

Отчет по выполнению лабораторной работы №8

Операционные системы

Монхжаргал Тувшинбаяр

Содержание

Список иллюстраций

• Запись в файл	8
• Вывод содержимого файла	8
• Добавление данных в файл	9
• Поиск файлов определенного расширения	9
• Запись в файл	9
• Поиск файлов, начинающихся с определенного элемента	10
• Поиск файлов, начинающихся с определенного элемента	10
• Поиск файлов, начинающихся с определенного элемента	10
• Создание фонового процесса.....	11
• Удаление файла	11
• Создание фонового процесса.....	11
• Поиск идентификатора процесса	11
• Чтение документации.....	12
• Удаление процесса	12
• Чтение документации.....	12
• Утилита df.....	13
• Утилита du	13
• Чтение документации.....	13
• Название рисунка.....	14

Список таблиц

- 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.

- Задание

- Осуществите вход в систему, используя соответствующее имя пользователя.
- Запишите в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.
- Выведите имена всех файлов из file.txt, имеющих расширение .conf, после чего запишите их в новый текстовый файл conf.txt.
- Определите, какие файлы в вашем домашнем каталоге имеют имена, начинающиеся с символа с? Предложите несколько вариантов, как это сделать.
- Выведите на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h.

- Запустите в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log.
- Удалите файл ~/logfile.
- Запустите из консоли в фоновом режиме редактор gedit.
- Определите идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Как ещё можно определить идентификатор процесса?
- Прочтите справку (man) команды kill, после чего используйте её для завершения процесса gedit.
- Выполните команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man.
- Воспользовавшись справкой команды find, выведите имена всех директо- рий, имею- щихся в вашем домашнем каталоге.

• Теоретическое введение

В интерфейсе командной строки есть очень полезная возможность перена- правления (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 выво- да (например, распечатать на принтере). Тоже самое и со вводом информации, вместо ввода данных с клавиатуры, для многих программ можно задать счи- тывание символов их файла. Кроме того, можно даже вывод одной программы передать на ввод другой программе.


К каждой программе, запускаемой в командной строке, по умолчанию под- ключено 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 '|’.

• 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. 4.1).



```
liveuser@tmunkhjargal:~$ sudo 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. 4.2).

```
liveuser@tmunkhjargal:~$ head file.txt
/etc:
total 1908
drwxr-xr-x. 3 root root 4096 Apr 14 18:59 abrt
-rw-r--r--. 1 root root 18 Apr 14 19:03 adjtime
-rw-r--r--. 1 root root 1529 Nov 29 2023 aliases
drwxr-xr-x. 3 root root 4096 Apr 14 19:00 alsa
drwxr-xr-x. 2 root root 4096 Apr 14 19:01 alternatives
drwxr-xr-x. 4 root root 4096 Apr 14 18:59 anaconda
-rw-r--r--. 1 root root 269 Mar 13 20:00 anthy-unicode.conf
-rw-r--r--. 1 root root 55 Jan 28 2024 asound.conf
```

Рис. 4.2: Вывод содержимого файла

Добавила в созданный файл имена файлов из домашнего каталога, используя перенаправление “>” в режиме добавления (рис. 4.3).

```
liveuser@tmunkhjargal:~$ ls -lR ~/ >> file.txt
liveuser@tmunkhjargal:~$
```

Рис. 4.3: Добавление данных в файл

Вывела на экран имена всех файлов, имеющих расширение “.conf” с помощью утилиты grep (рис. 4.4).

```
liveuser@tmunkhjargal:~$ cat file.txt
/etc:
total 1908
drwxr-xr-x. 3 root root 4096 Apr 14 18:59 abrt
-rw-r--r--. 1 root root 18 Apr 14 19:03 adjtime
-rw-r--r--. 1 root root 1529 Nov 29 2023 aliases
drwxr-xr-x. 3 root root 4096 Apr 14 19:00 alsa
drwxr-xr-x. 2 root root 4096 Apr 14 19:01 alternatives
drwxr-xr-x. 4 root root 4096 Apr 14 18:59 anaconda
-rw-r--r--. 1 root root 269 Mar 13 20:00 anthy-unicode.conf
-rw-r--r--. 1 root root 55 Jan 28 2024 asound.conf
```

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. 4.5).

