Отчёт по лабораторной работе № 6

*дисциплина:* Операционные системы

Андрианова Марина Георгиевна

Содержание

#### 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.

#### Выполнение лабораторной работы

1. Осуществим вход в систему,используя соответствующее имя пользователя.
2. Запишем в файл file.txt названия файлов,содержащихся в каталоге /etc. Для этого введём команду “ls -a /etc > file.txt”.Допишем в этот же файл названия файлов,содержащихся в нашем домашнем каталоге с помощью команды “ls -a ~ >> file.txt”. Затем просмотрим файл с помощью команды “cat file.txt”,чтобы убедиться в правильном выполнении действий(рис.1-2).

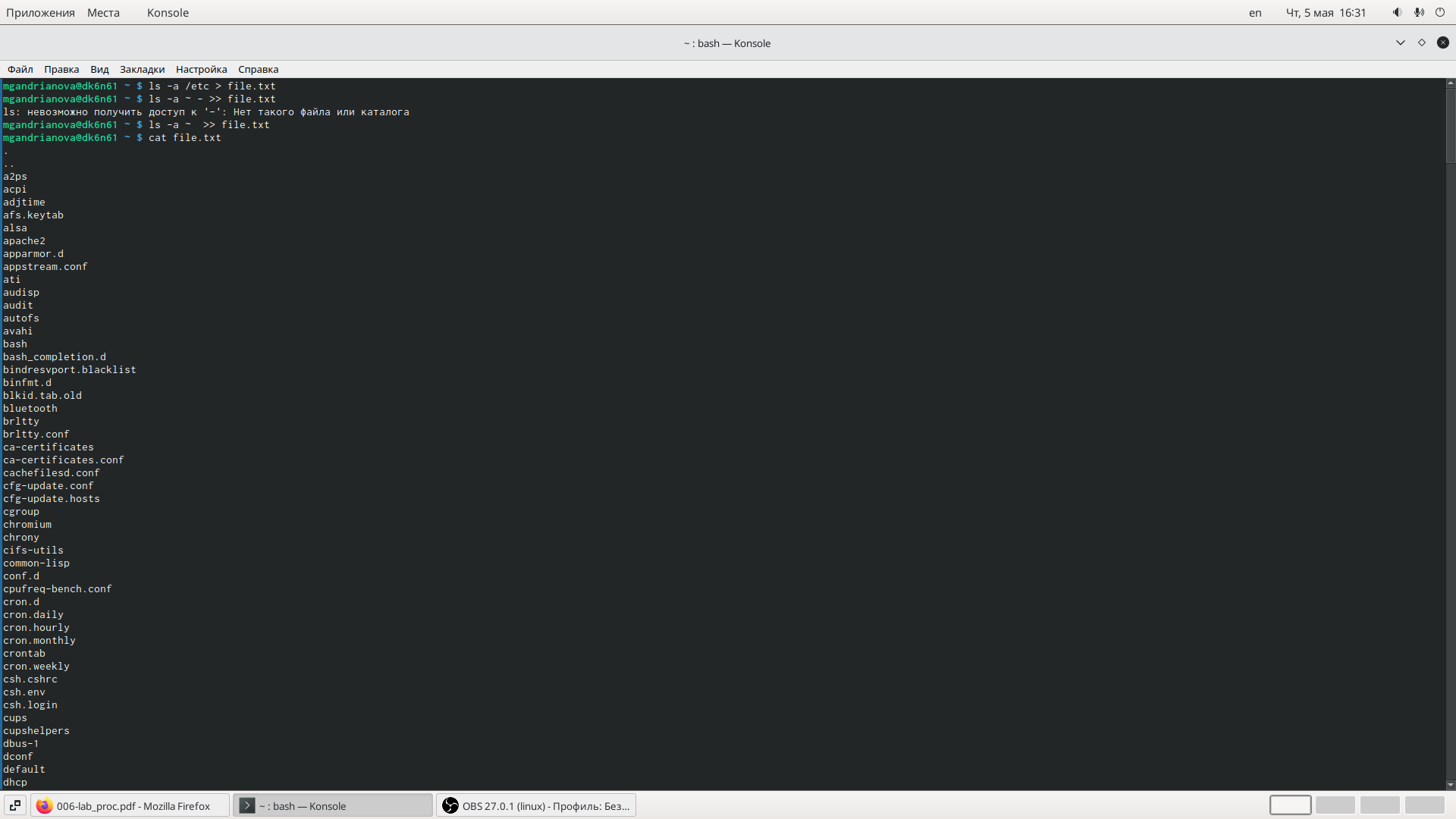


Рис.1

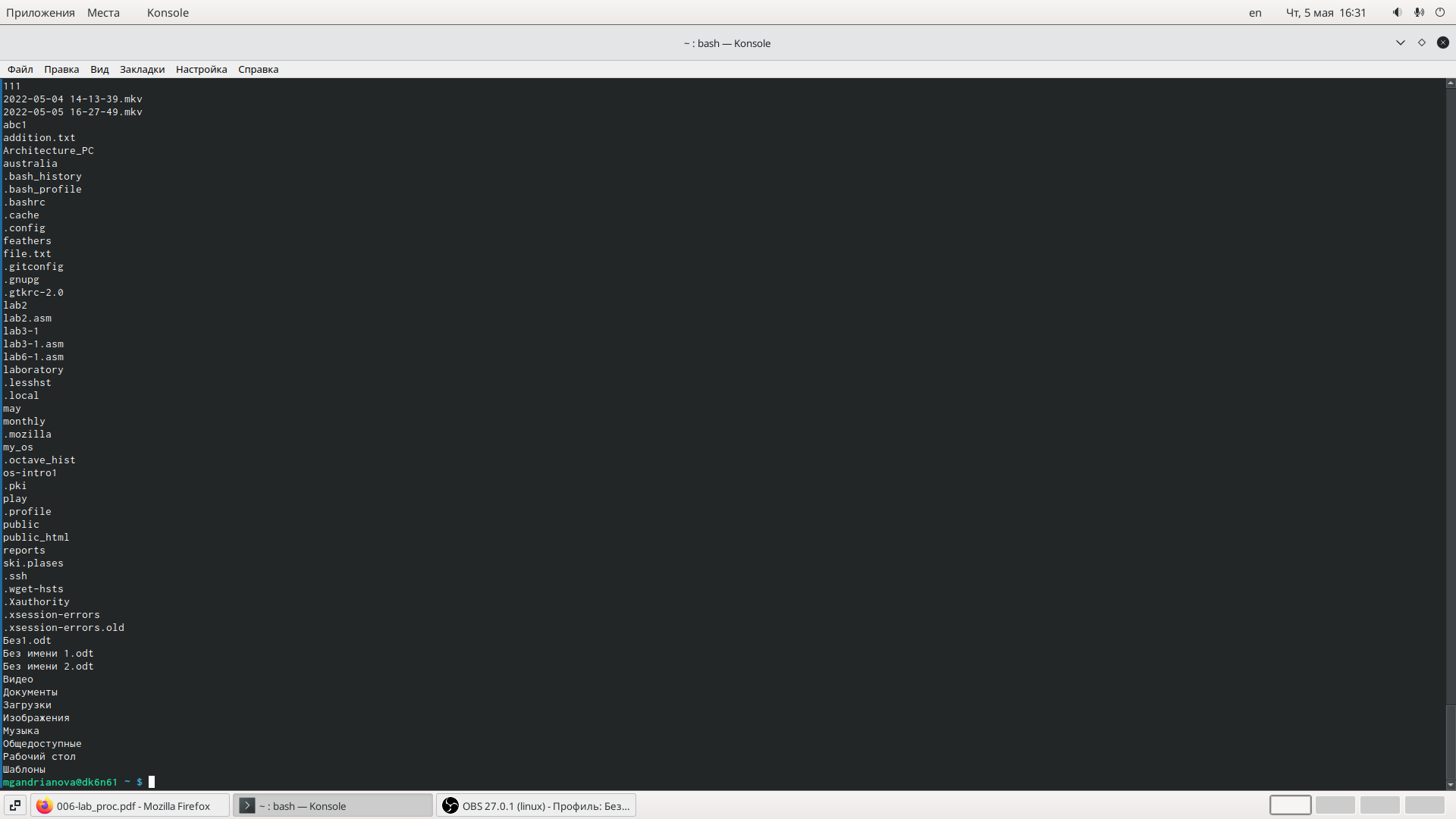


Рис.2

1. Выведем имена всех файлов из file.txt,имеющих расширение .conf,после чего запишем их в новый текстовой файл conf.txt, используя команду “grep -e ‘.conf$’ file.txt > conf.txt”. С помощью команды “cat conf.txt” проверяем правильность выполненных действий(рис.3-4).

