

**Лабораторная работа №6: Поиск файлов.
Перенаправление ввода-вывода.
Просмотр запущенных процессов**

Дисциплина: Операционные системы

Дарья Эдуардовна Ибатулина

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	12
5	Контрольные вопросы	22
6	Выводы	25
	Список литературы	26

Список иллюстраций

4.1	Вход в систему	12
4.2	Запись в файл file.txt всех файлов, содержащихся в каталоге /etc .	13
4.3	Запись в файл file.txt всех файлов, содержащихся в домашнем каталоге	13
4.4	Вывод имён файлов, имеющих расширение .conf	14
4.5	Запись имён данных файлов в текстовый файл conf.txt	14
4.6	Вывод имён файлов, начинающихся с символа с (вариант 1) . . .	14
4.7	Вывод имён файлов, начинающихся с символа с (вариант 2) . . .	15
4.8	Вывод имён файлов, начинающихся с символа h	15
4.9	Запуск в фоновом режиме процесса, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log	16
4.10	Удаление файла ~/logfile	16
4.11	Запуск из консоли в фоновом режиме редактора gedit	16
4.12	Определение идентификатора процесса gedit, используя команду ps, конвейер и фильтр grep (вариант 1)	16
4.13	Определение идентификатора процесса gedit, используя команду ps, конвейер и фильтр grep (вариант 2)	16
4.14	Справка по команде kill	17
4.15	Завершение процесса gedit с использованием команды kill . . .	17
4.16	Справка по команде df	18
4.17	Справка по команде du	19
4.18	Использование команды df	20
4.19	Использование команды du	20
4.20	Справка по команде find	21
4.21	Вывод имён всех директорий, имеющихся в домашнем каталоге .	21

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`.
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, имеющих в вашем домашнем каталоге.

3 Теоретическое введение

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".
```

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 > 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 быть 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 можно воспользоваться командой man.

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

```
grep begin f*
```

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

```
ls -l | grep лаб
```

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
```

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 текстовый 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
```

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. Осуществим вход в систему, используя соответствующее имя пользователя (рис. 4.1):

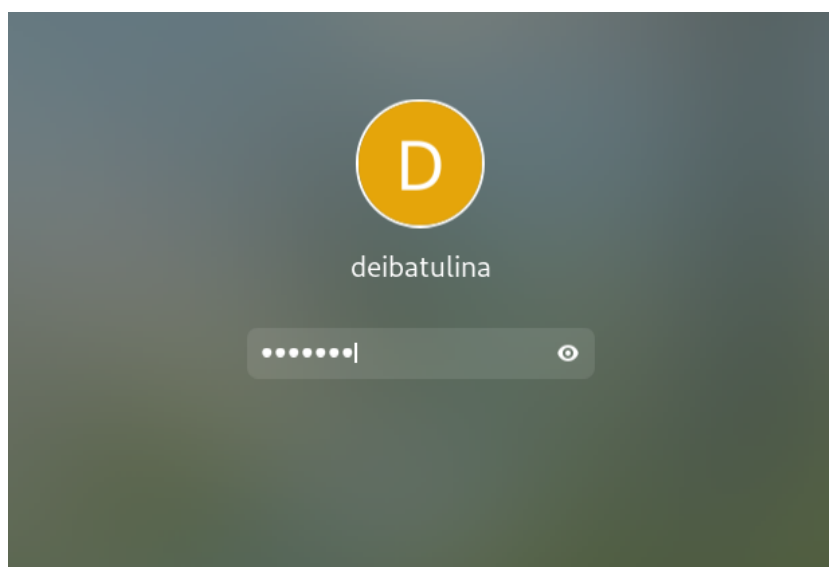


Рис. 4.1: Вход в систему

2. Запишем в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`. Допишем в этот же файл названия файлов, содержащихся в вашем домашнем каталоге (рис. 4.2, 4.3):

