

Лабораторная работа No 6.

Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов

Боровиков Даниил Александрович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	11
4	Контрольные вопросы	12

Список иллюстраций

2.1	Запись в файл file.txt названия файлов, содержащихся в каталоге /etc	6
2.2	Запись в файл conf.txt имена файлов, имеющих расширение .conf	7
2.3	Два варианта поиска файлов начинавшиеся с символа с	7
2.4	Файлы из каталога /etc, начинающиеся с символа h	7
2.5	Фоновый процесс, который записывает в файл ~/logfile файлы, имена которых начинаются с log	8
2.6	Удаление ~/logfile	8
2.7	gedit в фоновом режиме	8
2.8	Определение идентификатора процесса gedit	9
2.9	Завершение процесса gedit	9
2.10	Команды df и du	9
2.11	Вывод имен всех директорий домашнего каталога	10

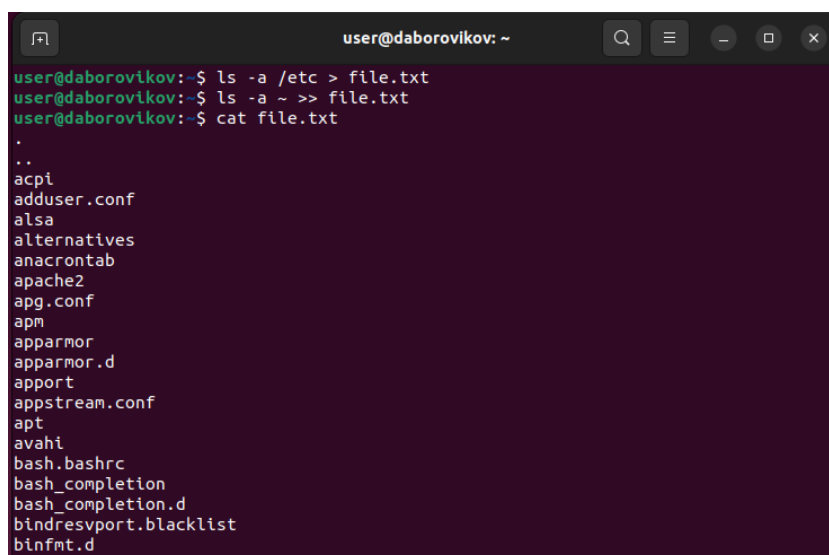
Список таблиц

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 Выполнение лабораторной работы

Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишем в этот же файл названия файлов, содержащихся в домашнем каталоге.(рис. fig. 2.1).



```
user@daborovikov: ~  
user@daborovikov:~$ ls -a /etc > file.txt  
user@daborovikov:~$ ls -a ~ >> file.txt  
user@daborovikov:~$ cat file.txt  
.  
..  
acpi  
adduser.conf  
alsa  
alternatives  
anacrontab  
apache2  
apg.conf  
apm  
apparmor  
apparmor.d  
appport  
appstream.conf  
apt  
avahi  
bash.bashrc  
bash_completion  
bash_completion.d  
bindresvport.blacklist  
binfmt.d
```

Рис. 2.1: Запись в файл file.txt названия файлов, содержащихся в каталоге /etc

Выведем имена всех файлов из file.txt, имеющих расширение .conf, после чего запишите их в новый текстовый файл conf.txt.(рис. fig. 2.2).

```
user@daborovikov: ~  
user@daborovikov:~$ grep -e '\.conf$' file.txt > conf.txt  
user@daborovikov:~$ cat file.txt  
..  
acpi  
adduser.conf  
alsa  
alternatives  
anacrontab  
apache2  
apgl.conf  
apm  
apparmor  
apparmor.d  
appport  
appstream.conf  
apt  
avahi  
bash.bashrc  
bash_completion  
bash_completion.d  
bindresvport.blacklist  
binfmt.d  
bluetooth
```

Рис. 2.2: Запись в файл conf.txt имена файлов, имеющих расширение .conf

Определим, какие файлы в домашнем каталоге имеют имена, начинавшиеся с символа с? Предложите несколько вариантов, как это сделать.(рис. fig. 2.3).

