

Отчет по лабораторной работе №6

дисциплина: Операционные системы

Егорова Юлия Владимировна

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Контрольные вопросы	12
4	Вывод	17

Список иллюстраций

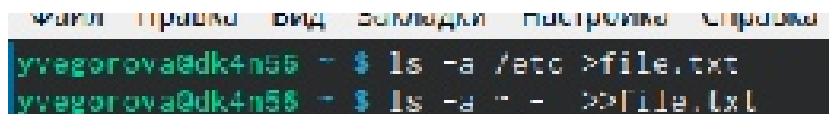
2.1	Запись в file.txt названия файлов из каталога /etc и домашнего каталога.	5
2.2	Вывод файлов.	6
2.3	Запись в новый текстовый файл.	7
2.4	Поиск файлов, начинающиеся с символа “с”.	7
2.5	Поиск файлов, начинающиеся с символа “h”	8
2.6	Запуск процесса записи в файл файлов, начинающихся с log. . . .	8
2.7	Удаление файла.	8
2.8	Запуск в фоновом режиме редактора gedit.	8
2.9	Определение идентификатора процесса gedit.	9
2.10	Справка команды kill.	9
2.11	Завершение процесса gedit.	9
2.12	Информация о команде df.	10
2.13	Информация о команде du.	10
2.14	Команда df.	10
2.15	Команда du.	10
2.16	Команда du.	11
2.17	Вывод всех директорий, имеющихся в домашнем каталоге.	11

1 Цель 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 Выполнение лабораторной работы

1)Записываем в файл file.txt названия файлов, содержащихся в каталоге /etc. Дописываем в этот же файл названия файлов, содержащихся в домашнем каталоге.

A screenshot of a terminal window showing two commands being executed. The first command is 'ls -la /etc >file.txt' and the second is 'ls -la ~ - >>file.txt'. The terminal output is not visible, only the commands are shown. The terminal has a dark background with light-colored text.

```
yvegorova@dk4n55 ~$ ls -la /etc >file.txt
yvegorova@dk4n55 ~$ ls -la ~ - >>file.txt
```

Рис. 2.1: Запись в file.txt названия файлов из каталога /etc и домашнего каталога.

2)Выводим имена всех файлов из file.txt, имеющих расширение .conf, после записываем их в новый текстовый файл conf.txt.

```
yvegorova@dk4n56 ~ $ cat file.txt
.  
..  
a2ps  
acpi  
adjtime  
afs.keytab  
alsa  
apache2  
apparmor.d  
appstream.conf  
ati  
audisp  
audit  
autofs  
avahi  
bash  
bash_completion.d  
bindresvport.blacklist  
binfmt.d  
blkid.tab.old  
bluetooth  
brltty  
brltty.conf  
ca-certificates  
ca-certificates.conf  
cachefilesd.conf  
cfg-update.conf  
cfg-update.hosts  
cgroup  
chromium  
chrony  
cifs-utils  
common-lisp  
conf.d  
cpufreq-bench.conf  
cron.d  
cron.daily  
cron.hourly  
cron.monthly  
crontab  
cron.weekly  
csh.cshrc  
csh.env  
csh.login  
cups  
cupshelpers  
dbus-1  
dconf  
default  
dhcp
```

Рис. 2.2: Вывод файлов.

```
yvegorova@dk4n56 ~ $ cat conf.txt
appstream.conf
brlty.conf
ca-certificates.conf
cachefilesd.conf
cfg-update.conf
cpufreq-bench.conf
dhcpcd.conf
dispatch-conf.conf
dleyna-server-service.conf
dnsmasq.conf
e2fsck.conf
e2scrub.conf
etc-update.conf
fluidsynth.conf
fuse.conf
gai.conf
genkernel.conf
gssapi_mech.conf
host.conf
idmapd.conf
idn2.conf
idnalias.conf
krb5.conf
ldap.conf
ld.so.conf
libaudit.conf
lightdm.conf
locale.conf
```

Рис. 2.3: Запись в новый текстовый файл.

