

Отчет по лабораторной работе №6

Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов

Галацан Николай, НПИбд-01-22

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	16
6	Ответы на контрольные вопросы	17

Список иллюстраций

4.1	Запись в файл file.txt и вывод содержимого	9
4.2	Запись в файл conf.txt и вывод содержимого	10
4.3	Поиск файлов, начинающихся с символа c	10
4.4	Поиск файлов в /etc, начинающихся с h	11
4.5	Вывод команды <code>find /etc -maxdepth 1 -name "h*" -print less</code>	11
4.6	Вывод файла logfile, удаление и проверка	12
4.7	Запуск и проверка идентификатора процесса gedit	13
4.8	<code>man kill</code>	13
4.9	Завершение процесса gedit	14
4.10	Использование команд <code>df</code> и <code>du</code>	14
4.11	<code>man find</code>	15
4.12	Поиск и вывод всех директорий домашнего каталога	15

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, начинающиеся 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 директо-
рий, имею- щихся в вашем домашнем каталоге.

3 Теоретическое введение

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`. 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 `>`, `>>`, `<`, `<<`.

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

```
1 команда 1 | команда 2
```

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

```
find путь [-опции]
```

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

гтер строка имя_файла

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

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

```
df [-опции] [файловая_система]
```

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

```
du [-опции] [имя_файла...]
```

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. Для завершения задачи необходимо выполнить команду

```
kill %номер задачи
```

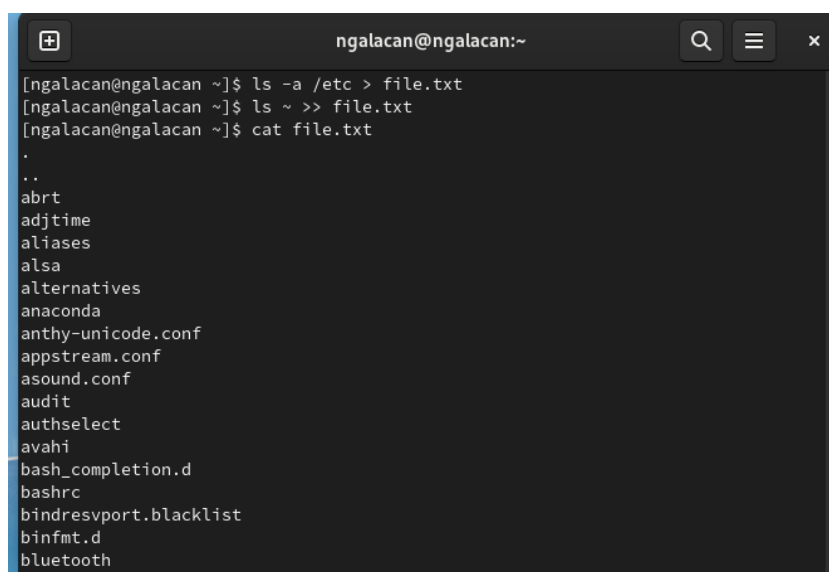
Любой команде, выполняемой в системе, присваивается *идентификатор процесса* (`process ID`). Получить информацию о процессе и управлять им, пользуясь идентификатором процесса, можно из любого окна командного интерпретатора.

