

Лабораторная работа №6

Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов

Поляков Глеб Сергеевич

Содержание

1 Цель работы	5
2 Задание	6
3 Теоретическое введение	8
3.0.1 Перенаправление ввода-вывода	8
3.0.2 Конвейер	9
3.0.3 Фильтрация текста	10
3.0.4 Проверка использования диска	11
3.0.5 Управление задачами	11
3.0.6 Управление процессами	12
4 Выполнение лабораторной работы	13
5 Выводы	20
Список литературы	21

Список иллюстраций

4.1	Вход в систему	13
4.2	Запись в файл	14
4.3	Вывод из файла	14
4.4	Выполнение команды	15
4.5	Выполнение команды	16
4.6	Выполнение команды	16
4.7	Выполнение команды	17
4.8	Выполнение команды	17
4.9	Выполнение команды	18
4.10	Выполнение команды	19

Список таблиц

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 текстовой файл conf.txt.
4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа c? Предложите несколько вариантов, как это сделать.
5. Выведите на экран (постранично) 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

3.0.1 Перенаправление ввода-вывода

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 вывода 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`. Например, команда `ls` выводит в стандартный поток вывода (консоль) список файлов в текущей директории. Потоки вывода и ввода можно перенаправлять на другие файлы или устройства. Проще всего это делается с помощью символов `>`, `>>`, `<`, `<<`. Рассмотрим пример. # Перенаправление `stdout` (вывода) в файл. # Если файл отсутствовал, то он создаётся, # иначе – перезаписывается. # Создаёт файл, содержащий список дерева каталогов. `ls -lR > dir-tree.list` 1`>filename` # Перенаправление вывода (`stdout`) в файл “filename”. 1`>>filename` # Перенаправление вывода (`stdout`) в файл “filename”, # файл открывается в режиме добавления. 2`>filename` # Перенаправление `stderr` в файл “filename”. 2`>>filename` # Перенаправление `stderr` в файл “filename”, # файл открывается в режиме добавления. &`>filename` # Перенаправление `stdout` и `stderr` в файл “filename”.

3.0.2 Конвейер

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 передаётся на ввод команде 2

Конвейеры можно группировать в цепочки и выводить с помощью перенаправления в файл, например:

```
ls -la |sort > sorting_list
```

вывод команды ls -la передаётся команде сортировки sort\verb, которая пишет результат в файл sorting_list\verb. Чаще всего скрипты на Bash используются в качестве автоматизации каких-то рутинных операций в консоли, отсюда иногда возникает необходимость в обработке stdout одной команды и передача на stdin другой команде, при этом результат выполнения команды должен обработан.

###Поиск файла

Команда find используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов. Формат команды:

```
find путь [-опции]
```

Путь определяет каталог, начиная с которого по всем подкаталогам будет вестись поиск. **Примеры:**

1. Вывести на экран имена файлов из вашего домашнего каталога и его подкаталогов, начинающихся на f:

```
find ~ -name "f*" -print
```

Здесь ~ — обозначение вашего домашнего каталога, -name — после этой опции указывается имя файла, который нужно найти, “f*” — строка символов, определяющая имя файла, -print — опция, задающая вывод результатов поиска на экран.

2. Вывести на экран имена файлов в каталоге /etc, начинающихся с символа p:

```
find /etc -name "p\*" -print
```

3. Найти в Вашем домашнем каталоге файлы, имена которых заканчиваются символом ~ и удалить их:

```
find ~ -name "*~" -exec rm "{}" \;
```

Здесь опция -exec rm "{}" ; задаёт применение команды rm ко всем файлам, имена которых соответствуют указанной после опции -name строке символов.

Для просмотра опций команды find воспользуйтесь командой man.

3.0.3 Фильтрация текста

Найти в текстовом файле указанную строку символов позволяет команда grep.
Формат команды:

```
grep строка имя_файла
```

Кроме того, команда grep способна обрабатывать стандартный вывод других команд (любой текст). Для этого следует использовать конвейер, связав вывод команды с вводом grep.

