

Лабораторная работа №8

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

Пакавира Арсениу

Нкабд-04-23

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	16
6	Контрольные вопросы	17

Список литературы	20
-------------------	----

Список иллюстраций

4.1 Запись названий файлов из каталога /etc в файл file.txt	7
4.2 Запись названий файлов из домашнего каталога в файл file.txt	7
4.3 Имена файлов с расширением .conf	8
4.4 Запись этих имен в файл conf.txt	8
4.5 Команда find	8
4.6 Команда grep	9
4.7 Имена файлов из каталога /etc, которые начинаются с символа h	9
4.8 Запуск процесса в фоновом режиме	9
4.9 Удаление файла ~/logfile	9
4.10 Запуск редактора gedit в фоновом режиме	10
4.11 Идентификатор процесса gedit	10
4.12 Справка man kill	11
4.13 Команда kill	11
4.14 man df	12
4.15 man du	12
4.16 Команда df -vi	13
4.17 Команда du -a	13

4.18 man find	13
4.19 Команда find -type d	14

Списоктаблиц

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, начинавшиеся с символа `s`? 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 Теоретическое введение

интерфейсе командной строки есть очень полезная возможность перенаправ-

ления (переадресации) ввода и вывода (англ. термин I/O Redirection). Как мы уже заметили, многие программы выводят данные на экран. А ввод данных в терминале осуществляется с клавиатуры. С помощью специальных обозначений можно перенаправить вывод многих команд в файлы или иные устройства вывода (например, распечатать на принтере). То же самое и со вводом информации, вместо ввода данных с клавиатуры, для многих программ можно задать считывание символов из файла. Кроме того, можно даже вывод одной программы передать на ввод другой программе.

[Entr: bash?]

К каждой программе, запускаемой в командной строке, по умолчанию подключено три потока данных:

STDIN (0) — стандартный поток ввода (данные, загружаемые в программу). STDOUT (1) — стандартный поток вывода (данные, которые выводит программа). По умолчанию — терминал. STDERR (2) — стандартный поток вывода диагностических и отладочных сообщений (например, сообщения об ошибках). По умолчанию — терминал.

[Conv: bash?]

Pipe (конвейер) — это однонаправленный канал межпроцессного взаимодействия. Термин был придуман Дугласом Макилроем для командной оболочки Unix и назван по аналогии с трубопроводом. Конвейеры чаще всего используются в shell-скриптах для связи нескольких команд путем перенаправления вывода одной команды (stdout) на вход (stdin) последующей, используя символ конвейера

'|'. [Pipes: bash?]

4 Выполнение лабораторной работы

1. Запишите в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге. (рис. fig. 4.1 ;fig. 4.2).

Рис. 4.3: Имена файлов с расширением .conf

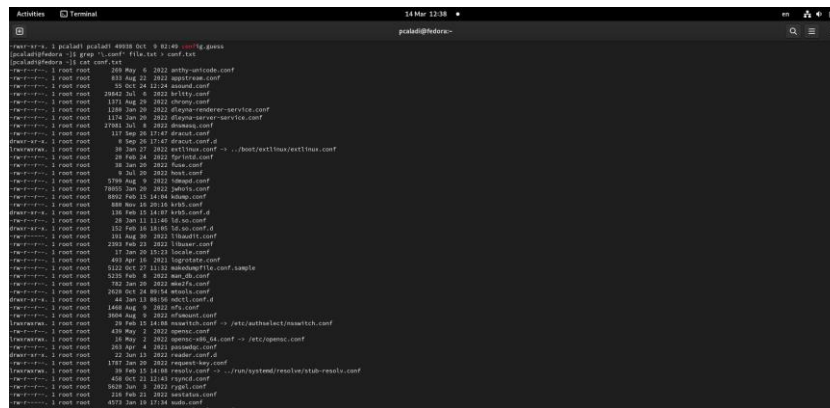


Рис. 4.4: Запись этих имен в файл conf.txt

3. Выведите имена всех файлов из file.txt, имеющих расширение .conf, после чего запишите их в новый текстовый файл conf.txt. (рис. fig. 4.5 ;fig. 4.6).