```
liveuser@tmunkhjargal:~$ grep .conf file.txt
-rw-r--r--. 1 root root 269 Mar 13 20:00 anthy-unicode.conf
-rw-r--r--. 1 root root 55 Jan 28 2024 asound.conf
-rw-r--r--. 1 root root 30583 Jan 30 2024 brltty.conf
drwxr-xr-x. 2 root root 4096 Jan 22 2024 chkconfig.d
-rw-r--r--. 1 root root 1365 Sep 4 14:37 chrony.conf
drwxr-xr-x. 4 root root 4096 Apr 14 18:57 dconf
-rw-r--r--. 1 root root 1174 Jan 23 2024 dleyna-server-service.conf
```

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. 4.6). Но таким образом я получаю информацию даже о файлах из подкаталогов домашнего каталога.

```
liveuser@tmunkhjargal:~$ find ~ -name "c*" -print
/home/liveuser/.password-store/.git/hooks/commit-msg.sample
/home/liveuser/.password-store/.git/config
/home/liveuser/.gnupg/common.conf
/home/liveuser/.config/sway/other/blocks/calendar
/home/liveuser/.config/sway/other/blocks/current_play.sh
/home/liveuser/.config/sway/other/waybar/config
/home/liveuser/.config/sway/other/tessen/config
/home/liveuser/.config/sway/other/wofi/config
/home/liveuser/.config/sway/config
/home/liveuser/.config/sway/config.d
```

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. 4.7).

```
liveuser@tmunkhjargal:~$ ls -ls -lR | grep c*
48 -rw-r--r--. 1 liveuser liveuser 49034 Sep  5 11:28 conf.txt
```

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. 4.8).

```
liveuser@tmunkhjargal:~$ sudo find /etc -name "h*" -print
/etc/hostname
/etc/containers/oci/hooks.d
/etc/sysconfig/htcacheclean
/etc/systemd/system/httpd.service.d
/etc/host.conf
/etc/avahi/hosts
/etc/logrotate.d/httpd
/etc/udev/hwdb.d
/etc/udev/hwdb.bin
/etc/firewalld/helpers
/etc/sane.d/hpsj5s.conf
/etc/sane.d/hp5400.conf
/etc/sane.d/hs2p.conf
```

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. 4.9).

```
liveuser@tmunkhjargal:~$ find ~ -name "log*" -print > logfile &
[1] 13507
```

Рис. 4.9: Создание фонового процесса

Проверяю, что файл создан, удаляю его, проверяю, что файл удален (рис. 4.10).

```
liveuser@tmunkhjargal:~$ ls
abc1 Desktop fff logfile Music reports work
bin Documents file.txt may Pictures Templates
conf.txt Downloads LICENSE monthly Public Videos
[1]+ Done find ~ -name "log*" -print > logfile
liveuser@tmunkhjargal:~$ rm logfile
liveuser@tmunkhjargal:~$ ls
abc1 Desktop fff may Pictures Templates
bin Documents file.txt monthly Public Videos
conf.txt Downloads LICENSE Music reports work
```

Рис. 4.10: 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. 4.11).

```
liveuser@tmunkhjargal:~$ mousepad &
[1] 13619
```

Рис. 4.11: Создание фонового процесса

С помощью утилиты `ps` определяю идентификатор процесса `mousepad`, его значение 13643 (рис. 4.12). Также мы можем определить идентификатор с помощью `pgrep`.

```
liveuser@tmunkhjargal:~$ ps aux | grep mousepad
liveuser  13643  0.4  1.2 834960 51072 pts/0    Sl   11:45   0:00 mousepad
liveuser  13656  0.0  0.0 227752  2176 pts/0    S+   11:46   0:00 grep --color=
auto mousepad
liveuser@tmunkhjargal:~$ pgrep mousepad
13643
liveuser@tmunkhjargal:~$ ps aux | grep mousepad | grep -v grep
liveuser  13643  0.2  1.2 834960 51072 pts/0    Sl   11:45   0:00 mousepad
```

Рис. 4.12: Поиск идентификатора процесса

Прочитала справку команды `kill` (рис. 4.13).

