

Отчет по лабораторной работе №6

Дисциплина: Операционные системы

Кабанова Варвара Дмитриевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	16
4	Контрольные вопросы	17
	Список литературы	23

Список иллюстраций

2.1	рис.1	6
2.2	рис.2	6
2.3	рис.3	6
2.4	рис.4	7
2.5	рис.5	7
2.6	рис.6	8
2.7	рис.7	8
2.8	рис.8	9
2.9	рис.9	9
2.10	рис.10	9
2.11	рис.11	10
2.12	рис.12	10
2.13	рис.13	10
2.14	рис.14	10
2.15	рис.15	11
2.16	рис.16	11
2.17	рис.17	11
2.18	рис.18	12
2.19	рис.19	12
2.20	рис.20	13
2.21	рис.21	13
2.22	рис.22	14
2.23	рис.23	14
2.24	рис.24	15
2.25	рис.25	15

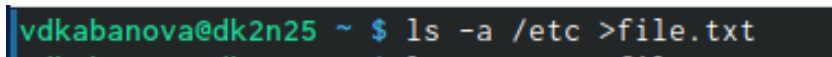
Список таблиц

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.

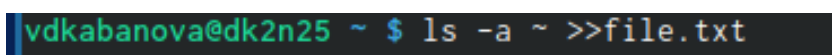
2 Выполнение лабораторной работы

Записываю в файл file.txt названия файлов, содержащихся в каталоге /etc (рис. 1). Дописываю в этот же файл названия файлов, содержащихся в вашем домашнем каталоге (рис.2).



```
vdkabanova@dk2n25 ~ $ ls -a /etc >file.txt
```

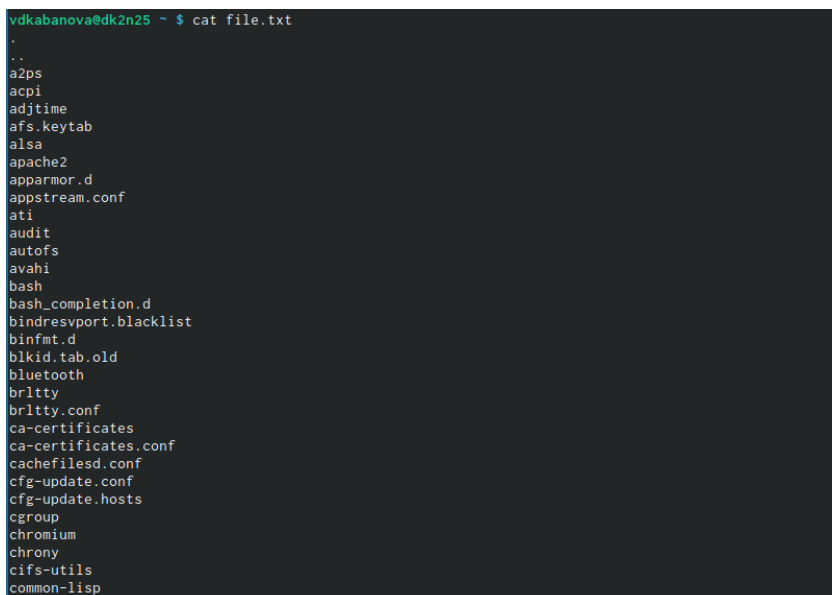
Рис. 2.1: рис.1



```
vdkabanova@dk2n25 ~ $ ls -a ~ >>file.txt
```

Рис. 2.2: рис.2

С помощью команды cat вывожу имена всех файлов из file.txt (рис. 3-4).



```
vdkabanova@dk2n25 ~ $ cat file.txt
.
..
a2ps
acpi
adjtime
afs.keytab
alsa
apache2
apparmor.d
appstream.conf
ati
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
```

Рис. 2.3: рис.3

```
file.txt
games
.gitconfig
.gnupg
GNUstep
.gphoto
.gtkrc-2.0
io.h
.lesshst
.local
may
monthly
.mozilla
my_os
.octave_hist
.pki
play
.profile
public
public_html
reports
ski_places
.ssh
.texlive2021
tmp
work
.Xauthority
.xsession-errors
.xsession-errors.old
Документы
Загрузки
Изображения
Музыка
Общедоступные
Рабочий стол
Шаблоны
```