Примеры:

- Показать строки во всех файлах в вашем домашнем каталоге с именами, начинающимися на f, в которых есть слово begin:

```
grep begin f*
```

- Найти в текущем каталоге все файлы, содержащих в имени «лаб»:

```
ls -l | grep лаб
```

3.0.4 Проверка использования диска

Команда df показывает размер каждого смонтированного раздела диска. Формат команды:

```
df [-опции] [файловая_система] Пример:  
df -vi
```

Команда du показывает число килобайт, используемое каждым файлом или каталогом. Формат команды:

```
du [-опции] [имя_файла...] Пример.  
du -a ~/
```

На afs можно посмотреть использованное пространство командой

```
fs quota
```

3.0.5 Управление задачами

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

```
gedit &
```

Будет запущен текстовой редактор gedit в фоновом режиме. Консоль при этом не будет заблокирована.

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

```
kill %номер задачи
```

3.0.6 Управление процессами

Любой команде, выполняемой в системе, присваивается идентификатор процесса (process ID). Получить информацию о процессе и управлять им, пользуясь идентификатором процесса, можно из любого окна командного интерпретатора.

###Получение информации о процессах Команда ps используется для получения информации о процессах. Формат команды:

```
ps [-опции]
```

Для получения информации о процессах, управляемых вами и запущенных (работающих или остановленных) на вашем терминале, используйте опцию aux.
Пример:

```
ps aux
```

Для запуска команды в фоновом режиме необходимо в конце командной строки указать знак & (амперсанд). Пример работы, требующей много машинного времени для выполнения, и которую целесообразно запустить в фоновом режиме:

```
find /var/log -name "*log" -print > l.log &
```

4 Выполнение лабораторной работы

1. Осуществил вход в систему, используя соответствующее имя пользователя (рис. fig. 4.1).

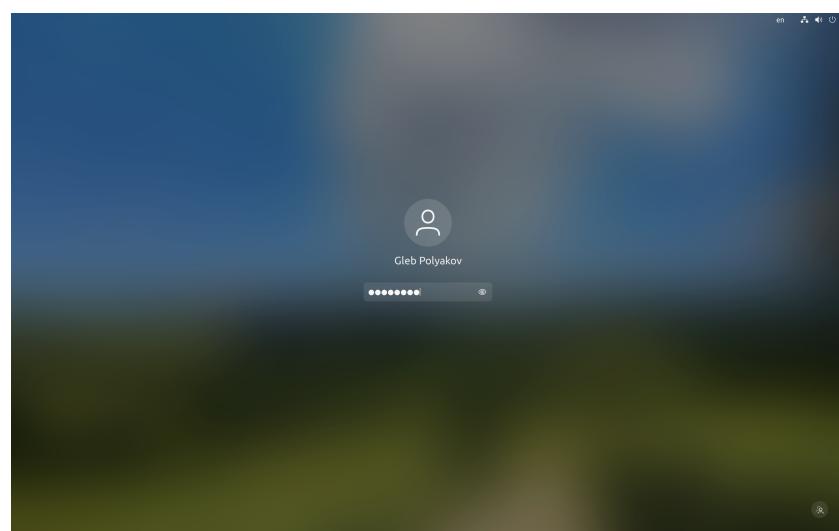
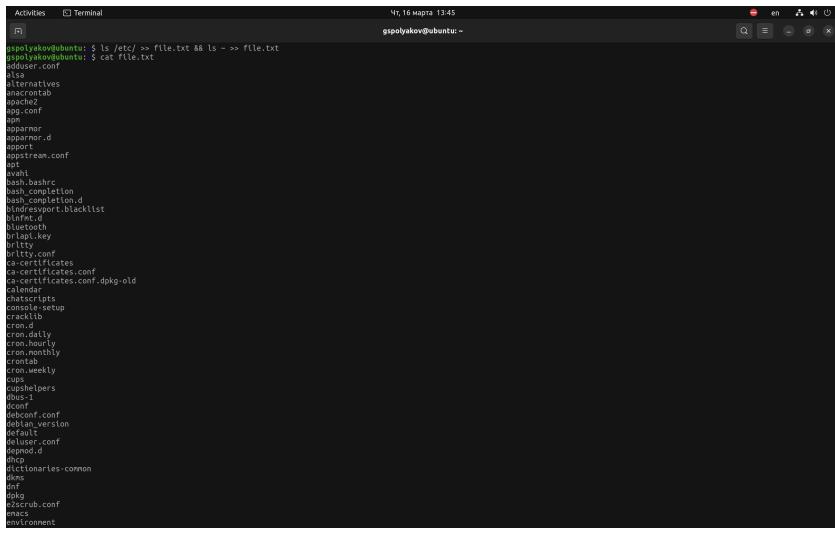


Рис. 4.1: Вход в систему

2. Записал в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишинал в этот же файл названия файлов, содержащихся в моём домашнем каталоге.(рис. fig. 4.2).