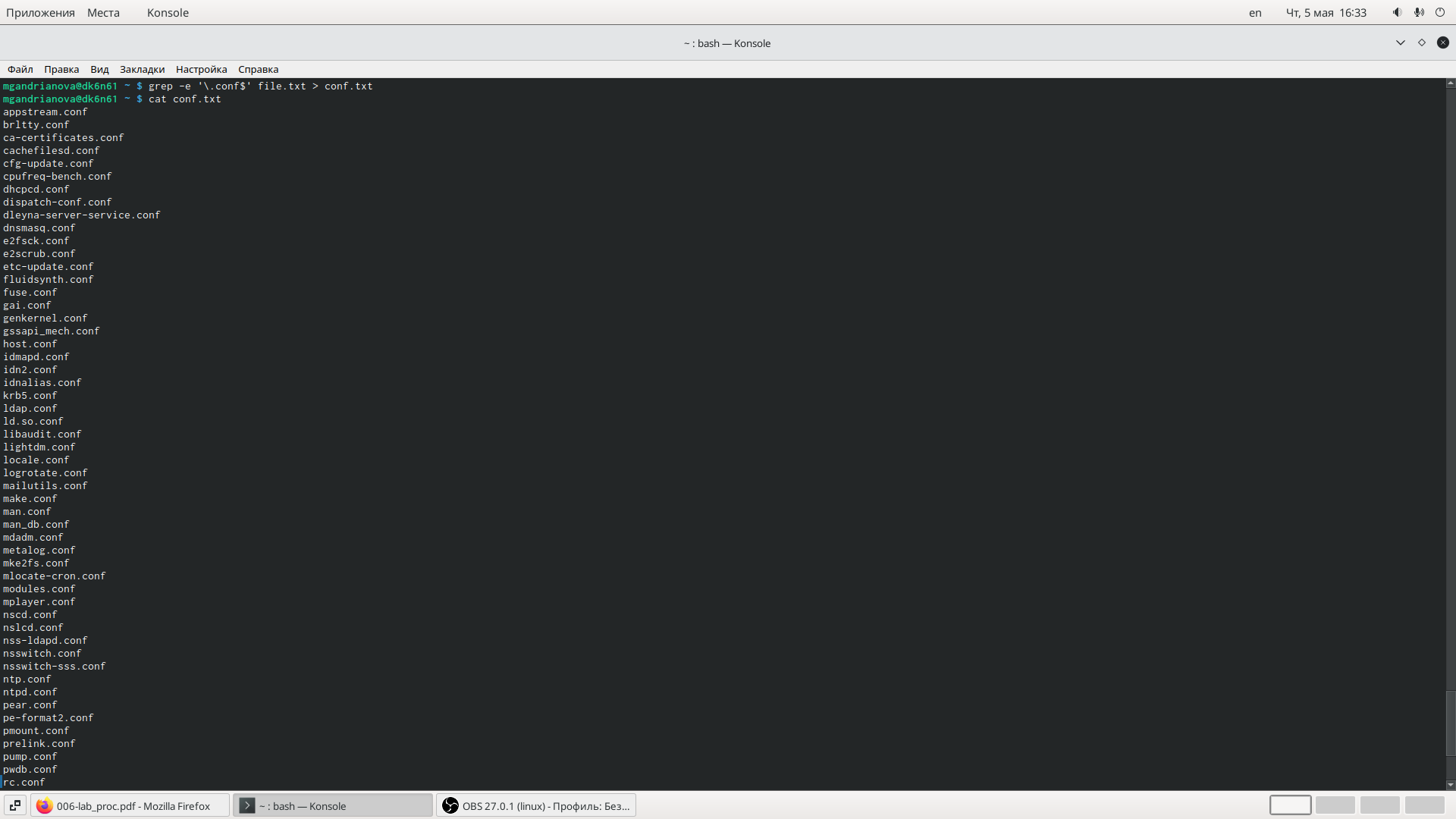


Рис.3

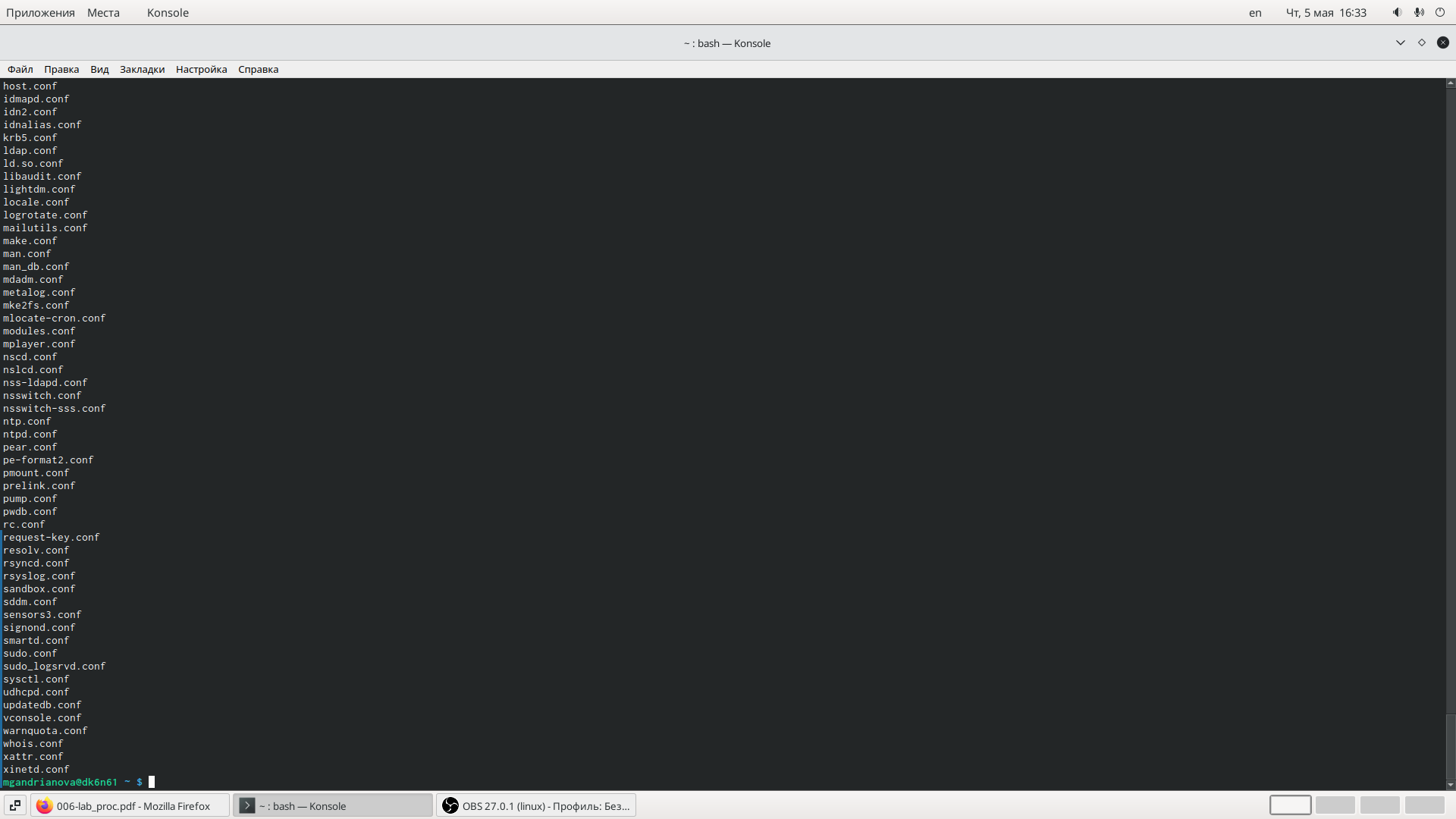


Рис.4

1. Определить,какие файлы в нашем домашнем каталоге имеют имена,начинающиеся с символа c, можно несколькими командами(рис.5). (Опция maxdepth 1 необходима для того, чтобы файлы находились строго только в домашнем каталоге):

find ~ -maxdepth 1 -name "c\*" -print  
 ls ~/c\*  
 ls -a ~ grep c\*

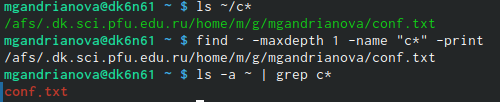


Рис.5

1. Чтобы вывести на экран (по странично) имена файлов из каталога /etc,начинающиеся с символа h, я использовала команду “find /etc -maxdepth 1 -name”h\*” | less” (рис.6-7).

Рис.6

Рис.6

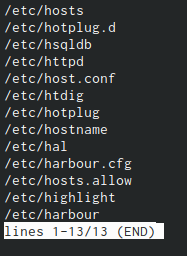


Рис.7

1. Запустим в фоновом режиме процесс,который будет записывать в файл ~/logfile файлы,имена которых начинаются с log, используя команду “find / -name”log\*” > logfile& “.(рис.8).

Рис.8

Рис.8

Командой “cat logfile” проверяю выполненные действия(рис.9-10)

