

Лабораторная работа №8

Отчёт

Андрюшин Никита Сергеевич

Содержание

1 Цель работы	5
2 Задание	6
3 Выполнение лабораторной работы	8
4 Контрольные вопросы	21
5 Выводы	24
Список литературы	25

Список иллюстраций

3.1	Перенаправление в файл	8
3.2	Перенаправление в файл в режиме дозаписи	9
3.3	Использование grep	10
3.4	grep и перенаправление файла	11
3.5	Использование find	11
3.6	Вывод команды find	12
3.7	find и less	13
3.8	Вывод find и less	13
3.9	Запуск фоновой задачи	14
3.10	Содержимое файла	15
3.11	Удаление файла	15
3.12	Запуск gedit в фоновом режиме	16
3.13	Использование ps	16
3.14	Завершение процесса	16
3.15	Использование df	17
3.16	Использование du	17
3.17	Вывод du	18
3.18	Использование find для вывода каталогов	19
3.19	Вывод find	20

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.
[1]

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 текстовой 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 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, имеющихся в вашем домашнем каталоге.

3 Выполнение лабораторной работы

Попробуем перенаправить вывод команды ls в файл с помощью > (рис. 3.1).

```
[nsandryushin@nsandryushin ~]$ ls /etc > file.txt
[nsandryushin@nsandryushin ~]$ cat file.txt
abrt
adjtime
aliases
alsa
alternatives
anaconda
anacrontab
appstream.conf
apt
asound.conf
at.deny
audit
authselect
avahi
bash_completion.d
bashrc
```

Рис. 3.1: Перенаправление в файл

Теперь дозапишем в наш файл содержимое нашего домашнего каталога с помощью » (рис. 3.2).

```
[nsandryushin@nsandryushin ~]$ ls >> file.txt
[nsandryushin@nsandryushin ~]$ tac file.txt
Шаблоны
Рабочий стол
Общедоступные
Музыка
Изображения
Загрузки
Документы
Доклад.zip
Доклад
Видео
work
ski.plases
reports
play
pass.txt
my_os
monthly
may
LICENSE
Lab1
git-extended
file.txt
feathers
Downloads
australia
abc1
yum.repos.d
xml
x12tdd
```

Рис. 3.2: Перенаправление в файл в режиме дозаписи

С помощью grep выведем содержимое нашего файла, куда мы записывали содержимое каталогов, таким образом, чтобы выводились только файлы с расширением conf (рис. 3.3).

```
[nsandryushin@nsandryushin ~]$ grep "\.conf" file.txt
appstream.conf
asound.conf
chrony.conf
dnsmasq.conf
dracut.conf
dracut.conf.d
fprintd.conf
fuse.conf
host.conf
idmapd.conf
ipsec.conf
kdump.conf
krb5.conf
krb5.conf.d
ld.so.conf
ld.so.conf.d
libaudit.conf
libuser.conf
locale.conf
logrotate.conf
makedumpfile.conf.sample
man_db.conf
mke2fs.conf
mtools.conf
ndctl.conf.d
nfs.conf
nfsmount.conf
nilfs_cleanerd.conf
nsswitch.conf
opensc.conf
```

Рис. 3.3: Использование grep

Выполним ту же команду, только перенаправим вывод в файл (рис. 3.4).

```
[nsandryushin@nsandryushin ~]$ grep "\.conf" file.txt > conf.tx  
t
```

Рис. 3.4: grep и перенаправление файла

Найдём в домашнем каталоге файлы, которые начинаются на “с” с помощью команды find (рис. 3.5).

```
[nsandryushin@nsandryushin ~]$ find ~ -name "c*" -print
```

Рис. 3.5: Использование find

Мы увидим следующее (рис. 3.6).

```
/home/nsandryushin/work/blog/public/media/icons/brands/coursera
.svg
/home/nsandryushin/work/blog/public/media/albums/demo/chris-mon
tgomery-smgtvepind4-unsplash.jpg
/home/nsandryushin/work/blog/public/media/albums/demo/chris-mon
tgomery-smgtvepind4-unsplash_hu68f1f4ae312ace90ad56e06a7b965ca9
_245831_750x750_fit_q75_h2_lanczos.webp
/home/nsandryushin/work/blog/public/post/ci-cd
/home/nsandryushin/work/blog/public/categories
/home/nsandryushin/work/blog/public/category
/home/nsandryushin/work/blog/public/publication/conference-pape
r
/home/nsandryushin/work/blog/public/publication/conference-pape
r/conference-paper.pdf
/home/nsandryushin/work/blog/public/publication/conference-pape
r/cite.bib
/home/nsandryushin/work/blog/public/publication/journal-article
/cite.bib
/home/nsandryushin/work/blog/public/ci-cd
/home/nsandryushin/work/DrNikiyProgrammingAccount.github.io/.gi
t/hooks/commit-msg.sample
/home/nsandryushin/work/DrNikiyProgrammingAccount.github.io/.gi
t/config
/home/nsandryushin/git-extended/.git/hooks/commit-msg.sample
/home/nsandryushin/git-extended/.git/objects/74/cc48134d95aabab
17eaf26da7dc201946a9455
/home/nsandryushin/git-extended/.git/objects/cb
/home/nsandryushin/git-extended/.git/config
/home/nsandryushin/.gnupg/common.conf
/home/nsandryushin/.password-store/.git/hooks/commit-msg.sample
```

