

# **Лабораторная работа №6**

**НКАбд-01-22**

Никита Михайлович Демидович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
<b>5</b>	<b>Контрольные вопросы</b>	<b>17</b>
<b>6</b>	<b>Выводы</b>	<b>20</b>
	<b>Список литературы</b>	<b>21</b>

## Список иллюстраций

4.1	Добавление содержимого каталога /etc и домашнего каталога в файл file.txt . . . . .	9
4.2	Новый файл conf.txt . . . . .	10
4.3	Первый вариант определение имён файлов, названия которых начинаются с символа “с” . . . . .	10
4.4	Второй вариант определение имён файлов, названия которых начинаются с символа “с” . . . . .	11
4.5	Вывод имён файлов, начинающихся на h . . . . .	11
4.6	Запуск фонового процесса . . . . .	12
4.7	Новый файл logfile.txt . . . . .	12
4.8	Удаление файла logfile.txt . . . . .	12
4.9	Команды ps aux   grep “gedit”, ps -fC gedit и pidof gedit для определения идентификатора процесса . . . . .	13
4.10	Справка (man) команды kill . . . . .	13
4.11	Справка (man) команды df . . . . .	14
4.12	Выполнение команды df . . . . .	14
4.13	Выполнение команды df . . . . .	15
4.14	Справка (man) команды du . . . . .	15
4.15	Выполнение команды du . . . . .	15
4.16	Справка (man) команды find . . . . .	16
4.17	Выполнение команды find . . . . .	16

## **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 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`. Определите идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`. Как ещё можно определить идентификатор процесса?
9. Прочтите справку (`man`) команды `kill`, после чего используйте её для завершения процесса `gedit`.
10. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.
11. Воспользовавшись справкой команды `find`, выведите имена всех директорий, имеющих- ся 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:

*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 > sortilg_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.

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

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

*grep*

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

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

*df <-опции>*

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

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>*

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

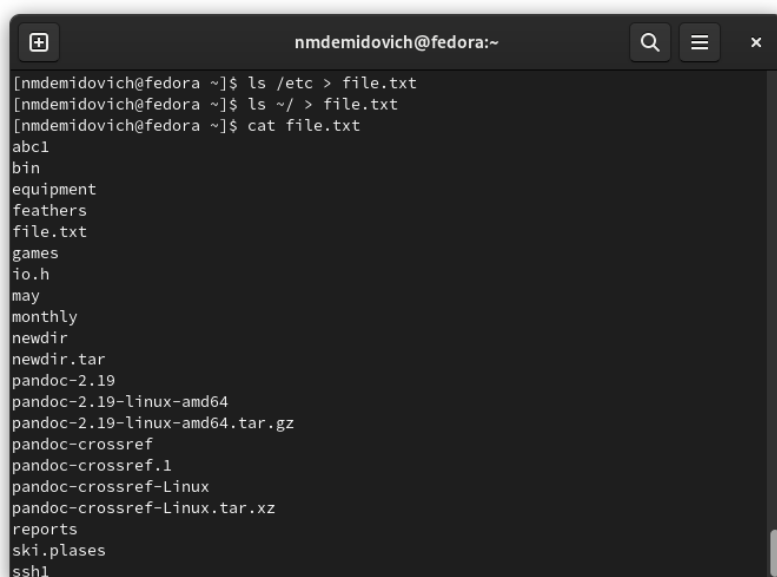
Пример:

*ps aux*



## 4 Выполнение лабораторной работы

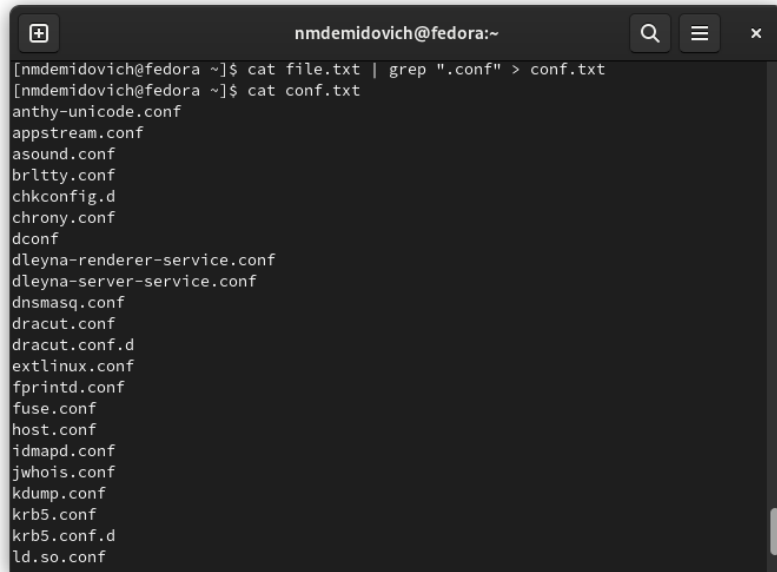
На первом этапе выполнения лабораторной работы я осуществил вход в систему, используя соответствующее имя пользователя - nmdemidovich. Далее я записал в файл file.txt названия файлов, содержащихся в каталоге /etc, после чего дописал в этот же файл названия файлов, содержащихся в своём домашнем каталоге (рис. 4.1).



```
nmdemidovich@fedora:~  
[nmdemidovich@fedora ~]$ ls /etc > file.txt  
[nmdemidovich@fedora ~]$ ls ~/ > file.txt  
[nmdemidovich@fedora ~]$ cat file.txt  
abc1  
bin  
equipment  
feathers  
file.txt  
games  
io.h  
may  
monthly  
newdir  
newdir.tar  
pandoc-2.19  
pandoc-2.19-linux-amd64  
pandoc-2.19-linux-amd64.tar.gz  
pandoc-crossref  
pandoc-crossref.1  
pandoc-crossref-Linux  
pandoc-crossref-Linux.tar.xz  
reports  
ski.places  
ssh1
```

Рис. 4.1: Добавление содержимого каталога /etc и домашнего каталога в файл file.txt

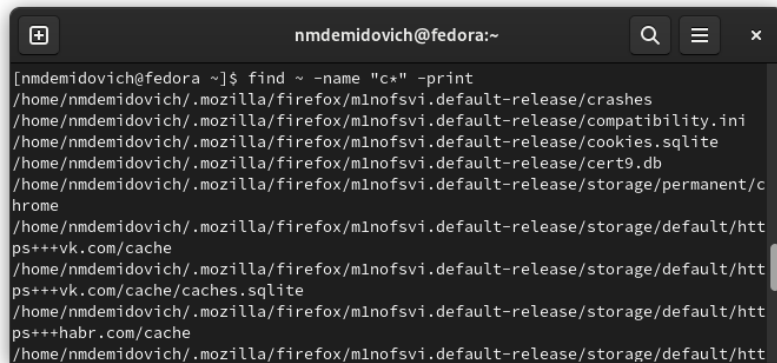
Затем я вывел имена всех файлов из file.txt, имеющих расширение .conf, после чего записал их в новый текстовый файл conf.txt (рис. 4.2).



```
nmdemidovich@fedora:~  
[nmdemidovich@fedora ~]$ cat file.txt | grep ".conf" > conf.txt  
[nmdemidovich@fedora ~]$ 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  
krb5.conf.d  
ld.so.conf
```

Рис. 4.2: Новый файл conf.txt

После этого я определил, какие файлы в моём домашнем каталоге имеют имена, начинавшиеся с символа с и предложил несколько вариантов, как это сделать (рис. 4.3) - (рис. 4.4).



```
nmdemidovich@fedora:~  
[nmdemidovich@fedora ~]$ find ~ -name "c*" -print  
/home/nmdemidovich/.mozilla/firefox/m1nofsvi.default-release/crashes  
/home/nmdemidovich/.mozilla/firefox/m1nofsvi.default-release/compatibility.ini  
/home/nmdemidovich/.mozilla/firefox/m1nofsvi.default-release/cookies.sqlite  
/home/nmdemidovich/.mozilla/firefox/m1nofsvi.default-release/cert9.db  
/home/nmdemidovich/.mozilla/firefox/m1nofsvi.default-release/storage/permanent/c  
hrome  
/home/nmdemidovich/.mozilla/firefox/m1nofsvi.default-release/storage/default/htt  
ps+++vk.com/cache  
/home/nmdemidovich/.mozilla/firefox/m1nofsvi.default-release/storage/default/htt  
ps+++vk.com/cache/caches.sqlite  
/home/nmdemidovich/.mozilla/firefox/m1nofsvi.default-release/storage/default/htt  
ps+++habr.com/cache  
/home/nmdemidovich/.mozilla/firefox/m1nofsvi.default-release/storage/default/htt
```