Рис. 2.4: рис.4

Вывожу имена всех файлов из file.txt, имеющих расширение .conf, после чего записываю их в новый текстовый файл conf.txt (рис.5-6).

```
vdkabanova@dk2n25 ~ $ grep -e '\.conf$' file.txt > conf.txt
vdkabanova@dk2n25 ~ $
```

Рис. 2.5: рис.5

```

vdkabanova@dk2n25 ~ $ cat conf.txt
appstream.conf
brlty.conf
ca-certificates.conf
cachefilesd.conf
cfg-update.conf
cpufreq-bench.conf
dhcpcd.conf
dispatch.conf
dleyna-server-service.conf
dnsmasq.conf
dracut.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
logrotate.conf

```

Рис. 2.6: рис.6

Определяю, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа с. Привожу пример двух команд, которые позволяют сделать это (рис.7-9).

```

vdkabanova@dk2n25 ~ $ find ~ -maxdepth 1 -name "c*" -print
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/assets/close-normal.svg
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/assets/close-active.svg
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/assets/close-hover.svg
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/assets/close-backdrop-normal.svg
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/assets/close-backdrop-active.svg
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/assets/close-backdrop-hover.svg
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/colors.css
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/kdeconnect/certificate.pem
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/kdeconnect/config
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/config
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.component.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.configuration.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.executable.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.help.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.script.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.sfwk.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.bundle.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.component.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.configuration.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.configuration.PackageRegistryBackend/configmgr.ini
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.executable.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.help.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.depl

```

Рис. 2.7: рис.7


```
vdkabanova@dk2n25 ~ $ find -name "cs" -print
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/assets/close-normal.svg
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/assets/close-active.svg
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/assets/close-hover.svg
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/assets/close-backdrop-normal.svg
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/assets/close-backdrop-active.svg
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/assets/close-backdrop-hover.svg
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/gtk-3.0/colors.css
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/kdeconnect/certificate.pem
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/kdeconnect/config
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/config
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.component.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.configuration.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.executable.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.help.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.script.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.sfwk.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/shared/registry/com.sun.star.comp.deployment.bundle.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.component.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.configuration.PackageRegistryBackend
/afs/.dk.sci.pfu.edu.ru/home/v/d/vdkabanova/.config/libreoffice/4/user/extensions/bundled/registry/com.sun.star.comp.deployment.configuration.PackageRegistryBackend/configmgr.ini
```

Рис. 2.8: рис.8

```
/etc/systemd/homed.conf
/etc/systemd/system/hddtemp.service.d
/etc/nvme/hostnqn
/etc/nvme/hostid
/etc/xdg/xfce4/helpers.rc
/etc/sane.d/hp3900.conf
/etc/sane.d/hp.conf
/etc/sane.d/hp5400.conf
/etc/sane.d/hs2p.conf
/etc/sane.d/hp4200.conf
/etc/hsqldb
/etc/udev/hwdb.d
/etc/udev/hwdb.bin
/etc/brltty/Contraction/ha.ctb
/etc/brltty/Text/hr.ttb
/etc/brltty/Text/hu.ttb
/etc/brltty/Text/hy.ttb
/etc/brltty/Text/he.ttb
/etc/brltty/Text/hi.ttb
/etc/brltty/Input/hd
/etc/brltty/Input/ht
/etc/brltty/Input/bm/horizontal.kti
/etc/brltty/Input/hm
/etc/brltty/Input/hw
/etc/distcc/hosts
/etc/X11/fontpath.d/hack
/etc/runlevels/default/hdparm
/etc/runlevels/boot/hwclock
/etc/runlevels/boot/hostname
/etc/httpd
find: '/etc/cron.weekly': Отказано в доступе
find: '/etc/fcron': Отказано в доступе
find: '/etc/cups/certs': Отказано в доступе
find: '/etc/cups/ssl': Отказано в доступе
find: '/etc/.git': Отказано в доступе
find: '/etc/multipath': Отказано в доступе
lines 1-36/66 61%
```