Рис. 3.6: Вывод команды find

Теперь выведем постранично файлы, которые начинаются на “h”, с помощью того же find. Для этого создадим конвеер, и передадим вывод в команду less (рис. 3.7).

```
[nsandryushin@nsandryushin ~]$ find ~ -name "h*" -print | less
```

Рис. 3.7: find и less

Увидим следующее (рис. 3.8).

```
/home/nsandryushin/.mozilla/firefox/evrl18ok.default-release/st  
orage/default/https+++www.google.com  
/home/nsandryushin/.mozilla/firefox/evrl18ok.default-release/st  
orage/default/https+++esystem.rudn.ru  
/home/nsandryushin/.mozilla/firefox/evrl18ok.default-release/st  
orage/default/https+++github.com  
/home/nsandryushin/.mozilla/firefox/evrl18ok.default-release/st  
orage/default/https+++fedoraproject.org  
/home/nsandryushin/.mozilla/firefox/evrl18ok.default-release/st  
orage/default/https+++fonts2u.com  
/home/nsandryushin/.mozilla/firefox/evrl18ok.default-release/st  
orage/default/https+++www.google.com^partitionKey=%28https%2Cfo  
nts2u.com%29  
/home/nsandryushin/.mozilla/firefox/evrl18ok.default-release/st  
orage/default/https+++ru.ffffonts.net  
/home/nsandryushin/.mozilla/firefox/evrl18ok.default-release/st  
orage/default/https+++www.google.com^partitionKey=%28https%2Cff  
onts.net%29  
/home/nsandryushin/.mozilla/firefox/evrl18ok.default-release/st  
orage/default/https+++fedora.pkgs.org  
/home/nsandryushin/.mozilla/firefox/evrl18ok.default-release/st  
orage/default/https+++www.google.com^partitionKey=%28https%2Cpk  
gs.org%29  
/home/nsandryushin/.mozilla/firefox/evrl18ok.default-release/st  
orage/default/https+++www.linux.org.ru
```

Рис. 3.8: Вывод find и less

Теперь запишем в файл имена файлов, начинающиеся с “log”, но в фоновом режиме с помощью & (рис. 3.9).

```
[nsandryushin@nsandryushin ~]$ find ~ -name "log*" -print >> logfile &
[1] 2734
```

Рис. 3.9: Запуск фоновой задачи

Содержимое будет выглядеть так (рис. 3.10).

```
@4.1.0/node_modules/log-symbols
/home/nsandryushin/.local/share/pnpm/global/5/.pnpm/node_modules/log-symbols
/home/nsandryushin/.local/share/pnpm/global/5/.pnpm/handlebars@4.7.8/node_modules/handlebars/lib/handlebars/helpers/log.js
/home/nsandryushin/.local/share/pnpm/global/5/.pnpm/handlebars@4.7.8/node_modules/handlebars/lib/handlebars/logger.js
/home/nsandryushin/.local/share/pnpm/global/5/.pnpm/handlebars@4.7.8/node_modules/handlebars/dist/amd/handlebars/helpers/log.js
/home/nsandryushin/.local/share/pnpm/global/5/.pnpm/handlebars@4.7.8/node_modules/handlebars/dist/amd/handlebars/logger.js
/home/nsandryushin/.local/share/pnpm/global/5/.pnpm/handlebars@4.7.8/node_modules/handlebars/dist/cjs/handlebars/helpers/log.js
/home/nsandryushin/.local/share/pnpm/global/5/.pnpm/handlebars@4.7.8/node_modules/handlebars/dist/cjs/handlebars/logger.js
/home/nsandryushin/.local/share/pnpm/global/5/.pnpm/neo-async@2.6.2/node_modules/neo-async/log.js
/home/nsandryushin/.local/share/chezmoi/.git/logs
/home/nsandryushin/work/study/2023-2024/Операционные системы/os-intro/.git/logs
/home/nsandryushin/work/study/2023-2024/Операционные системы/os-intro/.git/modules/template/presentation/logs
/home/nsandryushin/work/study/2023-2024/Операционные системы/os-intro/.git/modules/template/report/logs 】
/home/nsandryushin/work/blog/.git/logs
/home/nsandryushin/work/blog/.git/modules/public/logs
/home/nsandryushin/work/DrNikiyProgrammingAccount.github.io/.git/logs
```