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

Рис. 2.3: Два варианта поиска файлов начинавшиеся с символа с

Выведем на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h.(рис. fig. 2.4).

```
user@daborovikov:~$ find /etc ~ -maxdepth 1 -name "h*" -print | less  
user@daborovikov:~$
```

Рис. 2.4: Файлы из каталога /etc, начинающиеся с символа h

Запустим в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log.(рис. fig. 2.5).

```

user@daborovikov:~$ find / -name "log*" > logfile &
[1] 9530
user@daborovikov:~$ find: '/lost+found': Отказано в доступе
find: '/root': Отказано в доступе
find: '/run/wpa_supplicant': Отказано в доступе
find: '/run/gdm3': Отказано в доступе
find: '/run/udisks2': Отказано в доступе
find: '/run/cups/certs': Отказано в доступе
find: '/run/user/1000/systemd/inaccessible/dir': Отказано в доступе
find: '/run/sudo': Отказано в доступе
find: '/run/speech-dispatcher': Отказано в доступе
find: '/run/openvpn-server': Отказано в доступе
find: '/run/openvpn-client': Отказано в доступе
find: '/run/credentials/systemd-sysusers.service': Отказано в доступе
find: '/run/systemd/resolve/netif': Отказано в доступе
find: '/run/systemd/propagate': Отказано в доступе
find: '/run/systemd/unit-root': Отказано в доступе
find: '/run/systemd/inaccessible/dir': Отказано в доступе
find: '/run/initramfs': Отказано в доступе
find: '/snap/core20/1822/etc/ssl/private': Отказано в доступе
find: '/snap/core20/1822/root': Отказано в доступе
find: '/snap/core20/1822/var/cache/ldconfig': Отказано в доступе
find: '/snap/core20/1822/var/cache/private': Отказано в доступе

```

Рис. 2.5: Фоновый процесс, который записывает в файл ~/logfile файлы, имена которых начинаются с log

Удалим файл ~/logfile.(рис. fig. 2.6).

```

user@daborovikov:~$ rm logfile
[1]+  Выход 1          find / -name "log*" > logfile
user@daborovikov:~$ ls
'2023-03-18 16-58-48.mkv'  feathers      play          Документы
'2023-03-18 16-59-48.mkv'  file.old     mycharmProjects  Загрузки
abci                     file.txt     ski.plases.     Изображения
australia                my os       snap          Музыка
ElionProjects            parentdir   tmp           Общедоступные
conf.txt                 parentdir1  usr           Рабочий стол
danya                   parentdir2  work          Шаблоны
equipment                parentdir3  Видео

```

Рис. 2.6: Удаление ~/logfile

Запустим из консоли в фоновом режиме редактор gedit.(рис. fig. 2.7).

```

user@daborovikov:~$ gedit &
[1] 10101
user@daborovikov:~$

```

Рис. 2.7: gedit в фоновом режиме

Определим идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. (рис. fig. 2.8).


```

user@daborovikov:~$ ps
  PID TTY          TIME CMD
  7267 pts/0        00:00:00 bash
 10101 pts/0        00:00:01 gedit
 10234 pts/0        00:00:00 ps
user@daborovikov:~$ ps | grep -i "gedit"
 10101 pts/0        00:00:01 gedit
user@daborovikov:~$ ps | grep -i "gedit"
 10101 pts/0        00:00:01 gedit
user@daborovikov:~$ ps | grep -i "gedit"
[1]+  Завершён          gedit
user@daborovikov:~$

```

Рис. 2.8: Определение идентификатора процесса gedit

Прочитаем справку (man) команды kill, после чего используем её для завершения процесса gedit.(рис. fig. 2.9).

```

user@daborovikov:~$ man kill
user@daborovikov:~$ gedit &
[2] 11072
[1]  Завершён          gedit
user@daborovikov:~$ kill 11072
user@daborovikov:~$ ps | grep -i "gedit"
[2]+  Завершено          gedit
user@daborovikov:~$