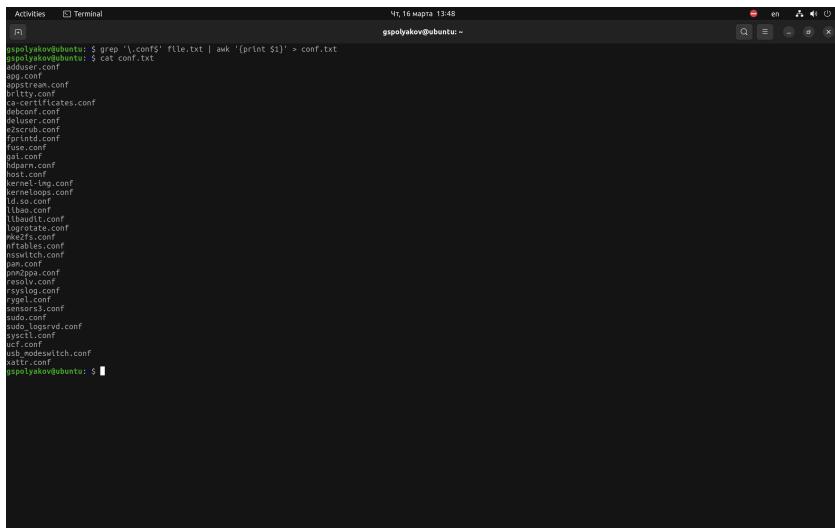


```
gspolyakov@ubuntu: ~
```

```
gspolyakov@ubuntu: $ ls /etc >> file.txt && ls - >> file.txt
alternatives
anacrontab
apt.conf
apt
aptaror
apptaror.d
apt-report
apt-recommends.conf
apt
avahi
bash.bashrc
bash_completion
bash_completion.d
bluetooth-blacklist
bluetooth.d
bluez
bluez-key
brltty
brltty.conf
ca-certificates
ca-certificates.conf
ca-certificates.conf.dpkg-old
calendar
chatscripts
control-setstep
cracklib
cron
cron.daily
cron.hourly
cron.monthly
cronatab
cron.weekly
cron
cupshelpers
dbus
dconf
debcnf.conf
debconf.version
default
deluser.conf
segond.d
mimic
dictionaries-common
tags
mfns
mfns
mhc
mhcscrub.conf
enacs
environment
```

Рис. 4.2: Запись в файл

3. Вывел имена всех файлов из file.txt, имеющих расширение .conf, после чего записал их в новый текстовой файл conf.txt. (рис. fig. 4.3).



```
gspolyakov@ubuntu: ~
```

```
gspolyakov@ubuntu: $ grep '\.conf$' file.txt | awk '{print $1}' > conf.txt
alternatives
anacrontab
apt.conf
apt
aptaror
apptaror.d
apt-report
apt-recommends.conf
apt
avahi
bash.bashrc
bash_completion
bash_completion.d
bluetooth-blacklist
bluetooth.d
bluez
bluez-key
brltty
brltty.conf
ca-certificates
ca-certificates.conf
ca-certificates.conf.dpkg-old
calendar
chatscripts
control-setstep
cracklib
cron
cron.daily
cron.hourly
cron.monthly
cronatab
cron.weekly
cron
cupshelpers
dbus
dconf
debcnf.conf
debconf.version
default
deluser.conf
segond.d
mimic
dictionaries-common
tags
mfns
mfns
mhc
mhcscrub.conf
enacs
environment
```

Рис. 4.3: Вывод из файла

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

6. Вывел на экран (постранично) имена файлов из каталога /etc, начинающиеся с символа h.(рис. fig. 4.4).

