Отчет по лабораторной работе №6

Дисциплина: операционные системы

Королев Федор Константинович

Содержание

[Цель работы 1](#_Toc102746104)

[Ход работы 1](#_Toc102746105)

[Контрольные вопросы 10](#_Toc102746106)

[Вывод 14](#_Toc102746107)

# 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.

# Ход работы

1. Осуществим вход в систему, исользуя соответствующее имя пользователя
2. Запишим в файл file.txt название файлов, содержащихся в каталоге /etc и в домашнем каталоге(Рис. 1):

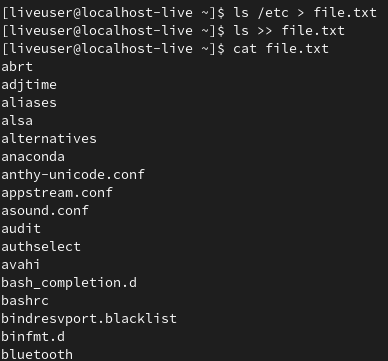


Рис. 1 запись содержимого /etc и домашнего каталога в file.txt

1. Выведем имена всех файлов из file.txt, имеющих расширение .conf, после чего запишим их в новый текстовый файл conf.txt(Рис. 2):

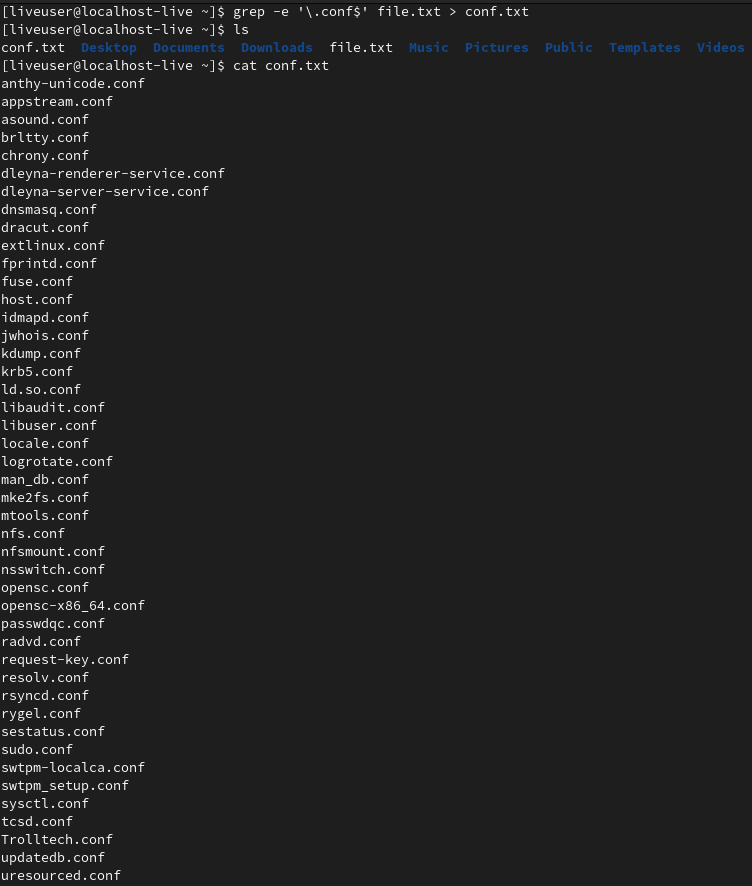


Рис. 2 запись имен из file.txt с расширением .conf в файл conf.txt

1. С помощью команд

ls -a ~ | grep c\*  
  
ls ~/c\*

Определим, какие файлы в домашнем каталоге начинаются с символа с(Рис. 3):