3) Определяем, какие файлы в домашнем каталоге имеют имена, начинающиеся с символа “с” с помощью команды `find`.

```
yvegorova@dk4n56 ~ $ find -name "c*" -print
./usr/share/evolution/addressbook/system/contacts.db
./usr/share/evolution/calendar/system/calendar.ics
./usr/share/evolution/mail/local/cur
./usr/share/evolution/mail/local/Drafts/cur
./usr/share/evolution/mail/local/Outbox/cur
./usr/share/evolution/mail/local/Sent/cur
./usr/share/evolution/mail/local/Templates/cur
./usr/share/evolution/Categories.xml
./usr/share/evolution/Cache.db
./usr/share/evolution/config/pulse/cookie
./usr/share/evolution/config/gtk-3.0/assets/close-normal.svg
./usr/share/evolution/config/gtk-3.0/assets/close-active.svg
./usr/share/evolution/config/gtk-3.0/assets/close-hover.svg
./usr/share/evolution/config/gtk-3.0/assets/close-backdrop-normal.svg
./usr/share/evolution/config/gtk-3.0/assets/close-backdrop-active.svg
./usr/share/evolution/config/gtk-3.0/assets/close-backdrop-hover.svg
./usr/share/evolution/config/gtk-3.0/colors.css
./usr/share/evolution/config/kdeconnect/certificate.pem
./usr/share/evolution/config/kdeconnect/config
./usr/share/evolution/config/libreoffice4/user/config
./usr/share/evolution/config/libreoffice4/user/extensions/shared/registry/com.sun.star.comp.deployment.component.PackageRegistryBackend
./usr/share/evolution/config/libreoffice4/user/extensions/shared/registry/com.sun.star.comp.deployment.configuration.PackageRegistryBackend
./usr/share/evolution/config/libreoffice4/user/extensions/shared/registry/com.sun.star.comp.deployment.executable.PackageRegistryBackend
./usr/share/evolution/config/libreoffice4/user/extensions/shared/registry/com.sun.star.comp.deployment.help.PackageRegistryBackend
./usr/share/evolution/config/libreoffice4/user/extensions/shared/registry/com.sun.star.comp.deployment.script.PackageRegistryBackend
./usr/share/evolution/config/libreoffice4/user/extensions/shared/registry/com.sun.star.comp.deployment.ui.PackageRegistryBackend
./usr/share/evolution/config/libreoffice4/user/extensions/bundled/registry/com.sun.star.comp.deployment.component.PackageRegistryBackend
./usr/share/evolution/config/libreoffice4/user/extensions/bundled/registry/com.sun.star.comp.deployment.configuration.PackageRegistryBackend
./usr/share/evolution/config/libreoffice4/user/extensions/bundled/registry/com.sun.star.comp.deployment.executable.PackageRegistryBackend
```

Рис. 2.4: Поиск файлов, начинающиеся с символа “с”.

4) Выводим на экран имена файлов, начинающиеся с символа `h` из каталога `/etc`.

```
yvegorova@dk4n56 ~ $ find /etc -name "h*" -print
find: '/etc/audisp/plugins.d': Отказано в доступе
/etc/conf.d/hdparm
/etc/conf.d/hsqldb
/etc/conf.d/hostname
/etc/conf.d/hddtemp
/etc/conf.d/hwclock
/etc/hotplug.d
/etc/hsqldb
/etc/hosts.allow
/etc/hostname
/etc/harbour.cfg
/etc/init.d/hdparm
/etc/init.d/hsqldb
/etc/init.d/hotplug
/etc/init.d/hddtemp
/etc/harbour
/etc/harbour/hb-charmap.def
/etc/avahi/hosts
/etc/distcc/hosts
find: '/etc/cron.daily': Отказано в доступе
find: '/etc/cron.monthly': Отказано в доступе
find: '/etc/lvm/cache': Отказано в доступе
/etc/sane.d/hp4200.conf
/etc/sane.d/hp3900.conf
/etc/sane.d/hp5400.conf
/etc/sane.d/hp.conf
/etc/sane.d/hs2p.conf
/etc/xdg/xfce4/helpers.rc
/etc/runlevels/default/hdparm
/etc/runlevels/boot/hostname
/etc/runlevels/boot/hwclock
find: '/etc/polkit-1/rules.d': Отказано в доступе
/etc/httpd
/etc/brltty/Input/hw
/etc/brltty/Input/hd
```

