

# **Отчет по выполнению лабораторной работы №6**

**Операционные системы**

Шуваев Сергей Александрович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>6</b>
<b>2</b>	<b>Задание</b>	<b>7</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>8</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
<b>5</b>	<b>Ответы на контрольные вопросы</b>	<b>17</b>

## Список иллюстраций

4.1	Я вошел в систему под соответствующим именем пользователя, открыл терминал, . . . . .	9
4.2	записал в файл file.txt названия файлов из каталога /etc с помощью перенаправления ">" (и файл создал, и записал в него то, что могло быть выведено ls -lR /etc). В файл я добавил также все файлы из подкаталогов, . . . . .	9
4.3	Вывел на экран имена всех файлов, имеющих расширение ".conf" с помощью утилиты grep, . . . . .	10
4.4	Определяю, какие файлы в домашнем каталоге начинаются с символа "с" с помощью утилиты find, прописываю ей в аргументах домашнюю директорию (тогда вывод относительно корневого каталога, а не домашнего будет), выбираю опцию -name (ищем по имени), и пишу маску, по которой будем искать имя, где * - любое кол-во любых символов, добавляю опцию -print, чтобы мне вывелся результат, . . . . .	11
4.5	Второй способ использовать утилиту ls -lR и использовать grep, чтобы найти элементы с первым символом с. Однако этот способ не работает для поиска файлов из подкаталогов каталога, . . . .	11
4.6	С помощью метода find, чьи опции я расписала ранее, ищу все файлы, начинающиеся с буквы "h", . . . . .	12
4.7	Запускаю в фоновом режиме (на это указывает символ &) процесс, который будет записывать в файл logfile (с помощью перенаправления >) файлы, имена которых начинаются с log (, . . . . .	12
4.8	Проверяю, что файл создан, удаляю его, проверяю, что файл удален,	12
4.9	. . . . .	12
4.10	Запускаю в консоли в фоновом режиме (с помощью символа &) редактор gedit . . . . .	13
4.11	С помощью утилиты ps определяю идентификатор процесса gedit его идентификатор 6619 . . . . .	13
4.12	Также мы можем определить идентификатор с помощью pgrep. .	13
4.13	Прочитал справку команды kill . . . . .	14
4.14	Прочитал документацию про функции df и du . . . . .	14
4.15	. . . . .	15

4.16	Используя утилиту df опции -iv позволяют увидеть информацию об инодах и сделать вывод читаемым, игнорируя сообщения системы о нем. Эта утилита нам нужна, чтобы выяснить, сколько свободного места есть у нашей системы., . . . . .	15
------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

## Список таблиц

# 1 Цель работы

Цель данной лабораторной работы - ознакомление с инструментами поиска файлов и фильтрации текстовых данных, а также приобретение практических навыков по управлению процессами (и заданиями), по проверке использования диска и по 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, начинавшиеся с символа `c`? Предложите несколько вариантов, как это сделать. 5.Выведите на экран (по страницно) 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

В интерфейсе командной строки есть очень полезная возможность перенаправления (переадресации) ввода и вывода (англ. термин I/O Redirection). Как мы уже заметили, многие программы выводят данные на экран. А ввод данных в терминале осуществляется с клавиатуры. С помощью специальных обозначений можно перенаправить вывод многих команд в файлы или иные устройства вывода (например, распечатать на принтере). Тоже самое и со вводом информации, вместо ввода данных с клавиатуры, для многих программ можно задать считывание символов их файла. Кроме того, можно даже вывод одной программы передать на ввод другой программе.

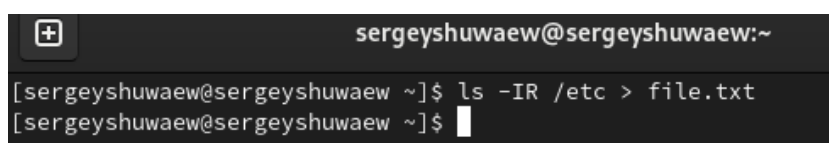
К каждой программе, запускаемой в командной строке, по умолчанию подключено три потока данных:

STDIN (0) — стандартный поток ввода (данные, загружаемые в программу). STDOUT (1) — стандартный поток вывода (данные, которые выводит программа). По умолчанию — терминал. STDERR (2) — стандартный поток вывода диагностических и отладочных сообщений (например, сообщениях об ошибках). По умолчанию — терминал.

