &
[1] 2581
max@evdokimov:~$ sudo find /etc -name "log*" -print > ~/logfile.txt &
[2] 2589
[1] Завершён      sudo find /etc -name "log*" -print > ~/logfile.txt
```

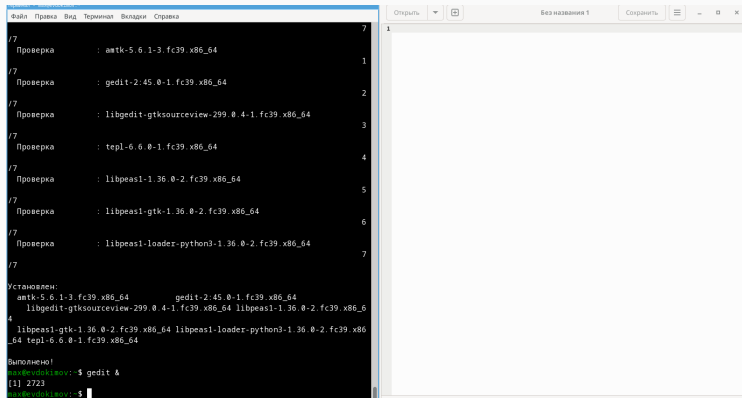
Рис. 8: Запуск процесса по поиску файлов с текстом "log"

7. Удалите файл ~/logfile

```
max@evdokimov:~$ rm ~/logfile.txt  
[2]+  Завершён      sudo find /etc -name "log*" -print > ~/logfile.txt  
max@evdokimov:~$
```

Рис. 9: Удаление, завершение процесса logfile

8. Запустите из консоли в фоновом режиме редактор gedit.



The screenshot shows a terminal window with a dark background and a light blue border. The terminal output is as follows:

```
/7
Проверка      : amtk-5.6.1-3.fc39.x86_64
/7
Проверка      : gedit-2:45.0-1.fc39.x86_64
/7
Проверка      : libgedit-gtksourceview-299.0.4-1.fc39.x86_64
/7
Проверка      : tepl-6.6.0-1.fc39.x86_64
/7
Проверка      : libpeas1-1.36.0-2.fc39.x86_64
/7
Проверка      : libpeas1-gtk-1.36.0-2.fc39.x86_64
/7
Проверка      : libpeas1-loader-python3-1.36.0-2.fc39.x86_64
/7

Установлен:
  amtk-5.6.1-3.fc39.x86_64      gedit-2:45.0-1.fc39.x86_64
  libgedit-gtksourceview-299.0.4-1.fc39.x86_64 libpeas1-1.36.0-2.fc39.x86_6
4
  libpeas1-gtk-1.36.0-2.fc39.x86_64 libpeas1-loader-python3-1.36.0-2.fc39.x86
_64 tepl-6.6.0-1.fc39.x86_64

Выполнено!
root@bevdokisov: ~$ gedit &
[1] 2723
root@bevdokisov: ~$
```

To the right of the terminal window, a portion of the gedit editor interface is visible. It shows a menu bar with 'Открыть', 'Сохранить', and window control buttons. Below the menu bar, the text 'Без названия 1' is displayed, followed by a large, empty white area representing the text editor's workspace.

Рис. 10: Запуск процесса gedit

9. Определите идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Как ещё можно определить идентификатор процесса?

```
max@evdokimov:~$ ps aux | grep "gedit"
max      2723  4.9  1.8 775568 58116 pts/0    Sl   20:14   0:03 gedit
max      2782  0.0  0.0 222456  2304 pts/0    S+   20:15   0:00 grep --col
or=auto gedit
max@evdokimov:~$ pidof gedit
2723
max@evdokimov:~$ ps -fc gedit
UID      PID    PPID  C  STIME TTY          TIME CMD
max      2723    2447  4  20:14 pts/0      00:00:03 gedit
max@evdokimov:~$
```

Рис. 11: Определение идентификатора процесса gedit разными способами

10. Прочтите справку (man) команды `kill`, после чего используйте 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] [--] pidname...

    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.

    Most modern shells have a builtin kill command, with a usage rather
    similar to that of the command described here. The --all, --pid, and
    --queue options, and the possibility to specify processes by command
    name, are local extensions.

    If signal is 0, then no actual signal is sent, but error checking is
    still performed.

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

Рис. 12: Просмотр мануал по команде `kill`

```
max@evdokimov:~$ man kill
max@evdokimov:~$ jobs; kill 2723
[1]+  Запущен          gedit &
[1]+  Завершено        gedit
max@evdokimov:~$
```

Рис. 13: Завершение процесса gedit с помощью kill

11. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.