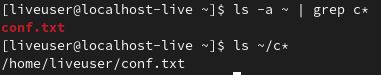


Рис. 3 файлы, начинающиеся с символа с

1. По странично выведем на экран имена файлов и каталога /etc, с помощью команды(Рис. 4 и 5)

find /etc -maxdepth 1 -name "h\*" | less

Рис. 4 команда выполнения

Рис. 4 команда выполнения

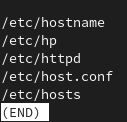


Рис. 5 постраничный вывод

1. Запустим в фоновом режиме процесс, который будет записывать в файл logfile файлы, имена которых начинаются с log, с помощью команды(Рис. 6)

find / -name "log\*" > logfile&

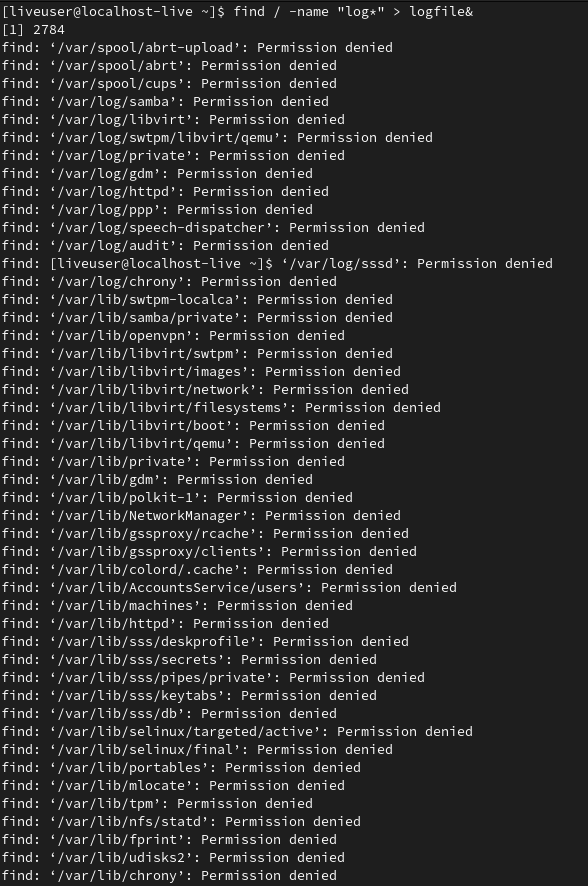


Рис. 6 запуск и исполнение программы в фоновом режиме

1. Удалим файл logfile(Рис. 7):

Рис. 7 удаление файла logfile

Рис. 7 удаление файла logfile

1. Запустим редактор gedit в фоновом режиме, с помощью команды(Рис. 8):

gedit&

Рис. 8 запуск gedit в фоновом режиме

Рис. 8 запуск gedit в фоновом режиме

1. Определим идентификатор процесса с помощью команды(Рис. 9)

ps | grep -i "gedit"

Видно, что он равен 2847. Также, идентификатор процесса можно определить с помощью команд(Рис. 9)

pgrep gedit  
  
pidof gedit

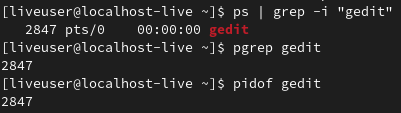


Рис. 9 определение идентификатора gedit

1. Прочитаем справку по команде kill(с помощью man, Рис. 10), далее завершим процесс gidit(Рис. 11):