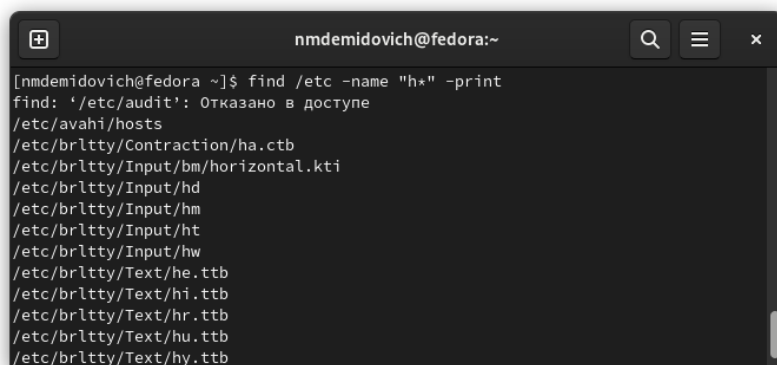
Рис. 4.3: Первый вариант определение имён файлов, названия которых начинаются с символа “с”



```
nmdemidovich@fedora:~  
[nmdemidovich@fedora ~]$ ls ~ | grep "c*"  
abc1  
bin  
cat  
conf.txt  
equipment  
feathers  
file.txt  
games  
io.h  
may  
monthly  
newdir  
newdir.tar
```

Рис. 4.4: Второй вариант определение имён файлов, названия которых начинаются с символа “с”

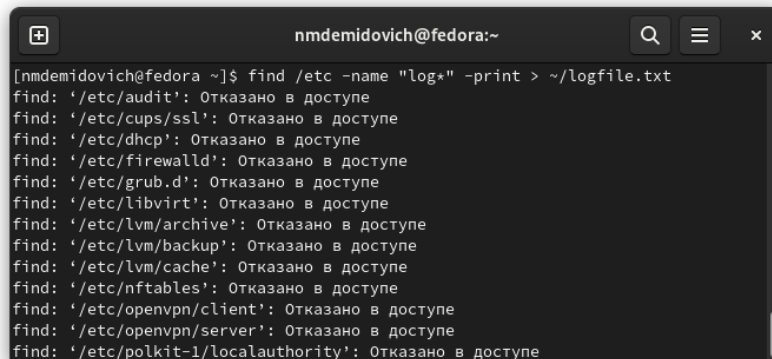
Далее я вывел на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h (рис. 4.5)



```
nmdemidovich@fedora:~  
[nmdemidovich@fedora ~]$ find /etc -name "h*" -print  
find: '/etc/audit': Отказано в доступе  
/etc/avahi/hosts  
/etc/brltty/Contraction/ha.ctb  
/etc/brltty/Input/bm/horizontal.kti  
/etc/brltty/Input/hd  
/etc/brltty/Input/hm  
/etc/brltty/Input/ht  
/etc/brltty/Input/hw  
/etc/brltty/Text/he.ttb  
/etc/brltty/Text/hi.ttb  
/etc/brltty/Text/hr.ttb  
/etc/brltty/Text/hu.ttb  
/etc/brltty/Text/hy.ttb
```

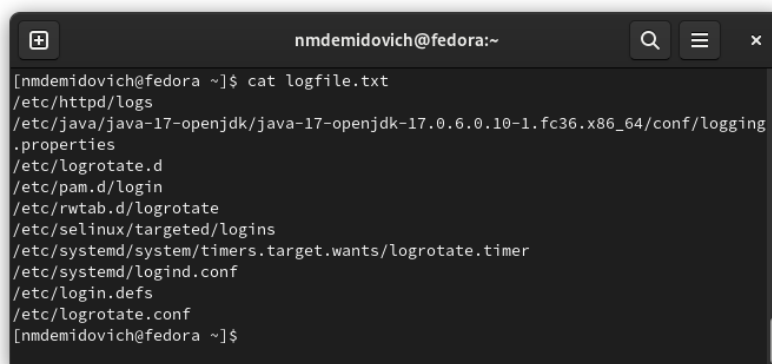
Рис. 4.5: Вывод имён файлов, начинающихся на h

Затем я запустил в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log, после чего удалил его (рис. 4.6) - (рис. 4.8).