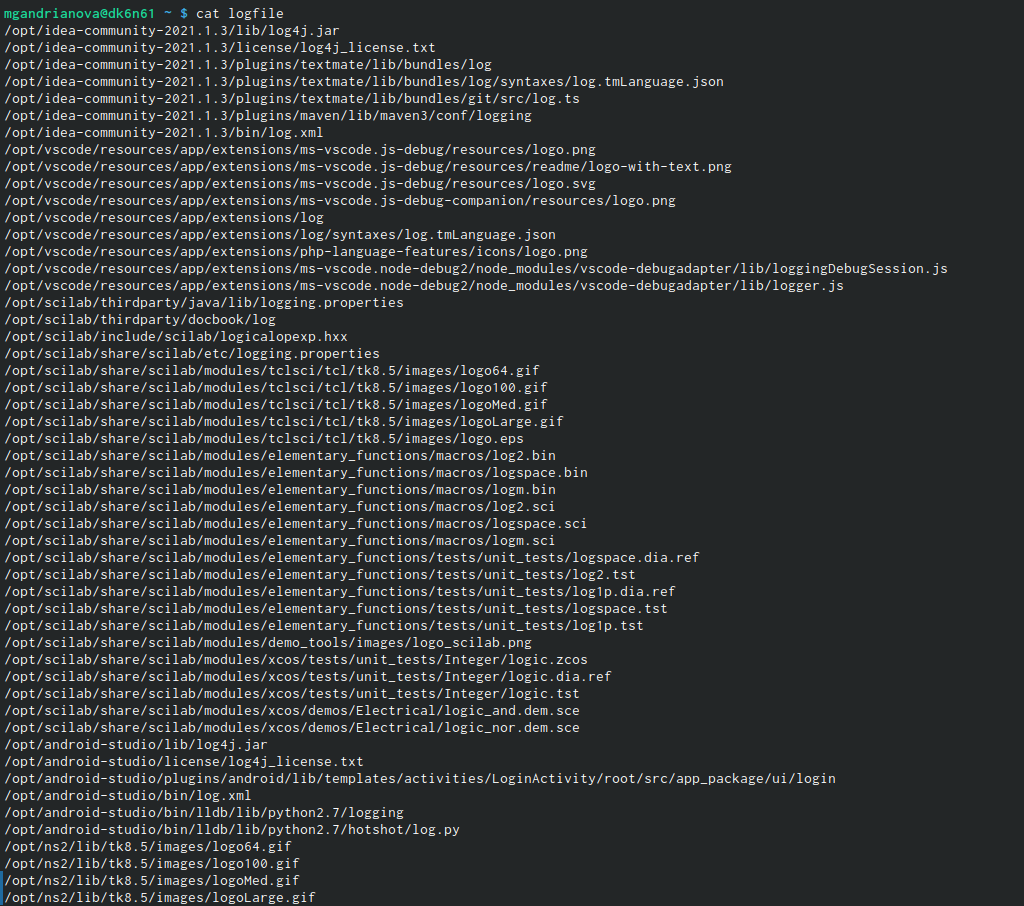


Рис.9

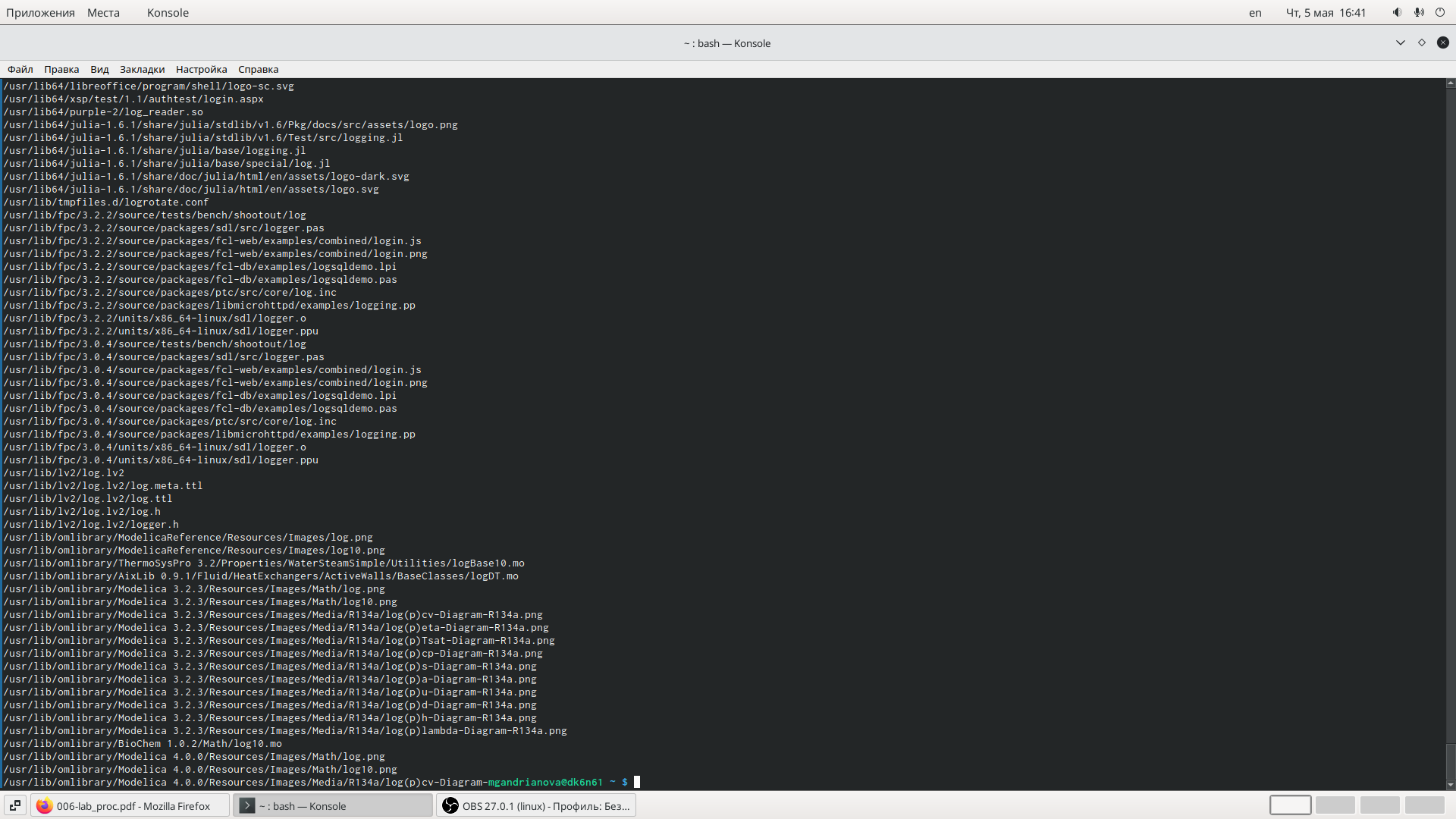


Рис.10

1. Удалим файл ~/logfile с помощью команды(рис.11):

rm logfile

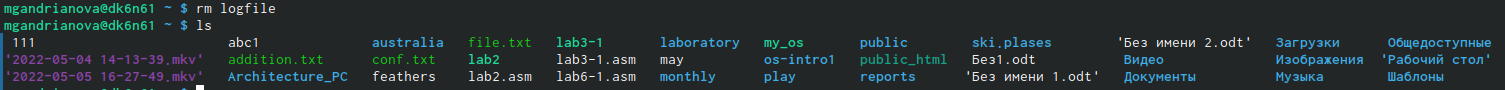


Рис.11

1. Запустим из консоли в фоновом режиме редактор gedit с помощью команды “gedit&”(рис.12). Появится окно редактора(рис.13).

Рис.12

Рис.12

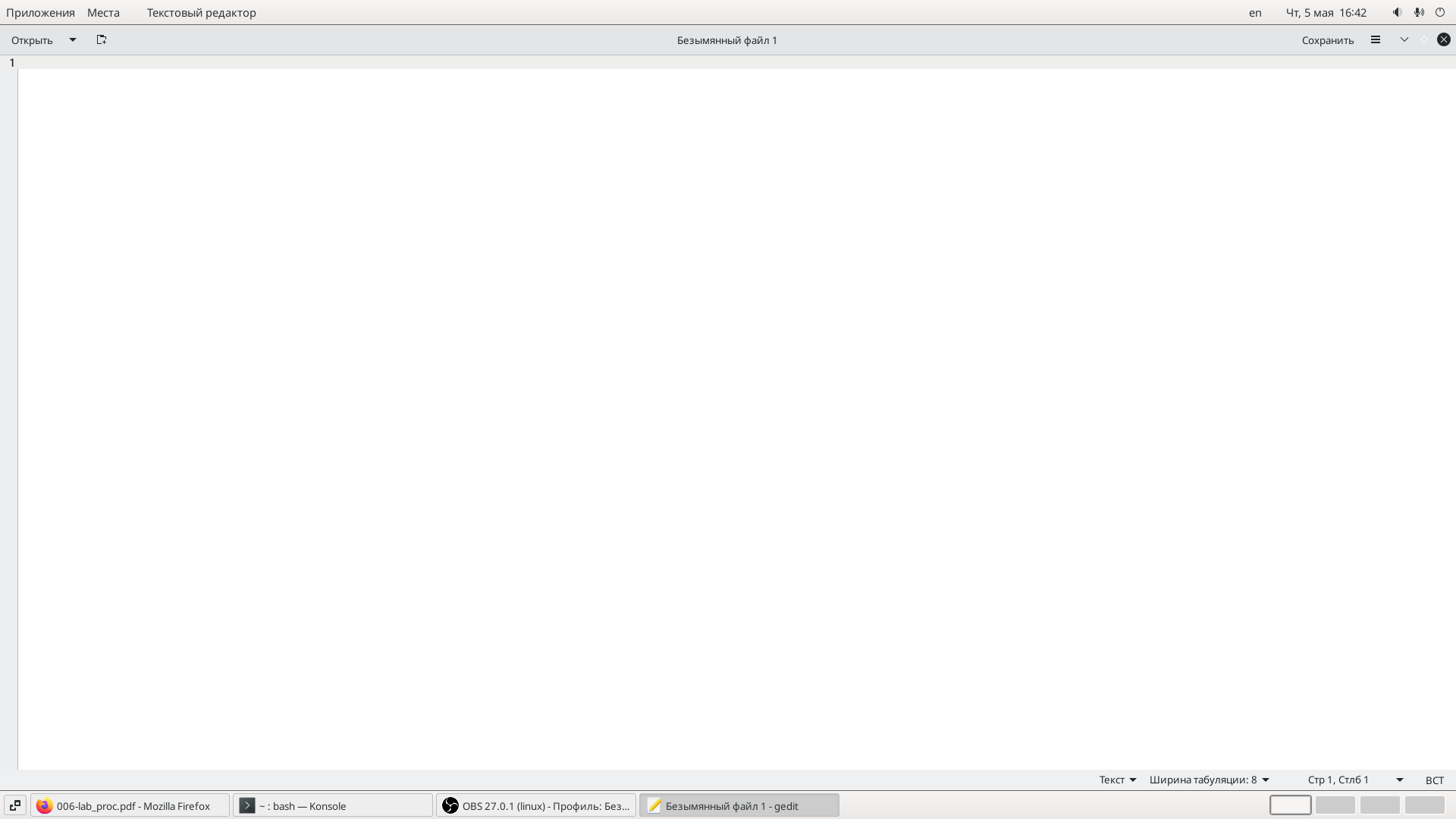


Рис.13

1. Чтобы определить идентификатор процесса gedit,используем команду ” ps | grep -i “gedit” “.Наш процесс имеет PID 10980.(рис.14).Определить идентификатор процесса можно и другими командами(рис.14):

pgrep gedit  
 pidof gedit

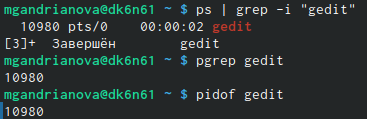


Рис.14

1. Прочитав справку (man) команды kill с помощью команды “man kill”(рис.15-17), используем её для завершения процесса gedit(команда “kill %10980”) (рис.18).