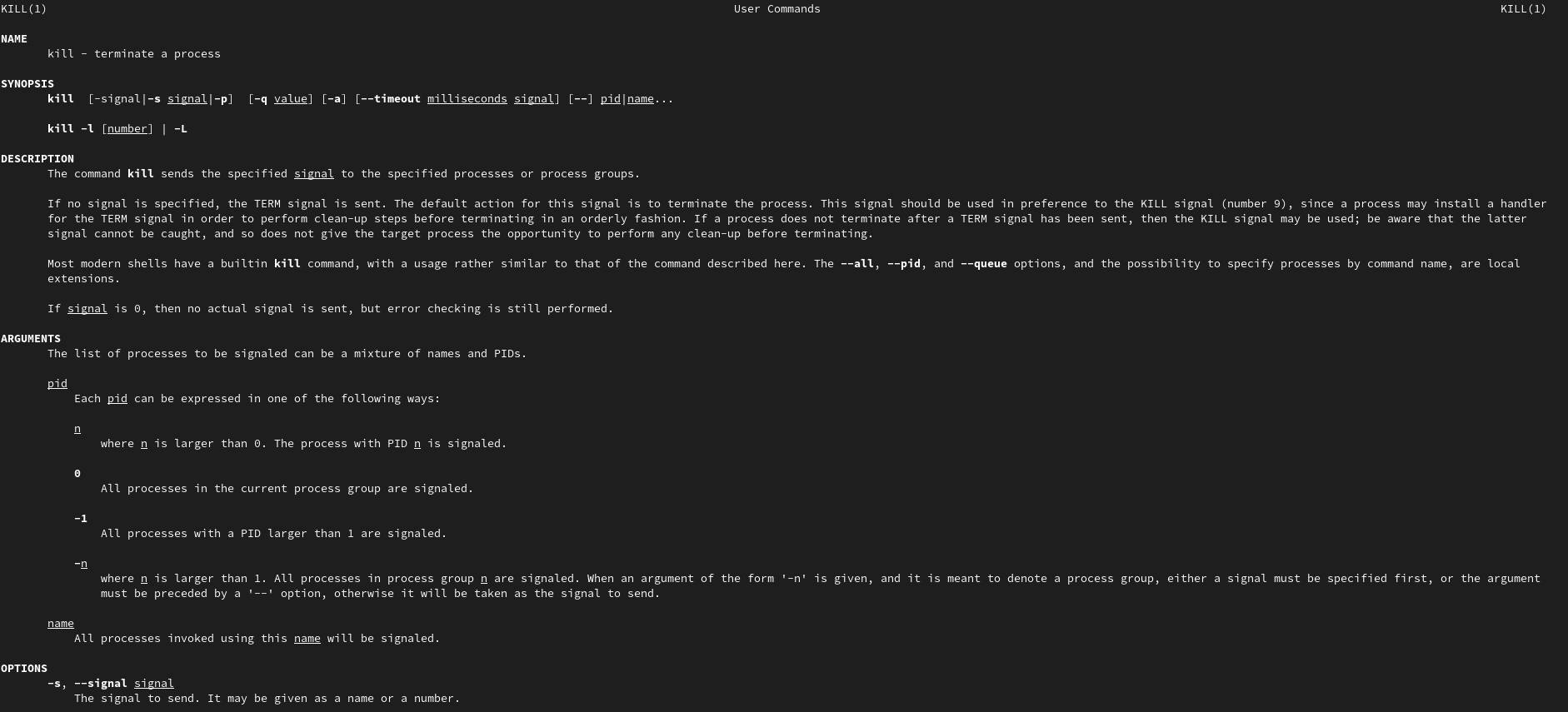


Рис. 10 man kill

Рис. 11 завершение процесса gidit

Рис. 11 завершение процесса gidit

1. Выполним команды df и du, предварительно получив более подробную с помощью man(Рис. 12, 13, 14):