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

```
ps [-опции]
```


4 Выполнение лабораторной работы

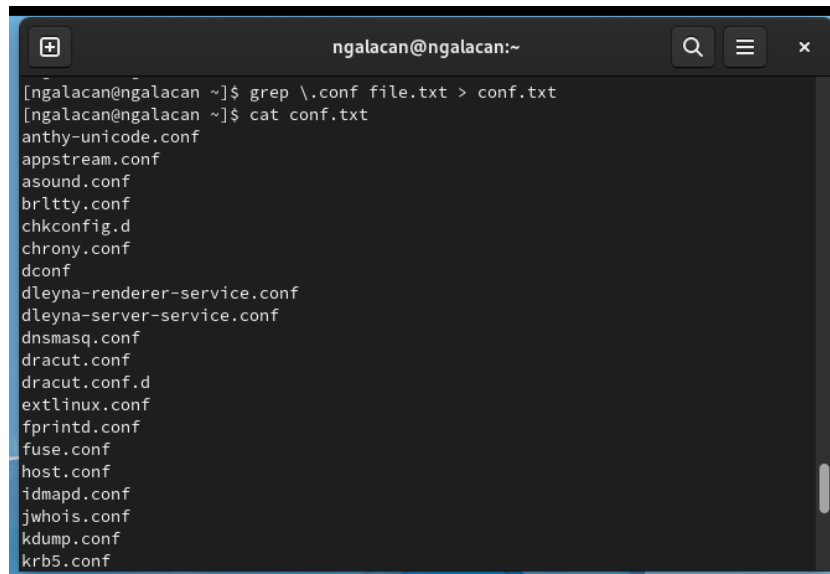
1. Вхожу в систему под именем пользователя `ngalacan`.
2. Записываю в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`. Допишу в этот же файл названия файлов, содержащихся в домашнем каталоге. Для этого использую перенаправление вывода с помощью `>`, чтобы открыть файл в режиме добавления. Вывожу содержимое файла (рис. 4.1).



```
ngalacan@ngalacan:~  
[ngalacan@ngalacan ~]$ ls -a /etc > file.txt  
[ngalacan@ngalacan ~]$ ls ~ >> file.txt  
[ngalacan@ngalacan ~]$ cat file.txt  
.  
..  
abrt  
adjtime  
aliases  
alsa  
alternatives  
anaconda  
anthy-unicode.conf  
appstream.conf  
asound.conf  
audit  
authselect  
avahi  
bash_completion.d  
bashrc  
bindresvport.blacklist  
binfmt.d  
bluetooth
```

Рис. 4.1: Запись в файл `file.txt` и вывод содержимого

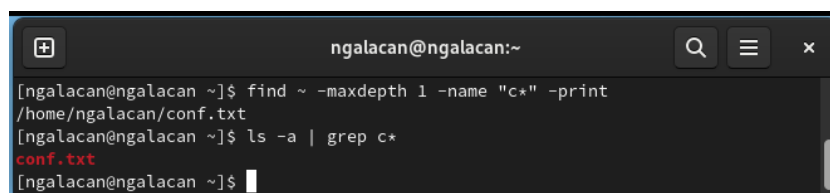
3. Вывожу имена всех файлов из `file.txt`, имеющих расширение `.conf` с помощью команды `grep`, после чего записываю их в новый текстовый файл `conf.txt` с помощью перенаправления вывода `>`. Вывожу содержимое нового файла (рис. 4.2).



```
ngalacan@ngalacan:~  
[ngalacan@ngalacan ~]$ grep \.conf file.txt > conf.txt  
[ngalacan@ngalacan ~]$ cat conf.txt  
anthy-unicode.conf  
appstream.conf  
asound.conf  
brltty.conf  
chkconfig.d  
chrony.conf  
dconf  
dleyna-renderer-service.conf  
dleyna-server-service.conf  
dnsmasq.conf  
dracut.conf  
dracut.conf.d  
extlinux.conf  
fprintd.conf  
fuse.conf  
host.conf  
idmapd.conf  
jwhois.conf  
kdump.conf  
krb5.conf
```

Рис. 4.2: Запись в файл conf.txt и вывод содержимого

4. Нахожу файлы в домашнем каталоге, начинающиеся с символа **с**. В первом случае использую команду `find`, где в опции с именем файла ввожу условие поиска `-name с*`. Также использую опцию `-maxdepth 1`, чтобы поиск осуществлялся только на первом уровне каталога, без поиска в подкаталогах. Во втором случае использую `ls -a`, конвейер и поиск с помощью `grep с*` (рис. 4.3).



```
ngalacan@ngalacan:~  
[ngalacan@ngalacan ~]$ find ~ -maxdepth 1 -name "c*" -print  
/home/ngalacan/conf.txt  
[ngalacan@ngalacan ~]$ ls -a | grep c*  
conf.txt  
[ngalacan@ngalacan ~]$
```

