

Отчет по лабораторной работе №8

Дисциплина: Операционные системы

Иванов Сергей Владимирович

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	7
4	Контрольные вопросы	14
5	Выводы	17

Список иллюстраций

3.1	Запись в файл	7
3.2	Вывод содержимого файла	7
3.3	Добавление данных в файл	8
3.4	Поиск файлов определенного расширения	8
3.5	Запись в файл	8
3.6	Поиск файлов, начинающихся с определенного элемента	9
3.7	Поиск файлов, начинающихся с определенного элемента	9
3.8	Поиск файлов, начинающихся с определенного элемента	9
3.9	Создание фонового процесса	10
3.10	Удаление файла	10
3.11	Создание фонового процесса	10
3.12	Поиск идентификатора процесса	10
3.13	Чтение документации	11
3.14	Удаление процесса	11
3.15	Чтение документации	11
3.16	Утилита df	12
3.17	Утилита du	12
3.18	Чтение документации	12
3.19	Вывод директорий	13

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, начинающиеся 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 директо-

рий, имею- щихся в вашем домашнем каталоге.

3 Выполнение лабораторной работы

Вхожу в систему под соответствующим именем пользователя, открываю терминал, записываю в файл file.txt названия файлов из каталога /etc с помощью перенаправления “>”. В файл я добавил также все файлы из подкаталогов (рис. 1).

```
[root@svivanov1 ~]# ls -lR /etc > file.txt  
[root@svivanov1 ~]#
```

Рис. 3.1: Запись в файл

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

```
[root@svivanov1 ~]# head file.txt  
/etc:  
итого 1364  
drwxr-xr-x. 1 root root    126 ноя  1 04:07 abrt  
-rw-r--r--. 1 root root     16 фев 15 17:42 adjtime  
-rw-r--r--. 1 root root  1529 июл 25 2023 aliases  
drwxr-xr-x. 1 root root     70 янв 29 03:00 alsa  
drwxr-xr-x. 1 root root  1400 фев 24 23:44 alternatives  
drwxr-xr-x. 1 root root     56 ноя  1 04:07 anaconda  
-rw-r--r--. 1 root root    541 июл 19 2023 anacrontab  
-rw-r--r--. 1 root root    833 фев 10 2023 appstream.conf
```

Рис. 3.2: Вывод содержимого файла

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

```
[root@svivanov1 ~]# ls -lR ~/ >> file.txt
[root@svivanov1 ~]#
```

Рис. 3.3: Добавление данных в файл

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

```
[root@svivanov1 ~]# grep .conf file.txt
-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 1372 дек 5 03:00 chrony.conf
drwxr-xr-x. 1 root root 18 ноя 1 04:05 dconf
-rw-r--r--. 1 root root 28601 янв 12 03:00 dnsmasq.conf
-rw-r--r--. 1 root root 117 ноя 16 03:00 dracut.conf
drwxr-xr-x. 1 root root 0 ноя 16 03:00 dracut.conf.d
-rw-r--r--. 1 root root 20 фев 24 2022 fprintd.conf
-rw-r--r--. 1 root root 38 авг 9 2023 fuse.conf
```

Рис. 3.4: Поиск файлов определенного расширения

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

```
[root@svivanov1 ~]# grep .conf file.txt > conf.txt
[root@svivanov1 ~]# head conf.txt
-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 1372 дек 5 03:00 chrony.conf
drwxr-xr-x. 1 root root 18 ноя 1 04:05 dconf
-rw-r--r--. 1 root root 28601 янв 12 03:00 dnsmasq.conf
-rw-r--r--. 1 root root 117 ноя 16 03:00 dracut.conf
drwxr-xr-x. 1 root root 0 ноя 16 03:00 dracut.conf.d
-rw-r--r--. 1 root root 20 фев 24 2022 fprintd.conf
-rw-r--r--. 1 root root 38 авг 9 2023 fuse.conf
-rw-r--r--. 1 root root 9 июл 25 2023 host.conf
```

Рис. 3.5: 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 будет относительно корневого каталога, а не домашнего), 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. Но таким образом я получаю информацию даже о файлах из подкаталогов домашнего каталога. (рис. 6)

```
[root@svivanov1 ~]# find ~ -name "c*" -print
/root/.config/gh/config.yml
/root/.gnupg/common.conf
/root/work/study/2023-2024/Операционные системы/os-intro/.git/hooks/commit-msg.sample
/root/work/study/2023-2024/Операционные системы/os-intro/.git/objects/d8/c0956b6b51002643
/root/work/study/2023-2024/Операционные системы/os-intro/.git/objects/9a/cdb3daebd97a5505
/root/work/study/2023-2024/Операционные системы/os-intro/.git/objects/c7
/root/work/study/2023-2024/Операционные системы/os-intro/.git/config
/root/work/study/2023-2024/Операционные системы/os-intro/.git/modules/template/presentati
```

Рис. 3.6: 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).

```
[root@svivanov1 ~]# ls -lR | grep c*
-rw-r--r--. 1 root root 42996 map 16 16:09 conf.txt
```

Рис. 3.7: Поиск файлов, начинающихся с определенного элемента

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

```
[root@svivanov1 ~]# find /etc -name "h*" -print
/etc/avahi/hosts
/etc/firewalld/helpers
/etc/libibverbs.d/hfi1verbs.driver
/etc/libibverbs.d/hns.driver
/etc/systemd/homed.conf
/etc/udev/hwdb.d
/etc/udev/hwdb.bin
/etc/host.conf
/etc/hosts
/etc/hostname
/etc/mercurial/hgrc.d
[root@svivanov1 ~]#
```