Рис. 4.4: Выполнение команды

7. Запустил в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log.`(рис. fig. 4.5).

```

Activities Unknown 47, 16 марта 13:57
gspolyakov@ubuntu: ~
[1] 31917 pts/0 0:00 nohup find / -name "log" > ~/logfile &
gspolyakov@ubuntu: ~
[1]+  Exit 1                  nohup find / -name "log" > ~/logfile
gspolyakov@ubuntu: ~
gspolyakov@ubuntu: ~ cat logfile
find: '/tmp/systemd-private-11e6552a328d434bd9f94e17a19175bf-sysvand-resolved.service-BuRfG3': Permission denied
find: '/tmp/systemd-private-11e6552a328d434bd9f94e17a19175bf-power-profiles-dæmon.service-Rug3M': Permission denied
find: '/tmp/systemd-private-11e6552a328d434bd9f94e17a19175bf-switcheroo-control.service-q5a10': Permission denied
find: '/tmp/systemd-private-11e6552a328d434bd9f94e17a19175bf-systemd-oom.service-p0rUJU': Permission denied
find: '/tmp/systemd-private-11e6552a328d434bd9f94e17a19175bf-timesyncd.service-8rkKdp': Permission denied
find: '/tmp/systemd-private-11e6552a328d434bd9f94e17a19175bf-colord.service-7lVHlI': Permission denied
find: '/tmp/sys/dev/scsi/logging_level': Permission denied
/proc/sys/net/ipv4/conf/default/log_martians
/proc/sys/net/ipv4/conf/all/log_martians
/proc/sys/net/ipv4/conf/eth0/log_martians
find: '/proc/tun/tun': Permission denied
find: '/proc/1/task/1/fd': Permission denied
read: '/proc/1/task/1/fd': Permission denied
find: '/proc/1/task/1/ns': Permission denied
/proc/1/task/1/loginuid
find: '/proc/1/task/1/fd/0': Permission denied
find: '/proc/1/task/1/fd/1': Permission denied
find: '/proc/1/task/1/fd/2': Permission denied
find: '/proc/1/task/1/fd/3': Permission denied
find: '/proc/1/task/1/fd/4': Permission denied
find: '/proc/1/task/1/fd/5': Permission denied
find: '/proc/1/task/1/fd/6': Permission denied
find: '/proc/1/task/1/fd/7': Permission denied
find: '/proc/1/task/1/fd/8': Permission denied
find: '/proc/1/task/1/fd/9': Permission denied
find: '/proc/1/task/1/fd/10': Permission denied
find: '/proc/1/task/1/fd/11': Permission denied
find: '/proc/1/task/1/fd/12': Permission denied
find: '/proc/1/task/1/fd/13': Permission denied
find: '/proc/1/task/1/fd/14': Permission denied
find: '/proc/1/task/1/fd/15': Permission denied
find: '/proc/1/task/1/fd/16': Permission denied
find: '/proc/1/task/1/fd/17': Permission denied
find: '/proc/1/task/1/fd/18': Permission denied
find: '/proc/1/task/1/fd/19': Permission denied
find: '/proc/1/task/1/fd/20': Permission denied
find: '/proc/1/task/1/fd/21': Permission denied
find: '/proc/1/task/1/fd/22': Permission denied
find: '/proc/1/task/1/fd/23': Permission denied
find: '/proc/1/task/1/fd/24': Permission denied
find: '/proc/1/task/1/fd/25': Permission denied
find: '/proc/1/task/1/fd/26': Permission denied
find: '/proc/1/task/1/fd/27': Permission denied
find: '/proc/1/task/1/fd/28': Permission denied
find: '/proc/1/task/1/fd/29': Permission denied
find: '/proc/1/task/1/fd/30': Permission denied
find: '/proc/1/task/1/fd/31': Permission denied
/proc/1/task/1/ns

```

Рис. 4.5: Выполнение команды

8. Удалил файл ~/logfile.(рис. fig. 4.6).