```
DE(1)                                     User Commands                               DE(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
    Show information about the file system on which each FILE resides, or all file systems by default.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        include pseudo, duplicate, inaccessible file systems

    -B, --block-size=SIZE
        scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576 bytes; see SIZE format below

    --direct
        show statistics for a file instead of mount point

    -h, --human-readable
        print sizes in powers of 1024 (e.g., 1023M)

Manual page df(1) line 1/100 39% (press h for help or q to quit)
```

Рис. 14: Просмотр мануал по команде `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.

    -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 the like

    -B, --block-size=SIZE
        scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576 bytes; see SIZE format below

    -b, --bytes
        equivalent to '--apparent-size --block-size=1'

    -c, --total
        produce a grand total

    -D, --dereference-args
Manual page du(1) line 1 (press h for help or q to quit)
```

Рис. 15: Просмотр мануал по команде du

```

max@evdokimov:~$ man df
max@evdokimov:~$ man du
max@evdokimov:~$ df -vi
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
/dev/sda3          0          0          0          - /
devtmpfs           377862      492      377370          1% /dev
tmpfs              383018       2      383016          1% /dev/shm
tmpfs              819200      826      818374          1% /run
tmpfs              1048576      31     1048545          1% /tmp
/dev/sda2          65536       395      65141          1% /boot
/dev/sda3          0           0           0          - /home
tmpfs              76603       90       76513          1% /run/user/1000
max@evdokimov:~$ df -cat
df: неверный ключ — «с»
По команде «df --help» можно получить дополнительную информацию.
max@evdokimov:~1$ du -a file.txt
4      file.txt
max@evdokimov:~$ █

```

Рис. 16: Примеры использования df и du

12. Воспользовавшись справкой команды `find`, выведите имена всех директорий, имеющих в вашем домашнем каталоге.

```
find(1)                                     General Commands Manual                               find(1)

NAME
    find - search for files in a directory hierarchy

SYNOPSIS
    find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [expression]

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 precedence (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 discussion than this manual page, so you may find it a more useful source of information.

OPTIONS
    The -H, -L and -P options control the treatment of symbolic links. Command-line arguments following these are taken to be names of files or directories to be examined, up to the first argument that begins with '-', or the argument '(' or '!'. That argument and any following arguments are taken to be the expression describing what is to be searched for. If no paths are given, the current directory is used. If no expression is given, the expression -print is used (but you should probably consider using -print0 instead, anyway).

    This manual page talks about 'options' within the expression list. These options control the behaviour of find but are specified immediately after the last path name. The five 'real' options -H, -L, -P, -D and -O must appear before the first path name, if at all. A double dash -- could theoretically be used to signal that any remaining arguments are not options, but this does not really work due to the way find determines the end of the following path arguments: it does that by reading until an expression argument comes (which also starts with a '-'). Now, if a path argument would start with a '-', then find would treat it as expression argument instead. Thus, to ensure that all start points are taken as such, and especially to prevent that wildcard patterns expanded by the calling shell are not mistakenly treated as expression arguments, it is generally safer to prefix wildcards or dubious path names with either './' or to use absolute path names starting with '/'. Alternatively, it is generally safe though non-portable to use the GNU option -files0-from to pass arbitrary starting points to find.

    -P      Never follow symbolic links. This is the default behaviour. When find examines or prints information about files, and the file is a sym-
```

