

Лабораторная работа 8

Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов

Мочалкина Софья Васильевна

Содержание

Цель работы	1
Задание.....	1
Теоретическое введение	2
Выполнение лабораторной работы	2
Выводы	10

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.

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.

Теоретическое введение

1 # Перенаправление stdout (вывода) в файл. 2 # Если файл отсутствовал, то он создаётся, 3 # иначе – перезаписывается. 4 5 # Создаёт файл, содержащий список дерева каталогов. 6 ls -lR > dir-tree.list 7 8 1>filename 9 # Перенаправление вывода (stdout) в файл "filename". 10 1>>filename 11 # Перенаправление вывода (stdout) в файл "filename", 12 # файл открывается в режиме добавления. 13 2>filename 14 # Перенаправление stderr в файл "filename". 15 2>>filename 16 # Перенаправление stderr в файл "filename", 17 # файл открывается в режиме добавления. 18 &>filename 19 # Перенаправление stdout и stderr в файл "filename"

Выполнение лабораторной работы

- 1) Осуществляю вход в систему, используя соответствующее имя пользователя.
- 2) Записываю в файл file.txt названия файлов, содержащихся в каталоге /etc. Дописываю в этот же файл названия файлов, содержащихся в домашнем каталоге.

```
[svmochalkina@vbox ~]$ ls /etc > file.txt
[svmochalkina@vbox ~]$ ls ~ >> file.txt
[svmochalkina@vbox ~]$
```

рис.1

- 3) Вывожу имена всех файлов из file.txt, имеющих расширение .conf, после чего записываю их в новый текстовый файл conf.txt.

```
[svmochalkina@vbox ~]$ grep '\.conf$' file.txt > conf.txt
[svmochalkina@vbox ~]$
```

рис.2

- 4) Определяю, какие файлы в домашнем каталоге имеют имена, начинавшиеся с символа с.

```

[svmochalkina@vbox ~]$ find ~ -name 'c*'
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/crashes
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/compatibility.ini
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/cookies.sqlite
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/cert9.db
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/security_state/crlite.filter
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/security_state/crlite.coverage
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/security_state/crlite.enrollment
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/permanent/chrome
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++www.youtube.com*partitionKey=%28https%2Cgoogle.com%29/cache
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++www.youtube.com*partitionKey=%28https%2Cgoogle.com%29/cache/caches.sqlite
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++www.youtube.com*partitionKey=%28https%2Cgoogle.com%29/cache/caches.sqlite-wal
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++web.telegram.org/cache
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++web.telegram.org/cache/caches.sqlite
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++web.telegram.org/cache/caches.sqlite-wal
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++web.telegram.org/cache/context_open.marker
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++www.youtube.com*partitionKey=%28https%2Cfedoraproject.org%29/cache
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++www.youtube.com*partitionKey=%28https%2Cfedoraproject.org%29/cache/caches.sqlite
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++www.youtube.com*partitionKey=%28https%2Cfedoraproject.org%29/cache/caches.sqlite-wal
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++products.aspose.app/cache
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++products.aspose.app/cache/caches.sqlite
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++products.aspose.app/cache/caches.sqlite-wal
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++www.googletagmanager.com*partitionKey=%28https%2Czamazar.com%29/cache
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++www.googletagmanager.com*partitionKey=%28https%2Czamazar.com%29/cache/caches.sqlite
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++www.googletagmanager.com*partitionKey=%28https%2Czamazar.com%29/cache/caches.sqlite-wal
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++www.googletagmanager.com*partitionKey=%28https%2Csmallpdf.com%29/cache
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++www.googletagmanager.com*partitionKey=%28https%2Csmallpdf.com%29/cache/caches.sqlite
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++www.googletagmanager.com*partitionKey=%28https%2Csmallpdf.com%29/cache/caches.sqlite-wal
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/storage/default/https+++www.googletagmanager.com*partitionKey=%28https%2Csmallpdf.com%29/cache/context_open.marker
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/content-prefs.sqlite
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/cookies.sqlite-wal
/home/svmochalkina/.mozilla/firefox/m43mn3nn.default-release/containers.json
/home/svmochalkina/.cache/mozilla/firefox/m43mn3nn.default-release/cache2
/home/svmochalkina/.cache/mozilla/firefox/m43mn3nn.default-release/safebrowsing/content-track-digest256.sbstore
/home/svmochalkina/.cache/mozilla/firefox/m43mn3nn.default-release/safebrowsing/content-track-digest256.vlpset
/home/svmochalkina/.cache/mozilla/firefox/m43mn3nn.default-release/safebrowsing/content-email-track-digest256.sbstore
/home/svmochalkina/.cache/mozilla/firefox/m43mn3nn.default-release/safebrowsing/content-email-track-digest256.vlpset
/home/svmochalkina/.cache/fontconfig/c9937410b59c5398b7630047d01f86f1e64.cache-9

