

Отчет по выполнению лабораторной работы №8

Операционные системы

Петрова Алевтина

Содержание

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 `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

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). То же 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 Макилро-ем 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

Я вошла в систему под соответствующим именем пользователя, открыла терминал, записала в файл file.txt названия файлов из каталога /etc с помощью перенаправления “>” (и файл создала, и записала в него то, что могло быть выведено ls -lR /etc). В файл я добавила также все файлы из подкаталогов (рис.1).

```
[petrovkina1002@fedora ~]$ sudo ls -lR /etc > file.txt
[sudo] пароль для petrovkina1002:
[petrovkina1002@fedora ~]$
```

Запись в файл

Проверила, что в файл записались нужные значения с помощью утилиты head, она выводит первые 10 строк файла на экран (рис.2).

```
[petrovkina1002@fedora ~]$ head file.txt
/etc:
итого 1452
drwxr-xr-x. 1 root root      126 июн 30  2023 abrt
-rw-r--r--. 1 root root       16 фев 21  13:25 adjtime
-rw-r--r--. 1 root root    1529 ноя 27  2022 aliases
drwxr-xr-x. 1 root root       70 янв 29  03:00 alsa
drwxr-xr-x. 1 root root    1536 мар 15  21:11 alternatives
drwxr-xr-x. 1 root root       56 апр 14  2023 anaconda
-rw-r--r--. 1 root root     269 янв 18  2023 anthy-unicode.conf
-rw-r--r--. 1 root root     833 фев 10  2023 appstream.conf
[petrovkina1002@fedora ~]$
```

Вывод содержимого файла

Добавила в созданный файл имена файлов из домашнего каталога, используя перенаправление “>>” в режиме добавления (рис.3).

```
[petrovkina1002@fedora ~]$ ls -lR ~/ >> file.txt  
[petrovkina1002@fedora ~]$
```

Добавление данных в файл

Вывела на экран имена всех файлов, имеющих расширение “.conf” с помощью утилиты grep (рис.4).

```
[petrovkina1002@fedora ~]$ grep .conf file.txt  
-rw-r--r--. 1 root root      269 янв 18  2023 anthy-unicode.conf  
-rw-r--r--. 1 root root      833 фев 10  2023 appstream.conf  
-rw-r--r--. 1 root root        55 янв 29 03:00 asound.conf  
-rw-r--r--. 1 root root    29842 янв 24  2023 brltty.conf  
drwxr-xr-x. 1 root root        0 янв 17 03:00 chkconfig.d  
-rw-r--r--. 1 root root    1423 фев 21 13:25 chrony.conf  
drwxr-xr-x. 1 root root        18 апр 14  2023 dconf  
-rw-r--r--. 1 root root    1280 сен 20  2023 dleyna-renderer-service.conf  
-rw-r--r--. 1 root root    1174 сен 20  2023 dleyna-server-service.conf  
-rw-r--r--. 1 root root   28485 апр 27  2023 dnsmasq.conf
```

Поиск файлов определенного расширения

Добавила вывод прошлой команды в новый файл conf.txt с помощью перенаправления “>” (файл создается при выполнении этой команды) (рис.5).

```
[petrovkina1002@fedora ~]$ grep .conf file.txt > conf.txt  
[petrovkina1002@fedora ~]$ head conf.txt  
-rw-r--r--. 1 root root      269 янв 18  2023 anthy-unicode.conf  
-rw-r--r--. 1 root root      833 фев 10  2023 appstream.conf  
-rw-r--r--. 1 root root        55 янв 29 03:00 asound.conf  
-rw-r--r--. 1 root root    29842 янв 24  2023 brltty.conf  
drwxr-xr-x. 1 root root        0 янв 17 03:00 chkconfig.d  
-rw-r--r--. 1 root root    1423 фев 21 13:25 chrony.conf  
drwxr-xr-x. 1 root root        18 апр 14  2023 dconf  
-rw-r--r--. 1 root root    1280 сен 20  2023 dleyna-renderer-service.conf  
-rw-r--r--. 1 root root    1174 сен 20  2023 dleyna-server-service.conf  
-rw-r--r--. 1 root root   28485 апр 27  2023 dnsmasq.conf  
[petrovkina1002@fedora ~]$
```