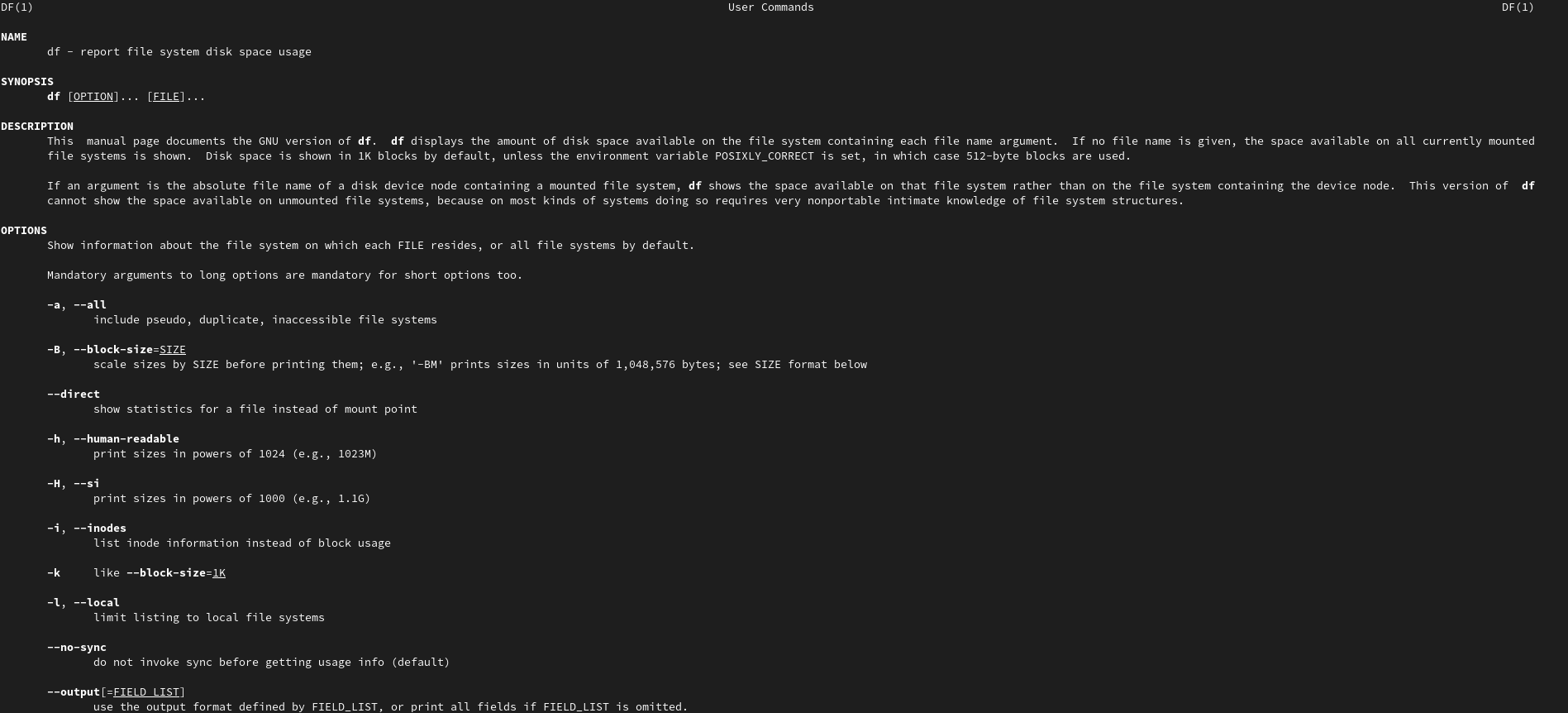


Рис. 12 man df

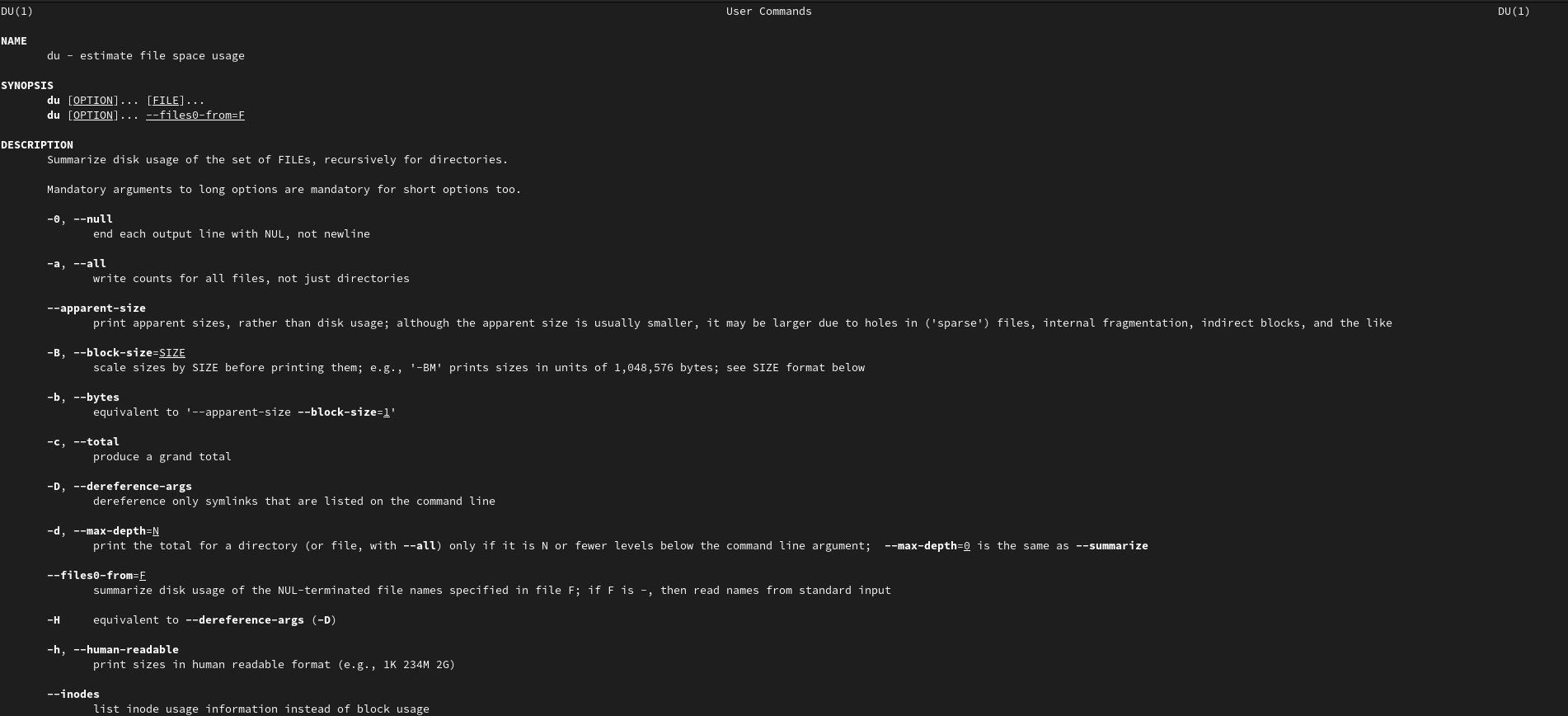


Рис. 13 man du

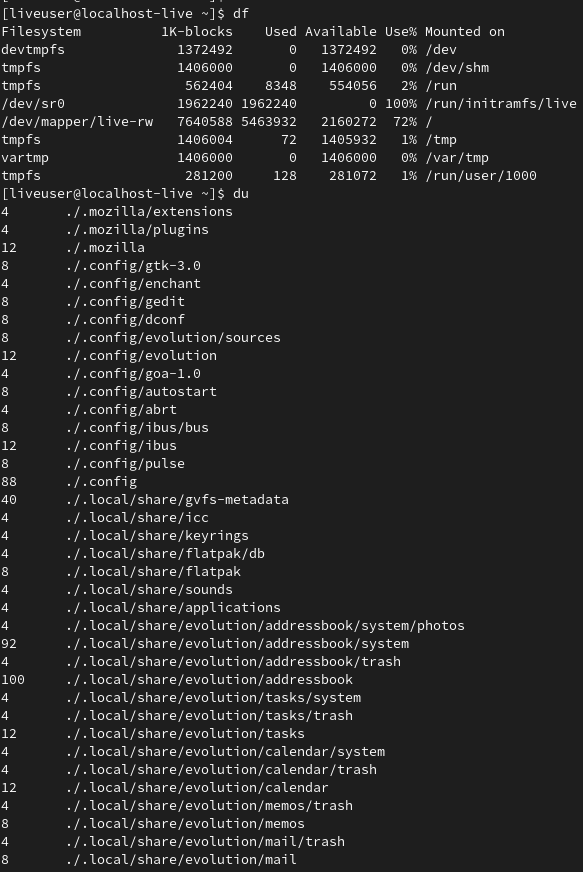


Рис. 14 выполнение df и часть du

1. Выведе имена всех директорий, имеющихся в домашнем каталоге(Рис. 16-18), предварительно посмотрев информацию о find, с помощью команды

