

Отчёт по лабораторной работе №8

Дисциплина: Операционные системы

Дарина Андреевна Куокконен

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 4 |
| 2 | Задание | 5 |
| 3 | Теоретическое введение | 6 |
| 4 | Выполнение лабораторной работы | 7 |
| 5 | Выводы | 13 |
| 6 | Ответы на контрольные вопросы | 14 |
| | Список литературы | 17 |

Список иллюстраций

| | | |
|------|--|----|
| 4.1 | Запись в файл | 7 |
| 4.2 | Проверка файла | 8 |
| 4.3 | Добавление в файл | 8 |
| 4.4 | Поиск файлов | 8 |
| 4.5 | Запись в файл | 9 |
| 4.6 | Поиск имен файлов | 9 |
| 4.7 | Поиск имен файлов | 9 |
| 4.8 | Поиск файлов | 10 |
| 4.9 | Запуск процесса в фоновом режиме | 10 |
| 4.10 | Удаление файла | 10 |
| 4.11 | Запуск процесса в фоновом режиме | 11 |
| 4.12 | Определение идентификатора разными способами | 11 |
| 4.13 | Удаление процесса | 11 |
| 4.14 | Команда df | 12 |
| 4.15 | Команда du | 12 |
| 4.16 | Выполнение команды | 12 |

1 Цель работы

Цель данной лабораторной 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

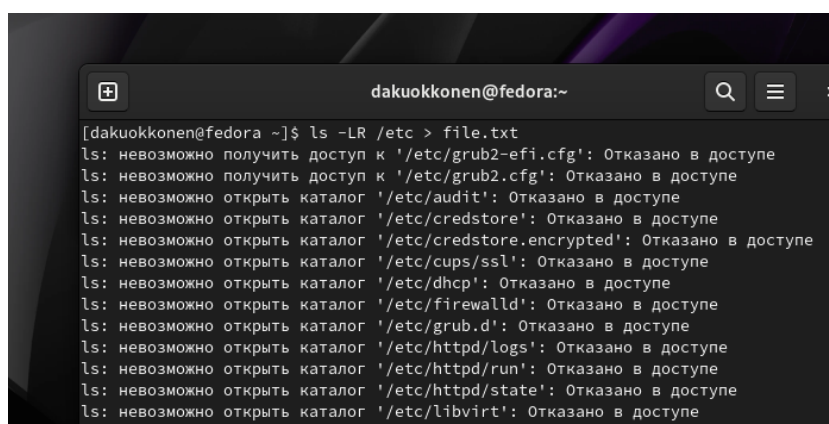
1. Осуществить вход в систему, используя соответствующее имя пользователя.
2. Записать в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`
3. Выведите имена всех файлов из `file.txt`, имеющих расширение `.conf`
4. Запустите в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`.
5. Удалить лишние файлы
6. Выполнить команды `df` и `du`
7. Ответы на контрольные вопросы

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 передается последующей. Синтаксис следующий: *команда 1* | *команда 2* - означает, что вывод команды 1 передается 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` передается 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.

4 Выполнение лабораторной работы

Вхожу в систему,используя соответствующее имя пользователя. С помощью перенаправления “>” добавляю в файл file.txt название файлов из /etc. (рис. 1).



```
dakuokkonen@fedora:~  
[dakuokkonen@fedora ~]$ ls -LR /etc > file.txt  
ls: невозможно получить доступ к '/etc/grub2-efi.cfg': Отказано в доступе  
ls: невозможно получить доступ к '/etc/grub2.cfg': Отказано в доступе  
ls: невозможно открыть каталог '/etc/audit': Отказано в доступе  
ls: невозможно открыть каталог '/etc/credstore': Отказано в доступе  
ls: невозможно открыть каталог '/etc/credstore.encrypted': 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/httpd/logs': Отказано в доступе  
ls: невозможно открыть каталог '/etc/httpd/run': Отказано в доступе  
ls: невозможно открыть каталог '/etc/httpd/state': Отказано в доступе  
ls: невозможно открыть каталог '/etc/libvirt': Отказано в доступе
```

Рис. 4.1: Запись в файл

С помощью команды head проверяю первые 10 строк файла (рис. 2).

```
[dakuokkonen@fedora ~]$ head file.txt
/etc:
abrt
adjtime
aliases
alsa
alternatives
anaconda
anthy-unicode.conf
appstream.conf
asound.conf
[dakuokkonen@fedora ~]$
```