Рис. 3.10: Содержимое файла

Удалим этот файл (рис. 3.11).

```
[nsandryushin@nsandryushin ~]$ rm logfile
```

Рис. 3.11: Удаление файла

Запустим gedit в фоновом режиме (рис. 3.12).

```
[nsandryushin@nsandryushin ~]$ gedit &  
[1] 2746
```

Рис. 3.12: Запуск gedit в фоновом режиме

Посмотрим на pid этого процесса с помощью ps (рис. 3.13).

```
[nsandryushin@nsandryushin ~]$ ps | grep gedit  
2746 pts/0    00:00:03 gedit -
```

Рис. 3.13: Использование ps

Убьём процесс gedit по его pid (рис. 3.14).

```
[nsandryushin@nsandryushin ~]$ kill 2746
```

Рис. 3.14: Завершение процесса

Посмотрим на размер доступного места в системе с помощью df (рис. 3.15).

[nsandryushin@nsandryushin ~]\$ df	Файловая система 1К-блоков	Использовано	Доступно	Использовано%
Смонтировано в				
/dev/sda3	82834432	16217644	65653508	20%
/				
devtmpfs	4096	0	4096	0%
/dev				
tmpfs	3088212	3412	3084800	1%
/dev/shm				
tmpfs	1235288	1184	1234104	1%
/run				
tmpfs	3088216	4	3088212	1%
/tmp				
/dev/sda3	82834432	16217644	65653508	20%
/home				
/dev/sda2	996780	273168	654800	30%
/boot				
work	481793020	435590764	46202256	91%
/media/sf_work				
tmpfs	617640	96	617544	1%
/run/user/1000				

Рис. 3.15: Использование df

И посмотрим на занимаемое место с помощью du (рис. 3.16).

```
[nsandryushin@nsandryushin ~]$ du
```

Рис. 3.16: Использование du

Увидим следующее (рис. 3.17).

```
4      ./password-store/.git/objects/5a
4      ./password-store/.git/objects/b2
4      ./password-store/.git/objects/e8
4      ./password-store/.git/objects/f9
4      ./password-store/.git/objects/ab
4      ./password-store/.git/objects/7c
4      ./password-store/.git/objects/c6
4      ./password-store/.git/objects/e6
4      ./password-store/.git/objects/63
4      ./password-store/.git/objects/5b
4      ./password-store/.git/objects/f2
4      ./password-store/.git/objects/91
48      ./password-store/.git/objects
4      ./password-store/.git/logs/refs/heads
4      ./password-store/.git/logs/refs/remotes/origin
4      ./password-store/.git/logs/refs/remotes
8      ./password-store/.git/logs/refs
12      ./password-store/.git/logs
164      ./password-store/.git
35184      ./password-store/bin
35360      ./password-store
32      ./bashrc.d
196      ./Downloads
0      ./reports/monthly/monthly
0      ./reports/monthly
0      ./reports
0      ./monthly
8      ./ski.plases/equipment
0      ./ski.plases/plans
8      ./ski.plases
0      ./australia
0      ./play/games
0      ./play
1764088 .
```

Рис. 3.17: Вывод du

Выведем все директории в домашнем каталоге с помощью find, указав в аргу-

менте -type букву “d” (directory) (рис. 3.18).

```
[nsandryushin@nsandryushin ~]$ find ~ -name "*" -type d -print
```

