

Отчет по лабораторной работе №8

Операционные системы

Чувакина Мария Владимировна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	9
5	Выводы	15
6	Ответы на контрольные вопросы	16

Список иллюстраций

4.1	Запись в файл	9
4.2	Вывод содержимого файла	9
4.3	Добавление данных в файл	9
4.4	Поиск файлов определенного расширения	10
4.5	Запись в файл	10
4.6	Поиск файлов, начинающихся с определенного элемента	10
4.7	Поиск файлов, начинающихся с определенного элемента	11
4.8	Поиск файлов, начинающихся с определенного элемента	11
4.9	Создание фонового процесса	11
4.10	Удаление файла	11
4.11	Создание фонового процесса	12
4.12	Поиск идентификатора процесса	12
4.13	Чтение документации	12
4.14	Удаление процесса	12
4.15	Чтение документации	12
4.16	Утилита df	13
4.17	Утилита du	13
4.18	Чтение документации	13
4.19	Вывод всех директорий	14

Список таблиц

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. 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 директо-

рий, имеющихся 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). Тоже 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 Макилроем 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).

```
mvchuvakina@dk2n22 ~ $ ls -lR /etc > file.txt
```

Рис. 4.1: Запись в файл

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

```
mvchuvakina@dk2n22 ~ $ head file.txt
/etc:
итого 3256
drwxr-xr-x  2 root  root    4096 дек 25 00:12 a2ps
drwxr-xr-x  4 root  root    4096 ноя 20  2022 acpi
-rw-r--r--  1 root  root      47 авг 28  2019 adjtime
-rw-----  1 root  root    494 мая 17  2007 afs.keytab
drwxr-xr-x  3 root  root    4096 июн  4  2019 alsa
drwxr-xr-x  4 root  root    4096 авг 28  2023 apache2
drwxr-xr-x  2 root  root    4096 авг 29  2023 apache2.d
drwxr-xr-x  2 root  root    4096 ноя 26 22:07 apparmor.d
mvchuvakina@dk2n22 ~ $
```

Рис. 4.2: Вывод содержимого файла

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

```
mvchuvakina@dk2n22 ~ $ ls -lR ~/ >> file.txt
```

Рис. 4.3: Добавление данных в файл

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

```
-rw-r--r-- 1 root root 58929 фев 27 2006 %gconf.xml
/etc/gconf/gconf.xml.defaults/schemas/apps/galeon/UI/Tabs:
-rw-r--r-- 1 root root 74671 фев 27 2006 %gconf.xml
/etc/gconf/gconf.xml.defaults/schemas/apps/galeon/UI/ToolBar:
-rw-r--r-- 1 root root 16565 фев 27 2006 %gconf.xml
/etc/gconf/gconf.xml.defaults/schemas/apps/galeon/UI/Windows:
-rw-r--r-- 1 root root 65773 фев 27 2006 %gconf.xml
```

Рис. 4.4: Поиск файлов определенного расширения

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

```
mvchuvakina@dk2n22 ~$ grep .conf file.txt > conf.txt
mvchuvakina@dk2n22 ~$ head conf.txt
-rw-r--r-- 1 root root 833 ноя 10 23:28 appstream.conf
-rw-r--r-- 1 root root 30597 ноя 30 15:59 brltty.conf
-rw-r--r-- 1 root root 5961 фев 3 19:08 ca-certificates.conf
-rw-r--r-- 1 root root 756 июл 19 2022 cachefilesd.conf
-rw-r--r-- 1 root root 10883 июл 19 2022 cfg-update.conf
-rw-r--r-- 1 root root 254 фев 8 12:45 chrome-flags.conf
lrwxrwxrwx 1 root root 18 янв 8 19:03 chrony.conf -> chrony/chrony.conf
drwxr-xr-x 2 root root 4096 янв 31 18:06 conf.d
drwxr-xr-x 4 root root 4096 июн 4 2019 dconf
-rw-r--r-- 1 root root 1429 фев 3 19:17 dhcpd.conf
mvchuvakina@dk2n22 ~$
```

