Отчёт по лабораторной работе №6

Операционные системы

Самсонова Мария Ильинична

Содержание

[Цель работы 1](#_Toc102846268)

[Задание 1](#_Toc102846269)

[Выполнение лабораторной работы 2](#_Toc102846270)

[Вывод 24](#_Toc102846271)

[Ответы на контрольные вопросы: 24](#_Toc102846272)

# Цель работы

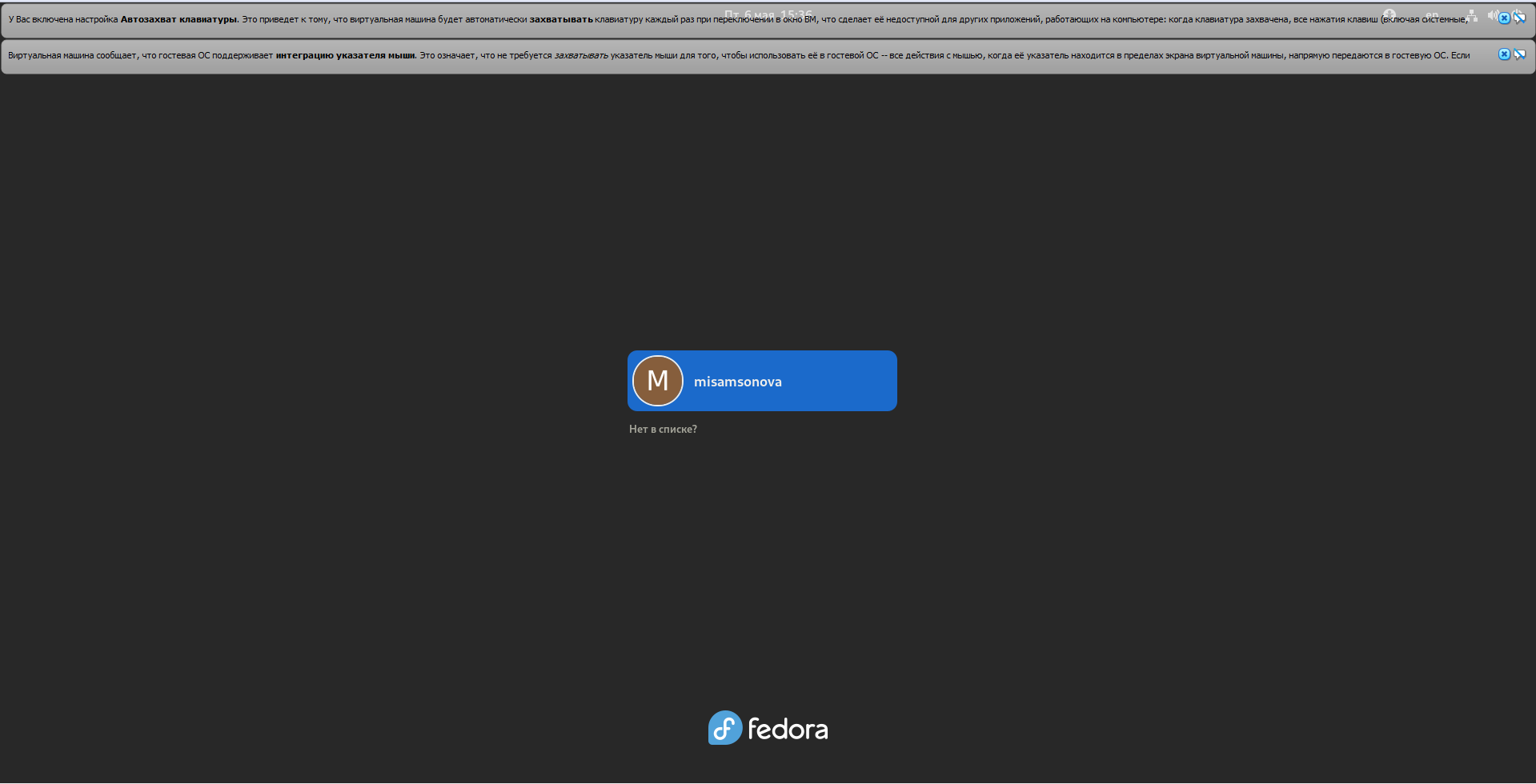
Ознакомление с инструментами поиска файлов и фильтрации текстовых данных. Приобретение практических навыков: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

# 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 .
4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся 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.