```
[deibatulina@10 ~]$ ls -lR /etc > file.txt
ls: невозможно открыть каталог '/etc/audit': Отказано в доступе
ls: невозможно открыть каталог '/etc/cups/ssl': Отказано в доступе
ls: невозможно открыть каталог '/etc/dhcp': Отказано в доступе
ls: невозможно открыть каталог '/etc/firewalld': Отказано в доступе
ls: невозможно открыть каталог '/etc/grub.d': Отказано в доступе
ls: невозможно открыть каталог '/etc/libvirt': Отказано в доступе
ls: невозможно открыть каталог '/etc/lvm/archive': Отказано в доступе
ls: невозможно открыть каталог '/etc/lvm/backup': Отказано в доступе
ls: невозможно открыть каталог '/etc/lvm/cache': Отказано в доступе
ls: невозможно открыть каталог '/etc/nftables': Отказано в доступе
ls: невозможно открыть каталог '/etc/openvpn/client': Отказано в доступе
ls: невозможно открыть каталог '/etc/openvpn/server': Отказано в доступе
ls: невозможно открыть каталог '/etc/polkit-1/localauthority': Отказано в доступе
ls: невозможно открыть каталог '/etc/polkit-1/rules.d': Отказано в доступе
ls: невозможно открыть каталог '/etc/sos/cleaner': Отказано в доступе
ls: невозможно открыть каталог '/etc/ssh/sshd_config.d': Отказано в доступе
ls: невозможно открыть каталог '/etc/ssh': Отказано в доступе
ls: невозможно открыть каталог '/etc/sudoers.d': Отказано в доступе
[deibatulina@10 ~]$ cat file.txt
/etc:
итого 1448
drwxr-xr-x. 1 root root      126 ноя  5 11:51 abrt
-rw-r--r--. 1 root root      16 фев 11 16:15 adjtime
-rw-r--r--. 1 root root    1529 июл 20 2022 aliases
drwxr-xr-x. 1 root root       70 окт 24 18:29 alsa
drwxr-xr-x. 1 root root    1554 фев 22 16:23 alternatives
drwxr-xr-x. 1 root root       56 ноя  5 11:51 anaconda
-rw-r--r--. 1 root root     269 июл 20 2022 anthy-unicode.conf
-rw-r--r--. 1 root root     833 авг 22 2022 appstream.conf
-rw-r--r--. 1 root root       55 дек  7 17:18 asound.conf
drwxr-x---. 1 root root     108 фев 11 16:21 audit
drwxr-xr-x. 1 root root     232 фев 11 18:43 authselect
drwxr-xr-x. 1 root root       66 ноя  7 20:25 avahi
drwxr-xr-x. 1 root root     144 фев 11 18:40 bash_completion.d
-rw-r--r--. 1 root root    2638 июл 20 2022 bashrc
-rw-r--r--. 1 root root     535 авг  8 2022 bindresvport.blacklist
drwxr-xr-x. 1 root root        0 дек 20 21:39 binfmt.d
```

Рис. 4.2: Запись в файл file.txt всех файлов, содержащихся в каталоге /etc

```
[deibatulina@10 ~]$ ls -lR ~/ >> file.txt
```

Рис. 4.3: Запись в файл file.txt всех файлов, содержащихся в домашнем каталоге

3. Выведем имена всех файлов из file.txt, имеющих расширение .conf, после чего запишем их в новый текстовый файл conf.txt (рис. 4.4, 4.5):

```
[deibatulina@10 ~]$ grep .conf file.txt
-rw-r--r--. 1 root root      269 июл 20  2022 anthy-unicode.conf
-rw-r--r--. 1 root root      833 авг 22  2022 appstream.conf
-rw-r--r--. 1 root root       55 дек 7 17:18 asound.conf
-rw-r--r--. 1 root root    29842 авг 2  2022 brltty.conf
drwxr-xr-x. 1 root root       0 окт 5 12:39 chkconfig.d
-rw-r--r--. 1 root root    1371 авг 29  2022 chrony.conf
drwxr-xr-x. 1 root root       18 ноя 5 11:38 dconf
-rw-r--r--. 1 root root    1280 июл 21  2022 dleyna-renderer-service.conf
-rw-r--r--. 1 root root    1174 июл 21  2022 dleyna-server-service.conf
-rw-r--r--. 1 root root   28442 дек 8 16:04 dnsmasq.conf
-rw-r--r--. 1 root root     117 ноя 16 21:00 dracut.conf
drwxr-xr-x. 1 root root       0 ноя 16 21:00 dracut.conf.d
-rw-r--r--. 1 root root      20 фев 24  2022 fprintd.conf
```

Рис. 4.4: Вывод имён файлов, имеющих расширение .conf

