

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 9

дисциплина: Операционные системы

Студент: Хосе Фернадо Леон Атупанья

Группа: НПМбд-02-20

МОСКВА 2021 г.

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.

Задание

1. Осуществите вход в систему, используя соответствующее имя пользователя.
2. Запишите в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.
3. Выведите имена всех файлов из file.txt, имеющих расширение .conf, после чего запишите их в новый текстовый файл conf.txt.
4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинающиеся с символа с? Предложите несколько вариантов, как это сделать.
5. Выведите на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h.
6. Запустите в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log.

7. Удалите файл ~/logfile.
8. Запустите из консоли в фоновом режиме редактор gedit.
9. Определите идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Можно ли определить этот идентификатор более простым способом?
10. Прочтите справку (man) команды kill, после чего используйте её для завершения процесса gedit.
11. Выполните команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man.
12. Воспользовавшись справкой команды find, выведите имена всех директорий, имеющих в вашем домашнем каталоге.

Выполнение лабораторной работы

1. Осуществите вход в систему, используя соответствующее имя пользователя.
2. Для записи в файл file.txt имена файлов, содержащихся в каталоге / и т.д., Я использую команду "ls - a / и т.д. > file.txt" (рис. -@рис.: 001). Затем с помощью команды "ls-a ~ >> file.txt" я добавляю в тот же файл имена файлов, содержащихся в моем домашнем каталоге (рис. - @рис.002). Команда "cat file.txt" я смотрю на файл, чтобы увидеть, правильны ли действия (рис . - @рис.003).

```
fernando@fernando-VirtualBox:~$ ls -a /etc > file.txt
fernando@fernando-VirtualBox:~$ cat file.txt
.
..
acpi
adduser.conf
alsa
alternatives
anacrontab
apg.conf
apm
apparmor
apparmor.d
appport
appstream.conf
apt
avahi
bash.bashrc
bash_completion
bash_completion.d
bindresvport.blacklist
binfmt.d
bluetooth
brlapi.key
brltty
brltty.conf
ca-certificates
ca-certificates.conf
ca-certificates.conf.dpkg-old
calendar
chatscripts
console-setup
cracklib
cron.d
cron.daily
cron.hourly
cron.monthly
crontab
cron.weekly
cups
cupshelpers
dbus-1
dconf
debconf.conf
debian_version
default
deluser.conf
depmod.d
dhcp
dictionaries-common
dpkg
e2scrub.conf
emacs
environment
environment.d
ethertypes
firefox
fonts
fprintd.conf
fstab
fuse.conf
```

{ #fig:001 width=70% }

```

fernando@fernando-VirtualBox:~$ ls -la --> file.txt
fernando@fernando-VirtualBox:~$ ls -la ~
.          .bash_history  .cache  docs      equipment  .gitconfig  lab2    may      .mozilla  Pictures  program  .ssh      .sudo_as_admin_successful  Videos  work
..         .bash_logout   .config Documents feathers    .gnupg   labor2  monthly  Music     '~/play'  Public   .sudo_as_admin_successful  .vininfo .Xauthority
01        australia .bashrc      Desktop  Downloads file.txt     images  .local  monthly.00 nyoa      .profile  ski_places  templates  .wget-hsts  .Xsession-errors
fernando@fernando-VirtualBox:~$ cat file.txt
.
..
acpi
adduser.conf
alsa
alternatives
anacrontab
apg.conf
apn
apparmor
apparmor.d
appport
appstream.conf
apt
avahi
bash.bashrc
bash_completion
bash_completion.d
bindresvport.blacklist
binfmt.d
bluetooth
brlapi.key
brltty
brltty.conf
ca-certificates
ca-certificates.conf
ca-certificates.conf.dpkg-old
calendar
chatscripts
console-setup
crackmap
cron.d
cron.daily
cron.hourly
cron.monthly
crontab
cron.weekly
cups
cupshelpers
dbus-1
dconf
debconf.conf
debian_version
default
deluser.conf
depmod.d
dhcp
dictionaries-common
dpkg
e2scrub.conf
emacs
environment
environment.d
ethertypes
firefox
fonts
fontconfig

```

{ #fig:002 width=70% }

```
usb_modeswitch.conf
usb_modeswitch.d
vdpau_wrapper.cfg
vim
vmware-tools
vtrgb
vulkan
wgetrc
whoopsie
wpa_supplicant
wvdial.conf
X11
xattr.conf
xdg
xml
zsh_command_not_found
.
..
01
abc1
april
australia
.bash_history
.bash_logout
.bashrc
.cache
.config
Desktop
docs
Documents
Downloads
equipment
feathers
file.txt
.gitconfig
.gnupg
images
lab2
labor2
.local
may
monthly
monthly.00
.mozilla
Music
myos
Pictures
~.play
.profile
program
Public
ski.places
.ssh
.sudo_as_admin_successful
Templates
Videos
.viminfo
.wget-hsts
work
.Xauthority
.xsession-errors
fernando@fernando-VirtualBox:~$
```