# Выполнение лабораторной работы

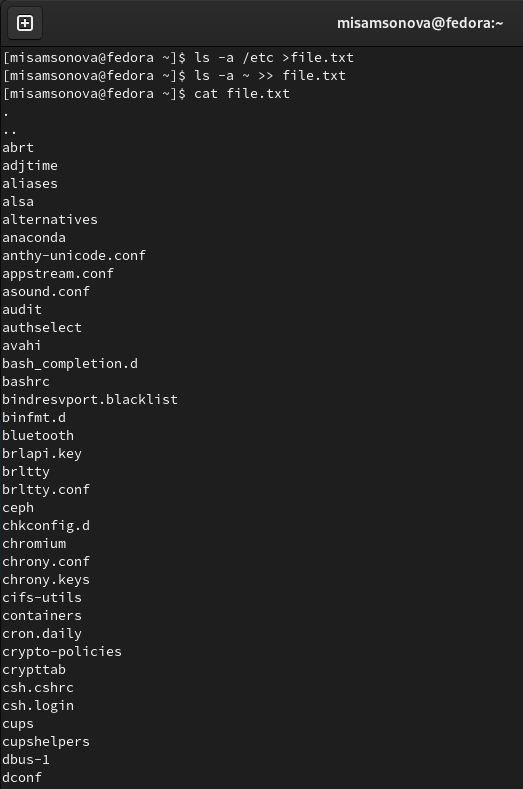
1. Для начала мы осуществим вход в систему, используя наше имя пользователя:(рис. [-@fig:001])



Вход в систему

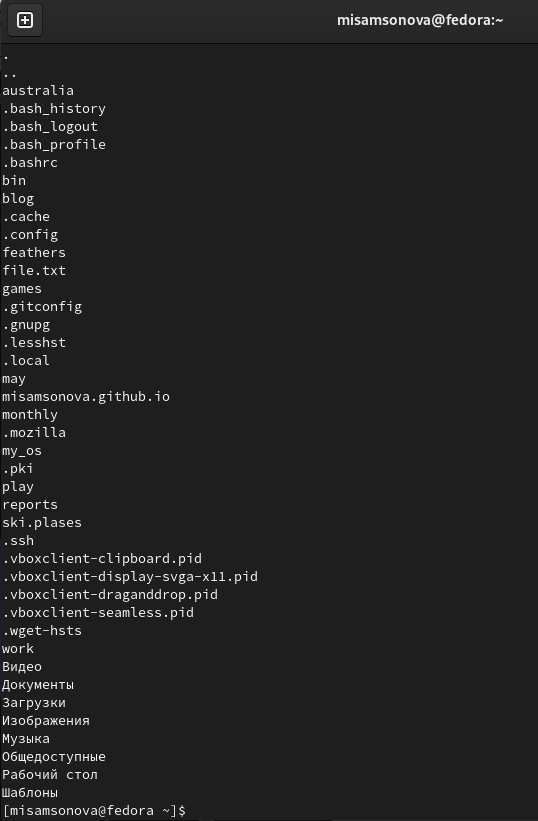
{ #fig:001 width=70% }

1. Зайдём в Терминал и с помощью команды *ls -a /etc > file.txt”* запишем в файл file.txt названия файлов, содержащихся в каталоге /etc. Также командой *ls -a ~ >> file.txt* допишем в этот же файл названия файлов, содержащихся в нашем домашнем каталоге, и командой *cat file.txt* проверим содержимое файла: (рис. [-@fig:002]) и (рис. [-@fig:003])



Запись названия файлов, содержащихся в каталоге /etc и в домашней каталоге, в конкретный файл, содержащийся в каталоге /etc, вывод содержимого файла file.txt