Рис. 2.5: Поиск файлов, начинающиеся с символа “h”

5)Запускаем в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log.

```
yvegorova@dk4n56 ~ $ find /var/log -name "log*" -print > logfile &
[1] 9942
```

Рис. 2.6: Запуск процесса записи в файл файлов, начинающихся с log.

6)Удаляем файл ~/logfile.

```
yvegorova@dk4n56 ~ $ rm logfile
```

Рис. 2.7: Удаление файла.

7)Запускаем из консоли в фоновом режиме редактор gedit.

```
yvegorova@dk4n56 ~ $ gedit &
[1] 10112
```

Рис. 2.8: Запуск в фоновом режиме редактора gedit.

8) Определяем идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`.

```
yvegorova@dk4n56 ~ $ ps aux | grep gedit
yvegorov+ 10170  0.0  0.0 10156  972 pts/0    S+   20:29   0:00 grep --colour=auto gedit
```

Рис. 2.9: Определение идентификатора процесса `gedit`.

9) Читаем справку команды `kill` с помощью команды `man`, после чего используем ее для завершения процесса `gedit`.

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

SYNOPSIS
    kill [options] qids [ ... ]

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: -s, -SIG#, or -#0#. Negative PID values may be used to check whole process groups; see the PIDs column in ps command output. A PID of -1 is special; it indicates all processes except the kill process itself and init.

OPTIONS
    -qids [ ... ]
        Send signal to every qid in listed.

    -s signal
    -signal
    -signal=signal
        Specify the signal to be sent. The signal can be specified by using name or number. The behavior of signals is explained in signal(7) manual page.

    -s, --queue value
        Use signalset(3) rather than kill(2) and the value argument is used to specify an integer to be sent with the signal. If the receiving process has installed a handler for that signal using the SA_SIGINFO flag to signalset(3), then it can obtain this data via the si_value field of the siginfo_t structure.

    -l, --list [table]
        List signal names. This option has optional argument, which will convert signal number to signal name, or other way round.

    -t, --table
        List signal names in a nice table.

NOTES
    Your shell (command line interpreter) may have a built-in kill command. You may need to run the command described here as /bin/kill to solve the conflict.

EXAMPLES
    kill -9 -1
        Kill all processes you can kill.

    kill -1 1
        Kill process 1 with signal 1.
```

Рис. 2.10: Справка команды `kill`.

```
yvegorova@dk4n56 ~ $ ps aux | grep gedit
[1]+  Завершён                  gedit
```

Рис. 2.11: Завершение процесса `gedit`.

10) Выполняем команды `df` и `du`, предварительно получаем более подробную информацию об этих командах с помощью команды `man`.

```

df(1)                                     User Commands                                df(1)

NAME
    df - report file system disk space usage

SYNOPSIS
    df [OPTION]... [FILE]...

DESCRIPTION
    This manual page documents the GNU version of df.  df displays the amount of disk space available on the file system containing each file name argument.  If no file
    name is given, the space available on all currently mounted file systems is shown.  Disk space is shown in 1K blocks by default, unless the environment variable
    POSIXLY_CORRECT is set, in which case 512 byte blocks are used.