```
[deibatulina@10 ~]$ grep '\.conf' file.txt > conf.txt
[deibatulina@10 ~]$ cat conf.txt
-rw-r--r--. 1 root root      269 июл 20  2022 anthy-unicode.conf
-rw-r--r--. 1 root root      833 авг 22  2022 appstream.conf
-rw-r--r--. 1 root root       55 дек 7 17:18 asound.conf
-rw-r--r--. 1 root root    29842 авг 2  2022 brltty.conf
-rw-r--r--. 1 root root    1371 авг 29  2022 chrony.conf
-rw-r--r--. 1 root root    1280 июл 21  2022 dleyna-renderer-service.conf
-rw-r--r--. 1 root root    1174 июл 21  2022 dleyna-server-service.conf
-rw-r--r--. 1 root root   28442 дек 8 16:04 dnsmasq.conf
-rw-r--r--. 1 root root     117 ноя 16 21:00 dracut.conf
drwxr-xr-x. 1 root root       0 ноя 16 21:00 dracut.conf.d
-rw-r--r--. 1 root root      20 фев 24  2022 fprintd.conf
-rw-r--r--. 1 root root       38 июл 21  2022 fuse.conf
-rw-r--r--. 1 root root       9 июл 20  2022 host.conf
-rw-r--r--. 1 root root    5799 янв 21 19:02 idmapd.conf
-rw-r--r--. 1 root root    8892 ноя 5 11:47 kdump.conf
-rw-r--r--. 1 root root     880 ноя 16 20:16 krb5.conf
drwxr-xr-x. 1 root root     106 ноя 16 20:21 krb5.conf.d
-rw-r--r--. 1 root root      28 янв 11 07:55 ld.so.conf
drwxr-xr-x. 1 root root     116 фев 11 19:50 ld.so.conf.d
-rw-r-----. 1 root root     191 авг 30  2022 libaudit.conf
```

Рис. 4.5: Запись имён данных файлов в текстовый файл conf.txt

4. Определим, какие файлы в домашнем каталоге имеют имена, начинающиеся с символа c (рис. 4.6, 4.7):

```
[deibatulina@10 ~]$ ls -l | grep c*
-rw-r--r--. 1 deibatulina deibatulina 41417 мар 12 13:16 conf.txt
[deibatulina@10 ~]$
```

Рис. 4.6: Вывод имён файлов, начинающихся с символа c (вариант 1)