Pipe (конвейер) – это однонаправленный канал межпроцессного взаимодействия. Термин был придуман Дугласом Макилроем для командной оболочки Unix и назван по аналогии с трубопроводом. Конвейеры чаще всего используются в shell-скриптах для связи нескольких команд путем перенаправления вывода одной команды (stdout) на вход (stdin) последующей, используя символ конвейера '|'.

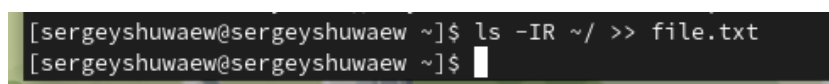


## 4 Выполнение лабораторной работы



```
sergeyshuwaew@sergeyshuwaew:~  
[sergeyshuwaew@sergeyshuwaew ~]$ ls -lR /etc > file.txt  
[sergeyshuwaew@sergeyshuwaew ~]$
```

Рис. 4.1: Я вошел в систему под соответствующим именем пользователя, открыл терминал,



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

Рис. 4.2: записал в файл file.txt названия файлов из каталога /etc с помощью перенаправления “>” (и файл создал, и записал в него то, что могло быть выведено ls -lR /etc). В файл я добавил также все файлы из подкаталогов ,

```
[sergeyshuwaew@sergeyshuwaew ~]$ grep .conf file.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
fprintd.conf
fuse.conf
host.conf
idmapd.conf
kdump.conf
krb5.conf
krb5.conf.d
ld.so.conf
```

Рис. 4.3: Вывел на экран имена всех файлов, имеющих расширение “.conf” с помощью утилиты grep,

```

[sergeyshuwaew@sergeyshuwaew ~]$ find ~ -name "c*" -print
/home/sergeyshuwaew/.mozilla/firefox/7f0i6xj2.default-release/compatibility.ini
/home/sergeyshuwaew/.mozilla/firefox/7f0i6xj2.default-release/cookies.sqlite
/home/sergeyshuwaew/.mozilla/firefox/7f0i6xj2.default-release/cert9.db
/home/sergeyshuwaew/.mozilla/firefox/7f0i6xj2.default-release/storage/permanent/
chrome
/home/sergeyshuwaew/.mozilla/firefox/7f0i6xj2.default-release/storage/default/ht
tps+++rutube.ru/cache
/home/sergeyshuwaew/.mozilla/firefox/7f0i6xj2.default-release/storage/default/ht
tps+++rutube.ru/cache/caches.sqlite
/home/sergeyshuwaew/.mozilla/firefox/7f0i6xj2.default-release/storage/default/ht
tps+++www.youtube.com/cache
/home/sergeyshuwaew/.mozilla/firefox/7f0i6xj2.default-release/storage/default/ht
tps+++www.youtube.com/cache/caches.sqlite
/home/sergeyshuwaew/.mozilla/firefox/7f0i6xj2.default-release/storage/default/ht
tps+++www.youtube.com*partitionKey=%28https%2Citsfoss.com%29/cache
/home/sergeyshuwaew/.mozilla/firefox/7f0i6xj2.default-release/storage/default/ht
tps+++www.youtube.com*partitionKey=%28https%2Citsfoss.com%29/cache/caches.sqlite
/home/sergeyshuwaew/.mozilla/firefox/7f0i6xj2.default-release/storage/default/ht
tps+++www.youtube.com*partitionKey=%28https%2Cgoogle.com%29/cache
/home/sergeyshuwaew/.mozilla/firefox/7f0i6xj2.default-release/storage/default/ht

```

Рис. 4.4: Определяю, какие файлы в домашнем каталоге начинаются с символа “с” с помощью утилиты `find`, прописываю ей в аргументах домашнюю директорию (тогда вывод относительно корневого каталога, а не домашнего будет), выбираю опцию `-name` (ищем по имени), и пишу маску, по которой будем искать имя, где `*` - любое кол-во любых символов, добавляю опцию `-print`, чтобы мне вывелся результат ,

```

[sergeyshuwaew@sergeyshuwaew ~]$ ls -lR | grep c*
ab c l
australia
bin
feathers
file.txt
fun
june
letters
may
memos
misk
monthly
montly.00

