

# **Отчет по лабораторной работе №6**

*дисциплина: Операционные системы*

Старков Никита Алексеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>5</b>
<b>3</b>	<b>Контрольные вопросы</b>	<b>16</b>
<b>4</b>	<b>Вывод</b>	<b>21</b>

## Список иллюстраций

2.1	Запись в file.txt названия файлов из каталога /etc и домашнего каталога . . . . .	5
2.2	Вывод файлов и их запись в новый текстовый файл . . . . .	6
2.3	Поиск файлов, начинающиеся с символа “с” . . . . .	7
2.4	Поиск файлов, начинающиеся с символа “h” . . . . .	8
2.5	Запускаем процесс записи в файл файлов, начинающихся с log . .	8
2.6	Удаление файла . . . . .	9
2.7	Запуск в фоновом режиме редактора gedit . . . . .	9
2.8	Определение идентификатора процесса gedit . . . . .	9
2.9	Справка команды kill . . . . .	10
2.10	Завершение процесса gedit . . . . .	10
2.11	Информация о команде df . . . . .	11
2.12	Информация о команде du . . . . .	12
2.13	Команда df в действии . . . . .	13
2.14	Команда du . . . . .	13
2.15	Команда du в действии . . . . .	14
2.16	Вывод всех директорий, имеющихся в домашнем каталоге . . . .	14
2.17	Вывод всех директорий, имеющихся в домашнем каталоге . . . .	15

# 1 Цель работы

**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

```
nastarkov@dk3n59 ~ $ ls /etc > file.txt  
nastarkov@dk3n59 ~ $ ls ~ > 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

```
nastarkov@dk3n59 ~ $ cat file.txt | grep .conf >> conf.txt
nastarkov@dk3n59 ~ $ cat conf.txt
appstream.conf
brltty.conf
ca-certificates.conf
cachefilesd.conf
cfg-update.conf
cpufreq-bench.conf
dconf
dhcpcd.conf
dispatch-conf.conf
dleyna-server-service.conf
dnsmasq.conf
e2fsck.conf
e2scrub.conf
etc-update.conf
fluidsynth.conf
fuse.conf
gai.conf
gconf
genkernel.conf
gssapi_mech.conf
host.conf
idmapd.conf
idn2.conf
idn2.conf.sample
idnalias.conf
idnalias.conf.sample
java-config-2
krb5.conf
krb5.conf.example
ldap.conf
ldap.conf.sudo
ld.so.conf
ld.so.conf.d
libaudit.conf
lightdm.conf
```

Рис. 2.2: Вывод файлов и их запись в новый текстовый файл

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

```
nastarkov@dk3n59 ~ $ find ~ -name "c*" -print
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/kdeconnect/certificate.pem
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/kdeconnect/config
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/gtk-3.0/assets/close-normal.svg
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/gtk-3.0/assets/close-active.svg
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/gtk-3.0/assets/close-hover.svg
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/gtk-3.0/assets/close-backdrop-normal.svg
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/gtk-3.0/assets/close-backdrop-active.svg
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/gtk-3.0/assets/close-backdrop-hover.svg
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/gtk-3.0/colors.css
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/pulse/cookie
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/config
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.component.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.configuration.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.executable.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.help.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.script.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.sfwk.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.bundle.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.component.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.configuration.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.configuration.PackageRegistryBackend/configmgr.ini
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.executable.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.help.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.script.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.sfwk.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.bundle.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/tmp/registry/com.sun.star.comp.deployment.component.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/tmp/registry/com.sun.star.comp.deployment.configuration.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/tmp/registry/com.sun.star.comp.deployment.executable.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/tmp/registry/com.sun.star.comp.deployment.help.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/tmp/registry/com.sun.star.comp.deployment.script.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/tmp/registry/com.sun.star.comp.deployment.sfwk.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/.config/libreoffice/4/user/extensions/tmp/registry/com.sun.star.comp.deployment.bundle.PackageRegistryBackend
```

Рис. 2.3: Поиск файлов, начинающиеся с символа “с”

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