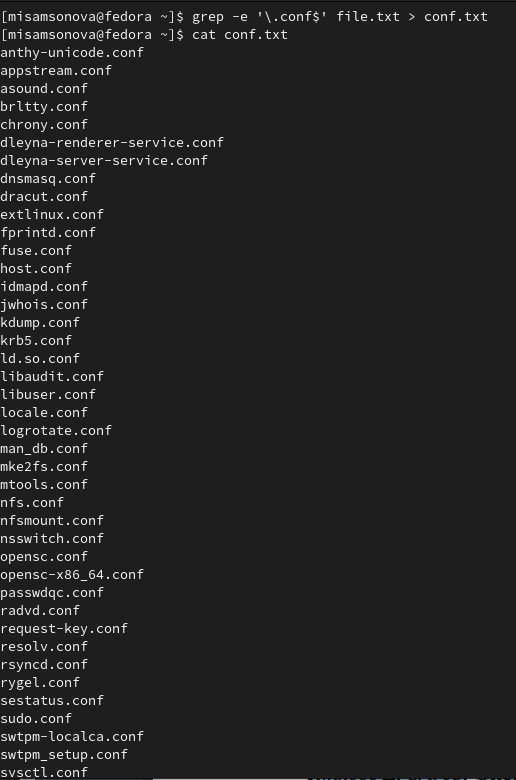
{ #fig:002 width=70% }



Вывод содержимого файла file.txt

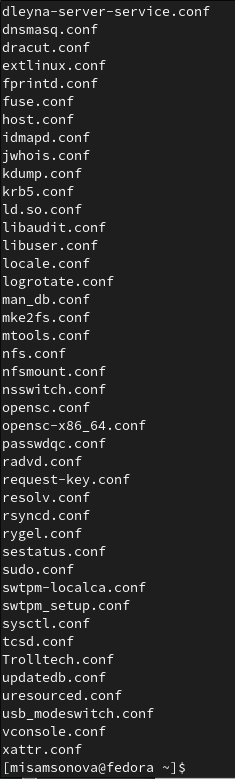
{ #fig:003 width=70% }

1. С помощью команды *grep -e ‘.conf$’ file.txt* выведем имена всех файлов из file.txt, имеющих расширение .conf, и проверим содержимое файла file.txt командой *cat file*: (рис. [-@fig:004]) и (рис. [-@fig:005])



Вывод имен всех файлов с расширением .conf и вывод содержимого файла file.txt

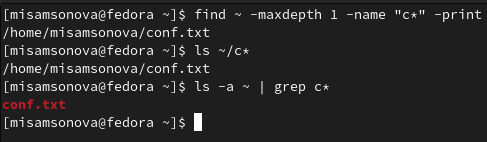
{ #fig:004 width=70% }



Вывод содержимого файла file.txt

{ #fig:005 width=70% }

1. Теперь определим, какие файлы в нашем домашнем каталоге имеют имена, начинавшиеся с символа “c” с помощью команд: *find ~ -maxdepth 1 -name “c \*” -print*, *ls ~/c*, *ls -a ~ | grep c*  :\* (причём опция maxdepth1 необходима для того, чтобы файлы находились строго только в домашнем каталоге) (рис. [-@fig:006])



Вывод содержимого файла file.txt

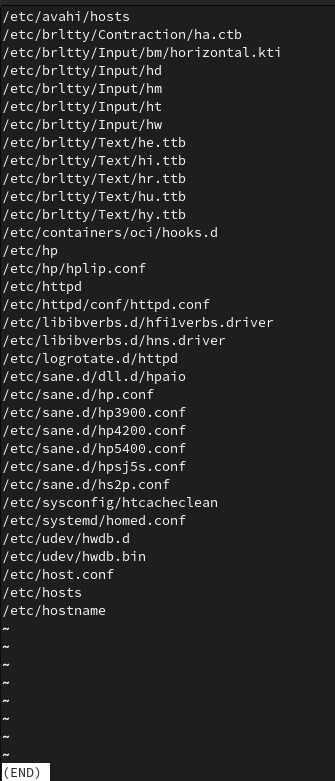
{ #fig:006 width=70% }

1. Выведем на экран (постранично) имена файлов из каталога /etc, начинающиеся с символа h с помощью команды *find /etc ~maxdepth 1 -name “h \*” | less* : (рис. [-@fig:007]) и (рис. [-@fig:008])