Рис. 4.5: 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 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.

```
mvchuvakina@dk2n22 ~$ find ~ -name "с*" -print
/afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/.local/share/evolution/addressbook/system/contacts.db
/afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/.local/share/evolution/calendar
/afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/.local/share/evolution/calendar/system/calendar.ics
/afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/.local/share/evolution/mail/local/cur
/afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/.local/share/evolution/mail/local/.Drafts/cur
/afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/.local/share/evolution/mail/local/.Outbox/cur
/afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/.local/share/evolution/mail/local/.Sent/cur
/afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/.local/share/evolution/mail/local/.Templates/cur
/afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/.local/share/evolution/categories.xml
```

Рис. 4.6: 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).

```
mvchuvakina@dk2n22 ~ $ ls -lR | grep c*
-rw-r--r-- 1 mvchuvakina studsci 141061 map 21 16:42 conf.txt
mvchuvakina@dk2n22 ~ $
```

Рис. 4.7: Поиск файлов, начинающихся с определенного элемента

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

```
mvchuvakina@dk2n22 ~ $ find /etc -name "h*" -print
/etc/gconf/gconf.xml.defaults/schemas/system/http_proxy
/etc/gconf/gconf.xml.defaults/schemas/desktop/gnome/applications/help_viewer
/etc/gconf/gconf.xml.defaults/schemas/desktop/gnome/url-handlers/http
/etc/gconf/gconf.xml.defaults/schemas/desktop/gnome/url-handlers/h323
/etc/gconf/gconf.xml.defaults/schemas/desktop/gnome/url-handlers/https
```

Рис. 4.8: 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).

```
mvchuvakina@dk2n22 ~ $ find ~ -name "log*" -print > logfile &
[1] 3849
mvchuvakina@dk2n22 ~ $
```

Рис. 4.9: Создание фонового процесса

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

```
mvchuvakina@dk2n22 ~ $ ls
bin      GNUstep  tmp      Документы  Музыка      'Снимки экрана'
conf.txt logfile  work     Загрузки   Общедоступные  Шаблоны
file.txt public   Видео    Изображения 'Рабочий стол'
[1]+  Завершён      find ~ -name "log*" -print > logfile
mvchuvakina@dk2n22 ~ $
```

Рис. 4.10: Удаление файла

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

```
mvchuvakina@dk2n22 ~ $ gedit &
[1] 3919
mvchuvakina@dk2n22 ~ $
```

Рис. 4.11: Создание фонового процесса

С помощью утилиты `ps` определяю идентификатор процесса `gedit`, его значение 3913 (рис.12). Также мы можем определить идентификатор с помощью `pgrep`.

```
mvchuvakina@dk2n22 ~ $ ps aux | grep gedit
mvchuvakina+ 3972 0.0 0.0 6800 2372 pts/3 S+ 16:48 0:00 grep --colour=auto gedit
[1]+ Завершён gedit
mvchuvakina@dk2n22 ~ $ pgrep gedit
mvchuvakina@dk2n22 ~ $ ps aux | grep gedit | grep -v grep
mvchuvakina@dk2n22 ~ $
```

Рис. 4.12: Поиск идентификатора процесса

Прочитала справку команды `kill` (рис.13).

```
KILL(1) User Commands KILL(1)
NAME
    kill - send a signal to a process

SYNOPSIS
    kill [options] <pid> [...]

DESCRIPTION
    The default signal for kill is TERM. Use -l or -L to list available signals. Particularly
    useful signals include HUP, INT, KILL, STOP, CONT, and 0. Alternate signals may be specified
    in three ways: -9, -SIGKILL or -KILL. Negative PID values may be used to choose whole process
    groups; see the PGID column in ps command output. A PID of -1 is special; it indicates all
    processes except the kill process itself and init.
```