Рис. 3.8: 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).

```
[root@svivanov1 ~]# find ~ -name "log*" -print > logfile &
[1] 2152
[root@svivanov1 ~]#
```

Рис. 3.9: Создание фонового процесса

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

```
[root@svivanov1 ~]# ls
anaconda-ks.cfg  conf.txt  file.txt  logfile  work
[1]+  Завершён      find ~ -name "log*" -print > logfile
[root@svivanov1 ~]# rm logfile
rm: удалить обычный файл 'logfile'? y
[root@svivanov1 ~]# ls
anaconda-ks.cfg  conf.txt  file.txt  work
[root@svivanov1 ~]#
```

Рис. 3.10: Удаление файла

Запускаю в консоли в фоновом режиме (с помощью символа &) редактор gedit (рис. 11).

```
[root@svivanov1 ~]# gedit &
[1] 2164
```

Рис. 3.11: Создание фонового процесса

С помощью утилиты ps определяю идентификатор процесса gedit, его значение 3338. Также мы можем определить идентификатор с помощью pgrep. (рис. 12)

```
[svivanov1@svivanov1 ~]$ ps aux | grep gedit
svivano+  3338  21.4  1.5 918328 63152 pts/0    S1   20:20   0:01 gedit
svivano+  3356   0.0  0.0 222456  2432 pts/0    S+   20:20   0:00 grep --color=auto gedit
[svivanov1@svivanov1 ~]$ ps aux | grep gedit | grep -v grep
svivano+  3338  3.3  1.5 918328 63152 pts/0    S1   20:20   0:01 gedit
[svivanov1@svivanov1 ~]$
```

Рис. 3.12: Поиск идентификатора процесса

Прочитал справку команды kill (рис. 13).

```
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
```

Рис. 3.13: Чтение документации

Использую команду kill и идентификатор процесса, чтобы его удалить. Замечаем, что закрылась программа gedit. (рис. 14)

```
[svivanov1@svivanov1 ~]$ man kill
[svivanov1@svivanov1 ~]$ kill 3338
[svivanov1@svivanov1 ~]$
```

Рис. 3.14: Удаление процесса

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

```
[1]+  Завершено      gedit
[svivanov1@svivanov1 ~]$ man df
[svivanov1@svivanov1 ~]$ man du
[svivanov1@svivanov1 ~]$
```

Рис. 3.15: Чтение документации

Использую 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. (рис. 16)

```
[svivanov1@svivanov1 ~]$ df -vi
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
/dev/sda3          0          0          0          -          /
devtmpfs           493273      529    492744          1% /dev
tmpfs              498519        8    498511          1% /dev/shm
efivarfs           0           0           0          - /sys/firmware/efi/efivars
tmpfs              819200      879    818321          1% /run
tmpfs              1048576      32   1048544          1% /tmp
/dev/sda3          0           0           0          - /home
/dev/sda2          65536       38    65498          1% /boot
/dev/sda1          0           0           0          - /boot/efi
work               1000      -999000   1000000          - /media/sf_work
tmpfs              99703       106    99597          1% /run/user/1000
[svivanov1@svivanov1 ~]$
```

Рис. 3.16: Утилита df

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

```
[svivanov1@svivanov1 ~]$ du -a work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/labs/lab05/pre
sentation/
28 work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/labs/lab05/presentation/image/1.png
12 work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/labs/lab05/presentation/image/10.png
20 work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/labs/lab05/presentation/image/11.png
12 work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/labs/lab05/presentation/image/12.png
28 work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/labs/lab05/presentation/image/13.png
40 work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/labs/lab05/presentation/image/14.png
12 work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/labs/lab05/presentation/image/15.png
60 work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/labs/lab05/presentation/image/16.png
12 work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/labs/lab05/presentation/image/17.png
16 work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/labs/lab05/presentation/image/18.png
28 work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/labs/lab05/presentation/image/19.png
```

Рис. 3.17: Утилита du

Прочитал документацию о команде find (рис. 18).

```
FIND(1)                                     General Commands Manual                                     FIND(1)

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
    each given starting-point by evaluating the given expression from left to right, according to the rules
    of precedence (see section OPERATORS), until the outcome is known (the left hand side is false for and
    operations, true for or), at which point find moves on to the next file name. If no starting-point is
    specified, . is assumed.
```

Рис. 3.18: Чтение документации

Вывожу 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. Утилита -a позволит увидеть размер всех файлов, а не только директорий. (рис. 19)

```
./work3/SergeyIvanov21.github.io/tag/开源/page  
./work3/SergeyIvanov21.github.io/tag/开源/page/1  
./work3/SergeyIvanov21.github.io/tags  
./work3/SergeyIvanov21.github.io/tags/page  
./work3/SergeyIvanov21.github.io/tags/page/1  
./work3/SergeyIvanov21.github.io/talk  
./work3/SergeyIvanov21.github.io/talk/example-talk  
./work3/SergeyIvanov21.github.io/uploads  
./work3/SergeyIvanov21.github.io/webfonts  
[svivanov1@svivanov1 ~]$ find -type d
```

Рис. 3.19: Вывод директорий

4 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.

5 Выводы

В результате выполнения 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.