```

Рис. 4.5: Второй способ использовать утилиту `ls -lR` и использовать `grep`, чтобы найти элементы с первым символом с. Однако этот способ не работает для поиска файлов из подкаталогов каталога ,

```
[sergeyshuwaew@sergeyshuwaew ~]$ sudo find /etc -name "h*" -print
[sudo] пароль для sergeyshuwaew:
/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
/etc/firewalld/helpers
/etc/hp
/etc/hp/hplip.conf
```

Рис. 4.6: С помощью метода find, чьи опции я расписала ранее, ищу все файлы, начинающиеся с буквы “h”,

```
[sergeyshuwaew@sergeyshuwaew ~]$ find ~ -name "log*" -print > logfile &
[1] 5654
```

Рис. 4.7: Запускаю в фоновом режиме (на это указывает символ &) процесс, который будет записывать в файл logfile (с помощью перенаправления >) файлы, имена которых начинаются с log (,

```
[sergeyshuwaew@sergeyshuwaew ~]$ ls
abcl fun memos play Видео Общедоступные
australia june misk reports Документы 'Рабочий стол'
bin letters monthly ski.plases Загрузки Шаблоны
feathers logfile montly.00 tutorial Изображения
file.txt may my_os work Музыка
[1]+ Завершён find ~ -name "log*" -print > logfile
```

Рис. 4.8: Проверяю, что файл создан, удаляю его, проверяю, что файл удален,

```
[sergeyshuwaew@sergeyshuwaew ~]$ rm logfile
[sergeyshuwaew@sergeyshuwaew ~]$ ls
abcl fun misk reports Документы 'Рабочий стол'
australia june monthly ski.plases Загрузки Шаблоны
bin letters montly.00 tutorial Изображения
feathers may my_os work Музыка
file.txt memos play Видео Общедоступные
```

Рис. 4.9:

```
[sergeyshuwaew@sergeyshuwaew ~]$ gedit
bash: gedit: команда не найдена...
Установить пакет «gedit», предоставляющий команду «gedit»?

* Ожидание в очереди...
Следующие пакеты должны быть установлены:
  gedit-2:43~alpha-2.fc37.x86_64 Text editor for the GNOME
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...

[sergeyshuwaew@sergeyshuwaew ~]$ gedit
```

Рис. 4.10: Запускаю в консоли в фоновом режиме (с помощью символа &) редактор gedit

```
[sergeyshuwaew@sergeyshuwaew ~]$ gedit
[sergeyshuwaew@sergeyshuwaew ~]$ ps aux | grep gedit
sergeys+  6619  0.0  0.0 222044  2312 pts/0    S+   13:20   0:00 grep --color=
auto gedit
```

Рис. 4.11: С помощью утилиты ps определяю идентификатор процесса gedit его идентификатор 6619

```
[sergeyshuwaew@sergeyshuwaew ~]$ pgrep gedit
[sergeyshuwaew@sergeyshuwaew ~]$ ps aux | grep gedit | grep -v grep
[sergeyshuwaew@sergeyshuwaew ~]$
```

Рис. 4.12: Также мы можем определить идентификатор с помощью pgrep.

```
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

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

Рис. 4.13: Прочитал справку команды kill

```
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 cur-
    rently 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 ver-
    sion of df cannot show the space available on unmounted file systems,
    because on most kinds of systems doing so requires very nonportable in-
    timate knowledge of file system structures.

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

Рис. 4.14: Прочитал документацию про функции df и 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
    Manual page du(1) line 1 (press h for help or q to quit)

```

Рис. 4.15:

Файловая система	Инодов	ИИспользовано	ИСвободно	ИИспользовано%	Смонтировано в
devtmpfs	1048576	485	1048091	1%	/dev
tmpfs	501464	224	501240	1%	/dev/shm
tmpfs	819200	984	818216	1%	/run
/dev/sda3	0	0	0	-	/
tmpfs	1048576	65	1048511	1%	/tmp
/dev/sda3	0	0	0	-	/home
/dev/sda2	65536	401	65135	1%	/boot
tmpfs	100292	162	100130	1%	/run/user/1000
/dev/sr0	0	0	0	-	/run/media/serge
yshuwaew/VBox_GAs_7.0.6					

Рис. 4.16: Используя утилиту df опции -iv позволяют увидеть информацию об инодах и сделать вывод читаемым, игнорируя сообщения системы о нем. Эта утилита нам нужна, чтобы выяснить, сколько свободного места есть у нашей системы.,