Рис. 4.2: Проверка файла

Добавляю в файл имена из домашнего каталога(рис. 3).

```
[dakuokkonen@fedora ~]$ ls -LR ~/ >> file.txt
```

Рис. 4.3: Добавление в файл

Вывожу на экран имена всех файлов,содержащих расщирение .conf, с помощью команды grep (рис. 4).

```
dakuokkonen@fedora:~
[dakuokkonen@fedora ~]$ grep .conf file.txt
anthy-unicode.conf
appstream.conf
asound.conf
brltty.conf
chkconfig.d
chrony.conf
dconf
dley-na-renderer-service.conf
dley-na-server-service.conf
dnsmasq.conf
dracut.conf
dracut.conf.d
fprintd.conf
fuse.conf
host.conf
```

Рис. 4.4: Поиск файлов

Записываю найденные файлы в новый файл (рис. 5).

```
[dakuokkonen@fedora ~]$ grep .conf file.txt > conf.txt
[dakuokkonen@fedora ~]$ head 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
[dakuokkonen@fedora ~]$
```

Рис. 4.5: Запись в файл

Нахожу в домашнем каталоге все файлы, которые начинаются на “с” (рис. 6).

```
dakuokkonen@fedora:~
[dakuokkonen@fedora ~]$ find ~ -name "с*" -print
/home/dakuokkonen/.mozilla/firefox/w76kvl2.default-release/compatibility.ini
/home/dakuokkonen/.mozilla/firefox/w76kvl2.default-release/cookies.sqlite
/home/dakuokkonen/.mozilla/firefox/w76kvl2.default-release/cert9.db
/home/dakuokkonen/.mozilla/firefox/w76kvl2.default-release/storage/permanent/ch
rome
/home/dakuokkonen/.mozilla/firefox/w76kvl2.default-release/storage/default/http
s+++www.microsoft365.com/cache
/home/dakuokkonen/.mozilla/firefox/w76kvl2.default-release/storage/default/http
s+++www.microsoft365.com/cache/caches.sqlite
/home/dakuokkonen/.mozilla/firefox/w76kvl2.default-release/storage/default/http
s+++www.microsoft365.com/cache/caches.sqlite-wal
/home/dakuokkonen/.mozilla/firefox/w76kvl2.default-release/storage/default/http
s+++www.youtube.com^partitionKey=%28https%2Cfedoraproject.org%29/cache
/home/dakuokkonen/.mozilla/firefox/w76kvl2.default-release/storage/default/http
s+++www.youtube.com^partitionKey=%28https%2Cfedoraproject.org%29/cache/caches.sq
lite
/home/dakuokkonen/.mozilla/firefox/w76kvl2.default-release/storage/default/http
s+++www.youtube.com^partitionKey=%28https%2Cfedoraproject.org%29/cache/caches.sq
lite-wal
```

Рис. 4.6: Поиск имен файлов

Еще один способ найти элементы с первым символом. Однако этот способ не работает для поиска файлов из подкаталогов (рис. 7).

```
[dakuokkonen@fedora ~]$ ls -LR | grep с*
conf.txt
```

Рис. 4.7: Поиск имен файлов

С помощью опции `find`, вывожу на экран имена файлов из каталог `/etc`, начинающихся с символа `h` (рис. 8).

```
[dakuokkonen@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
/etc/containers/oci/hooks.d
find: '/etc/credstore.encrypted': Отказано в доступе
```

Рис. 4.8: Поиск файлов

Запускаю в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`. (рис. 9).

```
[dakuokkonen@fedora ~]$ find ~ -name "log*" -print > logfile &
[1] 3917
```

Рис. 4.9: Запуск процесса в фоновом режиме

Проверяю, что файл создан и удаляю его (рис. 10).

```
[dakuokkonen@fedora ~]$ ls
abcl      fun      package.json      Видео      'Рабочий стол'
australia logfile  package-lock.json  Документы  Шаблоны
bin       may      play              Загрузки
conf.txt  monthly  reports           Изображения
feathers  my_os    ski.places        Музыка
file.txt  node_modules  work              Общедоступные
[dakuokkonen@fedora ~]$ rm logfile
[dakuokkonen@fedora ~]$ ls
abcl      file.txt  node_modules      ski.places  Изображения
australia fun      package.json      work        Музыка
bin       may      package-lock.json  Видео      Общедоступные
conf.txt  monthly  play              Документы  'Рабочий стол'
feathers  my_os    reports           Загрузки   Шаблоны
[dakuokkonen@fedora ~]$
```