Рис.15

Рис.15

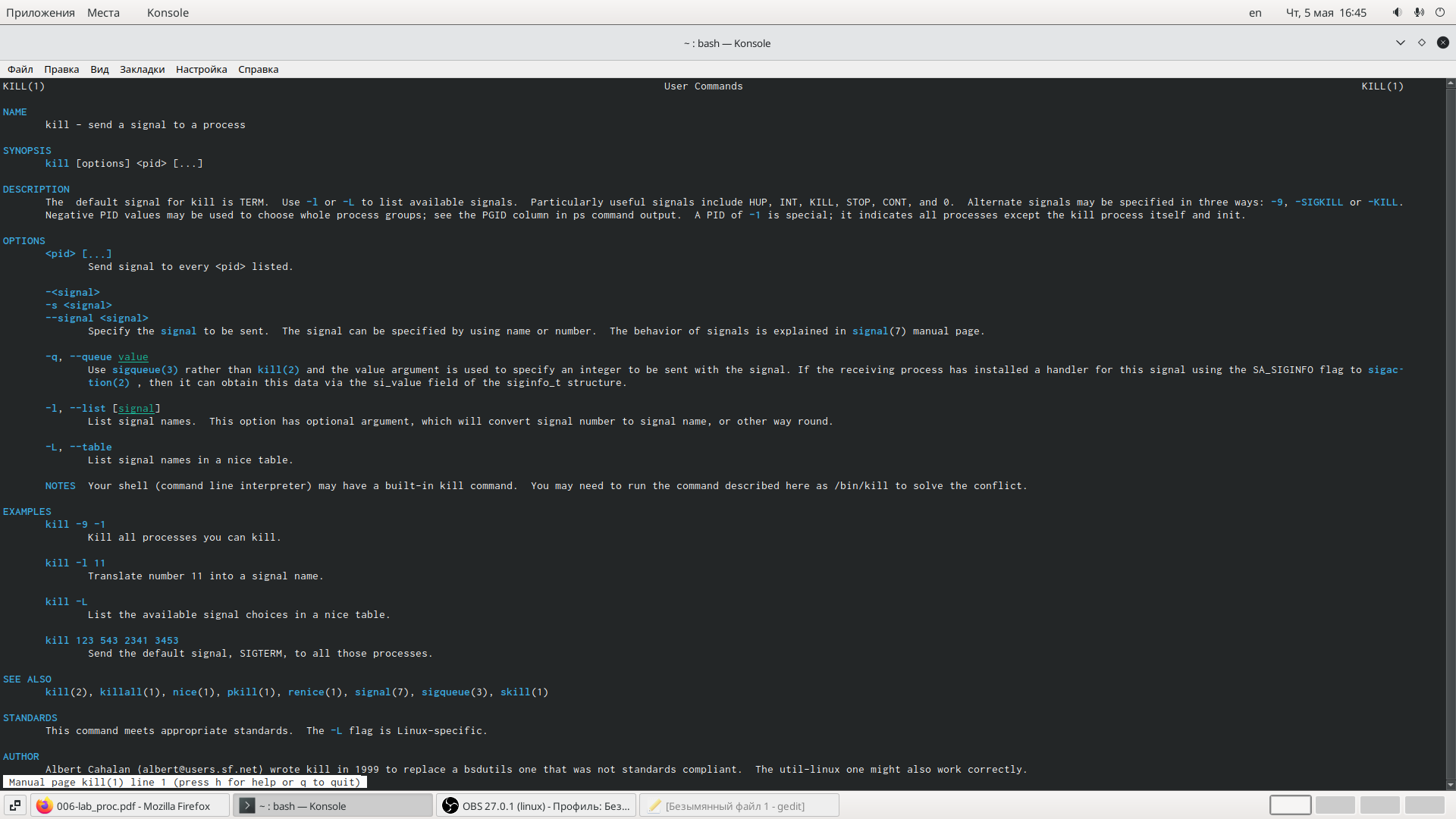


Рис.16

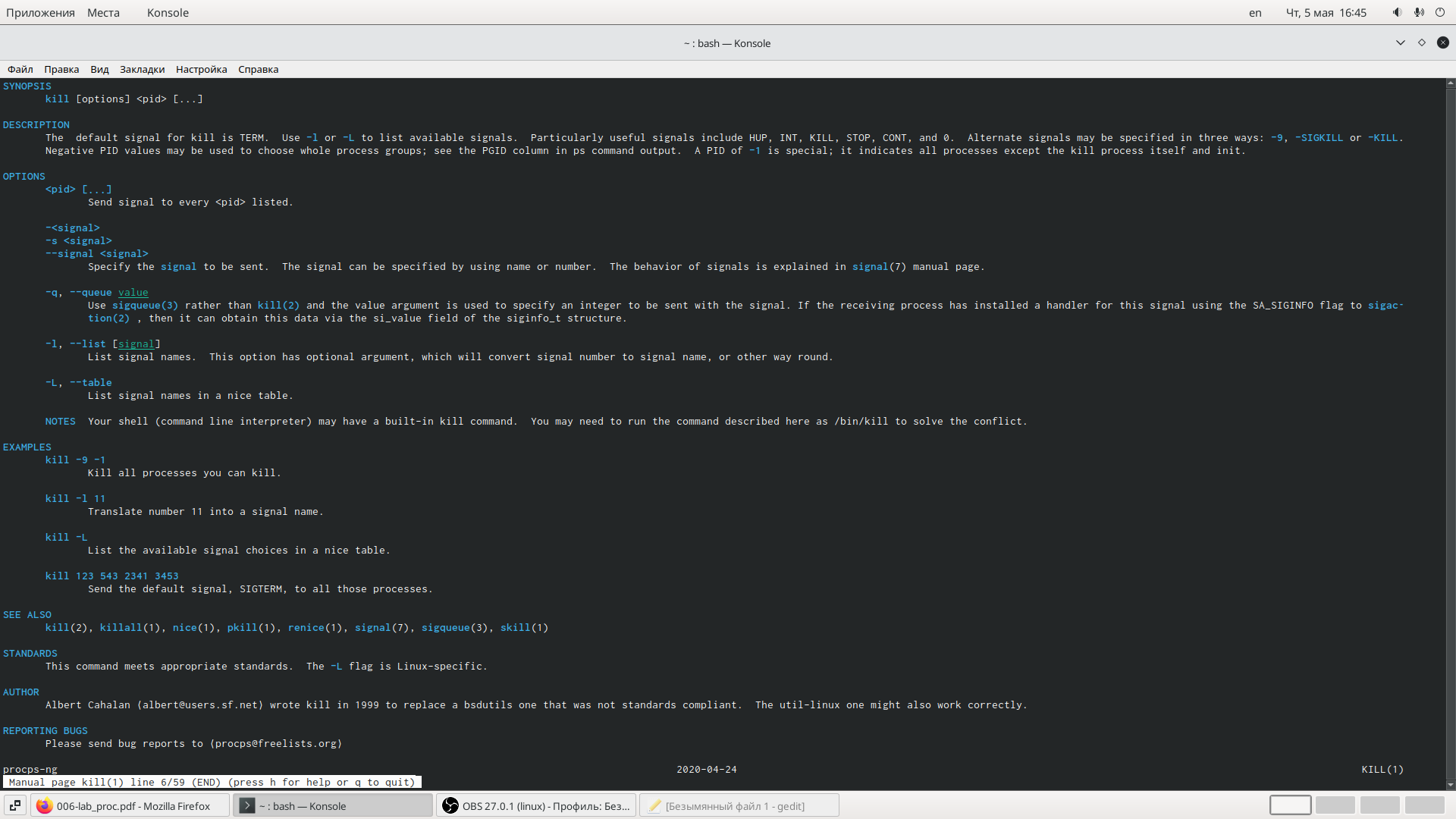


Рис.17

Рис.18

Рис.18

1. Выполним команды df(рис.22) и du(рис.23-24),предварительно получив более подробную информацию об этих командах(рис.20-21),с помощью команд “man df”(рис.19) и “man du”(рис.19).