    If an argument is the absolute file name of a disk device node containing a mounted file system, df shows the space available on that file system rather than on the
    file system containing the device node.  This version of df cannot show the space available on unmounted file systems, because it needs to know a system's data so it re-
    quires very considerable intimate knowledge of file system structures.

OPTIONS
    Show information about the file system on which each FILE resides, or all file systems by default.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        include pseudo, duplicate, inaccessible file systems

    -B, --block-size=UNIT
        scale sizes by UNIT before printing them; e.g., '10M' prints sizes in units of 1,048,576 bytes; see SIZE format below

    -h, --human-readable
        print sizes in powers of 1024 (e.g., 1024K)

    -H, --si
        print sizes in powers of 1000 (e.g., 1.0G)

    -l, --local
        list local information instead of block usage

    -k, --live --block-size=UNIT
        like --block-size=UNIT

Manual page df(1) lines 1 (press h for help or q to exit)

```

Рис. 2.12: Информация о команде df.

```

du(1)                                     User Commands                                du(1)

NAME
    du - estimate file space usage

SYNOPSIS
    du [OPTION]... [FILE]...
    du [OPTION]... filesystem

DESCRIPTION
    Summarize disk usage of the set of FILES, recursively for directories.
    Mandatory arguments to long options are mandatory for short options too.

    -d, --max-depth=NUMBER
        end each output line with NUMBER, not newline

    -a, --all
        write counts for all files, not just directories

    --apparent-size
        print apparent size, rather than disk usage; although the apparent size is usually smaller, it may be larger due to holes in ("sparse") files, internal frag-
        mentation, indirect blocks, and the like

    -B, --block-size=UNIT
        scale sizes by UNIT before printing them; e.g., '10M' prints sizes in units of 1,048,576 bytes; see SIZE format below

    -b, --bytes
        equivalent to "--apparent-size --block-size=1"

    -c, --total
        produce a grand total

    -D, --dereference-dereference
        dereference only symlinks that are listed on the command line

    -d, --max-depth=NUMBER
        print the total for a directory (or file, with --all) only if it is N or fewer levels below the command line argument; "--max-depth=0" is the same as "--summar-
        ize"

Manual page du(1) lines 1 (press h for help or q to exit)

```

Рис. 2.13: Информация о команде du.

```
yvegorova@dk4n56 ~$ df
```

Файловая система	1K-блоков	Использовано	Доступно	Использовано%	Смонтировано в
none	4000180	28460	3971720	1%	/run
udev	3890156	0	3890156	0%	/dev
tmpfs	4000180	326092	3674088	9%	/dev/shm
/dev/sda8	491812356	108701320	358105292	24%	/
/dev/sda6	91557952	851328	86032680	1%	/var/cache/openafs
tmpfs	4000180	175744	3824436	5%	/tmp
mark.sci.pfu.edu.ru:/com/lib/portage	1048320000	504344576	543975424	49%	/com/lib/portage
mark.sci.pfu.edu.ru:/usr/portage	18350080	5858816	11009792	35%	/usr/portage
mark.sci.pfu.edu.ru:/usr/local/share/portage	18350080	5858816	11009792	35%	/usr/local/share/portage
AFS	2147483647	0	2147483647	0%	/afs
tmpfs	800036	240	799796	1%	/run/user/4154
mark.sci.pfu.edu.ru:/usr/local/share/portage	18350080	5858816	11009792	35%	/usr/local/share/portage

Рис. 2.14: Команда df.

```

yvegorova@dk4n56 ~$ du

```

Рис. 2.15: Команда du.