```
./local/share/Trash/info/presentation.5.pdf.trashinfo
./local/share/Trash/info/12.5.png.trashinfo
./local/share/Trash/info/14.png.trashinfo
./local/share/Trash/info/report.3.docx.trashinfo
./local/share/Trash/info/report.5.pdf.trashinfo
./local/share/Trash/info/presentation.6.html.trashinfo
./local/share/Trash/info/presentation.6.pdf.trashinfo
./local/share/Trash/info/presentation.7.html.trashinfo
./local/share/Trash/info/presentation.7.pdf.trashinfo
./local/share/Trash/info/report.4.docx.trashinfo
./local/share/Trash/info/report.6.pdf.trashinfo
./local/share/Trash/info/presentation.8.html.trashinfo
./local/share/Trash/info/presentation.8.pdf.trashinfo
./local/share/Trash/info/parentdir.trashinfo
./local/share/Trash/info/presentation.9.html.trashinfo
./local/share/Trash/info/presentation.9.pdf.trashinfo
./local/share/Trash/info/1.6.png.trashinfo
./local/share/Trash/info/1.7.png.trashinfo
./local/share/Trash/info/ski.places equipment.trashinfo
./local/share/Trash/info/2.3.png.trashinfo
./local/share/Trash/info/3.3.png.trashinfo
./local/share/Trash/info/5.2.png.trashinfo
./local/share/Trash/info/8.3.png.trashinfo
./local/share/Trash/info/18.png.trashinfo
```

# Выводы

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



## 5 Ответы на контрольные вопросы

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 вывод сообщений об ошибках (по умолчанию: консоль), файловый дескриптор 2.

Объясните разницу между операцией `>` и `>>`. Этот знак `>` - перенаправление ввода/вывода, а `>>` - перенаправление в режиме добавления.

Что такое конвейер? Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.

Что такое процесс? Чем это понятие отличается от программы? Главное отличие между программой и процессом заключается в том, что программа - это набор инструкций, который позволяет ЦПУ выполнять определенную задачу, в то время как процесс - это исполняемая программа.

Что такое PID и GID? PPID - (`parent process ID`) идентификатор родительского процесса. Процесс может порождать и другие процессы. UID, GID - реальные идентификаторы пользователя и его группы, запустившего данный процесс.

Что такое задачи и какая команда позволяет ими управлять? Запущенные фоном программы называются задачами (`jobs`). Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач.

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

Команда `htop` похожа на команду `top` по выполняемой функции: они обе по-

казывают информацию о процессах в реальном времени, выводят данные о потреблении системных ресурсов и позволяют искать, останавливать и управлять процессами.

У обеих команд есть свои преимущества. Например, в программе `htop` реализован очень удобный поиск по процессам, а также их фильтрация. В команде `top` это не так удобно — нужно знать кнопку для вывода функции поиска.

Зато в `top` можно разделять область окна и выводить информацию о процессах в соответствии с разными настройками. В целом `top` намного более гибкая в настройке отображения процессов.

Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды. Команда `find` — это одна из наиболее важных и часто используемых утилит системы Linux. Это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям.

Утилита `find` предустановлена по умолчанию во всех Linux дистрибутивах, поэтому вам не нужно будет устанавливать никаких дополнительных пакетов. Это очень важная находка для тех, кто хочет использовать командную строку наиболее эффективно.

Команда `find` имеет такой синтаксис: `find [папка] [параметры] критерий шаблон [действие]` Пример: `find /etc -name "p*" -print`

Можно ли по контексту (содержанию) найти файл? Если да, то как? `find / -type f -exec grep -H 'текстДляПоиска' {} ;`

Как определить объем свободной памяти на жёстком диске? С помощью команды `df -h`.

Как определить объем вашего домашнего каталога? С помощью команды `du -s`.

Как удалить зависший процесс? С помощью команды `kill%` номер задачи.