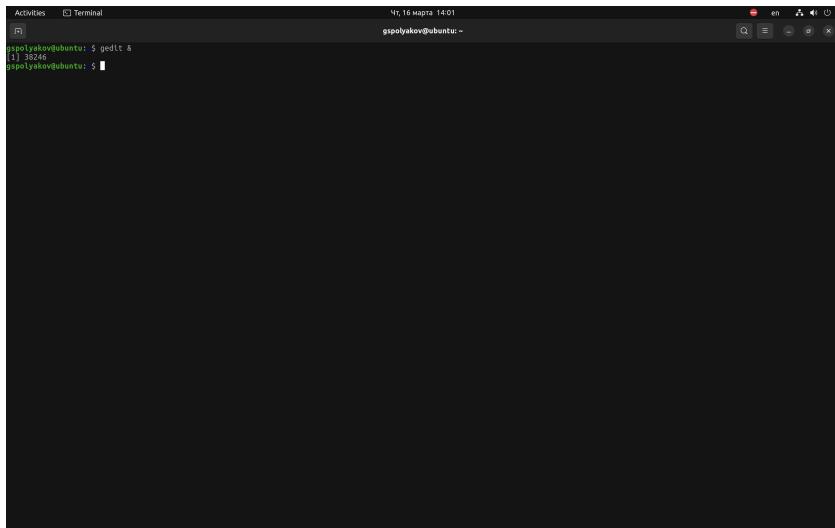
```

Activities Terminal 47, 16 марта 13:59
gspolyakov@ubuntu: ~
gspolyakov@ubuntu: ~ rm ~/logfile
gspolyakov@ubuntu: ~

```

Рис. 4.6: Выполнение команды

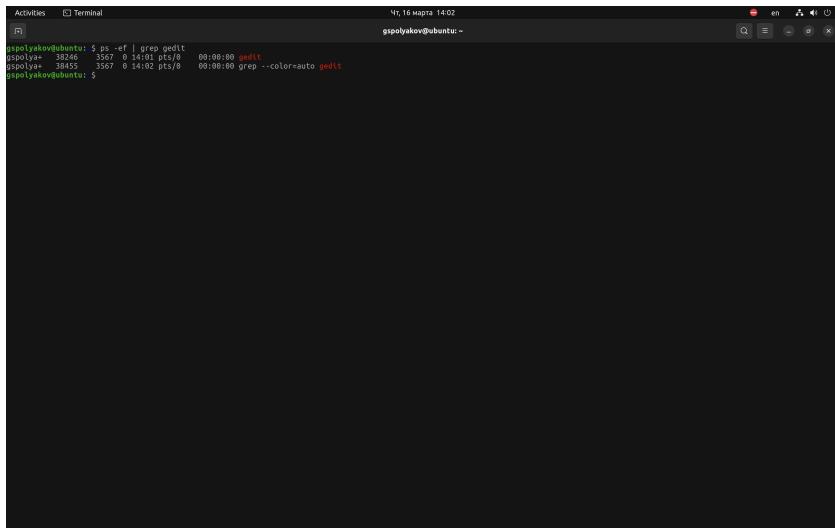
9. Запустил из консоли в фоновом режиме редактор gedit.(рис. fig. 4.7).



```
Activities Terminal Чт, 16 марта 14:01
gspolyakov@ubuntu: $ gedit &
[1] 38246
gspolyakov@ubuntu: $
```

Рис. 4.7: Выполнение команды

10. Определил идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Как ещё можно определить идентификатор процесса?(рис. fig. 4.8).



```
Activities Terminal Чт, 16 марта 14:02
gspolyakov@ubuntu: $ ps -ef | grep gedit
gspolyakov 38246 3567 0 14:01 pts/0 00:00:00 gedit
gspolyakov 38455 3567 0 14:02 pts/0 00:00:00 grep --color=auto gedit
gspolyakov@ubuntu: $
```

Рис. 4.8: Выполнение команды

11. Прочел справку (man) команды kill, после чего использовал её для завершения процесса gedit.(рис. fig. ??) (рис. fig. ??).