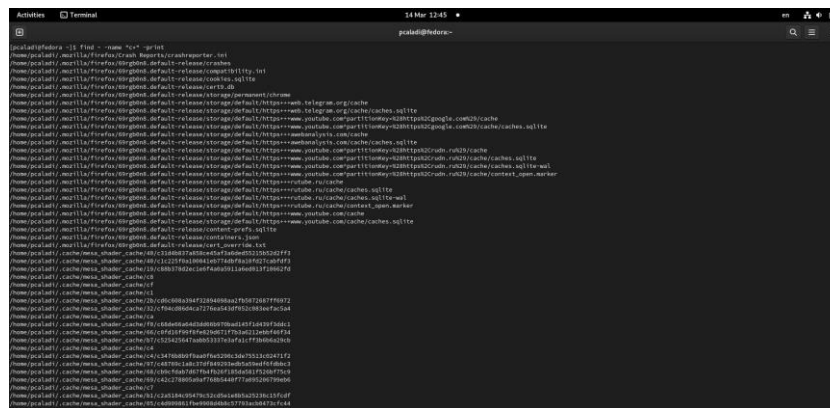


Рис. 4.5: Команда find

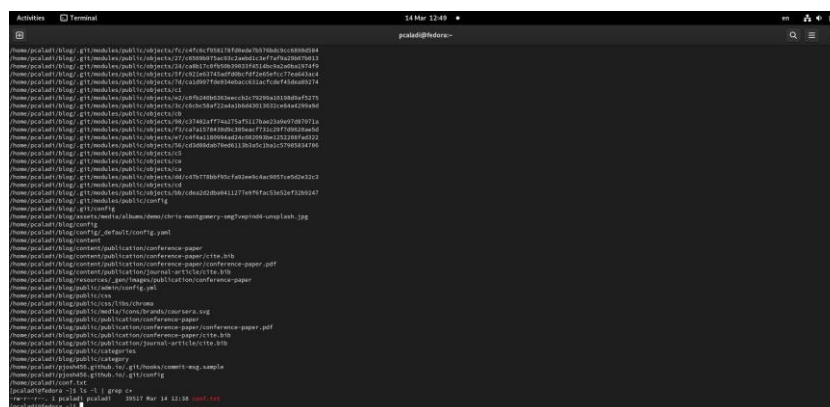


Рис. 4.6: Команда `grep`

4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа с? Предложите несколько вариантов, как это сделать.(рис. fig. 4.7).

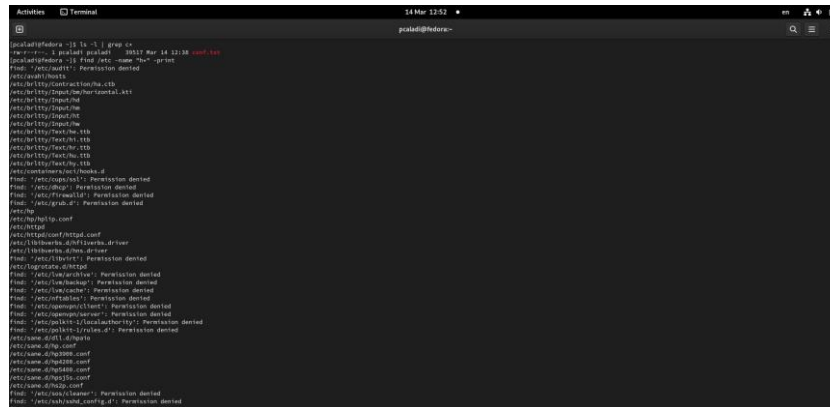


Рис. 4.7: Имена файлов из каталога /etc, которые начинаются с символа h

5. Запустите в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log. (рис. 4.8)