```
[deibatulina@i0 ~]$ find ~ -name "c*" -print
/home/deibatulina/.mozilla/firefox/wr4b0z2h.default-release/compatibility.ini
/home/deibatulina/.mozilla/firefox/wr4b0z2h.default-release/cookies.sqlite
/home/deibatulina/.mozilla/firefox/wr4b0z2h.default-release/cert9.db
/home/deibatulina/.mozilla/firefox/wr4b0z2h.default-release/storage/permanent/chrome
/home/deibatulina/.mozilla/firefox/wr4b0z2h.default-release/storage/default/https+++vk.com/cache
/home/deibatulina/.mozilla/firefox/wr4b0z2h.default-release/storage/default/https+++vk.com/cache/caches.sqlite
/home/deibatulina/.mozilla/firefox/wr4b0z2h.default-release/storage/default/https+++web.telegram.org/cache
/home/deibatulina/.mozilla/firefox/wr4b0z2h.default-release/storage/default/https+++web.telegram.org/cache/caches.sqlite
/home/deibatulina/.mozilla/firefox/wr4b0z2h.default-release/storage/default/https+++www.youtube.com^partitionKey=%28https%2Crudn.ru%29/cache
/home/deibatulina/.mozilla/firefox/wr4b0z2h.default-release/storage/default/https+++www.youtube.com^partitionKey=%28https%2Crudn.ru%29/cache/caches.sqlite
/home/deibatulina/.mozilla/firefox/wr4b0z2h.default-release/content-prefs.sqlite
/home/deibatulina/.mozilla/firefox/wr4b0z2h.default-release/containers.json
/home/deibatulina/.cache/mesa_shader_cache/bb/cdec1de012044c426a3b642d31ed751bad39f
/home/deibatulina/.cache/mesa_shader_cache/8f/cfa368106b05830df143f99aa43309f25fb978
/home/deibatulina/.cache/mesa_shader_cache/0d/ce17349c2f2a9caf6c2df7f850c717a03ccbec
/home/deibatulina/.cache/mesa_shader_cache/c6
```

Рис. 4.7: Вывод имён файлов, начинающихся с символа с (вариант 2)

5. Выведем на экран имена файлов из каталога /etc, начинающиеся с символа h (рис. 4.8):

```
[deibatulina@i0 ~]$ 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
/etc/containers/oci/hooks.d
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
/etc/hp
/etc/hp/hplip.conf
/etc/httpd
/etc/httpd/conf/httpd.conf
```

Рис. 4.8: Вывод имён файлов, начинающихся с символа h

6. Запустим в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log (рис. 4.9):

```
[deibatulina@10 ~]$ find ~ -name 'log*' -print > logfile &
[1] 68360
[deibatulina@10 ~]$
```

Рис. 4.9: Запуск в фоновом режиме процесса, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log

7. Удалим файл ~/logfile (рис. 4.10):

```
[deibatulina@10 ~]$ rm logfile
[1]+  Завершён      find ~ -name 'log*' -print > logfile
[deibatulina@10 ~]$
```

Рис. 4.10: Удаление файла ~/logfile

8. Запустим из консоли в фоновом режиме редактор gedit (рис. 4.11):

```
[deibatulina@10 ~]$ gedit &
[1] 68409
[deibatulina@10 ~]$
```

Рис. 4.11: Запуск из консоли в фоновом режиме редактора gedit

9. Определим идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep (рис. 4.12, 4.13):

```
[deibatulina@10 ~]$ ps aux | grep gedit
deibatu+  68409  1.3  3.7 785488 76564 pts/0    Sl   13:38   0:01 gedit
deibatu+  68464  0.0  0.1 222044  2244 pts/0    S+   13:39   0:00 grep --colo
r=auto gedit
[deibatulina@10 ~]$
```

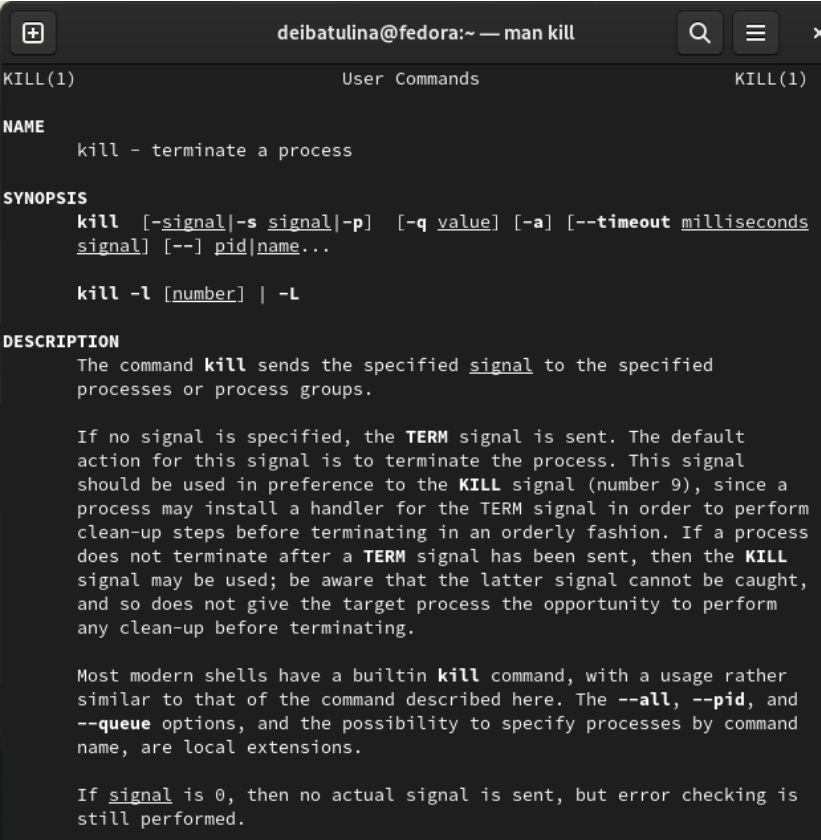
Рис. 4.12: Определение идентификатора процесса gedit, используя команду ps, конвейер и фильтр grep (вариант 1)