Рис.19

Рис.19



Рис.20

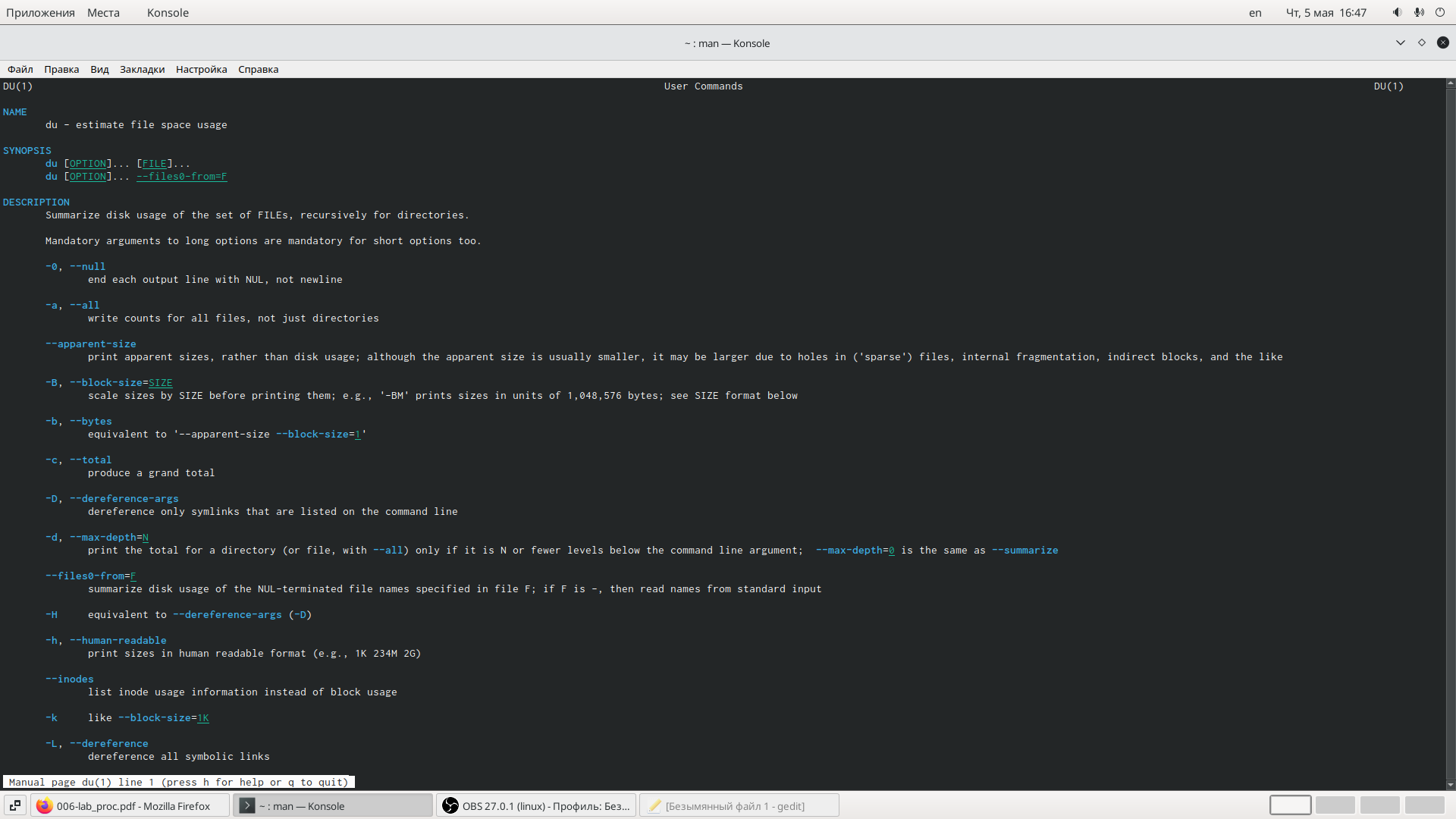


Рис.21

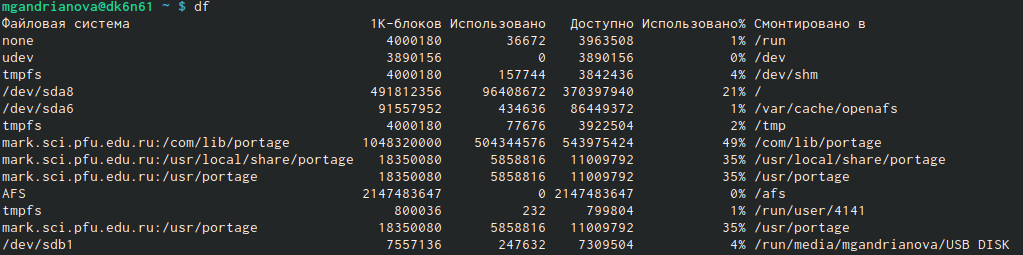


Рис.22

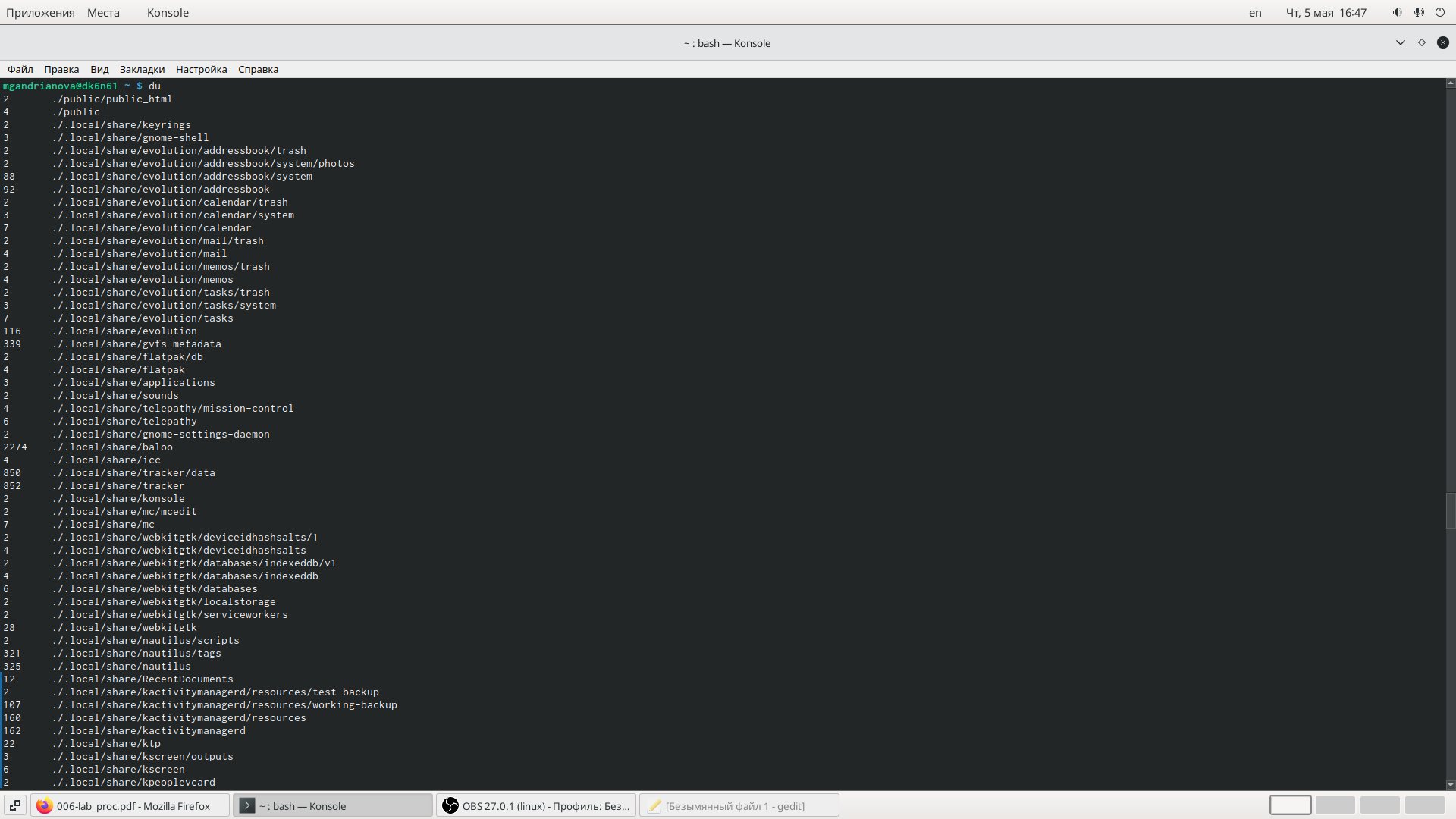


Рис.23

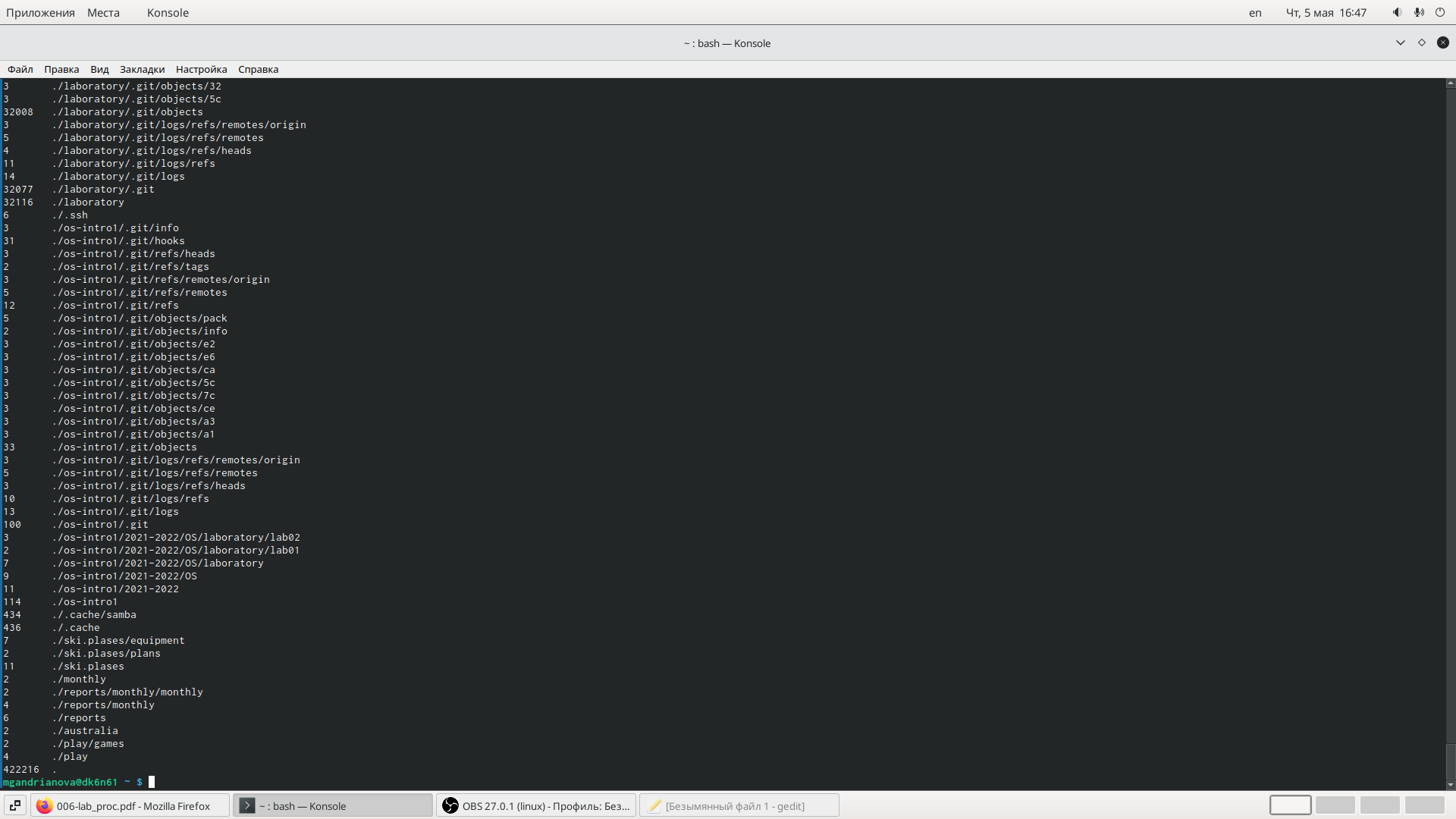


Рис.24

1. Воспользуемся справкой команды find(рис.26), введя команду “man find”(рис.25).Затем выведем имена всех директорий,имеющихся в нашем домашнем каталоге, с помощью команды “find ~ -type d” (рис.27-28).