Рис. 2.9: рис.9

Вывожу на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h (рис.10).

```
13. невозможно получить доступ к ... нет такого файла
vdkabanova@dk2n25 ~ $ find /etc -name "h*" | less
```

Рис. 2.10: рис.10

Запускаю в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log. (рис.11).

```
vdkabanova@dk2n25 ~ $ find /var/log -name "log*" -print > logfile &
[1] 9533
vdkabanova@dk2n25 ~ $ find: '/var/log/audit': Отказано в доступе
find: '/var/log/telnet': Отказано в доступе
find: '/var/log/munge': Отказано в доступе
find: '/var/log/mail': Отказано в доступе
find: '/var/log/sandbox': Отказано в доступе
find: '/var/log/pwdfail': Отказано в доступе
find: '/var/log/cron': Отказано в доступе
find: '/var/log/critical': Отказано в доступе
find: '/var/log/everything': Отказано в доступе
find: '/var/log/private': Отказано в доступе
find: '/var/log/hal': Отказано в доступе
find: '/var/log/kernel': Отказано в доступе
find: '/var/log/crond': Отказано в доступе
find: '/var/log/sshd': Отказано в доступе
find: '/var/log/portage': Отказано в доступе
find: '/var/log/apache2': Отказано в доступе
find: '/var/log/mysql': Отказано в доступе
```

Рис. 2.11: рис.11

Удаляю файл ~/logfile (рис.12).

```
vdkabanova@dk2n25 ~ $ rm logfile
```

Рис. 2.12: рис.12

Запускаю из консоли в фоновом режиме редактор gedit (рис. 13).

```
vdkabanova@dk2n25 ~ $ gedit &
[1] 9719
vdkabanova@dk2n25 ~ $
(gedit:9719): Gtk-WARNING **: 10:58:01.126: Negative content width -8 (allocation 16, extents 12x12) while allocating gad
get (node label, owner GtkLabel)
```

Рис. 2.13: рис.13

Определяю идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep (рис. 14-15).

```
vdkabanova@dk2n25 ~ $ ps aux | grep gedit
vdkaban+  9719  0.7  1.3 706596 108420 pts/2    Sl   18:57   0:01 gedit
vdkaban+  9924  0.0  0.0   8212  2384 pts/2    S+   19:00   0:00 grep --colour=auto gedit
```