Ввод команды для вывода на экран имен файлов из каталога /etc

Ввод команды для вывода на экран имен файлов из каталога /etc

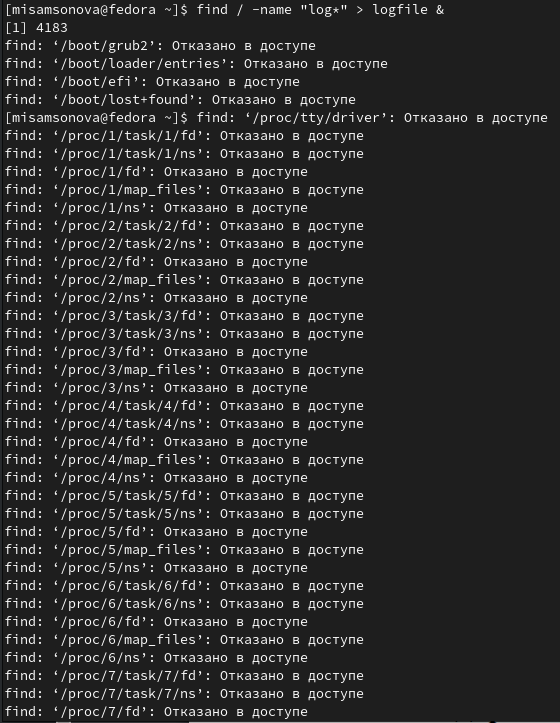
{ #fig:007 width=70% }



Ввод команды для вывода на экран имен файлов из каталога /etc

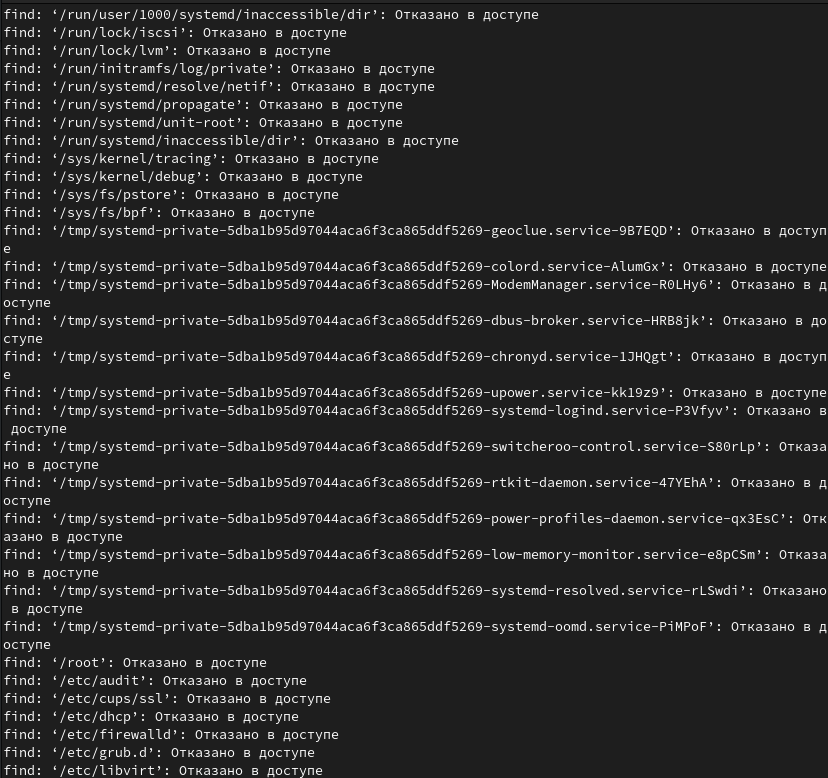
{ #fig:008 width=70% }

1. Запустим в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log, с помощью следующей команды: (рис. [-@fig:009])