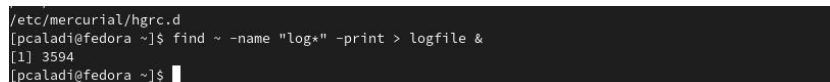


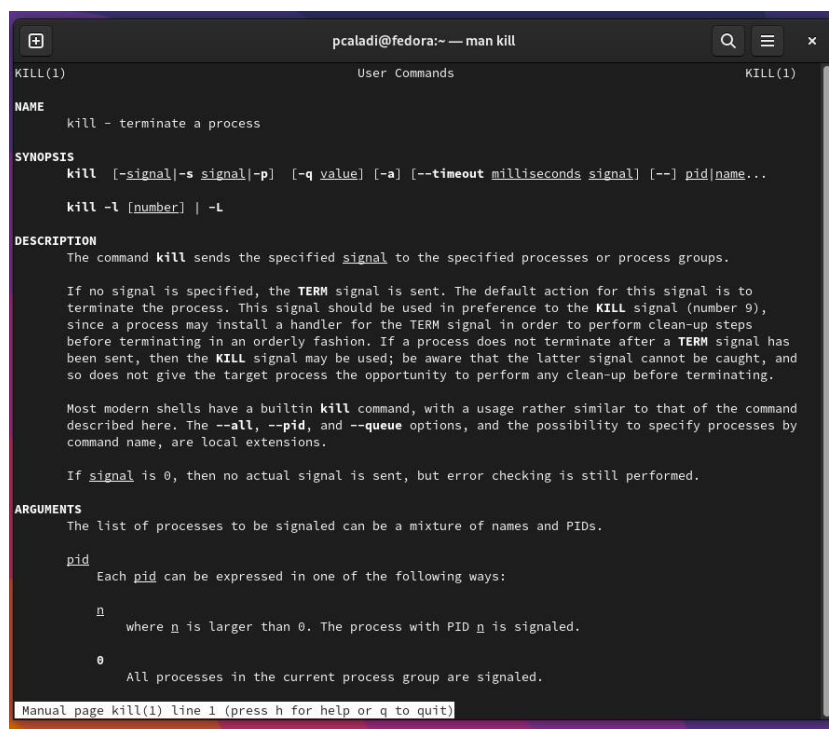
Рис. 4.8: Запуск процесса в фоновом режиме

6. Удалите файл `~/logfile`. (рис. 4.9)



Рис. 4.9: Удаление файла ~/logfile

7. Запустите из консоли в фоновом режиме редактор gedit. (рис. 4.10)



A terminal window showing the man page for the 'kill' command. The window title is 'pcaladi@fedora:~ — man kill'. The content includes sections for NAME, SYNOPSIS, DESCRIPTION, and ARGUMENTS. The SYNOPSIS shows the command syntax: `kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds signal] [--] pid|name...` and `kill -l [number] | -L`. The DESCRIPTION explains that the command sends a signal to specified processes, with `TERM` as the default. The ARGUMENTS section details how to specify processes by PID or name.

```
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.

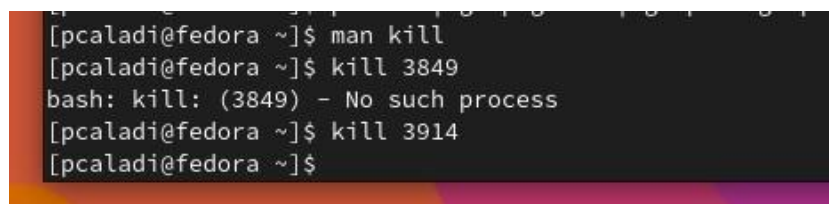
  pid
    Each pid can be expressed in one of the following ways:

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

    0
      All processes in the current process group are signaled.

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

Рис. 4.12: Справка man kill



A terminal window showing the execution of the 'kill' command. The user runs 'man kill' and then 'kill 3849'. The output shows an error: 'bash: kill: (3849) - No such process'. Then the user runs 'kill 3914' and the prompt returns.