```

рис.3

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

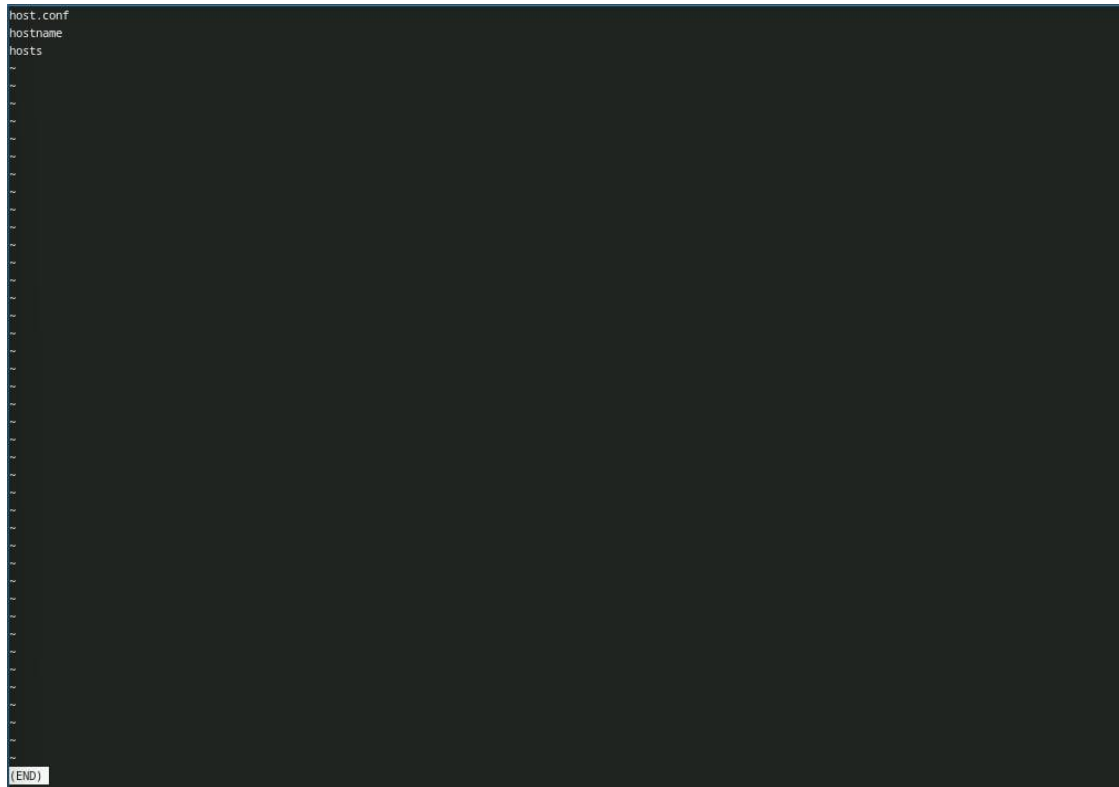


рис.4

```
[svmochalkina@vbox ~]$ ls /etc | grep '^h' | less
[svmochalkina@vbox ~]$
```

рис.5

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

```
[svmochalkina@vbox ~]$ find / -name 'log*' >> ~/logfile
```

рис.6

- 7) Удаляю файл ~/logfile.

```
[svmochalkina@vbox ~]$ rm ~/logfile
```

рис.7

- 8) Запускаю из консоли в фоновом режиме редактор gedit.

```
[svmochalkina@vbox ~]$ ps aux | grep gedit
svmocha+ 1648 2.3 1.3 774916 60552 pts/0 S1 19:09 0:01 gedit
svmocha+ 1664 0.0 0.0 230364 2420 pts/0 S+ 19:10 0:00 grep --color=auto
[svmochalkina@vbox ~]$
```

рис.8

- 9) Определяю идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Можно также использовать pgrep 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.