Рис. 2.14: рис.14

```

vdkabanova@dk2n25 ~ $ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0 169188 11836 ?        Ss   17:18   0:01 /usr/lib/systemd/systemd
root         2  0.0  0.0   0     0 ?        S    17:18   0:00 [kthreadd]
root         3  0.0  0.0   0     0 ?        Ic   17:18   0:00 [rcu_gp]
root         4  0.0  0.0   0     0 ?        Ic   17:18   0:00 [rcu_par_gp]
root         6  0.0  0.0   0     0 ?        Ic   17:18   0:00 [kworker/0:0H-events_highpri]
root         8  0.0  0.0   0     0 ?        Ic   17:18   0:00 [mm_percpu_wq]
root         9  0.0  0.0   0     0 ?        S    17:18   0:00 [rcu_tasks_kthre]
root        10  0.0  0.0   0     0 ?        S    17:18   0:00 [rcu_tasks_trace]
root        11  0.0  0.0   0     0 ?        S    17:18   0:02 [ksoftirqd/0]
root        12  0.0  0.0   0     0 ?        I    17:18   0:03 [rcu_preempt]
root        13  0.0  0.0   0     0 ?        S    17:18   0:00 [migration/0]
root        14  0.0  0.0   0     0 ?        S    17:18   0:00 [cpuhp/0]
root        15  0.0  0.0   0     0 ?        S    17:18   0:00 [cpuhp/1]
root        16  0.0  0.0   0     0 ?        S    17:18   0:00 [migration/1]
root        17  0.0  0.0   0     0 ?        S    17:18   0:00 [ksoftirqd/1]
root        19  0.0  0.0   0     0 ?        Ic   17:18   0:00 [kworker/1:0H-kblockd]
root        20  0.0  0.0   0     0 ?        S    17:18   0:00 [cpuhp/2]
root        21  0.0  0.0   0     0 ?        S    17:18   0:00 [migration/2]
root        22  0.0  0.0   0     0 ?        S    17:18   0:00 [ksoftirqd/2]
root        24  0.0  0.0   0     0 ?        Ic   17:18   0:00 [kworker/2:0H-kblockd]
root        25  0.0  0.0   0     0 ?        S    17:18   0:00 [cpuhp/3]
root        26  0.0  0.0   0     0 ?        S    17:18   0:00 [migration/3]
root        27  0.0  0.0   0     0 ?        S    17:18   0:00 [ksoftirqd/3]
root        29  0.0  0.0   0     0 ?        Ic   17:18   0:00 [kworker/3:0H-kblockd]
root        30  0.0  0.0   0     0 ?        S    17:18   0:00 [cpuhp/4]
root        31  0.0  0.0   0     0 ?        S    17:18   0:00 [migration/4]
root        32  0.0  0.0   0     0 ?        S    17:18   0:00 [ksoftirqd/4]
root        34  0.0  0.0   0     0 ?        Ic   17:18   0:00 [kworker/4:0H-kblockd]
root        35  0.0  0.0   0     0 ?        S    17:18   0:00 [cpuhp/5]
root        36  0.0  0.0   0     0 ?        S    17:18   0:00 [migration/5]
root        37  0.0  0.0   0     0 ?        S    17:18   0:00 [ksoftirqd/5]
root        39  0.0  0.0   0     0 ?        Ic   17:18   0:00 [kworker/5:0H-kblockd]
root        40  0.0  0.0   0     0 ?        S    17:18   0:00 [kdevtmpfs]
root        41  0.0  0.0   0     0 ?        Ic   17:18   0:00 [netns]
root        42  0.0  0.0   0     0 ?        Ic   17:18   0:00 [inet_frag_wq]
root        43  0.0  0.0   0     0 ?        S    17:18   0:00 [kauditd]
root        46  0.0  0.0   0     0 ?        S    17:18   0:00 [oom_reaper]
root        47  0.0  0.0   0     0 ?        Ic   17:18   0:00 [writeback]
root        48  0.0  0.0   0     0 ?        S    17:18   0:00 [kcompactd0]
root        49  0.0  0.0   0     0 ?        SN   17:18   0:00 [ksad]
root        50  0.0  0.0   0     0 ?        SN   17:18   0:00 [khugepaged]
root        51  0.0  0.0   0     0 ?        Ic   17:18   0:00 [pencrypt_serial]
root        52  0.0  0.0   0     0 ?        Ic   17:18   0:00 [pdecrypt_serial]
root        53  0.0  0.0   0     0 ?        Ic   17:18   0:00 [kintegrityd]
root        54  0.0  0.0   0     0 ?        Ic   17:18   0:00 [kblockd]
root        55  0.0  0.0   0     0 ?        Ic   17:18   0:00 [blkcg_punt_bio]
root        56  0.0  0.0   0     0 ?        Ic   17:18   0:00 [ata_sff]
root        57  0.0  0.0   0     0 ?        Ic   17:18   0:00 [edac-poller]

```

Рис. 2.15: рис.15

Прочитываю справку (man) команды kill (рис. 16-17), после чего используйте её для завершения процесса gedit (рис. 18)

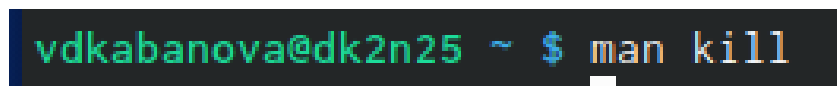


Рис. 2.16: рис.16

```

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, the -l or -L to list available signals. Particularly useful signals include SIGINT, KILL, STOP, CONT, and 0. Alternate signals may be specified in three ways: -s, -SIGNAME, or -KILL. Negative PID values may be used to choose whole process groups; see the PID2 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 qids listed.

    -s 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 signal pid
        Use signal(7) rather than kill(1) 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 format
        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 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
        Translate number 1 into a signal name.

    kill -1
        List the available signal choices in a nice table.

    kill 123 456 2345 3456
        Send the default signal, SIGTERM, to all these processes.

SEE ALSO
    kill(2), killall(1), nice(1), ps(1), renice(1), signal(7), sigaction(2), sh(1)(1)

PROGRAMS
    (banned) killall line 8 (press h for help or q to quit)
    (banned) killall line 8 (press h for help or q to quit)