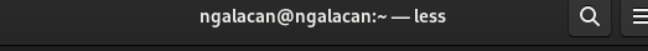
Рис. 4.3: Поиск файлов, начинающихся с символа **с**

5. Вывожу постранично на экран имена файлов из `/etc`, начинающиеся с символа **h**. Для этого воспользуюсь командой `find`, конвейером и командой `less` (рис. 4.4).

```
[ngalacan@ngalacan ~]$ find /etc -maxdepth 1 -name "h*" -print | less
[ngalacan@ngalacan ~]$
```

Рис. 4.4: Поиск файлов в /etc, начинающихся с **h**

Вывод введенной команды на рис. 4.5.



```
ngalacan@ngalacan:~ — less
/etc/hp
/etc/httpd
/etc/host.conf
/etc/hosts
/etc/hostname
~
~
~
(END)
```

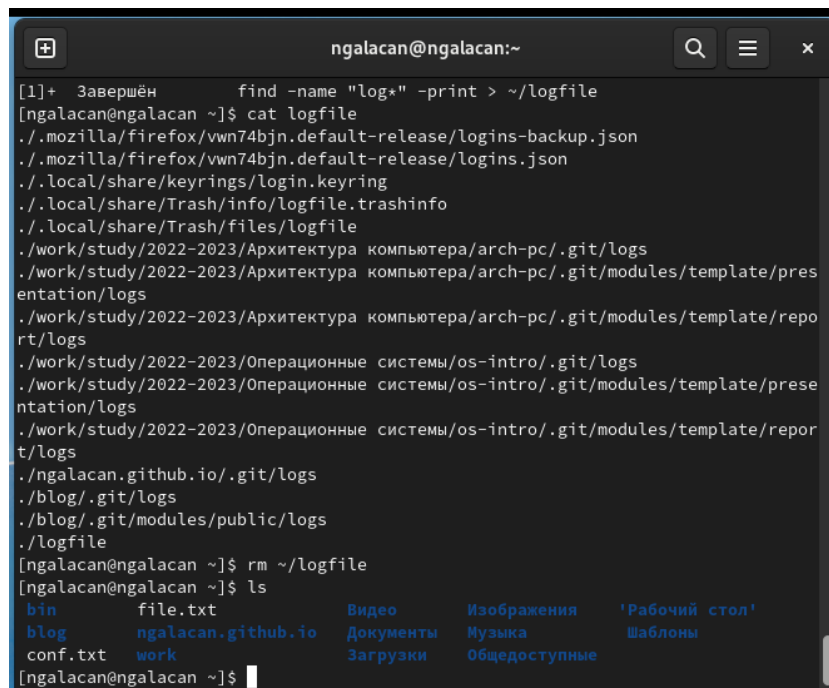
Рис. 4.5: Вывод команды `find /etc -maxdepth 1 -name "h*" -print | less`

6. Запускаю в фоновом режиме процесс для записи в файл `~/logfile` файлов, имена которых начинаются с `log`. Для запуска в фоновом режиме ставлю `&`:

```
find -name "log*" -print > ~/logfile &
```