```
nastarkov@dk3n59 ~ $ find /etc -name "h*" -print
find: '/etc/munge': Отказано в доступе
/etc/hosts.allow
find: '/etc/multipath': Отказано в доступе
/etc/httpd
/etc/hosts
/etc/hotplug
find: '/etc/cron.monthly': Отказано в доступе
find: '/etc/lvm/cache': Отказано в доступе
find: '/etc/cron.weekly': Отказано в доступе
find: '/etc/cron.daily': Отказано в доступе
find: '/etc/audit/plugins.d': Отказано в доступе
/etc/highlight
/etc/mercurial/hgrc.d
/etc/harbour.cfg
/etc/hotplug.d
/etc/conf.d/hdparm
/etc/conf.d/hwclock
/etc/conf.d/hddtemp
/etc/conf.d/hsqldb
/etc/conf.d/hostname
find: '/etc/cups/certs': Отказано в доступе
find: '/etc/cups/ssl': Отказано в доступе
/etc/init.d/hotplug
/etc/init.d/hdparm
/etc/init.d/hddtemp
/etc/init.d/hsqldb
/etc/sane.d/hp.conf
/etc/sane.d/hp3900.conf
/etc/sane.d/hp4200.conf
/etc/sane.d/hs2p.conf
/etc/sane.d/hp5400.conf
/etc/nvme/hostnqn
/etc/nvme/hostid
/etc/apache2/httpd.conf
/etc/brltty/Input/hw
/etc/brltty/Input/hd
/etc/brltty/Input/hm
/etc/brltty/Input/ht
/etc/brltty/Input/bm/horizontal.kti
/etc/brltty/Text/hi.ttb
/etc/brltty/Text/hy.ttb
/etc/brltty/Text/hu.ttb
/etc/brltty/Text/he.ttb
/etc/brltty/Text/hr.ttb
```

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

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

```
nastarkov@dk3n59 ~ $ find /var/log -name "log*" -print > logfile &
```

Рис. 2.5: Запускаем процесс записи в файл файлов, начинающихся с log

6) Удаляем файл ~/logfile



```
nastarkov@dk3n59 ~ $ rm logfile
```

Рис. 2.6: Удаление файла

7)Запускаем из консоли в фоновом режиме редактор gedit

```
nastarkov@dk3n59 ~ $ gedit &  
[1] 20898
```

Рис. 2.7: Запуск в фоновом режиме редактора gedit

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

```
nastarkov@dk3n59 ~ $ ps aux | grep gedit  
nastark+  20898  2.5  1.3 745472 107048 pts/2  S1   14:51   0:00 gedit  
nastark+  20953  0.0  0.0  10156   912 pts/2  S+   14:52   0:00 grep --colour=aut  
gedit
```

Рис. 2.8: 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] <pid> [...]

DESCRIPTION
    The default signal for kill is TERM. Use -l or -L to list available sig-
    nals. 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 pro-
    cesses except the kill process itself and init.

OPTIONS
    <pid> [...]
        Send signal to every <pid> listed.

    -<signal>
    -s <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.

    -q, --queue value
        Use sigqueue(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 this signal using the SA_SIGINFO
        flag to sigaction(2), then it can obtain this data via the si_value
        field of the siginfo_t structure.

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

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

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

EXAMPLES
    kill -9 1234
```

Рис. 2.9: Справка команды kill

```
nastarkov@dk3n59 ~ $ man kill
[2]+  Остановлен   man kill
nastarkov@dk3n59 ~ $ kill 20898
nastarkov@dk3n59 ~ $ jobs
[1]-  Завершено    gedit
[2]+  Остановлен   man kill
```

Рис. 2.10: 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 can-
    not show the space available on unmounted file systems, because on most
    kinds of systems doing so requires very nonportable 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=SIZE
        scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in
        units of 1,048,576 bytes; see SIZE format below

    -h, --human-readable
        print sizes in powers of 1024 (e.g., 1023M)

    -H, --si
        print sizes in powers of 1000 (e.g., 1.1G)

    -i, --inodes
        list inode information instead of block usage

    -k
        like --block-size=1K

```

Рис. 2.11: Информация о команде df

```

DU(1)                                User Commands                                DU(1)

NAME
    du - estimate file space usage

SYNOPSIS
    du [OPTION]... [FILE]...
    du [OPTION]... --files0-from=F

DESCRIPTION
    Summarize disk usage of the set of FILES, recursively for directories.