The image shows two side-by-side terminal windows. Both windows have a title bar 'Activities Terminal' and a status bar at the bottom indicating '4ч, 16 марта 14:19' and 'gspolyakov@ubuntu'. The left window displays the full man page for the 'kill' command, which includes sections for NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXAMPLES, SEE ALSO, and STANDARDS. The right window shows a portion of the same man page, specifically the 'KILL(1)' section.

```

NAME
    kill - send a signal to a process

SYNOPSIS
    kill [options] <pid> [...]

DESCRIPTION
    The default signal for kill is TERM. Use -l or -L to list available signals. Particularly useful signals include HUP, INT, KILL, STOP, CONT, and Q. Alternate signals may be specified in three ways: -9, -SIGKILL or -KILL. Negative PID values may be used to choose whole process groups; see the PGID column in ps command output. A PID of -1 is special; it indicates all processes except the kill process itself and init.

OPTIONS
    <pid> [...]          Send signal to every <pid> listed.
    -c <signal>
    -s <signal>
    --signal <signal>
        Specify the signal to be sent. The signal can be specified by using name or number. The behavior of signals is explained in signal(7) manual page.
    -q, --queue value
        Queue the signal rather than kill(2) and the value argument is used to specify an integer to be sent with the signal. If the receiving process has installed a handler for this signal using the SA_SIGINFO flag to sigaction(2), then it can obtain this data via the si_value field of the siginfo_t structure.
    -l, --list [signal]
        List signal names. This option has optional argument, which will convert signal number to signal name, or other way round.
    -L, --table
        List signal names in a nice table.

    NOTES Your shell (command line interpreter) may have a built-in kill command. You may need to run the command described here as /bin/kill to solve the conflict.

EXAMPLES
    kill -9 -1
        Kill all processes you can kill.
    kill -l 11
        Translate number 11 into a signal name.
    kill -L
        List the available signal choices in a nice table.
    kill 123 543 2341 3453
        Send the default signal, SIGTERM, to all those processes.

SEE ALSO
    kill(2), killall(1), nice(1), pkill(1), renice(1), signal(7), sigqueue(3), skill(1)

STANDARDS
    This command meets appropriate standards. The -L flag is Linux-specific.

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

```

12. Выполнил команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man.(рис. fig. 4.9).

The image shows a single terminal window with a title bar 'Activities Terminal' and a status bar at the bottom indicating '4ч, 16 марта 14:14' and 'gspolyakov@ubuntu'. The user runs the 'df' command to show disk usage and the 'du' command to show disk usage per directory.

```

Activities Unknown
gspolyakov@ubuntu: ~
gspolyakov@ubuntu: $ man df
gspolyakov@ubuntu: $ df
Filesystem      1K-blocks   Used Available Use% Mounted on
tmpfs            281372    1712   199468  1% /run
/dev/mapper/server--vg-root 6413940 1229496 4684444 2% /
tmpfs             1085848     0   1085848  0% /dev/shn
tmpfs              5128       4   5116  1% /run/lock
tmpfs             512000     0   512000  0% /run/lock/tmp
tmpfs            201168   2456   198712  2% /run/user/1000
gspolyakov@ubuntu: ~
gspolyakov@ubuntu: $ du
./reports/monthly/monthly
8 ./reports/monthly
24 ./reports
4 ./thumbnails
392 ./cache/gstreamer-1.0
24 ./cache/gstreamer-1.0.4
12 ./cache/update-manager-core
492 ./cache/thumbnails/large
16 ./cache/thumbnails/large/none-thumbnail-factory
12 ./cache/thumbnails/fat
508 ./cache/thumbnails
40 ./cache/mesa_shader_cache/db
8 ./cache/mesa_shader_cache/dd
8 ./cache/mesa_shader_cache/dl
8 ./cache/mesa_shader_cache/ds
8 ./cache/mesa_shader_cache/dt
8 ./cache/mesa_shader_cache/dv
8 ./cache/mesa_shader_cache/dz
8 ./cache/mesa_shader_cache/e
8 ./cache/mesa_shader_cache/ea
12 ./cache/mesa_shader_cache/e9
8 ./cache/mesa_shader_cache/e90
8 ./cache/mesa_shader_cache/e91
8 ./cache/mesa_shader_cache/e92
8 ./cache/mesa_shader_cache/e93
8 ./cache/mesa_shader_cache/e94
8 ./cache/mesa_shader_cache/e95
8 ./cache/mesa_shader_cache/e96
8 ./cache/mesa_shader_cache/e97
8 ./cache/mesa_shader_cache/e98
8 ./cache/mesa_shader_cache/e99
8 ./cache/mesa_shader_cache/e9a
8 ./cache/mesa_shader_cache/e9b
8 ./cache/mesa_shader_cache/e9c
8 ./cache/mesa_shader_cache/e9d
8 ./cache/mesa_shader_cache/e9e
8 ./cache/mesa_shader_cache/e9f
8 ./cache/mesa_shader_cache/e9g
8 ./cache/mesa_shader_cache/e9h
8 ./cache/mesa_shader_cache/e9i
8 ./cache/mesa_shader_cache/e9j
8 ./cache/mesa_shader_cache/e9k
8 ./cache/mesa_shader_cache/e9l
8 ./cache/mesa_shader_cache/e9m
8 ./cache/mesa_shader_cache/e9n
8 ./cache/mesa_shader_cache/e9o
8 ./cache/mesa_shader_cache/e9p
8 ./cache/mesa_shader_cache/e9q
8 ./cache/mesa_shader_cache/e9r
8 ./cache/mesa_shader_cache/e9s
8 ./cache/mesa_shader_cache/e9t
8 ./cache/mesa_shader_cache/e9u
8 ./cache/mesa_shader_cache/e9v
8 ./cache/mesa_shader_cache/e9w
8 ./cache/mesa_shader_cache/e9x
8 ./cache/mesa_shader_cache/e9y
8 ./cache/mesa_shader_cache/e9z
8 ./cache/mesa_shader_cache/e9