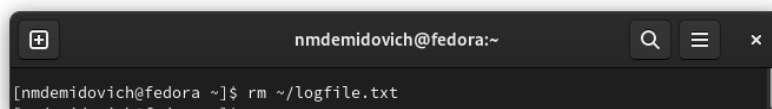
```
nmdemidovich@fedora:~  
[nmdemidovich@fedora ~]$ find /etc -name "log*" -print > ~/logfile.txt  
find: '/etc/audit': Отказано в доступе  
find: '/etc/cups/ssl': Отказано в доступе  
find: '/etc/dhcp': Отказано в доступе  
find: '/etc/firewalld': Отказано в доступе  
find: '/etc/grub.d': Отказано в доступе  
find: '/etc/libvirt': Отказано в доступе  
find: '/etc/lvm/archive': Отказано в доступе  
find: '/etc/lvm/backup': Отказано в доступе  
find: '/etc/lvm/cache': Отказано в доступе  
find: '/etc/nftables': Отказано в доступе  
find: '/etc/openvpn/client': Отказано в доступе  
find: '/etc/openvpn/server': Отказано в доступе  
find: '/etc/polkit-1/localauthority': Отказано в доступе
```

Рис. 4.6: Запуск фонового процесса



```
nmdemidovich@fedora:~  
[nmdemidovich@fedora ~]$ cat logfile.txt  
/etc/httpd/logs  
/etc/java/java-17-openjdk/java-17-openjdk-17.0.6.0.10-1.fc36.x86_64/conf/logging  
.properties  
/etc/logrotate.d  
/etc/pam.d/login  
/etc/rwtab.d/logrotate  
/etc/selinux/targeted/logins  
/etc/systemd/system/timers.target.wants/logrotate.timer  
/etc/systemd/logind.conf  
/etc/login.defs  
/etc/logrotate.conf  
[nmdemidovich@fedora ~]$
```

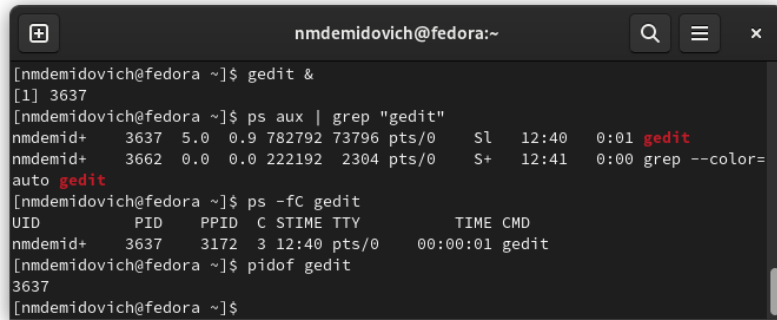
Рис. 4.7: Новый файл logfile.txt



```
nmdemidovich@fedora:~  
[nmdemidovich@fedora ~]$ rm ~/logfile.txt
```

Рис. 4.8: Удаление файла logfile.txt

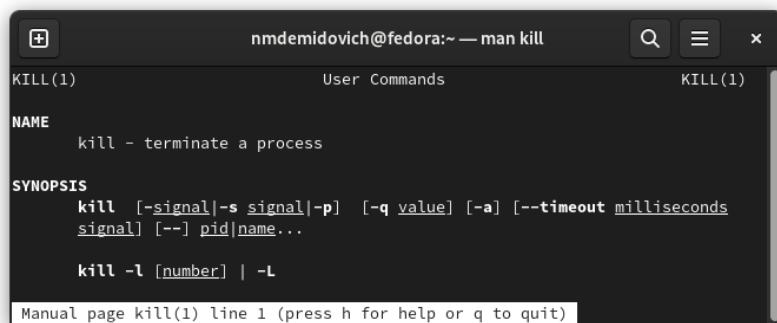
После этого я запустил из консоли в фоновом режиме редактор gedit. И определил идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep (рис. 4.9). Идентификатор процесса можно определить с помощью команд ps aux | grep “gedit”, ps -fC gedit и pidof gedit.



```
nmdemidovich@fedora:~  
[nmdemidovich@fedora ~]$ gedit &  
[1] 3637  
[nmdemidovich@fedora ~]$ ps aux | grep "gedit"  
nmdemid+ 3637 5.0 0.9 782792 73796 pts/0 Sl 12:40 0:01 gedit  
nmdemid+ 3662 0.0 0.0 222192 2304 pts/0 S+ 12:41 0:00 grep --color=  
auto gedit  
[nmdemidovich@fedora ~]$ ps -fc gedit  
UID      PID    PPID  C  STIME TTY      TIME CMD  
nmdemid+ 3637   3172  3 12:40 pts/0    00:00:01 gedit  
[nmdemidovich@fedora ~]$ pidof gedit  
3637  
[nmdemidovich@fedora ~]$
```