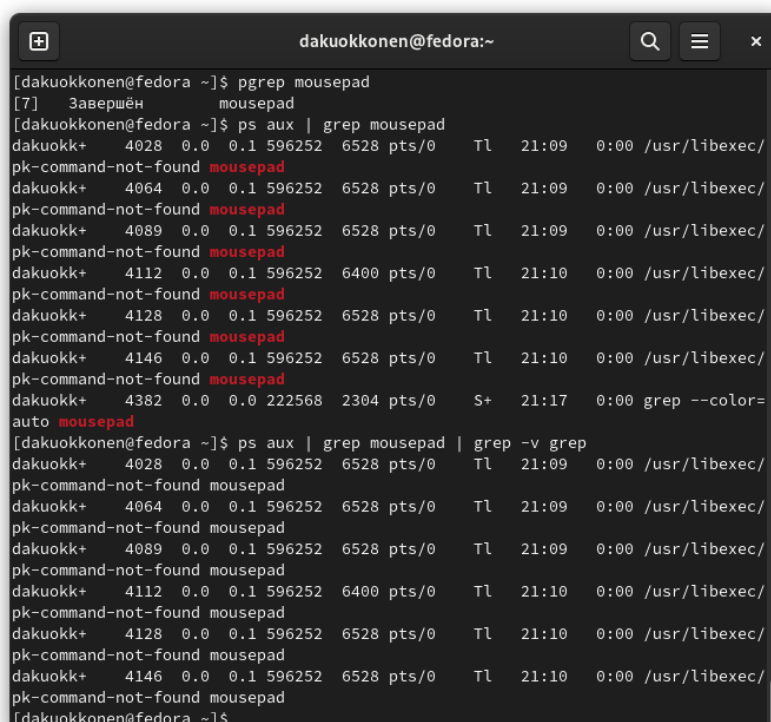
Рис. 4.10: Удаление файла

Запускаю в фоновом режиме редактор `mouesrad` (рис. 11).

```
[dakuokkonen@fedora ~]$ mousepad &  
[7] 4256  
[dakuokkonen@fedora ~]$
```

Рис. 4.11: Запуск процесса в фоновом режиме

Определяю идентификатор процесса mousepad, используя команду ps, конвейер и фильтр grep. (рис. 12).



```
dakuokkonen@fedora:~  
[dakuokkonen@fedora ~]$ pgrep mousepad  
[7] Завершён mousepad  
[dakuokkonen@fedora ~]$ ps aux | grep mousepad  
dakuokk+ 4028 0.0 0.1 596252 6528 pts/0 Tl 21:09 0:00 /usr/libexec/  
pk-command-not-found mousepad  
dakuokk+ 4064 0.0 0.1 596252 6528 pts/0 Tl 21:09 0:00 /usr/libexec/  
pk-command-not-found mousepad  
dakuokk+ 4089 0.0 0.1 596252 6528 pts/0 Tl 21:09 0:00 /usr/libexec/  
pk-command-not-found mousepad  
dakuokk+ 4112 0.0 0.1 596252 6400 pts/0 Tl 21:10 0:00 /usr/libexec/  
pk-command-not-found mousepad  
dakuokk+ 4128 0.0 0.1 596252 6528 pts/0 Tl 21:10 0:00 /usr/libexec/  
pk-command-not-found mousepad  
dakuokk+ 4146 0.0 0.1 596252 6528 pts/0 Tl 21:10 0:00 /usr/libexec/  
pk-command-not-found mousepad  
dakuokk+ 4382 0.0 0.0 222568 2304 pts/0 S+ 21:17 0:00 grep --color=  
auto mousepad  
[dakuokkonen@fedora ~]$ ps aux | grep mousepad | grep -v grep  
dakuokk+ 4028 0.0 0.1 596252 6528 pts/0 Tl 21:09 0:00 /usr/libexec/  
pk-command-not-found mousepad  
dakuokk+ 4064 0.0 0.1 596252 6528 pts/0 Tl 21:09 0:00 /usr/libexec/  
pk-command-not-found mousepad  
dakuokk+ 4089 0.0 0.1 596252 6528 pts/0 Tl 21:09 0:00 /usr/libexec/  
pk-command-not-found mousepad  
dakuokk+ 4112 0.0 0.1 596252 6400 pts/0 Tl 21:10 0:00 /usr/libexec/  
pk-command-not-found mousepad  
dakuokk+ 4128 0.0 0.1 596252 6528 pts/0 Tl 21:10 0:00 /usr/libexec/  
pk-command-not-found mousepad  
dakuokk+ 4146 0.0 0.1 596252 6528 pts/0 Tl 21:10 0:00 /usr/libexec/  
pk-command-not-found mousepad  
[dakuokkonen@fedora ~]$
```