Вывожу содержимое файла после завершения процесса

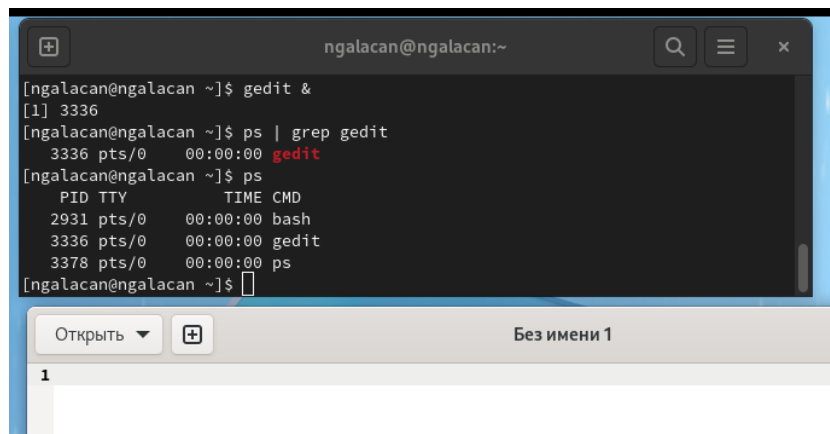
7. Удаляю файл и проверяю (рис. 4.6).



```
[1]+  Завершён      find -name "log*" -print > ~/logfile
[ngalacan@ngalacan ~]$ cat logfile
./.mozilla/firefox/vwn74bjn.default-release/logins-backup.json
./.mozilla/firefox/vwn74bjn.default-release/logins.json
./.local/share/keyrings/login.keyring
./.local/share/Trash/info/logfile.trashinfo
./.local/share/Trash/files/logfile
./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/logs
./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/modules/template/presentation/logs
./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/modules/template/report/logs
./work/study/2022-2023/Операционные системы/os-intro/.git/logs
./work/study/2022-2023/Операционные системы/os-intro/.git/modules/template/presentation/logs
./work/study/2022-2023/Операционные системы/os-intro/.git/modules/template/report/logs
./ngalacan.github.io/.git/logs
./blog/.git/logs
./blog/.git/modules/public/logs
./logfile
[ngalacan@ngalacan ~]$ rm ~/logfile
[ngalacan@ngalacan ~]$ ls
bin      file.txt      Видео      Изображения  'Рабочий стол'
blog     ngalacan.github.io  Документы  Музыка       Шаблоны
conf.txt work          Загрузки  Общедоступные
```

Рис. 4.6: Вывод файла logfile, удаление и проверка

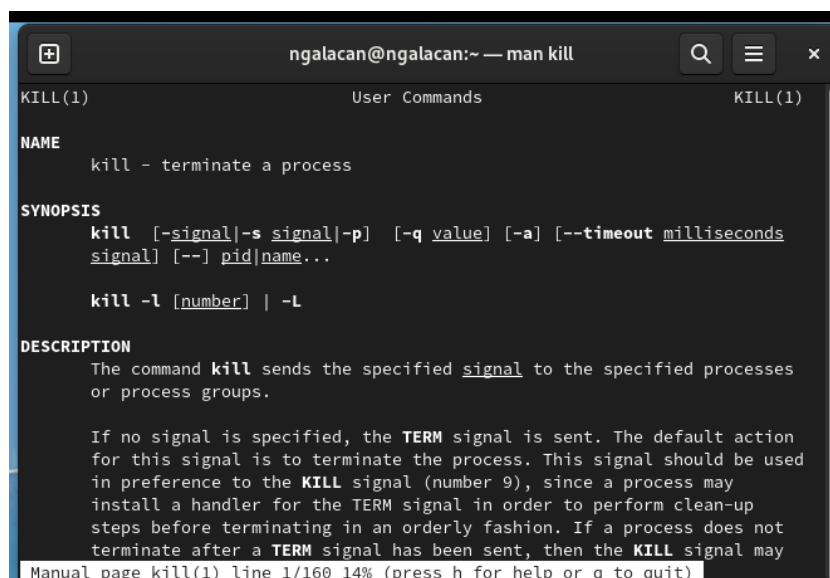
8. Запускаю в фоновом режиме редактор текста: `gedit &`. Открывается окно редактора и выводится идентификатор запущенного процесса.
9. Определяю идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`. Также идентификатор процесса можно определить после запуска процесса в фоновом режиме, а также применив команду `ps` и найдя соответствующую строку (рис. 4.7).



```
ngalacan@ngalacan:~$ gedit &
[1] 3336
ngalacan@ngalacan ~]$ ps | grep gedit
 3336 pts/0    00:00:00 gedit
ngalacan@ngalacan ~]$ ps
  PID TTY          TIME CMD
 2931 pts/0    00:00:00 bash
  3336 pts/0    00:00:00 gedit
  3378 pts/0    00:00:00 ps
ngalacan@ngalacan ~]$
```