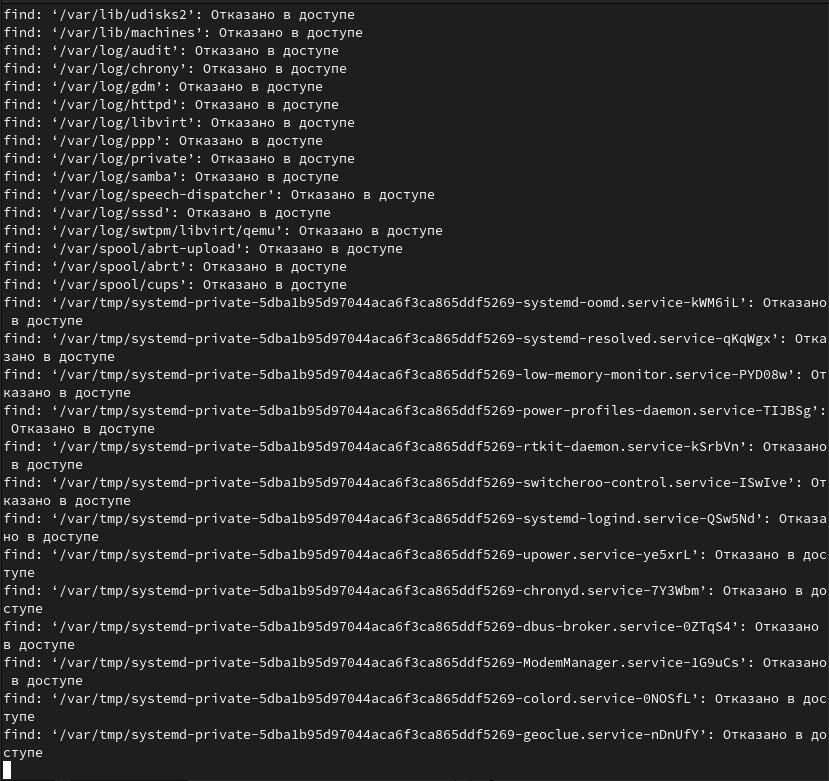
Запуск в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log