    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
    The list of processes to be signaled can be a mixture of names and PIDs.

    pid
        Each pid can be expressed in one of the following ways:

        n
            where n is larger than 0. The process with PID n is signaled.

        0
            All processes in the current process group are signaled.

        -1
            All processes with a PID larger than 1 are signaled.

        -n
            where n is larger than 1. All processes in process group n are
    Manual page kill(1) line 1 (press h for help or q to quit)
```

рис.9

- 10) Читаю справку (man) команды kill, после чего использую её для завершения процесса gedit.


```
[svmochalkina@vbox ~]$ man kill
[svmochalkina@vbox ~]$ kill 1648
[svmochalkina@vbox ~]$
```

рис.10

```
[svmochalkina@vbox ~]$ man df
```

рис.11

- 11) Выполняю 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 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.

  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)

  -H, --si
      print sizes in powers of 1000 (e.g., 1.1G)

  -i, --inodes
      list inode information instead of block usage

  -k
      like --block-size=1K

  -l, --local
      limit listing to local file systems

  --no-sync
      do not invoke sync before getting usage info (default)

  --output=FIELD LIST
      Manual page df(1) line 1 (press h for help or q to quit)
```

рис.12

```
[svmochalkina@vbox ~]$ df -h
Файловая система  Размер  Использовано  Дост  Использовано%  Смонтировано в
/dev/sda3          14G       9,5G       4,1G       71% /
devtmpfs           4,0M         0       4,0M         0% /dev
tmpfs              2,2G       3,4M       2,2G         1% /dev/shm
tmpfs              893M       1,2M       892M         1% /run
tmpfs              1,0M         0       1,0M         0% /run/credentials/systemd-journald.service
tmpfs              1,0M         0       1,0M         0% /run/credentials/systemd-network-generator.service
tmpfs              1,0M         0       1,0M         0% /run/credentials/systemd-udev-load-credentials.service
tmpfs              1,0M         0       1,0M         0% /run/credentials/systemd-tmpfiles-setup-dev-early.service
tmpfs              1,0M         0       1,0M         0% /run/credentials/systemd-sysctl.service
tmpfs              1,0M         0       1,0M         0% /run/credentials/systemd-tmpfiles-setup-dev.service
tmpfs              2,2G       4,0K       2,2G         1% /tmp
tmpfs              1,0M         0       1,0M         0% /run/credentials/systemd-vconsole-setup.service
/dev/sda3          14G       9,5G       4,1G       71% /home
/dev/sda2          974M      323M       584M       36% /boot
tmpfs              1,0M         0       1,0M         0% /run/credentials/systemd-tmpfiles-setup.service
tmpfs              1,0M         0       1,0M         0% /run/credentials/systemd-resolved.service
tmpfs              447M       96K       447M         1% /run/user/1000
[svmochalkina@vbox ~]$
```

рис.13

```
[svmochalkina@vbox ~]$ man du
```

pic.14

```
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
      dereference only symlinks that are listed on the command line

  -d, --max-depth=N
      print the total for a directory (or file, with --all) only if it is N or fewer levels below the command line argument; --max-depth=2 is the same as --summarize

  --files0-from=F
      summarize device usage of the NUL-terminated file names specified in file F; if F is -, then read names from standard input

  -H
      equivalent to --dereference-args (-D)

  -h, --human-readable
      print sizes in human readable format (e.g., 1K 234M 2G)

  --inodes
      print inode counts instead of block usage

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

pic.15

pic.16

pic.16

```

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 symbolic link, the information used shall be taken from the properties of the symbolic link itself.

    -L Follow symbolic links. When find examines or prints information about files, the information used shall be taken from the properties of the file to which the link points, not from the link itself (unless it is a broken symbolic link or find is unable to examine the file to which the link points). Use of this option implies -noleaf. If you later use the -P option, -noleaf will still be in effect. If -L is in effect and find discovers a symbolic link to a subdirectory during its search, the subdirectory pointed to by the symbolic link will be searched.