```

Рис. 4.9: Выполнение команды

13. Воспользовавшись справкой команды find, вывел имена всех директорий, имеющихся в вашем домашнем каталоге.(рис. fig. 4.10).

```
[root@ip-172-31-10-10 ~]# find -type d
/home/gspolyakov
/home/gspolyakov/reports
/home/gspolyakov/reports/monthly
/home/gspolyakov/reports/monthly/monthly
/home/gspolyakov/Templates
/home/gspolyakov/.cache/gstremer-1.0
/home/gspolyakov/.cache/ubuntu-report
/home/gspolyakov/.cache/ubuntu-report-core
/home/gspolyakov/.cache/thumbnails
/home/gspolyakov/.cache/thumbnails/fail
/home/gspolyakov/.cache/thumbnails/fail/gnome-thumbnail-factory
/home/gspolyakov/.cache/thumbnails/fail/gnome-thumbnail-factory
/home/gspolyakov/.cache/mesa_shader_cache/db
/home/gspolyakov/.cache/mesa_shader_cache/dd
/home/gspolyakov/.cache/mesa_shader_cache/dx
/home/gspolyakov/.cache/mesa_shader_cache/9a
/home/gspolyakov/.cache/mesa_shader_cache/9d
/home/gspolyakov/.cache/mesa_shader_cache/9e
/home/gspolyakov/.cache/mesa_shader_cache/9f
/home/gspolyakov/.cache/mesa_shader_cache/6a
/home/gspolyakov/.cache/mesa_shader_cache/6d
/home/gspolyakov/.cache/mesa_shader_cache/ae
/home/gspolyakov/.cache/mesa_shader_cache/af
/home/gspolyakov/.cache/mesa_shader_cache/30
/home/gspolyakov/.cache/mesa_shader_cache/97
/home/gspolyakov/.cache/mesa_shader_cache/3c
/home/gspolyakov/.cache/mesa_shader_cache/39
/home/gspolyakov/.cache/mesa_shader_cache/e5
/home/gspolyakov/.cache/mesa_shader_cache/6c
/home/gspolyakov/.cache/mesa_shader_cache/68
/home/gspolyakov/.cache/mesa_shader_cache/ad
/home/gspolyakov/.cache/mesa_shader_cache/de
/home/gspolyakov/.cache/mesa_shader_cache/4f
/home/gspolyakov/.cache/mesa_shader_cache/67
/home/gspolyakov/.cache/mesa_shader_cache/b4
/home/gspolyakov/.cache/mesa_shader_cache/20
/home/gspolyakov/.cache/mesa_shader_cache/16
/home/gspolyakov/.cache/mesa_shader_cache/53
/home/gspolyakov/.cache/mesa_shader_cache/3b
/home/gspolyakov/.cache/mesa_shader_cache/89
/home/gspolyakov/.cache/mesa_shader_cache/9a
/home/gspolyakov/.cache/mesa_shader_cache/53
/home/gspolyakov/.cache/mesa_shader_cache/85
/home/gspolyakov/.cache/mesa_shader_cache/r5
/home/gspolyakov/.cache/mesa_shader_cache/47
/home/gspolyakov/.cache/mesa_shader_cache/59
/home/gspolyakov/.cache/mesa_shader_cache/10
/home/gspolyakov/.cache/mesa_shader_cache/2a
/home/gspolyakov/.cache/mesa_shader_cache/e6
/home/gspolyakov/.cache/mesa_shader_cache/60
```

Рис. 4.10: Выполнение команды

5 Выводы

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

Список литературы