```

Рис. 2.17: рис.17

```

vdkabanova@dk2n25 ~ $ ps aux | grep gedit
vdkaban+  9719  0.3  1.3 706736 108420 pts/2    Sl   18:57   0:01 gedit
vdkaban+ 10237  0.0  0.0   8212  2360 pts/2    S+   19:04   0:00 grep --colour=auto gedit
vdkabanova@dk2n25 ~ $ kill 9719
vdkabanova@dk2n25 ~ $

```

Рис. 2.18: рис.18

Получаю более подробную информацию о командах `df`, `du`, с помощью команды `man` (рис. 19-21).

```

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 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=K

    -l, --local
        limit listing to local file systems

    --no-sync
        do not invoke sync before getting usage info (default)

Manual page df(1) line 1 (press h for help or q to quit)

```

Рис. 2.19: рис.19

```
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

    -H
        equivalent to --dereference-args (-D)

    -h, --human-readable
        Manual page du(1) line 1 (press h for help or q to quit)
```

Рис. 2.20: рис.20

```
vdkabanova@dk2n25 ~ $ man df
[1]+  Завершено      gedit
vdkabanova@dk2n25 ~ $ man du
```

Рис. 2.21: рис.21

Выполняю эти команды (рис. 22-24)

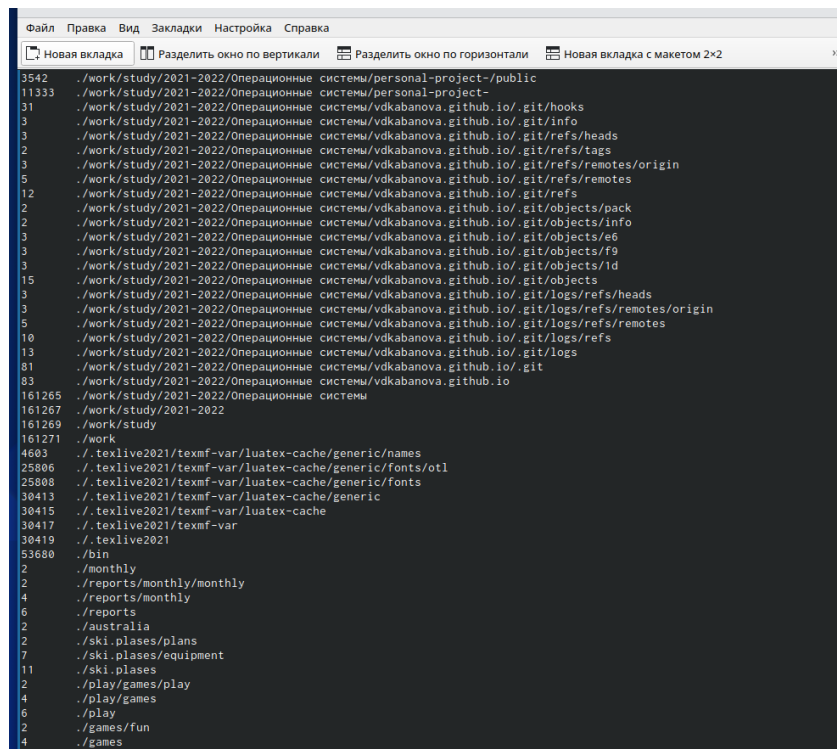


Рис. 2.22: рис.22

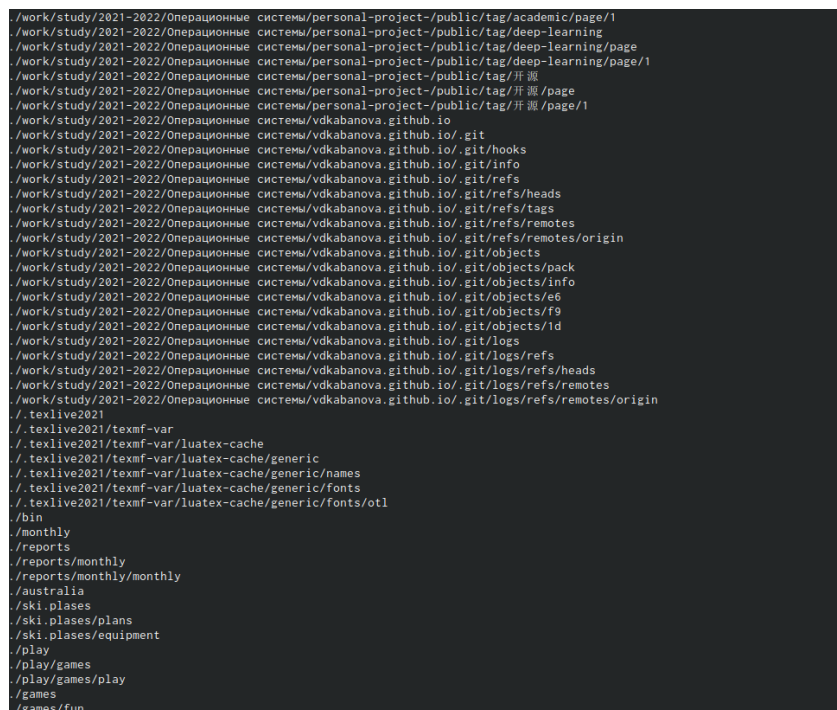


Рис. 2.23: рис.23

```
vdkabanova@dk2n25 ~ $ df \
```

Файловая система	1K-блоков	Использовано	Доступно	Использовано	Смонтировано в
none	3999736	20288	3979448	1%	/run
udev	10240	0	10240	0%	/dev
tmpfs	3999736	16	3999720	1%	/dev/shm
/dev/sda8	485013560	73680084	386626420	17%	/
tmpfs	3999740	74944	3924796	2%	/tmp
/dev/sda6	50143784	337492	47229396	1%	/var/cache/openafs
mark.sci.pfu.edu.ru:/usr/portage	18350080	5853440	11013376	35%	/usr/portage
mark.sci.pfu.edu.ru:/usr/local/share/portage	18350080	5853440	11013376	35%	/usr/local/share/portage