```
[pcaladi@fedora ~]$ man kill
[pcaladi@fedora ~]$ kill 3849
bash: kill: (3849) - No such process
[pcaladi@fedora ~]$ kill 3914
[pcaladi@fedora ~]$
```

Рис. 4.13: Команда kill

10. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`. (рис. 4.14, 4.15, 4.16, 4.17)

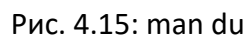
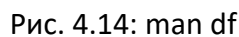


Рис. 4.16: Команда df -vi

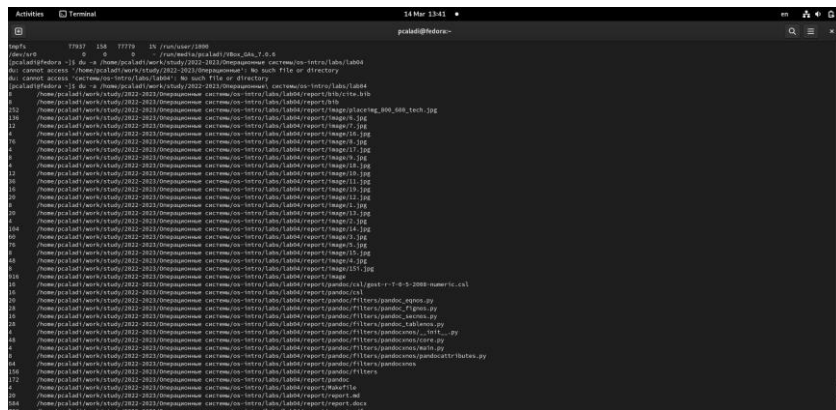


Рис. 4.17: Команда du -a

11. Воспользовавшись справкой команды find, выведите имена всех директорий, имеющих в вашем домашнем каталоге. (рис. 4.18, 4.19)

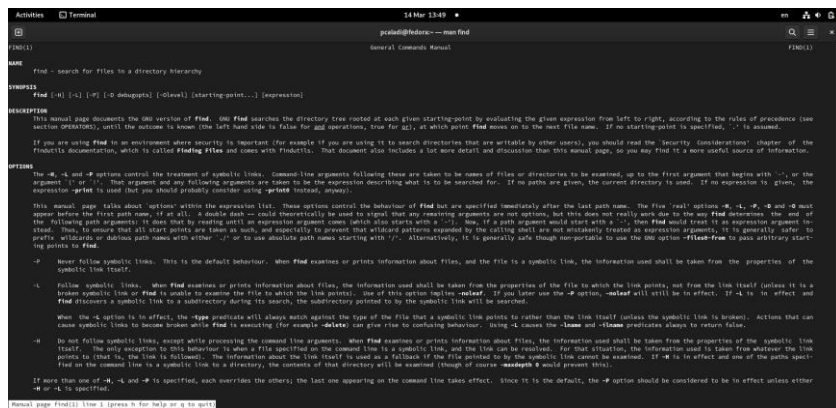


Рис. 4.18: man find

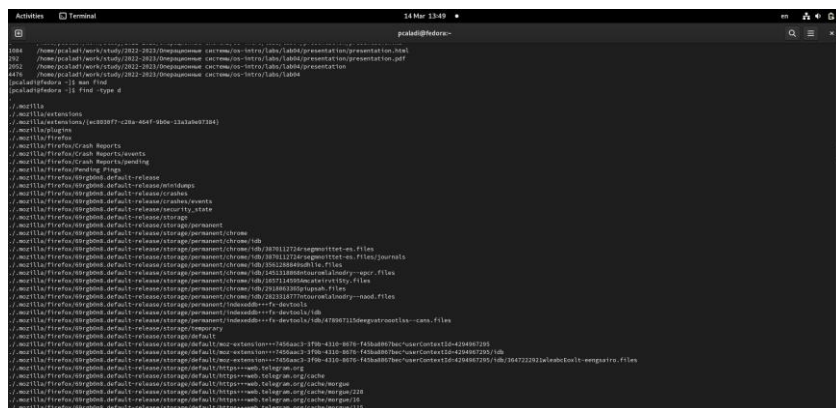


Рис. 4.19: Команда `find -type d`

5 Выводы

В процессе выполнения лабораторной работы ознакомилась с инструментами поиска файлов и фильтрации текстовых данных. Приобрела практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

6 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?

В системе по умолчанию открыто три специальных потока: – `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 `>>`.

Этот знак `>` - перенаправление ввода/вывода, а `>>` - перенаправление в режиме добавления.

3. Что такое конвейер?

Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передается последующей.

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

Главное отличие между программой и процессом заключается в том, что программа - это набор инструкций, который позволяет ЦПУ выполнять определенную задачу, в то время как процесс - это исполняемая программа.

5. Что такое PID и GID?

PPID - (`parent process ID`) идентификатор родительского процесса. Процесс может порождать и другие процессы. UID, GID - реальные идентификаторы пользователя и его группы, запустившего данный процесс.

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

Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач.

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

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.

Зато в `top` можно разделять область окна и выводить информацию о процессах в соответствии с разными настройками. В целом `top` намного более гибкая в настройке отображения процессов.

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

Команда `find` - это одна из наиболее важных и часто используемых утилит системы Linux. Это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям.

Утилита `find` предустановлена по умолчанию во всех Linux дистрибутивах, поэтому вам не нужно будет устанавливать никаких дополнительных пакетов.

Это очень важная находка для тех, кто хочет использовать командную строку наиболее эффективно.

Команда `find` имеет такой синтаксис: `find [папка] [параметры] критерий шаблон [действие]` Пример: `find /etc -name "p*" -print`

9. Можно ли по контексту (содержанию) найти файл? Если да, то как? `find / -type f -`

`exec grep -H 'текстДляПоиска' {} ;`

10. Как определить объем свободной памяти на жёстком диске?

С помощью команды `df -h`.

11. Как определить объем вашего домашнего каталога?

С помощью команды `du -s`.

12. Как удалить зависший процесс?

С помощью команды `kill%` номер задачи.

Список литературы