find ~ -type d

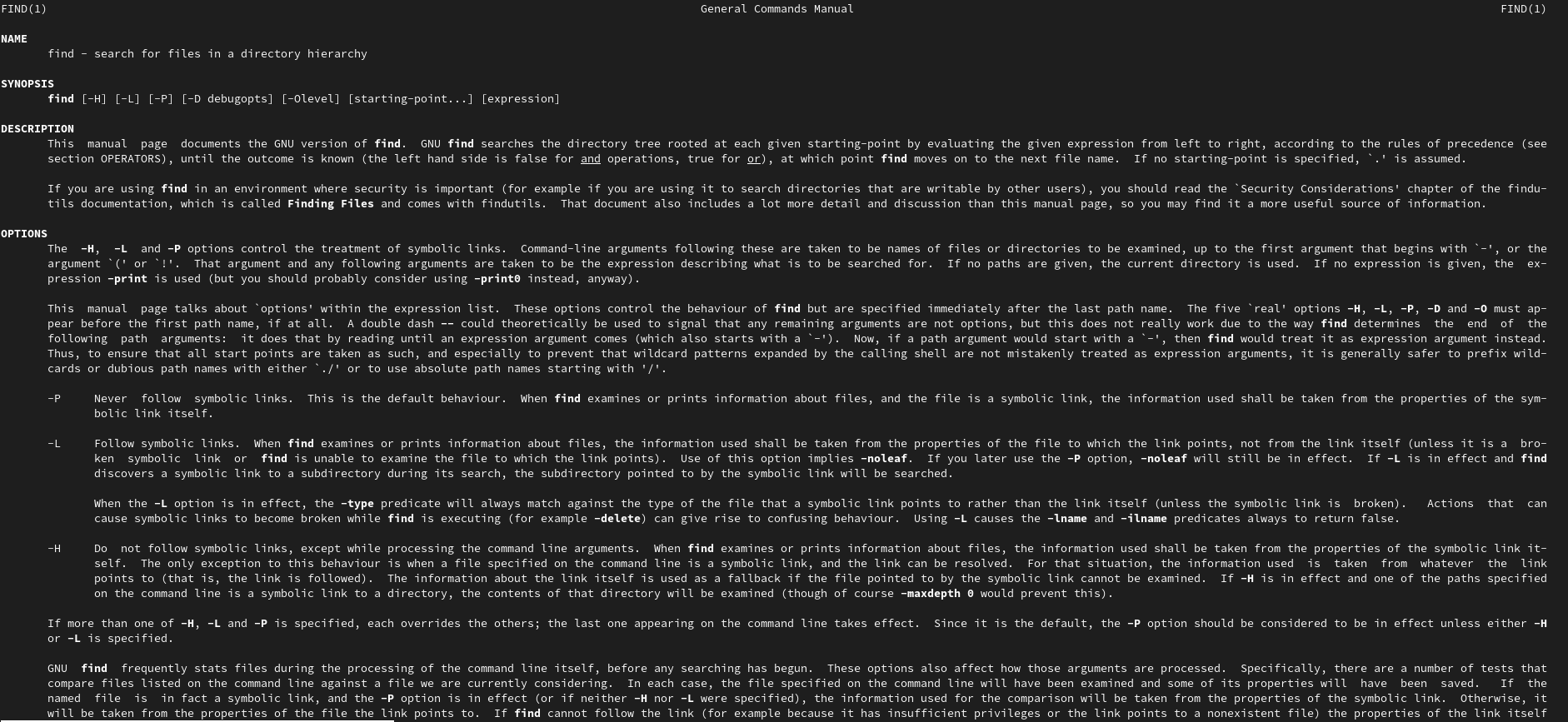
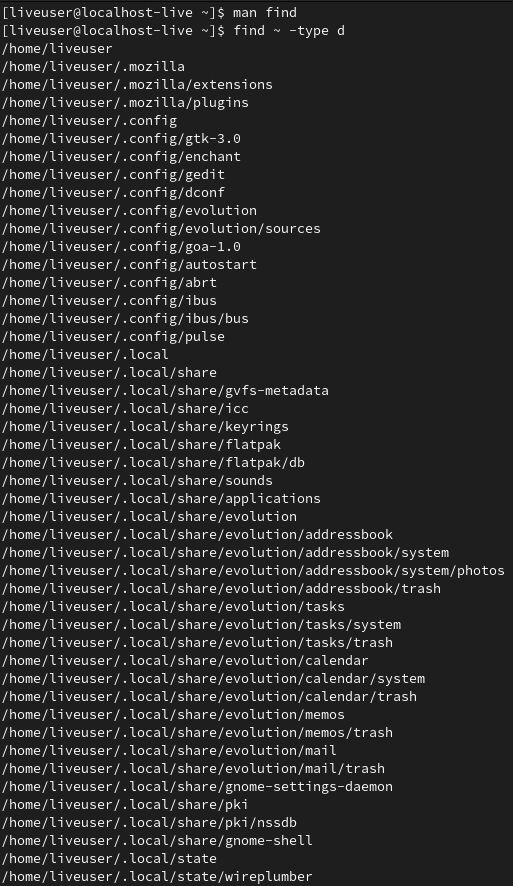
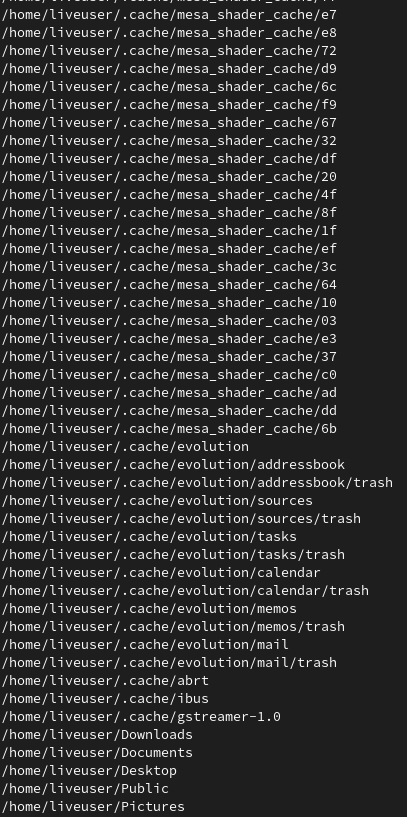