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

Рис. 17: Просмотр мануал по команде `find`


```
max@evdokinov:~$ man find
max@evdokinov:~$ find ~ -type d
/home/max
/home/max/.mozilla
/home/max/.mozilla/extensions
/home/max/.mozilla/plugins
/home/max/.cache
/home/max/.cache/imsettings
/home/max/.cache/abrt
/home/max/.cache/ mesa_shader_cache
/home/max/.config
/home/max/.config/imsettings
/home/max/.config/i3
/home/max/.config/abrt
/home/max/.config/volumeicon
/home/max/.config/dconf
/home/max/.config/xfce4
/home/max/.config/xfce4/xfconf
/home/max/.config/xfce4/xfconf/xfce-perchannel-xml
/home/max/.config/enchant
/home/max/Рабочий стол
/home/max/Загрузки
/home/max/Шаблоны
/home/max/Общедоступные
/home/max/Документы
/home/max/Музыка
/home/max/Изображения
/home/max/Видео
/home/max/.local
/home/max/.local/state
/home/max/.local/state/wireplumber
/home/max/.local/share
/home/max/new
/home/max/times
/home/max/ski_places
/home/max/ski_places/equipment
```

Рис. 18: Вывод всех директорий домашнего каталога

Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?

- `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 ».

>filename - Перенаправление вывода (stdout) в файл "filename". »filename - Перенаправление вывода (stdout) в файл "filename", файл открывается 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. (команда1 | команда2)

4. Что такое процесс? Чем это понятие отличается от программы?

Компьютерная программа сама по себе — лишь пассивная последовательность инструкций. В то время как процесс — непосредственное выполнение этих инструкций.

5. Что такое PID и GID?

Идентификатор процесса (PID). Каждому новому процессу ядро присваивает уникальный идентификационный номер. В любой момент времени идентификатор процесса является уникальным, хотя после завершения процесса он может использоваться снова для другого процесса. Некоторые идентификаторы зарезервированы системой для особых процессов. Так, процесс с идентификатором 1 - это процесс инициализации `init`, являющийся предком всех других процессов в системе. Идентификатор группы GID и эффективный идентификатор группы (EGID) GID - это идентификационный номер группы данного процесса. EGID связан с GID также, как EUID с UID.

6. Что такое задачи и какая команда позволяет ими управлять?

Задачи - это то, что мы подаем на выполнение системе, какой-то процесс, который она начинает выполнять, чтобы проследить за ними можно использовать Команду "jobs".

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

`top` (table of processes) — консольная команда, которая выводит список работающих в системе процессов и информацию о них. По умолчанию она в реальном времени сортирует их по нагрузке на процессор. Программа написана для UNIXсовместимых операционных систем и опубликована под свободной лицензией GNU FDL. `htop` — продвинутый монитор процессов, написанный для Linux. Он был задуман заменить стандартную программу `top`. `Htop` показывает динамический список системных процессов, список обычно выравнивается по использованию ЦПУ. В отличие от `top`, `htop` показывает все процессы в системе. Также показывает время непрерывной работы, использование процессоров и памяти. `Htop` часто применяется в тех случаях, когда информации даваемой утилитой `top` недостаточно, например при поиске утечек памяти в процессах. `Htop` написан на языке Си и использует для отображения библиотеку `Ncurses`.

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

Команда `find` используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов. Формат команды: `find <-опции>`, пример: `find /etc -name "p*" -print`.

9. Можно ли по контексту (содержанию) найти файл? Если да, то как?

Для поиска файла по содержимому проще всего воспользоваться командой “grep”, пример:
“grep -r строка_поиска каталог”.

10. Как определить объем свободной памяти на жёстком диске?

При помощи команды `df` (аббревиатура от `disk free`) — утилита в UNIX и UNIX-подобных системах, показывает список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования.

11. Как определить объем вашего домашнего каталога?

Для этого есть команда “du -a ~”.

12. Как удалить зависший процесс?

Для завершения процесса нужно вызвать утилиту `kill` с параметром `-9` или с параметром `"ping"`.

Выводы по проделанной работе

В ходе выполнения лабораторной работы были изучены и применены на практике базовые команды по работе с процессами, а также с методами и командами поиска и по файловой системе и записи данных в файлы.