Рис.25

Рис.25



Рис.26

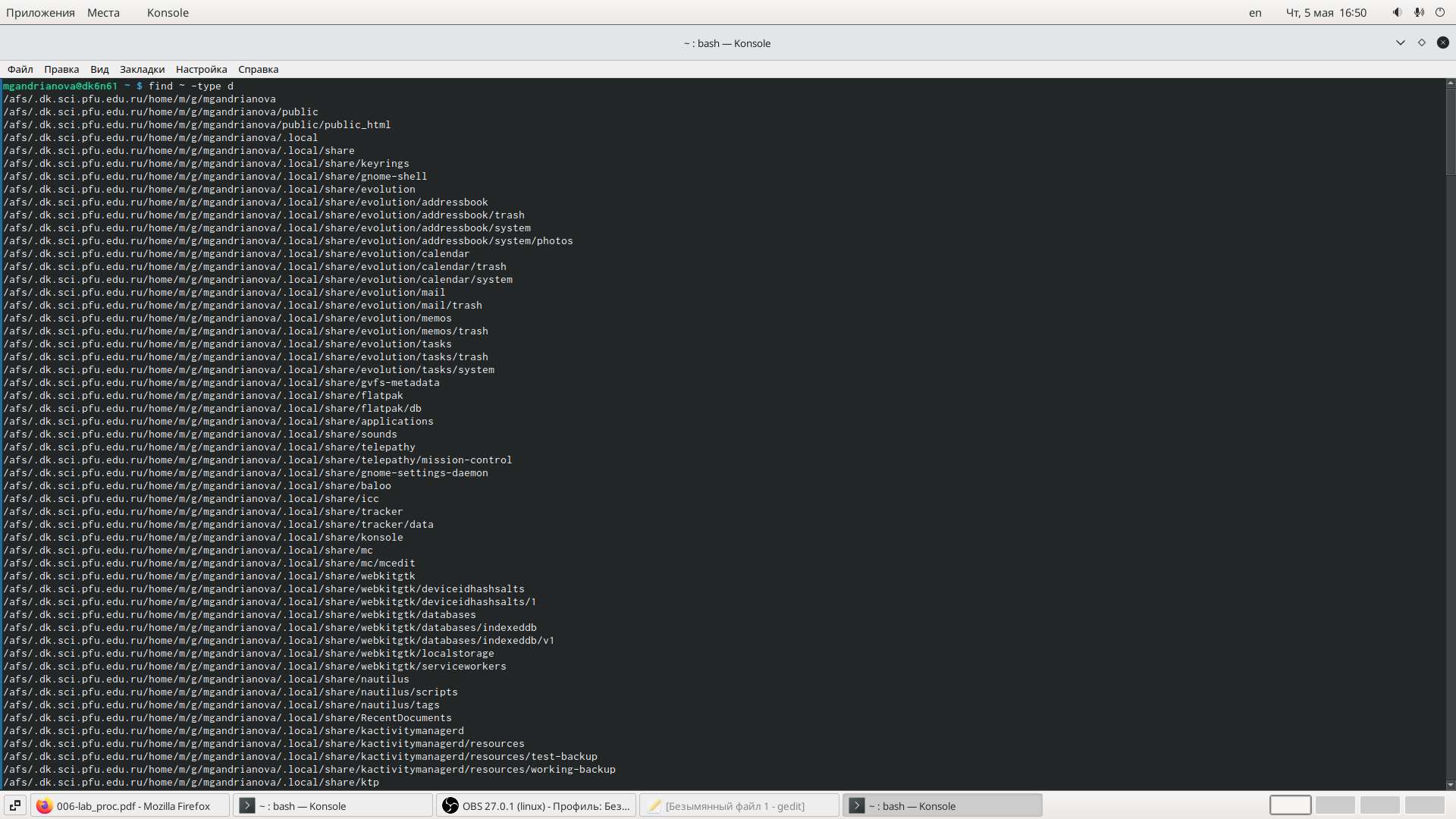


Рис.27

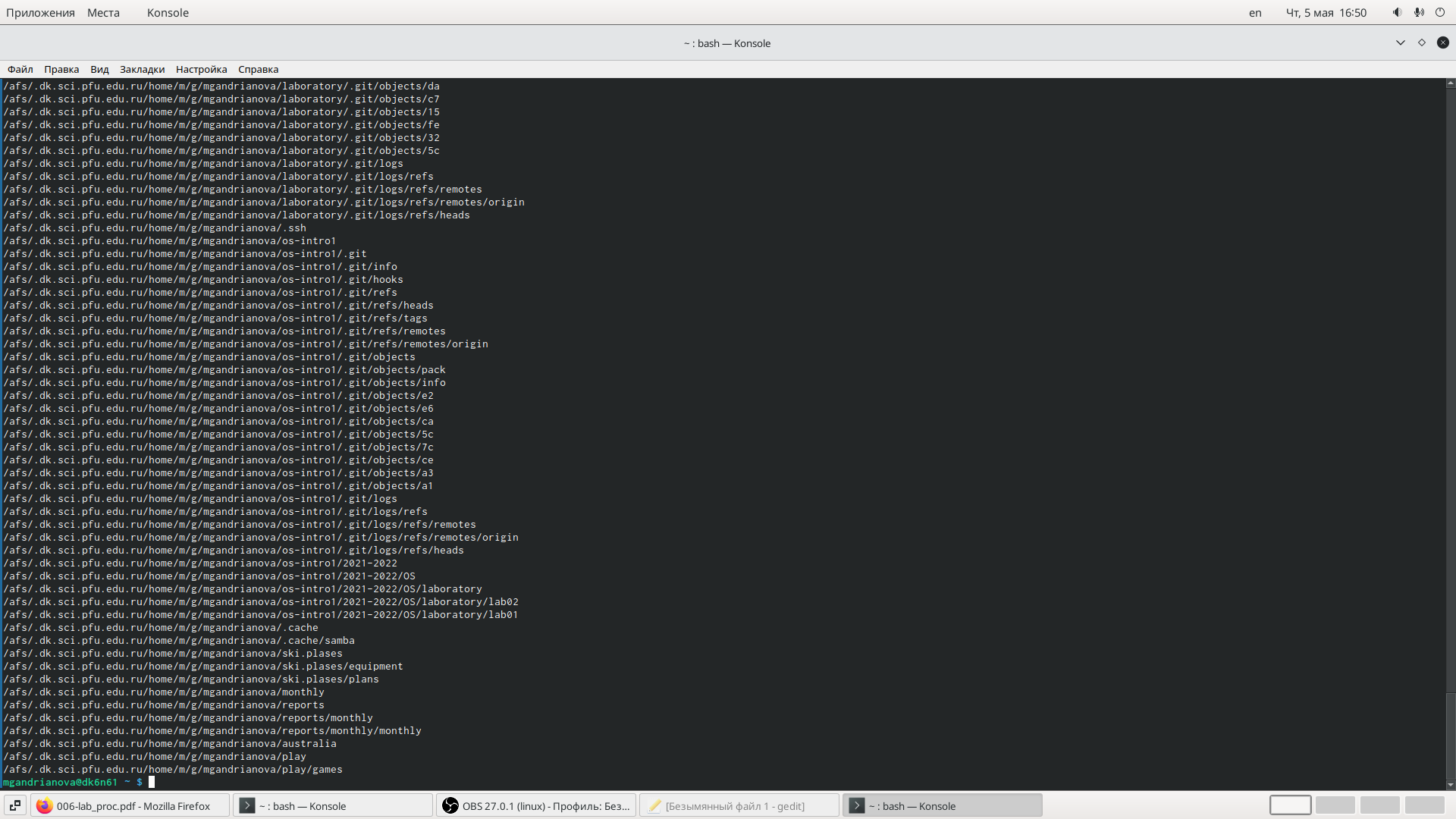


Рис.28

#### Выводы

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

## 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 (по умолчанию: консоль),файловый дескриптор 2.

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

1. ‘>’ означает перенаправление вывода (stdout) в файл.

‘>>’ означает перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла).

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

команда 1 | команда 2  
 # означает, что вывод команды 1 передастся на ввод команде 2

1. 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.

1. pid: идентификатор процесса (PID) процесса (processID), к которому вызывают метод gid: идентификатор группы UNIX, в котором работает программа.
2. 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.
3. 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 с top, то htop показывает абсолютно все процессы в системе, время их непрерывного использования, загрузку процессоров и расход оперативной памяти.
4. 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

-type f - искать только файлы

-type d - 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 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 его имя.

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.