{ #fig:009 width=70% }



Запуск в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log

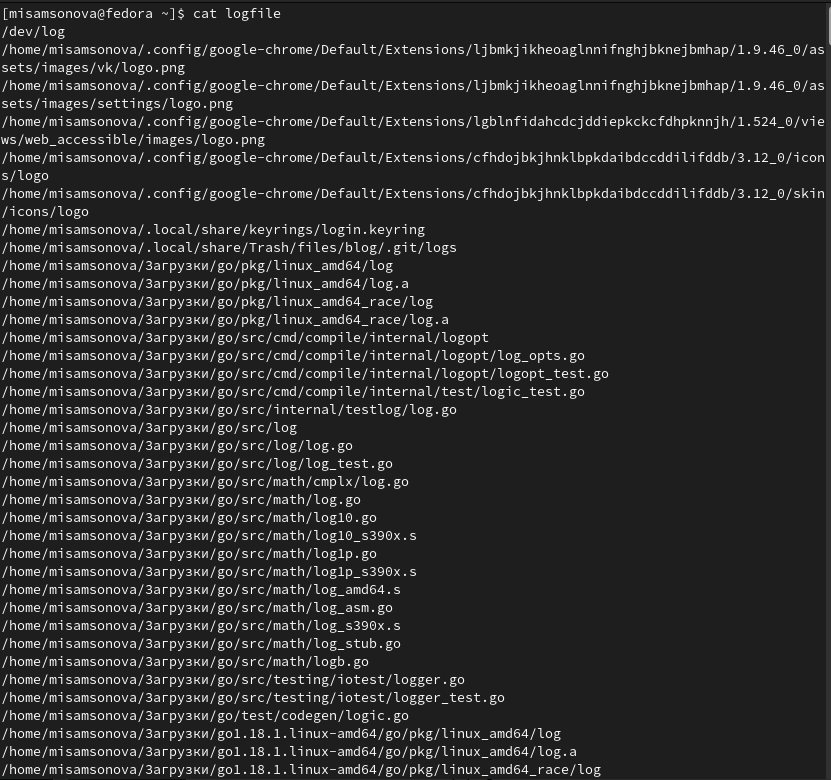
{ #fig:010 width=70% }



Запуск в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log

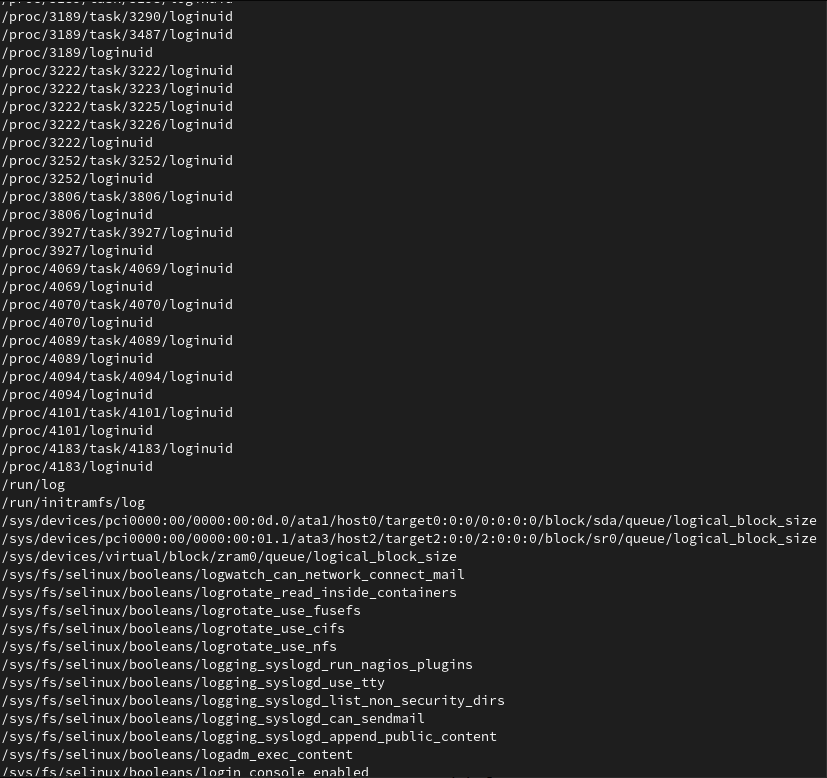
{ #fig:011 width=70% }

* Проверим содержимое файла logfile с помощью команды cat: (рис. [-@fig:012]), (рис. [-@fig:013]), (рис. [-@fig:014])



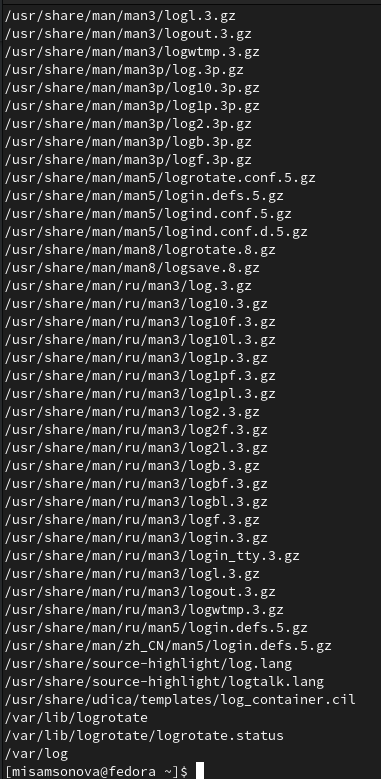
Содержимое logfile

{ #fig:012 width=70% }



Содержимое logfile

{ #fig:013 width=70% }



Содержимое logfile

{ #fig:014 width=70% }

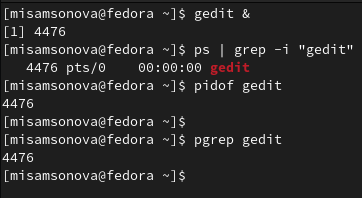
1. Далее удалим файл ~/logfile с помощью команды rm: (рис. [-@fig:015])

Удаление файла logfile

Удаление файла logfile

{#fig:015 width=70%}

1. Запускаем из консоли в фоновом режиме редактор gedit и определяем идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep с помощью следующих команд:(рис. [-@fig:016])



Запуск gedit и определение идентификатора процесса gedit

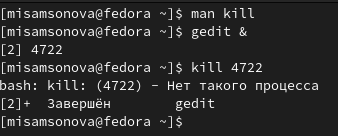
{#fig:016 width=70%}

1. После этого обращаемся к справке (man) команды kill и используем её для завершения процесса gedit:(рис. [-@fig:017]) и (рис. [-@fig:018])