Рис. 4.9: Команды `ps aux | grep "gedit"`, `ps -fc gedit` и `pidof gedit` для определения идентификатора процесса

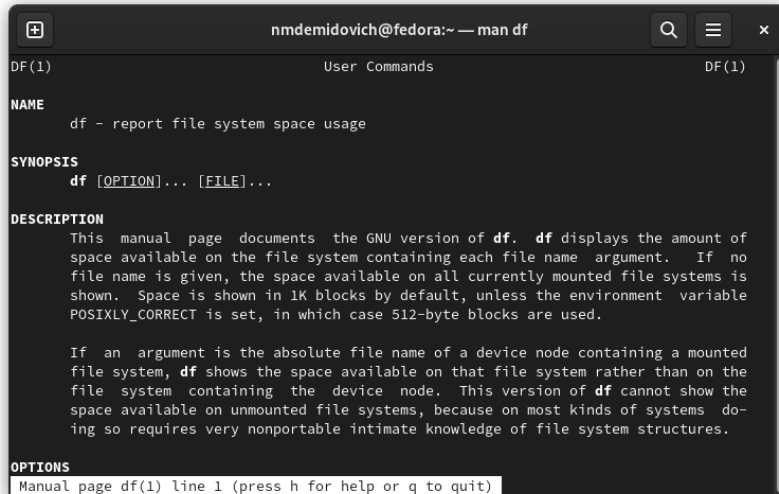
Далее я прочёл справку (man) команды `kill` (рис. 4.10).



```
nmdemidovich@fedora:~ — 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  
Manual page kill(1) line 1 (press h for help or q to quit)
```

Рис. 4.10: Справка (man) команды `kill`

Я не использовал её для завершения процесса, т.к он был завершён мною ранее.  
Затем я выполнил команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man` (рис. 4.11) - (рис. 4.15)



```
nmdemidovich@fedora:~ — man df
DF(1)                                User Commands                                DF(1)

NAME
    df - report file system space usage

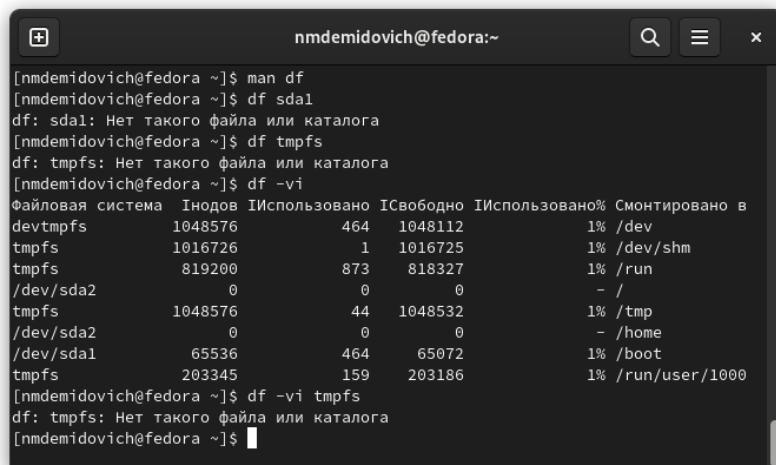
SYNOPSIS
    df [OPTION]... [FILE]...

DESCRIPTION
    This manual page documents the GNU version of df. df displays the amount of
    space available on the file system containing each file name argument. If no
    file name is given, the space available on all currently mounted file systems is
    shown. Space is shown in 1K blocks by default, unless the environment variable
    POSIXLY_CORRECT is set, in which case 512-byte blocks are used.

    If an argument is the absolute file name of a device node containing a mounted
    file system, df shows the space available on that file system rather than on the
    file system containing the device node. This version of df cannot show the
    space available on unmounted file systems, because on most kinds of systems do-
    ing so requires very nonportable intimate knowledge of file system structures.

OPTIONS
    Manual page df(1) line 1 (press h for help or q to quit)
```