```

2  ./course-directory-student-template/.git/modules/template/presentation/refs/tags
3  ./course-directory-student-template/.git/modules/template/presentation/refs/remotes/origin
5  ./course-directory-student-template/.git/modules/template/presentation/refs/remotes
12 ./course-directory-student-template/.git/modules/template/presentation/refs
37 ./course-directory-student-template/.git/modules/template/presentation/objects/pack
2  ./course-directory-student-template/.git/modules/template/presentation/objects/info
41 ./course-directory-student-template/.git/modules/template/presentation/objects
3  ./course-directory-student-template/.git/modules/template/presentation/logs/refs/remotes/origin
5  ./course-directory-student-template/.git/modules/template/presentation/logs/refs/remotes
3  ./course-directory-student-template/.git/modules/template/presentation/logs/refs/heads
10 ./course-directory-student-template/.git/modules/template/presentation/logs/refs
13 ./course-directory-student-template/.git/modules/template/presentation/logs
167 ./course-directory-student-template/.git/modules/template/presentation
3  ./course-directory-student-template/.git/modules/template/report/info
31 ./course-directory-student-template/.git/modules/template/report/hooks
3  ./course-directory-student-template/.git/modules/template/report/refs/heads
2  ./course-directory-student-template/.git/modules/template/report/refs/tags
3  ./course-directory-student-template/.git/modules/template/report/refs/remotes/origin
5  ./course-directory-student-template/.git/modules/template/report/refs/remotes
12 ./course-directory-student-template/.git/modules/template/report/refs
299 ./course-directory-student-template/.git/modules/template/report/objects/pack
2  ./course-directory-student-template/.git/modules/template/report/objects/info
303 ./course-directory-student-template/.git/modules/template/report/objects
3  ./course-directory-student-template/.git/modules/template/report/logs/refs/remotes/origin
5  ./course-directory-student-template/.git/modules/template/report/logs/refs/remotes
3  ./course-directory-student-template/.git/modules/template/report/logs/refs/heads
10 ./course-directory-student-template/.git/modules/template/report/logs/refs
13 ./course-directory-student-template/.git/modules/template/report/logs
370 ./course-directory-student-template/.git/modules/template/report

```

Рис. 2.16: Команда du.

11) Воспользовавшись справкой команды find, выводим имена всех директорий, имеющих в домашнем каталоге.