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

    -0, --null
        end each output line with NUL, not newline

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

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

    -B, --block-size=SIZE
        scale sizes by SIZE before printing them; e.g., '-BM' 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-args
        dereference only symlinks that are listed on the command line

    -d, --max-depth=N
        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 --summarize

    --files0-from=F
        summarize disk usage of the NUL-terminated file names specified in
        file F; if F is -, then read names from standard input

```

Рис. 2.12: Информация о команде du

```
[4]+ Остановлен man du
nastarkov@dk3n59 ~ $ df
```

Файловая система	1К-блоков	Использовано	Доступно	Испол
none	4000180	20292	3979888	
1% /run				
udev	3890156	0	3890156	
0% /dev				
tmpfs	4000180	95284	3904896	
3% /dev/shm				
/dev/sda8	491812356	92419284	374387328	
20% /				
/dev/sda6	91557952	492000	86392008	
1% /var/cache/openafs				
tmpfs	4000180	155316	3844864	
4% /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	260	799776	
1% /run/user/4272				
mark.sci.pfu.edu.ru:/usr/local/share/portage	18350080	5858816	11009792	
35% /usr/local/share/portage				

Рис. 2.13: Команда df в действии

```
nastarkov@dk3n59 ~ $ du
```

Рис. 2.14: Команда du

```

11      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
120
9      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
231
1101    ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
41
3      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
2
9      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
40
24      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
179
5      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
52
41      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
91
49      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
53
20      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
135
21      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
109
33      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
134
22      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
163
183      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
77
155      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
186
14      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
92
13      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
102
152      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
245
120      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
4
18      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
12
4      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
145
4      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
229
13      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
6
4      ./mozilla/firefox/nwnnnews.default-esr/storage/default/https+++web.telegram.org/cache/morgue/
51

```

Рис. 2.15: Команда du в действии

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

```

nastarkov@dk3n59 ~ $ find type d

```

Рис. 2.16: Вывод всех директорий, имеющих в домашнем каталоге

```
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/b3/18add34024fa2835f7
6bd98116573bab3587d4
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/b3/1fb3e46bc0381fe5a2
94eb6a9e0897d2127045
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/24
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/24/d14bfc9e2cab9601b1
09d6cf2ff289aac1e8c0
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/5f
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/5f/c921e63745adfd0bcf
df2e65efcc77ea643ac4
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/5f/82f51f03cd88a2de9f
0879dec6a98fd62aa86
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/51
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/51/04413ddfa2a2873c62
f2cb686f9361722d1e00
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/51/97613ce7c56243b2e3
b2c72ed056103bc28ad3
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/2e
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/2e/7053c52dd30d7e670b
44e6d9261092be05bf76
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/2e/bab27a0f81ad443961
847a72c852558a29e133
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/04
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/04/a7bae547f3cd518bd0
361f4839741f5a5e39f4
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/d6
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/d6/318fddc0c25645fb92
8cf0dfdb9c9ce94a809d
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/14
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/14/e9e7e994075c5bddd4
b67cb6d116c16c4e8a3b
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/a2
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/a2/e1d7688c64be868d3f
f19cfebe0ede595900e5
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/62
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/62/d11e02cf6555d67c6c
ee0427cad56a9aa85378
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/73
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/73/a303139785c64d3ada
d30dc9ad428a5d40e92b
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/73/c5c129404948346d4a
152dd336960e6dd00dd3
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/73/81f7b779c8013c4020
3dbe2296467743519805
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/c1
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/c1/954b5eb61bd31680ba
0b12b290f00db824103
/afs/.dk.sci.pfu.edu.ru/home/n/a/nastarkov/work/blog/.git/modules/public/objects/5d
```

Рис. 2.17: Вывод всех директорий, имеющиххся 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

**Вывод:** в ходе выполнения лабораторной работы я ознакомился с инструментами поиска файлов и фильтрации текстовых данных, приобрел практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.