Рис. 15 man find

# Контрольные вопросы

1). 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 (поумолчанию: консоль), файловый дескриптор 2.

Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода stdout.

2). ‘>’ Перенаправление вывода в файл

‘»’ Перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла)/

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

Синтаксис следующий:

команда1|команда2 (это означает, что вывод команды 1 передастся на ввод команде 2)

4). Процесс рассматривается операционной системой как заявка на потребление всех видов ресурсов, кроме одного − процессорного времени. Этот последний важнейший ресурс распределяется операционной системой между другими единицами работы − потоками, которые и получили свое название благодаря тому, что они представляют собой последовательности (потоки выполнения) команд.

Процесс − это выполнение программы. Он считается активной сущностью и реализует действия, указанные в программе.

Программа представляет собой статический набор команд, а процесс это набор ресурсов и данных, использующихся при выполнении программы.

5). pid: идентификатор процесса (PID) процесса (processID), к которому вызывают метод

gid: идентификатор группы UNIX, в котором работает программа.

6). Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда &.

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

7). top − это консольная программа, которая показывает список работающих процессов в системе. Программа в реальном времени отсортирует запущенные процессы по их нагрузке на процессор.

htop − это продвинутый консольный мониторинг процессов. Утилита выводит постоянно меняющийся список системных процессов, который сортируется в зависимости от нагрузки на ЦПУ. Если делать сравнение сtop, то htop показывает абсолютно все процессы в системе, время их непрерывного использования, загрузку процессоров и расход оперативной памяти.