Рис. 4.13: Чтение документации

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

```
mvchuvakina@dk2n22 ~ $ man kill
mvchuvakina@dk2n22 ~ $ kill 3913
mvchuvakina@dk2n22 ~ $
```

Рис. 4.14: Удаление процесса

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

```
mvchuvakina@dk2n22 ~ $ man df
mvchuvakina@dk2n22 ~ $ man du
mvchuvakina@dk2n22 ~ $
```

Рис. 4.15: 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.

```
mvchuvakina@dk2n22 ~$ df -v1
```

Файловая система	Инодов	Использовано	ISвободно	ИИспользовано%	Смонтировано в
/dev/sda8	33447936	1779363	31668573	6%	/
devtmpfs	986741	1628	985113	1%	/dev
tmpfs	999906	1	999905	1%	/dev/shm
tmpfs	819200	1901	817299	1%	/run
tmpfs	1048576	575	1048001	1%	/tmp
AFS	2147483647	0	2147483647	0%	/afs
tmpfs	199981	182	199799	1%	/run/user/5212

Рис. 4.16: Утилита df

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

```
mvchuvakina@dk2n22 ~$ du -a work/study/2023-2024/"Операционные системы"/os-intro/labs/lab08/presentation/
```

47	work/study/2023-2024/Операционные системы/os-intro/labs/lab08/presentation/image/kulyabov.jpg
49	work/study/2023-2024/Операционные системы/os-intro/labs/lab08/presentation/image
1	work/study/2023-2024/Операционные системы/os-intro/labs/lab08/presentation/Makefile
8	work/study/2023-2024/Операционные системы/os-intro/labs/lab08/presentation/presentation.md
60	work/study/2023-2024/Операционные системы/os-intro/labs/lab08/presentation/

Рис. 4.17: Утилита du

Прочитала 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	<p>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.</p> <p>If you are using find in an environment where security is important (for example if you are using it to search directories that are writable by other users), you should read the 'Security Considerations' chapter of the findutils documentation, which is called Finding Files and comes with findutils. That document also includes a lot more detail and discussion than this manual page, so you may find it a more useful source of information.</p>	

Рис. 4.18: 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 директорий.

```
./work/mvchuvakina.github.io/.git/logs/refs/heads
./work/mvchuvakina.github.io/.git/logs/refs/remotes
./work/mvchuvakina.github.io/.git/logs/refs/remotes/origin
./GNUstep
./GNUstep/Library
./GNUstep/Library/Services
./git
./git/info
./git/hooks
./git/refs
./git/refs/heads
./git/refs/tags
./git/objects
./git/objects/pack
./git/objects/info
./texlive2023
./texlive2023/texmf-var
./texlive2023/texmf-var/luatex-cache
./texlive2023/texmf-var/luatex-cache/generic
./texlive2023/texmf-var/luatex-cache/generic/names
./texlive2023/texmf-var/luatex-cache/generic/fonts
./texlive2023/texmf-var/luatex-cache/generic/fonts/otl
./fltk
./fltk/fltk.org
./fltk/fltk.org/fluid
./Снимки экрана
mvchuvakina@dk2n22 ~ $ find -type d
```

Рис. 4.19: Вывод всех директорий

5 Выводы

В результате данной лабораторной работы 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. 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. Этот знак `>` - перенаправление ввода/вывода, а `>>` - перенаправление в режиме добавления.
3. Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся следующей.
4. 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. 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. Запущенные фоном программы называются задачами (`jobs`). Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач.
7. Команда `htop` похожа на команду `top` по выполняемой функции: они обе показывают информацию о процессах в реальном времени, выводят дан-

ные 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. 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. find / -type f -exec grep -H 'текстДляПоиска' {} ; 10. С помощью команды df -h. 11. С помощью команды du -s. 12. С помощью команды kill% номер задачи.