```
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 may
    install a handler for the TERM signal in order to perform clean-up
    steps before terminating in an orderly fashion. If a process does not
    terminate after a TERM signal has been sent, then the KILL signal may
    be used: be aware that the latter signal cannot be caught, and so does
```

Рис. 4.13: Чтение документации

Использую команду `kill` и идентификатор процесса, чтобы его удалить (рис. 4.14). Заметила, как у меня закрылась программа `mousepad`.

```
liveuser@tmunkhjargal:~$ man kill
liveuser@tmunkhjargal:~$ kill 13643
liveuser@tmunkhjargal:~$
```

Рис. 4.14: Удаление процесса

Прочитала документацию про функции `df` и `du` (рис. 4.15).

```
liveuser@tmunkhjargal:~$ man df
[2]+  Done                  mousepad
liveuser@tmunkhjargal:~$ man du
liveuser@tmunkhjargal:~$
```

Рис. 4.15: Чтение документации

Используя утилиту `df` опции `-iv` позволяют увидеть информацию об инодах и сделать вывод читаемым, игнорируя сообщения системы о нем (рис. 4.16). Эта утилита нам нужна, чтобы выяснить, сколько свободного места есть у нашей системы.

```
liveuser@tmunkhjargal:~$ df -vi
Filesystem      Inodes    IUsed   IFree IUse% Mounted on
/dev/mapper/live-rw 525312 161608 363704   31% /
devtmpfs         482490     550  481940    1% /dev
tmpfs            500664        5  500659    1% /dev/shm
tmpfs            819200    1070  818130    1% /run
/dev/sr0          0          0        0    - /run/initramfs/live
tmpfs            1048576     63 1048513    1% /tmp
vartmp           500664     32  500632    1% /var/tmp
tmpfs            100132     195   99937    1% /run/user/1000
```

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. 4.17).

```
liveuser@tmunkhjargal:~$ du -a work/study/2023-2024/'Operating systems'/os-intro/
/labs/lab03/presentation/
8    work/study/2023-2024/Operating systems/os-intro/labs/lab03/presentation/
presentation.md
48   work/study/2023-2024/Operating systems/os-intro/labs/lab03/presentation/
image/kulyabov.jpg
52   work/study/2023-2024/Operating systems/os-intro/labs/lab03/presentation/
image
0    work/study/2023-2024/Operating systems/os-intro/labs/lab03/presentation/
.texlabroot
0    work/study/2023-2024/Operating systems/os-intro/labs/lab03/presentation/
.projectile
4    work/study/2023-2024/Operating systems/os-intro/labs/lab03/presentation/
```

Рис. 4.17: Утилита `du`

Прочитала документацию о команде find (рис. 4.18).

```
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...] [ex-
    pression]

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.

    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 de-
```

Рис. 4.18: Чтение документации

Вывела имена всех директорий, имеющихя в моем домашнем каталоге, ис- пользуя аргумент **d** у утилиты **find** опции **-type**, то есть указываю тип файлов, который мне нужен и этот тип Директория (рис. 4.19). Утилита **-a** позволит уви- деть размер всех файлов, а не только диреткорий.

```
liveuser@tmunkhjargal:~$ find -type d
.
./Desktop
./Downloads
./.password-store
./.password-store/BrowserPass
./.password-store/.git
./.password-store/.git/logs
./.password-store/.git/logs/refs
./.password-store/.git/logs/refs/heads
./.password-store/.git/hooks
./.password-store/.git/branches
./.password-store/.git/info
./.password-store/.git/refs
./password-store/.git/refs/heads
```

Рис. 4.19: Название рисунка

• 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.

• Ответы на контрольные вопросы

- 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.
- Объясните разницу между операцией `>` и `>>`. Этот знак `>` - перенаправление ввода/вывода, а `>>` - перенаправление в режиме добавления.
- Что такое конвейер? Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.
- 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.

- 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.
- Что такое задачи и какая команда позволяет ими управлять? Запущенные фонов программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач.
- 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.

- 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`

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