Запись в файл

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 (рис.6). Но таким образом я получаю информацию даже о файлах из подкаталогов домашнего каталога.

```
[petrovkina1002@fedora ~]$ find ~ -name "c*" -print
/home/petrovkina1002/.mozilla/firefox/ddf65akg.default-release/crashes
/home/petrovkina1002/.mozilla/firefox/ddf65akg.default-release/compatibility.ini
/home/petrovkina1002/.mozilla/firefox/ddf65akg.default-release/cookies.sqlite
/home/petrovkina1002/.mozilla/firefox/ddf65akg.default-release/cert9.db
/home/petrovkina1002/.mozilla/firefox/ddf65akg.default-release/storage/permanent/chrome
/home/petrovkina1002/.mozilla/firefox/ddf65akg.default-release/storage/default/https+++web.telegram.org/cache/
/home/petrovkina1002/.mozilla/firefox/ddf65akg.default-release/storage/default/https+++web.telegram.org/cache/
/home/petrovkina1002/.mozilla/firefox/ddf65akg.default-release/storage/default/https+++vk.com/cache/caches.sqlite-wal
/home/petrovkina1002/.mozilla/firefox/ddf65akg.default-release/storage/default/https+++vk.com/cache/caches.sqlite-wal
/home/petrovkina1002/.mozilla/firefox/ddf65akg.default-release/storage/default/https+++vk.com/cache/caches.sqlite-wal
```

Поиск файлов, начинающихся с определенного элемента

Второй способ использовать утилиту `ls -lR` и использовать `grep`, чтобы найти элементы с первым символом `c`. Однако этот способ не работает для поиска файлов из подкаталогов каталога (рис.7).

```
[petrovkina1002@fedora ~]$ ls -lR | grep c*
-rw-r--r--. 1 petrovkina1002 petrovkina1002 51871 map 28 13:24 conf.txt
[petrovkina1002@fedora ~]$
```

Поиск файлов, начинающихся с определенного элемента

С помощью метода `find`, чьи опции я расписала ранее, ищу все файлы, начинающиеся с буквы `"h"` (рис.8).

```
[petrovkina1002@fedora ~]$ sudo find /etc -name "h*" -print
[sudo] пароль для petrovkina1002:
/etc/avahi/hosts
/etc/brltty/Contraction/ha.ctb
/etc/brltty/Input/bm/horizontal.kti
/etc/brltty/Input/hd
/etc/brltty/Input/hm
```

Поиск файлов, начинающихся с определенного элемента

Запускаю в фоновом режиме (на это указывает символ `&`) процесс, который будет записывать в файл `logfile` (с помощью перенаправления `>`) файлы, имена которых начинаются с `log` (рис.9).

```
[petrovkina1002@fedora ~]$ find ~ -name "log*" -print > logfile &
[1] 3296
[petrovkina1002@fedora ~]$
```

Создание фонового процесса

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

```
[1] 3296
[petrovkina1002@fedora ~]$ ls
abc1      Downloads  logfile    play      Видео      'Рабочий стол'
ASD       feathers   may        presentation_example.md  Документы  Шаблоны
australia file.txt   monthly   reports   Загрузки
bin       fun        my_os     ski.places  Изображения
conf.txt  git-extended  os-intro  study_2023-2024_os-intro  Музыка
Desktop  LICENSE     Pictures  work       Общедоступные
[1]+  Завершен                  find ~ -name "log*" -print > logfile
[petrovkina1002@fedora ~]$
```

Удаление файла

Запускаю в консоли в фоновом режиме (с помощью символа &) редактор mousepad, потому что редактора gedit у меня, к сожалению, но работают они идентично (рис.11).

```
[1]+  Завершён      find -name logx -print > log.txt
[petrovkina1002@fedora ~]$ mousepad &
[1] 3448
[petrovkina1002@fedora ~]$
```

Создание фонового процесса

С помощью утилиты ps определяю идентификатор процесса mousepad(рис.12). Также мы можем определить идентификатор с помощью pgrep.

```
[petrovkina1002@fedora ~]$ ps aux | grep mousepad
petrovk+  3448  0.3  1.2 903488 51936 pts/0    Sl   13:43   0:00 mousepad
petrovk+  3522  0.0  0.0 222568 2432 pts/0    S+   13:45   0:00 grep --color=auto mousepad
[petrovkina1002@fedora ~]$
```

Поиск идентификатора процесса

Прочитала справку команды kill (рис.13).

```
KILL(1)                                User Commands
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.
```

Чтение документации

Использую команду kill и идентификатор процесса, чтобы его удалить (рис.14). Заметила, как у меня закрылась программа mousepad.

```
[petrovkina1002@fedora ~]$ man kill
[petrovkina1002@fedora ~]$ kill 3448
[petrovkina1002@fedora ~]$
```