Рис. 3.18: Использование find для вывода каталогов

Увидим следующее (рис. 3.19).

```
/home/nsandryushin/.password-store/.git/objects  
/home/nsandryushin/.password-store/.git/objects/pack  
/home/nsandryushin/.password-store/.git/objects/info  
/home/nsandryushin/.password-store/.git/objects/5a  
/home/nsandryushin/.password-store/.git/objects/b2  
/home/nsandryushin/.password-store/.git/objects/e8  
/home/nsandryushin/.password-store/.git/objects/f9  
/home/nsandryushin/.password-store/.git/objects/ab  
/home/nsandryushin/.password-store/.git/objects/7c  
/home/nsandryushin/.password-store/.git/objects/c6  
/home/nsandryushin/.password-store/.git/objects/e6  
/home/nsandryushin/.password-store/.git/objects/63  
/home/nsandryushin/.password-store/.git/objects/5b  
/home/nsandryushin/.password-store/.git/objects/f2  
/home/nsandryushin/.password-store/.git/objects/91  
/home/nsandryushin/.password-store/.git/logs  
/home/nsandryushin/.password-store/.git/logs/refs  
/home/nsandryushin/.password-store/.git/logs/refs/heads  
/home/nsandryushin/.password-store/.git/logs/refs/remotes  
/home/nsandryushin/.password-store/.git/logs/refs/remotes/origi  
n  
/home/nsandryushin/.password-store/bin  
/home/nsandryushin/.bashrc.d  
/home/nsandryushin/Downloads  
/home/nsandryushin/reports  
/home/nsandryushin/reports/monthly  
/home/nsandryushin/reports/monthly/monthly  
/home/nsandryushin/monthly  
/home/nsandryushin/ski.plases  
/home/nsandryushin/ski.plases/equipment  
/home/nsandryushin/ski.plases/plans  
/home/nsandryushin/australia  
/home/nsandryushin/play  
/home/nsandryushin/play/games
```

Рис. 3.19: Вывод find

4 Контрольные вопросы

1. В системе по умолчанию открыты три особых потока: `stdin` — это стандартный поток ввода (по умолчанию это клавиатура), его файловый дескриптор равен 0.
`stdout` — это стандартный поток вывода (по умолчанию это консоль), его файловый дескриптор равен 1.
`stderr` — это стандартный поток вывода сообщений об ошибках (по умолчанию это консоль), его файловый дескриптор равен 2.
2. Символ `>` используется для перенаправления ввода/вывода, а символ `»` используется для перенаправления в режиме добавления.
3. Конвейер (`r|p`) используется для объединения отдельных команд или утилит в цепочку, в которой вывод одной команды передается на вход следующей команды.
4. Основное различие между программой и процессом заключается в том, что программа представляет собой набор инструкций, предназначенных для выполнения определенной задачи центральным процессором (ЦПУ), в то время как процесс - это экземпляр исполняемой программы, который активно выполняется в операционной системе.
5. PID (Process ID) - это идентификатор процесса, который уникально идентифицирует каждый запущенный процесс в операционной системе.
GID (Group ID) - это идентификатор группы, который определяет принад-

лежность процесса к определенной группе пользователей в операционной системе.

6. Программы, запущенные в фоновом режиме, действительно называются задачами (*jobs*). Управлять ими можно с помощью команды *jobs*, которая выводит список запущенных в данный момент задач.
7. Команда *htop* и команда *top* выполняют аналогичные функции, показывая информацию о процессах в реальном времени и отображая данные о потреблении системных ресурсов. Обе команды также предоставляют возможность поиска, остановки и управления процессами.
Однако у них есть различия и преимущества. Например, в *htop* реализован более удобный поиск и фильтрация процессов, что делает его использование более интуитивно понятным по сравнению с *top*, где для активации функции поиска требуется знать соответствующую комбинацию клавиш. С другой стороны, в *top* можно разделить область окна и настроить отображение информации о процессах согласно различным настройкам, что делает его более гибким в настройке отображения.
8. Команда *find* является одной из наиболее важных и часто используемых утилит в системе Linux. Она предназначена для поиска файлов и каталогов на основе определенных условий. *find* можно применять в различных сценариях, таких как поиск файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям.

Утилита *find* по умолчанию предустановлена во всех дистрибутивах Linux, что обеспечивает готовность к использованию без необходимости установки дополнительных пакетов. Это делает *find* важным инструментом для эффективной работы в командной строке Linux.

Синтаксис команды *find* следующий: *find* путь параметры критерий действие. Например: *find /etc -name "p*" -print* – это команда, которая ищет файлы, начинающиеся с символа “*p*” в каталоге */etc* и выводит результаты

поиска.

9. Да, можно использовать команду find в сочетании с grep для поиска текста в файлах. Пример использования:

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

Эта команда будет рекурсивно искать файлы в корневом каталоге / и его подкаталогах. Затем она передаст каждый найденный файл в качестве аргумента команде grep, которая выполнит поиск строки ‘ТЕКСТ’ в каждом файле. Результатом будут строки с соответствующим текстом и именами файлов, в которых он найден.

10. С помощью df -h

11. С помощью команды du -s

12. С помощью команды kill PID

5 Выводы

В результате выполнения лабораторной работы были получены навыки работы с конвеером и перенаправлением потока вывода

Список литературы

1. Kulyabov. Лабораторная работа № 8. Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов. RUDN.