Рис. 4.7: Запуск и проверка идентификатора процесса gedit

10. Читаю справку по команде kill (рис. 4.8).



```
ngalacan@ngalacan:~ — man kill
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

Manual page kill(1) line 1/160 14% (press h for help or q to quit)
```

Рис. 4.8: man kill

Завершаю процесс gedit с помощью команды kill 3336, где в качестве аргумента выступает идентификатор процесса. Проверяю, завершен ли процесс (рис. 4.9).

```
3336 pts/0    00:00:00 ps
[ngalacan@ngalacan ~]$ man kill
[ngalacan@ngalacan ~]$ kill 3336
[ngalacan@ngalacan ~]$ ps
  PID TTY          TIME CMD
 2931 pts/0    00:00:00 bash
 3416 pts/0    00:00:00 ps
[1]+  Завершено      gedit
[ngalacan@ngalacan ~]$
```

Рис. 4.9: Завершение процесса gedit

11. Читаю справки по командам `df` и `du`. Выполняю команду `df` для определения размера каждого смонтированного раздела диска. Выполняю `du` для определения числа Кб, используемых каталогом `~/work` (рис. 4.10).

```
ngalacan@ngalacan:~
[1]+  Завершено      gedit
[ngalacan@ngalacan ~]$ man df
[ngalacan@ngalacan ~]$ man du
[ngalacan@ngalacan ~]$ df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs           4096            0      4096            0% /dev
tmpfs              2006760        11196   1995564           1% /dev/shm
tmpfs              802704         1392    801312           1% /run
/dev/sda2          82836480      12762444 69030452          16% /
tmpfs              2006760         16    2006744           1% /tmp
/dev/sda2          82836480      12762444 69030452          16% /home
/dev/sda1          996780        225796   702172           25% /boot
tmpfs              401352         160    401192           1% /run/user/1000
[ngalacan@ngalacan ~]$ du ~/work
0      /home/ngalacan/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/
branches
60     /home/ngalacan/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/
hooks
4      /home/ngalacan/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/
info
4      /home/ngalacan/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/
refs/heads
0      /home/ngalacan/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/
safe/branches
```

Рис. 4.10: Использование команд `df` и `du`

Весь каталог занимает более 102 тыс. килобайт

12. Читаю справку команды `find`, чтобы найти опцию для вывода всех директорий (рис. 4.11).

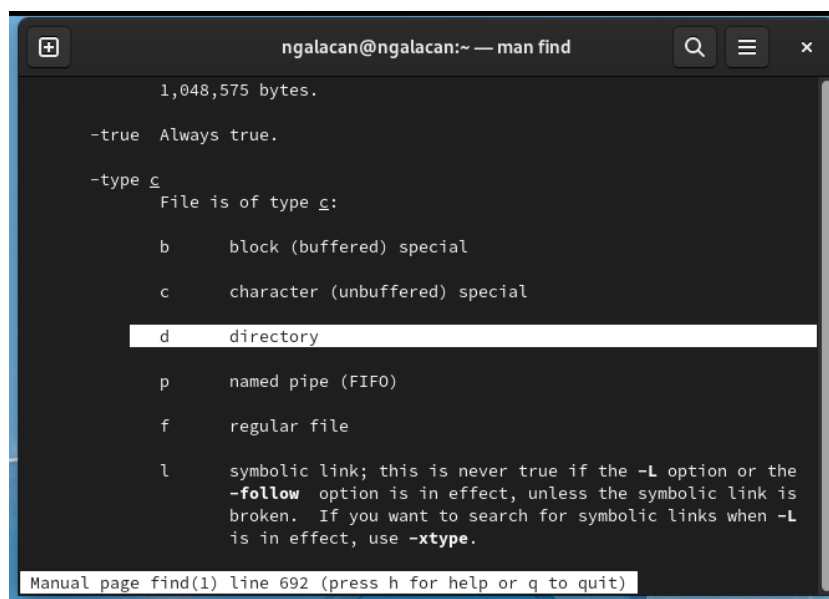


Рис. 4.11: man find

Используя опцию `-type d` для вывода всех директорий домашнего каталога на первом уровне (`-maxdepth 1`) (рис. 4.12).