Рис. 4.12: Определение идентификатора разными способами

Удаляю процесс (рис. 13).

```
[dakuokkonen@fedora ~]$ kill 4028  
[dakuokkonen@fedora ~]$
```

Рис. 4.13: Удаление процесса

Изучаю справку команды `df` и выполняю её,использовав утилиту `-iv`, которая позволяет увидеть информацию об инодах и сделать вывод читаемым.игнорирую сообщение системы о нем,так мы узнаем,сколько у нашей системы места (рис. 14).

```
[dakuokkonen@fedora ~]$ df -vi
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
devtmpfs          483250      508      482742          1% /dev
tmpfs             488376        2      488374          1% /dev/shm
tmpfs            819200      972      818228          1% /run
/dev/sda3          0            0            0             - /
tmpfs            1048576       48     1048528          1% /tmp
/dev/sda2          65536      401      65135          1% /boot
/dev/sda3          0            0            0             - /home
tmpfs             97675      156      97519          1% /run/user/1000
/dev/sr0           0            0            0             - /run/media/dakuokkonen/VB
_GAs_6.1.30
```

Рис. 4.14: Команда `df`

Изучаю справку команды `du` и выполняю её,использовав утилиту `-a`,которая позволяет увидеть,сколько памяти занимают все файлы (рис. 15).

```
[dakuokkonen@fedora ~]$ du -a
```

Рис. 4.15: Команда `du`

Выполняю команду `du` (рис. 16).

```
dakuokkonen@fedora:~ — du -a
16  ./cache/mozilla/firefox/w76kvl2.default-release/cache2/entries/DCEA136206AE2FE685
B560FB14E9852B17470437
12  ./cache/mozilla/firefox/w76kvl2.default-release/cache2/entries/6FC561E177CCE31E3E
324C6935D6162C925E2DFC
12  ./cache/mozilla/firefox/w76kvl2.default-release/cache2/entries/B5F948E27A5E468196
45A016C52D1B64408E5873
12  ./cache/mozilla/firefox/w76kvl2.default-release/cache2/entries/668B42BCD47FA09041
E1FE4358068865FFEDAC1B
12  ./cache/mozilla/firefox/w76kvl2.default-release/cache2/entries/5D58492F6B464F095F
60508DB23722989C55913E
20  ./cache/mozilla/firefox/w76kvl2.default-release/cache2/entries/CB055B4477CF85FF3F
B3FF90C1E7493055D6CFBE
12  ./cache/mozilla/firefox/w76kvl2.default-release/cache2/entries/68155D6325CECAD845
69C360B88925ABF51B10C4
12  ./cache/mozilla/firefox/w76kvl2.default-release/cache2/entries/79F114DCAA4C08C032
A1084481FD219755CDF718
52  ./cache/mozilla/firefox/w76kvl2.default-release/cache2/entries/17D50EFACDC01CFF3A
B65B12FB1F693C7DD7216F
12  ./cache/mozilla/firefox/w76kvl2.default-release/cache2/entries/768A27C5E7B34FBB35
AF9D66B420995519B1C229
12  ./cache/mozilla/firefox/w76kvl2.default-release/cache2/entries/D77DA62C66D0991BB7
910960CF53E069E7EC841
12  ./cache/mozilla/firefox/w76kvl2.default-release/cache2/entries/372D5A91B562FCEAFE
9C806EC745A444457ED8FE
```

Рис. 4.16: Выполнение команды

5 Выводы

Я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных, приобрела практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

6 Ответы на 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 вывод сообщений об ошибках (по умолчанию: консоль), файловый дескриптор 2.
2. Объясните разницу между операцией `>` и `»`. Этот знак `>` - перенаправление ввода/вывода, а `»` - перенаправление 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.
4. 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.
5. 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.
6. 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.

7. Найдите информацию об утилитах `top` и `htop`. Каковы их функции? Команда `htop` похожа на команду `top` по выполняемой функции: они обе показывают 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.

8. 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`

9. Можно ли по контексту (содержанию) найти файл? Если да, то как? `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? С помощью команды `kill%` номер задачи.

Список литературы

Архитектура ЭВМ