```
yvegorova@dk4n56 ~ $ find type d
```

Рис. 2.17: Вывод всех директорий, имеющих в домашнем каталоге.

3 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. Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода stdout.
- 2) Перенаправление вывода в файл > Перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла).
- 3) Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей. Синтаксис следующий: команда 1 | команда 2 (это означает, что вывод команды 1 передаётся на ввод команде 2)
- 4) Процесс рассматривается операционной системой как заявка на потребление всех видов ресурсов, кроме одного – процессорного времени. Этот последний важнейший ресурс распределяется операционной системой между другими единицами работы – потоками, которые и получили свое название благодаря тому, что они представляют собой последовательности (потоки выполнения) команд. Процесс – это выполнение программы. Он считается активной сущностью и реализует действия, указанные в программе. Про-

грамма представляет собой статический набор команд, а процесс это набор ресурсов и данных, использующихся при выполнении программы.

- 5) `pid`: идентификатор процесса (PID) процесса (process ID), к которому вызывают метод `gid`: идентификатор группы UNIX, в котором работает программа.
- 6) Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда `&`. Запущенные фоном программы называются задачами (`jobs`). Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач.
- 7) `top` – это консольная программа, которая показывает список работающих процессов в системе. Программа в реальном времени отсортирует запущенные процессы по их нагрузке на процессор. `htop` – это продвинутый консольный мониторинг процессов. Утилита выводит постоянно меняющийся список системных процессов, который сортируется в зависимости от нагрузки на ЦПУ. Если делать сравнение с `top`, то `htop` показывает абсолютно все процессы в системе, время их непрерывного использования, загрузку процессоров и расход оперативной памяти.
- 8) `find` – 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. Команда `find` имеет такой синтаксис: `find [папка] [параметры] критерий шаблон [действие]` Папка – каталог в котором будем искать Параметры – дополнительные параметры, например, глубина поиска, и т.д. Критерий – по какому критерию будем искать: имя, дата создания, права, владелец и т.д. Шаблон – непосредственно значение по которому будем отбирать файлы. Основные параметры:

- -P никогда не открывать символические ссылки
 - -L - получает информацию о файлах по символическим ссылкам. Важно для дальнейшей обработки, чтобы обрабатывалась не ссылка, а сам файл.
 - -maxdepth - максимальная глубина поиска по подкаталогам, для поиска только в текущем каталоге установите 1.
 - -depth - искать сначала в текущем каталоге, а потом в подкаталогах
 - -mount искать файлы только в этой файловой системе.
 - -version - показать версию утилиты find
 - -print - выводить полные имена файлов
 - -type f - искать только файлы
 - -type d - поиск папки в Linux
- Основные критерии:
- -name - поиск файлов по имени
 - -perm - поиск файлов в Linux по режиму доступа
 - -user - поиск файлов по владельцу
 - -group - поиск по группе
 - -mtime - поиск по времени модификации файла
 - -atime - поиск файлов по дате последнего чтения
 - -nogroup - поиск файлов, не принадлежащих ни одной группе
 - -nouser - поиск файлов без владельцев
 - -newer - найти файлы новее чем указанный
 - -size - поиск файлов в Linux по их размеру
- Примеры: `find ~ -type d` поиск директорий в домашнем каталоге
`find ~ -type f -name ".*"` поиск скрытых файлов в домашнем каталоге
- 9) Файл по его содержимому можно найти с помощью команды `grep`: «`grep -r "слово/выражение, которое нужно найти"`».
 - 10) Утилита `df`, позволяет проанализировать свободное пространство на всех подключенных к системе разделах.
 - 11) При выполнении команды `du` (без указания папки и опции) можно получить все файлы и папки текущей директории с их размерами. Для домашнего

каталога: `du ~/`

12) Основные сигналы (каждый сигнал имеет свой номер), которые используются для завершения процесса:

- **SIGINT** – самый безобидный сигнал завершения, означает Interrupt. Он отправляется процессу, запущенному из терминала с помощью сочетания клавиш Ctrl+C. Процесс правильно завершает все свои действия и возвращает управление;
 - **SIGQUIT** – это еще один сигнал, который отправляется с помощью сочетания клавиш, программе, запущенной в терминале. Он сообщает ей что нужно завершиться и программа может выполнить корректное завершение или проигнорировать сигнал. В отличие от предыдущего, она генерирует дампы памяти. Сочетание клавиш Ctrl+Q;
 - **SIGHUP** – сообщает процессу, что соединение с управляющим терминалом разорвано, отправляется, в основном, системой при разрыве соединения с интернетом;
 - **SIGTERM** – немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы;
 - **SIGKILL** – тоже немедленно завершает процесс, но, в отличие от предыдущего варианта, он не передается самому процессу, а обрабатывается ядром. Поэтому ресурсы и дочерние процессы остаются запущенными. Также для передачи сигналов процессам в Linux используется утилита `kill`, её синтаксис: `kill [-сигнал] [pid_процесса]` (PID – уникальный идентификатор процесса). Сигнал представляет собой один из выше перечисленных сигналов для завершения процесса.
- 2.14 Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого используют команды `ps` и `grep`. Команда `ps` предназначена для вывода списка активных процессов в системе и информации о них. Команда `grep` запускается одновременно с `ps` (в канале) и будет выполнять поиск по результатам команды `ps`. Утилита

`pskill` – это оболочка для `kill`, она ведет себя точно так же, и имеет тот же синтаксис, только в качестве идентификатора процесса ей нужно передать его имя. `killall` работает аналогично двум предыдущим утилитам. Она тоже принимает имя процесса в качестве параметра и ищет его PID в директории `/proc`. Но эта утилита обнаружит все процессы с таким именем и завершит их.

4 Вывод

В ходе 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.