8). find − 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.

Команда find имеет такой синтаксис:

find[папка][параметры] критерий шаблон [действие]

Папка − каталог в котором будем искать

Параметры − дополнительные параметры, например, глубина поиска, и т д.

Критерий − по какому критерию будем искать: имя, дата создания, права, владелец и т д.

Шаблон – непосредственно значение по которому будем отбирать файлы.

Основные параметры:

-P никогда не открывать символические ссылки

-L - получает информацию о файлах по символическим ссылкам. Важно для дальнейшей обработки, чтобы обрабатывалась не ссылка, а сам файл.

-maxdepth - максимальная глубина поиска по подкаталогам,для поиска только в текущем каталоге установите 1.

-depth - искать сначала в текущем каталоге, а потом в подкаталогах

-mount искать файлы только в этой файловой системе.

-version - показать версию утилиты find

-print - выводить полные имена файлов

-typef - искать только файлы

-typed - поиск папки в Linux

Основные критерии:

-name - поиск файлов по имени

-perm - поиск файлов в Linux по режиму доступа

-user - поиск файлов по владельцу

-group - поиск по группе

-mtime - поиск по времени модификации файла

-atime - поиск файлов по дате последнего чтения

-nogroup - поиск файлов, не принадлежащих ни одной группе

-nouser - поиск файлов без владельцев

-newer - найти файлы новее чем указанный

-size - поиск файлов в Linux по их размеру

Примеры:

find~ -type d поиск директорий в домашнем каталоге

find~ -type f -name “.\*” поиск скрытых файлов в домашнем каталоге

9). Файл по его содержимому можно найти с помощью команды grep: «grep -r” слово/выражение, которое нужно найти”».

10). Утилита df, позволяет проанализировать свободное пространство на всех подключенных к системе разделах.

11). При выполнении команды du (без указания папки и опции) можно получить все файлы и папки текущей директории с их размерами. Для домашнего каталога: du ~/

12). Основные сигналы (каждый сигнал имеет свой номер), которые используются для завершения процесса:

* SIGINT–самый безобидный сигнал завершения, означает Interrupt. Он отправляется процессу, запущенному из терминала с помощью сочетания клавиш Ctrl+C. Процесс правильно завершает все свои действия и возвращает управление;
* SIGQUIT–это еще один сигнал, который отправляется с помощью сочетания клавиш, программе, запущенной в терминале. Он сообщает ей что нужно завершиться и программа может выполнить корректное завершение или проигнорировать сигнал. В отличие от предыдущего, она генерирует дамп памяти. Сочетание клавиш Ctrl+/;
* SIGHUP–сообщает процессу, что соединение с управляющим терминалом разорвано, отправляется, в основном, системой при разрыве соединения с интернетом;
* SIGTERM–немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы;
* SIGKILL–тоже немедленно завершает процесс, но, в отличие от предыдущего варианта, он не передается самому процессу, а обрабатывается ядром. Поэтому ресурсы и дочерние процессы остаются запущенными.

Также для передачи сигналов процессам в Linux используется утилита kill, её синтаксис: kill [-сигнал] [pid\_процесса] (PID – уникальный идентификатор процесса). Сигнал представляет собой один из выше перечисленных сигналов для завершения процесса.

Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого используют команды ps и grep. Команда ps предназначена для вывода списка активных процессов в системе и информации о них. Команда grep запускается одновременно с ps (вканале) и будет выполнять поиск по результатам команды ps.

Утилита pkill – это оболочка для kill, она ведет себя точно так же, и имеет тот же синтаксис, только в качестве идентификатора процесса ей нужно передать ег оимя.

killall работает аналогично двум предыдущим утилитам. Она тоже принимает имя процесса в качестве параметра и ищет его PID в директории /proc. Но эта утилита обнаружит все процессы с таким именем и завершит их.

# Вывод

В ходе данной лабораторной работы я научился перенаправлять ввод, работать с конвейером, искать файлы, фильтровать текст, а также проверять использование диска и управлять процессами, получать о них информацию.