    When the -L option is in effect, the -type predicate will always match against the type of the file that a symbolic link points to rather than the link itself (unless the symbolic link is broken). Actions that can cause symbolic links to become broken while find is executing (for example -delete) can give rise to confusing behaviour. Using -L causes the -lname and -ilname predicates always to return false.

    -H Do not follow symbolic links, except while processing the command line arguments. When find examines or prints information about files, the information used shall be taken from the properties of the symbolic link itself. The only exception to this behaviour is when a file specified on the command line is a symbolic link, and the link can be resolved. For that situation, the information used is taken from whatever the link points to (that is, the link is followed). The information about the link itself is used as a fallback if the file pointed to by the symbolic link cannot be examined. If -H is in effect and one of the paths specified on the command line is a symbolic link to a directory, the contents of that directory will be examined (though of course -maxdepth 0 would prevent this).

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

рис.17

12) Воспользовавшись справкой команды **find**, выведу имена всех директорий, имеющих в домашнем каталоге.

```
[svmochalkina@vbox ~]$ find ~ -type d
```

рис.18

Ответы на контрольные вопросы:

1) Потоки ввода-вывода:

- Стандартный ввод (stdin): Обычно это клавиатура. Используется для ввода данных в программу.
- Стандартный вывод (stdout): Обычно это экран. Используется для вывода данных из программы.
- Стандартный поток ошибок (stderr): Также выводит сообщения об ошибках, но отдельно от стандартного вывода.

2) Разница между операцией **>** и **>>**:

- **>**: Перенаправляет стандартный вывод команды в файл, заменяя его содержимое. Если файл не существует, он будет создан.
- **>>**: Перенаправляет стандартный вывод команды в файл, добавляя данные в конец файла. Если файл не существует, он будет создан.

- 3) Конвейер: Конвейер (pipe) — это механизм, который позволяет передавать вывод одной команды в качестве ввода для другой команды. В Unix-подобных системах это реализуется с помощью символа |. Например: `ls -l | grep "txt"`.
- 4) Процесс: Процесс — это выполняемая программа, которая имеет свое собственное пространство памяти и системные ресурсы. Программа — это статический набор инструкций, тогда как процесс — это динамическое состояние выполнения этих инструкций. Процесс может состоять из одной или нескольких программ.
- 5) PID и GID:
 - PID (Process ID): Уникальный идентификатор процесса, который используется операционной системой для управления процессами.
 - GID (Group ID): Идентификатор группы, к которой принадлежит процесс. Он используется для управления правами доступа к ресурсам.
- 6) Задачи: Задачи — это процессы, которые выполняются в системе. Команда `ps` позволяет просматривать текущие задачи, а команда `kill` позволяет управлять ими (например, завершать).
- 7) Утилиты `top` и `htop`:
 - `top`: Команда, которая показывает текущие активные процессы и их использование ресурсов в реальном времени. Она предоставляет информацию о загрузке процессора, использовании памяти и т.д.
 - `htop`: Улучшенная версия `top`, которая имеет более удобный интерфейс, позволяет прокручивать список процессов и управлять ими более интуитивно.
- 8) Команда поиска файлов:
 - `find`: Команда для поиска файлов и каталогов в файловой системе по различным критериям (имя, размер, дата и т.д.).
 - Пример: `find /path/to/search -name "*.txt"` — ищет все текстовые файлы в указанном каталоге.
- 9) Поиск файла по контексту: Да, можно найти файл по содержимому с помощью команды `grep`. Например:
 - `grep -r "нужный текст" /path/to/search` — ищет текст в файлах в указанном каталоге и подкаталогах.
- 10) Определение объема свободной памяти на жёстком диске: Команда `df -h` показывает информацию о файловых системах, включая объем свободного места на диске.
- 11) Определение объема домашнего каталога: Команда `du -sh ~` показывает общий объем всех файлов и подкаталогов в вашем домашнем каталоге.

- 12) Удаление зависшего процесса: Для удаления зависшего процесса можно использовать команду `kill` с PID процесса. Если процесс не реагирует, можно использовать `kill -9 PID` для принудительного завершения.

Выводы

Я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных. Приобрела практических навыков: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.