```
[deibatulina@10 ~]$ pgrep gedit
68409
[deibatulina@10 ~]$
```

Рис. 4.13: Определение идентификатора процесса gedit, используя команду ps, конвейер и фильтр grep (вариант 2)

Получим, что его ID равен 68409.

10. Прочтём справку (man) команды `kill`, после чего используем её для завершения процесса `gedit` (рис. 4.14, 4.15):



```
deibatulina@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

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 be used; be aware that the latter signal cannot be caught,
    and so does not give the target process the opportunity to perform
    any clean-up before terminating.

    Most modern shells have a builtin kill command, with a usage rather
    similar to that of the command described here. The --all, --pid, and
--queue options, and the possibility to specify processes by command
    name, are local extensions.

    If signal is 0, then no actual signal is sent, but error checking is
    still performed.
```

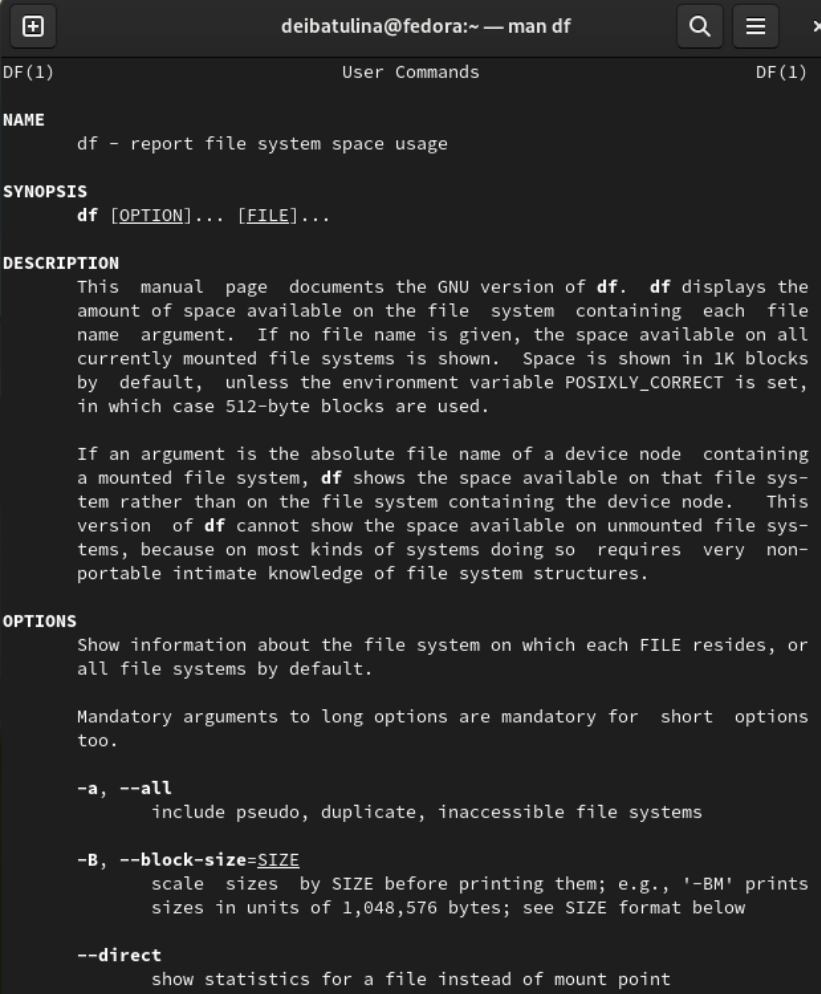
Рис. 4.14: Справка по команде `kill`