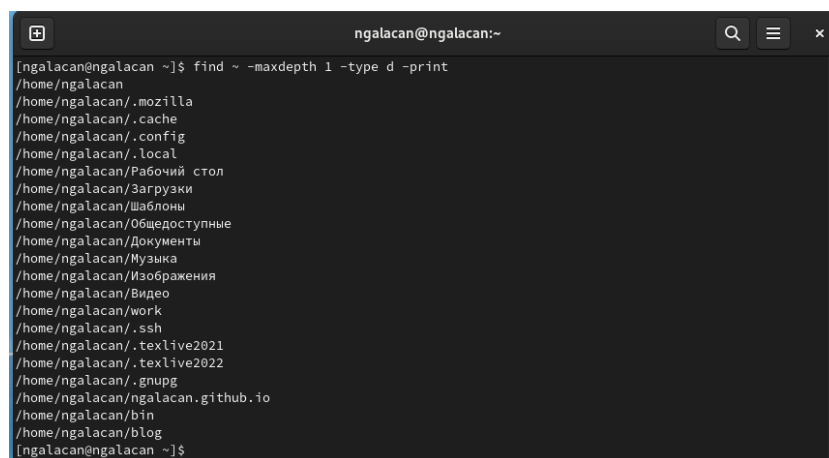


Рис. 4.12: Поиск и вывод всех директорий домашнего каталога

5 Выводы

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

6 Ответы на контрольные вопросы

1. Какие потоки ввода вывода вы знаете?

Ввод и вывод распределяется между тремя стандартными потоками: - `stdin` — стандартный ввод (клавиатура), - `stdout` — стандартный вывод (экран), - `stderr` — стандартная ошибка (вывод ошибок на экран).

2. Объясните разницу между операцией `>` и `>>`.

В первом случае файл открывается либо создается для записи, во втором случае - для добавления новых данных в тот же самый файл без потери уже содержащихся в нем.

3. Что такое конвейер?

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

1 команда 1 | команда 2

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

Процесс - это: - программа на стадии выполнения - “объект”, которому выделено процессорное время - асинхронная работа

5. Что такое PID и GID?

Идентификатор процесса (PID). Каждому новому процессу ядро присваивает уникальный идентификационный номер. В любой момент времени идентификатор процесса является уникальным, хотя после завершения процесса он может использоваться снова для другого процесса. Некоторые идентификаторы зарезервированы системой для особых процессов. Так, процесс с идентификатором 1 - это процесс инициализации `init`, являющийся предком всех других процессов в системе.

Идентификатор группы `GID` и эффективный идентификатор группы (`EGID`)
`GID` - это идентификационный номер группы данного процесса. `EGID` связан с `GID` также, как `EUID` с `UID`.

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

Задача - команда, запущенная в фоновом режиме. 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?

`top` - интерактивный просмотр процессов. `htop` - аналог `top`. Программа `top` динамически выводит в режиме реального времени информации о работающей системе, т.е. о фактической активности процессов. По умолчанию она выдает задачи, наиболее загружающие процессор сервера, и обновляет список каждые две секунды.

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

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

Формат команды: `find путь [-опции]`

Пример. Поиск файлов в `/etc`, начинающихся с `h`: `find /etc -name "h*" -print`

9. Можно ли по контексту (содержанию) найти файл? Если да, то как?

Для поиска файла по содержимому проще всего воспользоваться командой `grep` (вместо `find`).

Пример: `grep -r строка_поиска каталог`

10. Как определить объем свободной памяти на жёстком диске?

С помощью команды `df`.

11. Как определить объем вашего домашнего каталога?

С помощью команды `du ~`.

12. Как удалить зависший процесс?

С помощью команды `kill`, найдя PID процесса.