Запуск gedit и определение идентификатора процесса gedit

{#fig:017 width=70%}



Запуск gedit и определение идентификатора процесса gedit

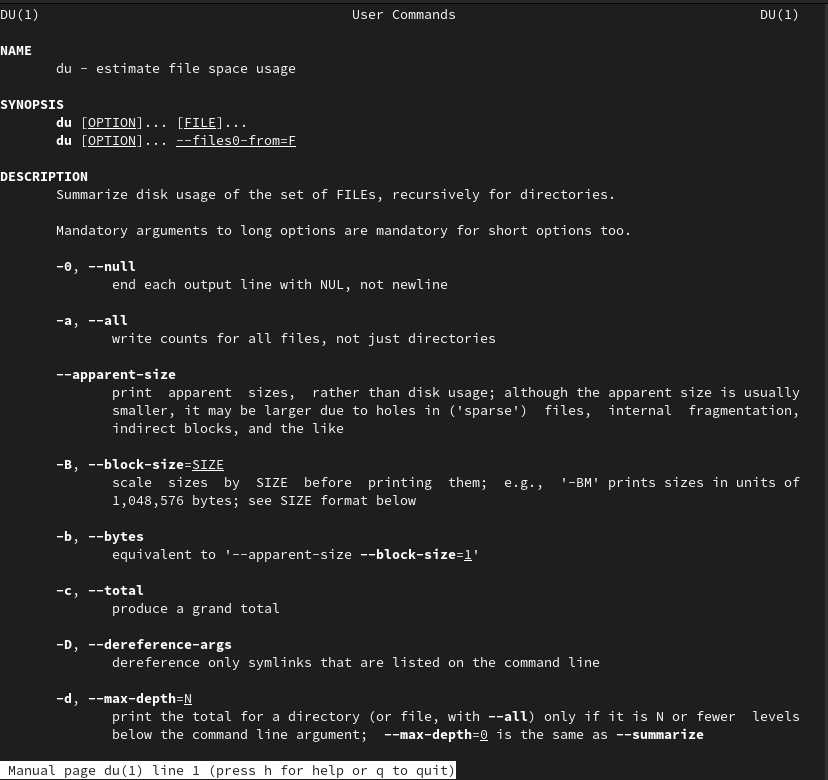
{#fig:018 width=70%}

1. Выполняем команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man: (рис. [-@fig:019]), (рис. [-@fig:020]) и (рис. [-@fig:021])



Справка о команде df

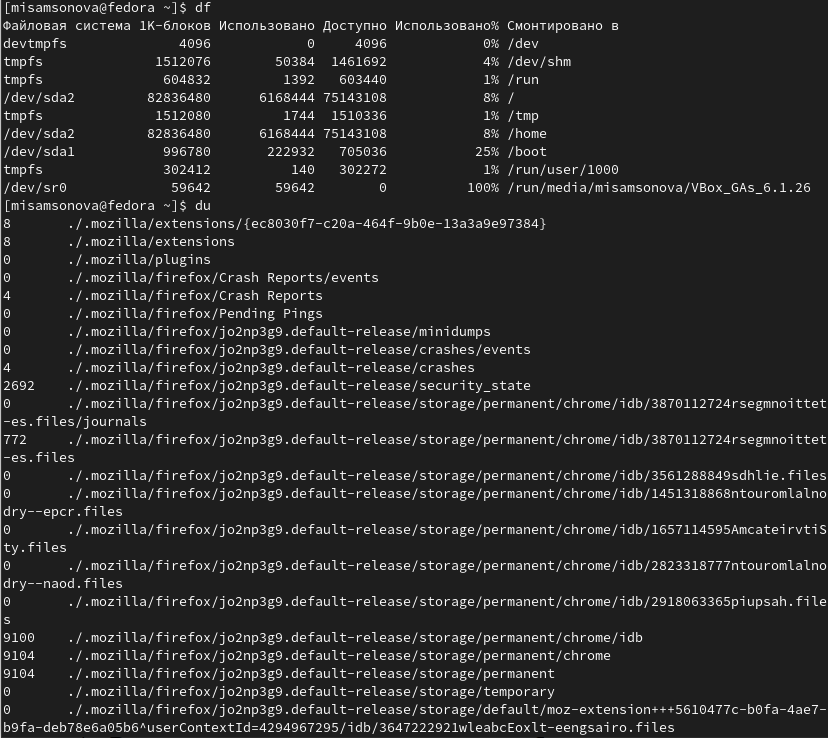
{#fig:019 width=70%}



Справка о команде du

{#fig:020 width=70%}

* Имеем: df – утилита, показывающая список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования. Синтаксис: df[опции] устройств. du – утилита, предназначенная для вывода информации об объеме дискового пространства, занятого файлами и директориями. Она принимает путь к элементу файловой системы и выводит информацию о количестве байт дискового пространства или блоков диска, задействованных для его хранения. Синтаксис: du [опции] каталог\_или\_файл