```
[deibatulina@10 ~]$ man kill
[deibatulina@10 ~]$ kill 68409
[deibatulina@10 ~]$
```

Рис. 4.15: Завершение процесса `gedit` с использованием команды `kill`

11. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man` (рис. 4.16, 4.17, 4.18, 4.19):



```
deibatulina@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 sys-
  tem rather than on the file system containing the device node. This
  version of df cannot show the space available on unmounted file sys-
  tems, because on most kinds of systems doing so requires very non-
  portable intimate knowledge of file system structures.

OPTIONS
  Show information about the file system on which each FILE resides, or
  all file systems by default.

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

  -a, --all
      include pseudo, duplicate, inaccessible file systems

  -B, --block-size=SIZE
      scale sizes by SIZE before printing them; e.g., '-BM' prints
      sizes in units of 1,048,576 bytes; see SIZE format below

  --direct
      show statistics for a file instead of mount point
```

Рис. 4.16: Справка по команде df

Получаем, что команда `df` используется для отображения использования дис-
кового пространства файловой системой Linux в целом.

```
deibatulina@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

    -a, --all
        write counts for all files, not just directories

    --apparent-size
        print apparent sizes rather than device usage; although the apparent size is usually smaller, it may be larger due to holes in ('sparse') files, internal fragmentation, indirect blocks, and the like

    -B, --block-size=SIZE
        scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576 bytes; see SIZE format below

    -b, --bytes
        equivalent to '--apparent-size --block-size=1'

    -c, --total
        produce a grand total

    -D, --dereference-args
        dereference only symlinks that are listed on the command line
```

Рис. 4.17: Справка по команде du

В свою очередь, команда du используется для отображения используемых файлов и каталогов на диске в файловой системе.

```
[deibatulina@10 ~]$ df -vi
Файловая система  Инодов  ИИспользовано  ISвободно  ИИспользовано%  Смонтировано в
devtmpfs           1048576      470      1048106          1% /dev
tmpfs              252088        1      252087          1% /dev/shm
tmpfs              819200      932      818268          1% /run
/dev/sda3           0            0            0            - /
tmpfs              1048576      743      1047833          1% /tmp
/dev/sda3           0            0            0            - /home
/dev/sda2           65536       394       65142          1% /boot
tmpfs              50417       159       50258          1% /run/user/1000
/dev/sr0            0            0            0            - /run/media/dei
batulina/VBox_GAs_7.0.6
[deibatulina@10 ~]$
```

Рис. 4.18: Использование команды df