mark.sci.pfu.edu.ru:/com/lib/portage	1048320000	504342016	543977984	49%	/com/lib/portage
AFS	2147483647	0	2147483647	0%	/afs
tmpfs	799944	244	799700	1%	/run/user/4160
mark.sci.pfu.edu.ru:/usr/local/share/portage	18350080	5853440	11013376	35%	/usr/local/share/portage

Рис. 2.24: рис.24

Воспользовавшись справкой команды `find`, вывожу имена всех директорий, имеющих в моем домашнем каталоге (рис. 24).

```
./games/ron
vdkabanova@dk2n25 ~ $ find -type d
```

Рис. 2.25: рис.25

3 Выводы

В ходе выполнения данной лабораторной работы я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных; приобретение практические навыков: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем

4 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?
2. Объясните разницу между операцией `>` и `»`.
3. Что такое конвейер?
4. Что такое процесс? Чем это понятие отличается от программы?
5. Что такое PID и GID?
6. Что такое задачи и какая команда позволяет ими управлять?
7. Найдите информацию об утилитах `top` и `htop`. Каковы их функции?
8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.
9. Можно ли по контексту (содержанию) найти файл? Если да, то как?
10. Как определить объем свободной памяти на жёстком диске?
11. Как определить объем вашего домашнего каталога?
12. Как удалить зависший процесс?

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:

команда1|команда2 (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.

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

5) pid: идентификатор процесса (PID) процесса (processID), 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
- typef - искать только файлы
- typed - 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.

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 pkill – 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.

Список литературы