Рис. 4.11: Справка (man) команды df



```
nmdemidovich@fedora:~
[nmdemidovich@fedora ~]$ man df
[nmdemidovich@fedora ~]$ df sda1
df: sda1: Нет такого файла или каталога
[nmdemidovich@fedora ~]$ df tmpfs
df: tmpfs: Нет такого файла или каталога
[nmdemidovich@fedora ~]$ df -vi
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
devtmpfs          1048576      464    1048112           1% /dev
tmpfs              1016726        1    1016725           1% /dev/shm
tmpfs              819200      873    818327           1% /run
/dev/sda2           0            0         0           - /
tmpfs              1048576      44    1048532           1% /tmp
/dev/sda2           0            0         0           - /home
/dev/sda1           65536      464     65072           1% /boot
tmpfs              203345      159    203186           1% /run/user/1000
[nmdemidovich@fedora ~]$ df -vi tmpfs
df: tmpfs: Нет такого файла или каталога
[nmdemidovich@fedora ~]$
```

Рис. 4.12: Выполнение команды df

```
nmdemidovich@fedora:~$ df -v
Файловая система 1К-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs           4096             0      4096             0% /dev
tmpfs              4066904          0    4066904             0% /dev/shm
tmpfs              1626764        1384    1625380             1% /run
/dev/sda2          82836480      14632404    67219212            18% /
tmpfs              4066908          16    4066892             1% /tmp
/dev/sda2          82836480      14632404    67219212            18% /home
/dev/sda1          996780        294444    633524             32% /boot
tmpfs              813380         508     812872             1% /run/user/1000
nmdemidovich@fedora ~]$
```

Рис. 4.13: Выполнение команды df

```
nmdemidovich@fedora:~ — man du
DU(1)                                User Commands                                DU(1)

NAME
    du - estimate file space usage

SYNOPSIS
    du [OPTION]... [FILE]...
    du [OPTION]... --files0-from=F

DESCRIPTION
    Summarize device usage of the set of FILES, recursively for directories.

    Mandatory arguments to long options are mandatory for short options too.

    -0, --null
        end each output line with NUL, not newline
Manual page du(1) line 1 (press h for help or q to quit)
```

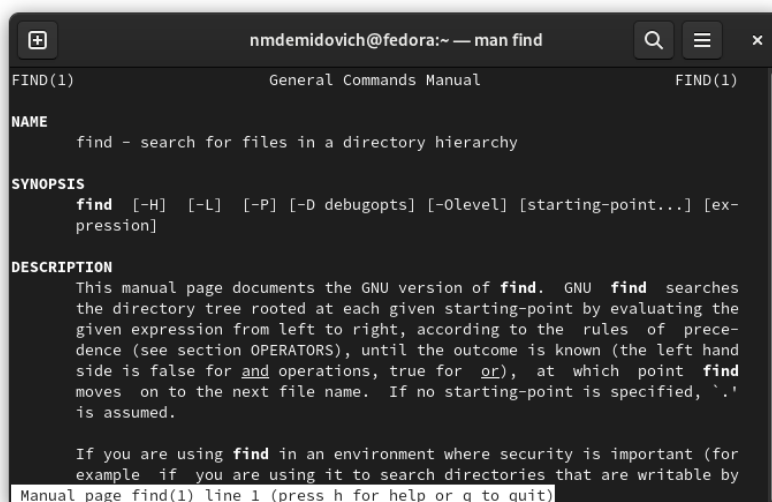
Рис. 4.14: Справка (man) команды du

```
nmdemidovich@fedo...
[nmdemidovich@fedora ~]$ man du
[nmdemidovich@fedora ~]$ du -a file.txt
4      file.txt
[nmdemidovich@fedora ~]$
```

Рис. 4.15: Выполнение команды du

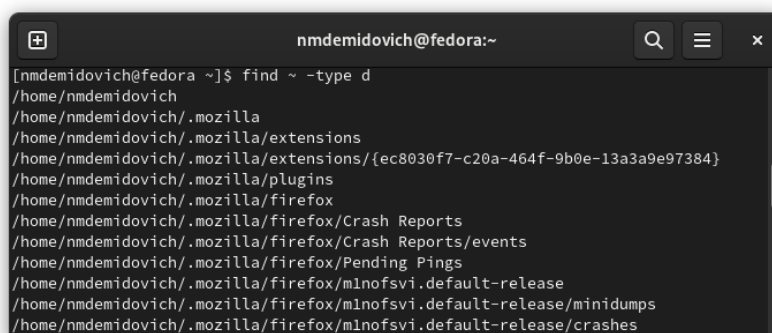
И на заключительном этапе выполнения работы я, воспользовавшись справкой