```
[deibatulina@10 ~]$ du -a /home/deibatulina/work/study/2022-2023/'Операционные
системы'/os-intro/labs/lab01
8      /home/deibatulina/work/study/2022-2023/Операционные системы/os-intro/l
abs/lab01/report/bib/cite.bib
8      /home/deibatulina/work/study/2022-2023/Операционные системы/os-intro/l
abs/lab01/report/bib
252    /home/deibatulina/work/study/2022-2023/Операционные системы/os-intro/l
abs/lab01/report/image/placeimg_800_600_tech.jpg
64     /home/deibatulina/work/study/2022-2023/Операционные системы/os-intro/l
abs/lab01/report/image/1.jpg
36     /home/deibatulina/work/study/2022-2023/Операционные системы/os-intro/l
abs/lab01/report/image/2.jpg
48     /home/deibatulina/work/study/2022-2023/Операционные системы/os-intro/l
abs/lab01/report/image/3.jpg
52     /home/deibatulina/work/study/2022-2023/Операционные системы/os-intro/l
abs/lab01/report/image/4.jpg
72     /home/deibatulina/work/study/2022-2023/Операционные системы/os-intro/l
abs/lab01/report/image/5.jpg
56     /home/deibatulina/work/study/2022-2023/Операционные системы/os-intro/l
abs/lab01/report/image/6.jpg
44     /home/deibatulina/work/study/2022-2023/Операционные системы/os-intro/l
abs/lab01/report/image/7.jpg
28     /home/deibatulina/work/study/2022-2023/Операционные системы/os-intro/l
abs/lab01/report/image/8.jpg
```

Рис. 4.19: Использование команды du

12. Воспользовавшись справкой команды find, выведем имена всех директорий, имеющих в домашнем каталоге (рис. 4.20, 4.21):

```
deibatulina@fedora:~ — man find
FIND(1)                                General Commands Manual                                FIND(1)

NAME
    find - search for files in a directory hierarchy

SYNOPSIS
    find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [ex-
    pression]

DESCRIPTION
    This manual page documents the GNU version of find. GNU 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.

    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 other users), you should read the 'Security Considera-
    tions' chapter of the findutils documentation, which is called Find-
    ing Files and comes with findutils. That document also includes a
    lot more detail and discussion than this manual page, so you may find
    it a more useful source of information.
```

Рис. 4.20: Справка по команде find

```
[deibatulina@l0 ~]$ find -type d
.
./mozilla
./mozilla/extensions
./mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
./mozilla/plugins
./mozilla/firefox
./mozilla/firefox/wr4b0z2h.default-release
./mozilla/firefox/wr4b0z2h.default-release/security_state
./mozilla/firefox/wr4b0z2h.default-release/storage
./mozilla/firefox/wr4b0z2h.default-release/storage/permanent
./mozilla/firefox/wr4b0z2h.default-release/storage/permanent/chrome
./mozilla/firefox/wr4b0z2h.default-release/storage/permanent/chrome/idb
./mozilla/firefox/wr4b0z2h.default-release/storage/permanent/chrome/idb/38701
12724rsegmnoittet-es.files
./mozilla/firefox/wr4b0z2h.default-release/storage/permanent/chrome/idb/35612
88849sdhlie.files
./mozilla/firefox/wr4b0z2h.default-release/storage/permanent/chrome/idb/14513
18868ntouromlalnody--epcr.files
./mozilla/firefox/wr4b0z2h.default-release/storage/permanent/chrome/idb/16571
```

Рис. 4.21: Вывод имён всех директорий, имеющихсся в домашнем каталоге

5 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?

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.

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

`>` записывает в новый файл или перезаписывает в уже существующий, в свою очередь `»` добавляет в уже созданный файл (файл открывается в режиме дозаписи).

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`

`#` означает, что вывод команды 1 передаётся на ввод команде 2

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

Процесс - это любая команда, выполняемая в системе. Программа - это набор инструкций, позволяющих ЦПУ выполнять определённую задачу.

5. Что такое PID и GID?

PID - идентификатор родительского процесса, UID, GID - реальные идентификаторы пользователя/группы, запустивших данный процесс.

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

Задачи - это запущенные фоном программы. 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` - 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. `Htop` - компьютерная программа, предназначенная для вывода на терминал списка запущенных процессов и информации о них. Создана как альтернатива программы `top`. Написана на языке Си.

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]
```

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 можно воспользоваться командой man.

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

Да, можно. С помощью команды:

```
find / -type f -exec grep -H 'текстДляПоиска' {} \;
```

10. Как определить объем свободной памяти на жёстком диске?

С помощью команды *df -h*.

11. Как определить объем вашего домашнего каталога?

С помощью команды *du -s*.

12. Как удалить зависший процесс?

Использовать команду *kill номер_задачи*.

6 Выводы

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

Список литературы