Удаление процесса

Прочитала документацию про функции df и du (рис.15).

```
[petrovkina1002@fedora ~]$ man df
[1]+  Завершён      mousepad
[petrovkina1002@fedora ~]$ man du
[petrovkina1002@fedora ~]$
```

Чтение документации

Использую утилиту df опции -iv позволяют увидеть информацию об инодах и сделать вывод читаемым, игнорируя сообщения системы о нем (рис.16). Эта утилита нам нужна, чтобы выяснить, сколько свободного места есть у нашей системы.

```
[petrovkina1002@fedora ~]$ df -v1
Файловая система  Инодов  Ииспользовано  Исвободно  Ииспользовано%  Смонтировано в
devtmpfs          495403      527      494876          1% /dev
tmpfs             500532        2      500530          1% /dev/shm
tmpfs             819200      986      818214          1% /run
/dev/sda4          0            0            0            - /
tmpfs            1048576       46     1048530          1% /tmp
/dev/sda4          0            0            0            - /home
/dev/sda3         65536       401      65135           1% /boot
tmpfs            100106       164      99942           1% /run/user/1000
[petrovkina1002@fedora ~]$
```

Утилита df

Использую утилиту du. Она нужна чтобы просмотреть, сколько места занимают файлы в определенной директории и найти самые большие из них (рис.17).

```
[petrovkina1002@fedora ~]$ du -a ~/study_2023-2024_os-intro/labs/lab01/presentation
48 /home/petrovkina1002/study_2023-2024_os-intro/labs/lab01/presentation/image/kulyabov.jpg
12 /home/petrovkina1002/study_2023-2024_os-intro/labs/lab01/presentation/image/1.png
12 /home/petrovkina1002/study_2023-2024_os-intro/labs/lab01/presentation/image/2.png
48 /home/petrovkina1002/study_2023-2024_os-intro/labs/lab01/presentation/image/3.png
72 /home/petrovkina1002/study_2023-2024_os-intro/labs/lab01/presentation/image/4.png
36 /home/petrovkina1002/study_2023-2024_os-intro/labs/lab01/presentation/image/5.png
48 /home/petrovkina1002/study_2023-2024_os-intro/labs/lab01/presentation/image/6.png
84 /home/petrovkina1002/study_2023-2024_os-intro/labs/lab01/presentation/image/7.png
24 /home/petrovkina1002/study_2023-2024_os-intro/labs/lab01/presentation/image/8.png
28 /home/petrovkina1002/study_2023-2024_os-intro/labs/lab01/presentation/image/9.png
20 /home/petrovkina1002/study_2023-2024_os-intro/labs/lab01/presentation/image/10.png
```

Утилита du

Прочитала документацию о команде find (рис.18).

```
FIND(1)                                General Commands Manual

NAME
    find - search for files in a directory hierarchy

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

DESCRIPTION
    This manual page documents the GNU version of find. GNU find searches the directory tree rooted at the
    given starting-point by evaluating the given expression from left to right, according to the operator
    precedence (see section OPERATORS), until the outcome is known (the left hand side is false or the
    right hand side is true).
```

Чтение документации

Вывела имена всех директорий, имеющиххся в моем домашнем каталоге, используя аргумент d у утилиты find опции -type, то есть указываю тип файлов, который мне нужен и этот тип Директория (рис.19). Утилита -a позволит увидеть размер всех файлов, а не только диреткорий.

```
[petrovkina1002@fedora ~]$ find -type d
.
./.mozilla
./.mozilla/extensions
./.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
./.mozilla/plugins
./.mozilla/firefox
./.mozilla/firefox/Crash Reports
./.mozilla/firefox/Crash Reports/events
./.mozilla/firefox/Pending Pings
```

Название рисунка

5 Выводы

В результате данной лабораторной работы я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных, а также приобрела практические навыки по управлению процессами (и заданиями), по проверке использования диска и по 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 следующей.
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.

1. 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

1. Можно ли по контексту (содержанию) найти файл? Если да, то как? find / -type f -exec grep -H 'текстДляПоиска' {} ;
2. Как определить объем свободной памяти на жёстком диске? С помощью команды df -h.
3. Как определить объем вашего домашнего каталога? С помощью команды du -s.
4. Как удалить зависший процесс? С помощью команды kill% номер задачи.