Выполнение команд df и du

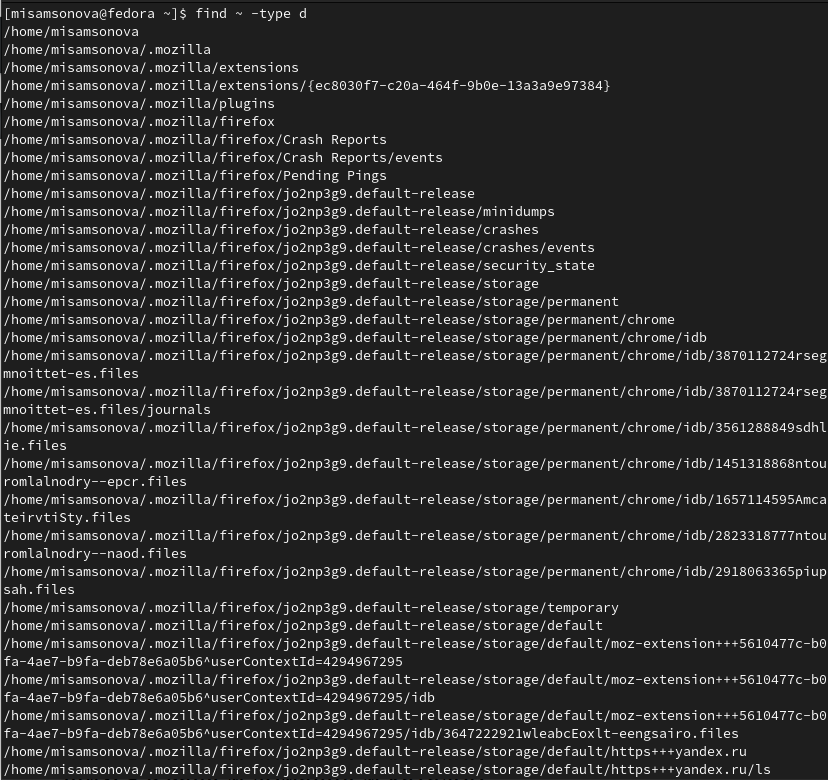
{#fig:021 width=70%}

1. Воспользовавшись справкой команды find, выведём имена всех директорий, имеющихся в нашем домашнем каталоге: (рис. [-@fig:022]), (рис. [-@fig:023]), (рис. [-@fig:024]), (рис. [-@fig:025])



Справка о команде find

{#fig:022 width=70%}



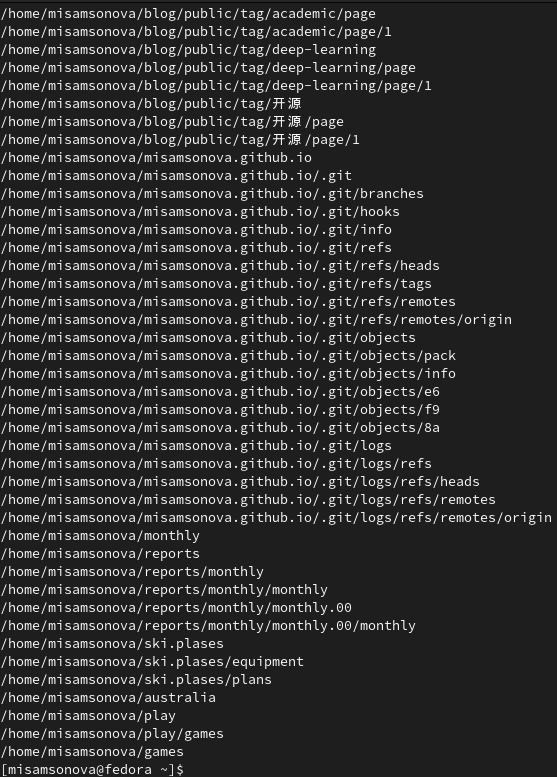
Вывод имен всех директорий, имеющихся в нашем домашнем каталоге

{#fig:023 width=70%}



Вывод имен всех директорий, имеющихся в нашем домашнем каталоге

{#fig:024 width=70%}



Вывод имен всех директорий, имеющихся в нашем домашнем каталоге

{#fig:025 width=70%}

# Вывод

В процессе выполнения лабораторной работы №6 мы ознакомились с инструментами поиска файлов и фильтрации текстовых данных и приобрели практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

# Ответы на 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.