команды `find`, вывел имена всех директорий, имеющихсся в моём домашнем каталоге (рис. 4.16) - (рис. 4.17)



The screenshot shows a terminal window titled "nmdemidovich@fedora:~ — man find". The window displays the manual page for the `find` command. The page is divided into sections: NAME, SYNOPSIS, and DESCRIPTION. The NAME section states "find - search for files in a directory hierarchy". The SYNOPSIS section shows the command syntax: `find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [expression]`. The DESCRIPTION section explains that the GNU version of `find` searches the directory tree rooted at each given starting-point by evaluating the given expression from left to right, according to the rules of precedence (see section OPERATORS), until the outcome is known (the left hand side is false for `and` operations, true for `or`), at which point `find` moves on to the next file name. If no starting-point is specified, `.'` is assumed. At the bottom, there is a note: "If you are using `find` in an environment where security is important (for example if you are using it to search directories that are writable by Manual page find(1) line 1 (press h for help or q to quit)".

Рис. 4.16: Справка (man) команды `find`



The screenshot shows a terminal window titled "nmdemidovich@fedora:~". The prompt is `[nmdemidovich@fedora ~]$`. The command `find ~ -type d` has been executed. The output lists the following directories: `/home/nmdemidovich`, `/home/nmdemidovich/.mozilla`, `/home/nmdemidovich/.mozilla/extensions`, `/home/nmdemidovich/.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}`, `/home/nmdemidovich/.mozilla/plugins`, `/home/nmdemidovich/.mozilla/firefox`, `/home/nmdemidovich/.mozilla/firefox/Crash Reports`, `/home/nmdemidovich/.mozilla/firefox/Crash Reports/events`, `/home/nmdemidovich/.mozilla/firefox/Pending Pings`, `/home/nmdemidovich/.mozilla/firefox/mInofsvi.default-release`, `/home/nmdemidovich/.mozilla/firefox/mInofsvi.default-release/minidumps`, and `/home/nmdemidovich/.mozilla/firefox/mInofsvi.default-release/crashes`.

Рис. 4.17: Выполнение команды `find`



## 5 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?

– `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.

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

`>filename` - Перенаправление вывода (`stdout`) в файл “filename”.

`>>filename` - Перенаправление вывода (`stdout`) в файл “filename”, файл открывается в режиме добавления.

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:

команда 1 | команда 2

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

Компьютерная программа сама по себе — лишь пассивная последовательность инструкций. В то время как процесс — непосредственное выполнение этих инструкций.

## 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?

Задачи - это то, что мы подаем на выполнение системе, какой-то процесс, который она начинает выполнять.

Команда - `jobs`.

## 7. Найдите информацию об утилитах `top` и `htop`. Каковы их функции?

`top` (table of processes) — консольная команда, которая выводит список работающих в системе процессов и информацию о них. По умолчанию она в реальном времени сортирует их по нагрузке на процессор. Программа написана для UNIX-совместимых операционных систем и опубликована под свободной лицензией GNU FDL.

`htop` — продвинутый монитор процессов, написанный для Linux. Он был задуман заменить стандартную программу `top`. `Htop` показывает динамический список системных процессов, список обычно выравнивается по использованию ЦПУ. В отличие от `top`, `htop` показывает все процессы в системе. Также показывает время непрерывной работы, использование процессоров и памяти. `Htop` часто

применяется в тех случаях, когда информации даваемой утилитой `top` недостаточно, например при поиске утечек памяти в процессах. `htop` написан на языке Си и использует для отображения библиотеку `Ncurses`.

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>  
find /etc -name "p*" -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?

При помощи команды `df` (аббревиатура от `disk free`) — утилита в UNIX и UNIX-подобных системах, показывает список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования.

11. Как определить объем вашего домашнего каталога?

```
du -a ~
```

12. Как удалить зависший процесс?

Для завершения процесса нужно вызвать утилиту `kill` с параметром `"-9"`.

## 6 Выводы

В ходе данной лабораторной работы я ознакомился с инструментами поиска файлов и фильтрации текстовых данных и приобрел практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

# Список литературы

Лабораторная работа №6 (Архитектура ОС)