```

Рис. 2.9: Завершение процесса gedit

Выполним команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man(рис. fig. 2.10).

```

user@daborovikov:~$ man df
user@daborovikov:~$ man du
user@daborovikov:~$ df
Файл.система  1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
tmpfs          363096      2080          361016      1% /run
/dev/sda6      34854488    31506596      1545140      96% /
tmpfs          1815460     216           1815244      1% /dev/shm
tmpfs          5120        4             5116         1% /run/lock
/dev/sda2      98304       30964         67340        32% /boot/efi
tmpfs          363092      4752          358340       2% /run/user/1000
user@daborovikov:~$ du
72  ./pki/nssdb
76  ./pki
1996 ./texlive2021/texmf-var/luatex-cache/generic/names
25708 ./texlive2021/texmf-var/luatex-cache/generic/fonts/otl
25712 ./texlive2021/texmf-var/luatex-cache/generic/fonts
27712 ./texlive2021/texmf-var/luatex-cache/generic
27716 ./texlive2021/texmf-var/luatex-cache
27720 /texlive2021/texmf-var

```

Рис. 2.10: Команды df и du

Воспользовавшись справкой команды find, выведем имена всех директорий,

имею- щихся в домашнем каталоге(рис. fig. 2.11).

```
user@daborovikov:~$ find ~/ -maxdepth 1 -type d -print
/home/user/
/home/user/.pki
/home/user/.texlive2021
/home/user/.thunderbird
/home/user/.ssh
/home/user/.java
/home/user/Шаблоны
/home/user/snap
/home/user/Музыка
/home/user/.local
/home/user/ski.places.
/home/user/usr
/home/user/Загрузки
/home/user/.android
/home/user/Видео
/home/user/parentdir3
/home/user/.cache
/home/user/Рабочий стол
/home/user/CLionProjects
/home/user/.texlive2022
/home/user/parentdir2
/home/user/Изображения
/home/user/parentdir
/home/user/work
/home/user/parentdir1
/home/user/tmp
/home/user/.gnupg
/home/user/Документы
/home/user/play
/home/user/PycharmProjects
/home/user/.config
/home/user/.mozilla
/home/user/Общедоступные
user@daborovikov:~$
```

Рис. 2.11: Вывод имен всех директорий домашнего каталога

3 Выводы

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

4 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 `>>`. Основное отличие: `>` : Перезаписывает существующий файл или создает файл, если файл с указанным именем отсутствует в каталоге. `>>` : добавляет существующий файл или создает файл, если файл с указанным именем отсутствует в каталоге.
3. Что такое конвейер? Конвейер (англ. *pipeline*) в терминологии операционных систем семейства Unix — некоторое множество процессов, для которых выполнено следующее перенаправление ввода-вывода: то, что выводит на поток стандартного вывода предыдущий процесс, попадает в поток стандартного ввода следующего процесса.
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? Принудительное завершение процесса и изменение его приоритета) можно выполнить и без команды `top`. Процессы в Linux имеют возможность обмениваться так называемыми “сигналами” с ядром и другими процессами. При получении сигнала процессом, управление передается подпрограмме его обработки или ядру, если такой подпрограммы не существует. В Linux имеется команда `kill`, которая позволяет послать заданному процессу любой сигнал.
 7. Найдите информацию об утилитах `top` и `htop`. Каковы их функции? `top` - интерактивный просмотрщик 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. `find` : Для поиска файлов из командной строки вы можете использовать команду “`find`”. У этой команды следующий синтаксис: `find path criteria action` - “`path`” - Секция для указания директории поиска. Если ничего не указано поиск идет по текущей директории. - “`criteria`” - Опции поиска.
- “`action`” - Опции, которые влияют на состояние поиска или контролируют его, например, - “`-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? Самый простой способ найти свободное место на диске в Linux - это используйте команду `df`. Команда `df` означает «свободное от диска» и, очевидно, показывает вам свободное и доступное дисковое пространство в системах Linux. Работы С Нами `-h` вариант, он показывает дисковое пространство в удобочитаемом формате (МБ и ГБ).
11. Как определить объем вашего домашнего каталога? В операционных системах на базе Linux посмотреть размер папки (директории) можно с помощью команды `du`. Эта команда, выполняемая в консоли, позволяет оценить используемый объем места на жестком диске отдельно по папкам и файлам, просуммировать результат, узнать общий размер папки.
12. Как удалить зависший процесс? Убиваем процессы в Linux — команды `ps`, `kill` и `killall` • Находим PID зависшего процесса Каждый процесс в Linux имеет свой идентификатор, называемый PID. • «Убиваем» процесс командой `kill`. Когда известен PID процесса, мы можем убить его командой `kill`. • Убиваем процессы командой `killall`.