{ #fig:003 width=70% }

3. Я имена всех файлов file.txt которые имеют расширение .conf и записать их в новый текстовый файл conf.txt с помощью команды 'grep-e'.conf\$' file.txt >

conf.txt». "Кот conf.txt" проверяю правильность выполненных действий (рис . - @рис.004).

```
fernando@fernando-VirtualBox:~$ find file.txt -name | "*.conf" -print > conf.txt
find: missing argument to '-name'
*.conf: command not found
fernando@fernando-VirtualBox:~$ find file.txt -name "*.conf" -print > conf.txt
fernando@fernando-VirtualBox:~$ cat conf.txt
fernando@fernando-VirtualBox:~$ ls
.  ..  australia  conf.txt  Desktop  docs  Documents  Downloads  file.txt  images  lab2  labor2  monthly  monthly.00  Music  Pictures  '!.play'
fernando@fernando-VirtualBox:~$ grep -e '\.conf$' file.txt > conf.txt
fernando@fernando-VirtualBox:~$ cat conf.txt
adduser.conf
apg.conf
appstream.conf
brltty.conf
ca-certificates.conf
debconf.conf
deluser.conf
e2scrub.conf
fprintd.conf
fuse.conf
gal.conf
hdparm.conf
host.conf
kernel-img.conf
kerneloops.conf
ld.so.conf
libao.conf
libaudit.conf
logrotate.conf
ltrace.conf
mke2fs.conf
ntools.conf
nsswitch.conf
pam.conf
pnm2ppa.conf
popularity-contest.conf
resolv.conf
rsyslog.conf
rygel.conf
sensors3.conf
sysctl.conf
ucf.conf
uniconf.conf
usb_modeswitch.conf
wvdial.conf
xattr.conf
fernando@fernando-VirtualBox:~$
```

{ #fig:004 width=70% }

4. Чтобы определить, какие файлы в моем домашнем каталоге имеют имена, начинающиеся с символа с, вы можете использовать несколько команд: "find ~ - maxdepth 1-name" с "-print "(опция maxdepth 1 необходима, чтобы файлы находились только в домашнем каталоге (а не в его подкаталогах)), "ls ~/C "и" ls-a ~ | grep C* " (рис. - @рис.005).

```
fernando@fernando-VirtualBox:~$ find ~ -maxdepth 1 -name "C*" -print
fernando@fernando-VirtualBox:~$ find ~ -maxdepth 1 -name "C*" -print
fernando@fernando-VirtualBox:~$ ls ~/C*
ls: cannot access '/home/fernando/C*': No such file or directory
fernando@fernando-VirtualBox:~$ find ~ -maxdepth 1 -name "C*" -print
fernando@fernando-VirtualBox:~$ ls ~/C*
/home/fernando/conf.txt
```

{ #fig:005 width=70% }

5. Чтобы отобразить (страница за страницей) имена файлов каталога / etc, начинающиеся с символа h, используем команду "find | etc -maxdepth 1 -name "h*" / less" (рис. - @рис.006).

```
/etc/hp
/etc/hdparm.conf
/etc/hosts.deny
/etc/hostid
/etc/hosts
/etc/hosts.allow
/etc/host.conf
/etc/hostname
(END)
```

{ #fig:006 width=70% }

6. Я запускаю фоновый процесс, который будет записывать в файл ~ / logfile файлы, имена которых начинаются с log, используя команду "find / - name log * "> logfile & " (рис. - @рис.007). Команда "cat logfile" проверяет выполненные действия (рис. - @рис.008).
7. Удаляю файл ~/logfile командой «rm logfile».

```
Fernando@fernando-VirtualBox:~$ find / -name "long*" > logfile &
[1] 11452
Fernando@fernando-VirtualBox:~$ find: '/var/tmp/systemd-private-d4fa4f14046746cc8412107e6c830499-systemd-logind.service-EKdCre': Permission denied
find: '/var/tmp/systemd-private-d4fa4f14046746cc8412107e6c830499-ModemManager.service-TbCj4h': Permission denied
find: '/var/tmp/systemd-private-d4fa4f14046746cc8412107e6c830499-systemd-resolved.service-3N5W0g': Permission denied
find: '/var/tmp/systemd-private-d4fa4f14046746cc8412107e6c830499-systemd-timesyncd.service-PSYwKg': Permission denied
find: '/var/tmp/systemd-private-d4fa4f14046746cc8412107e6c830499-colord.service-ub4U5h': Permission denied
find: '/var/tmp/systemd-private-d4fa4f14046746cc8412107e6c830499-upower.service-K3yIjg': Permission denied
find: '/var/tmp/systemd-private-d4fa4f14046746cc8412107e6c830499-switcheroo-control.service-1k3ZQh': Permission denied
find: '/var/cache/private': Permission denied
find: '/var/cache/apparmor/83ba7c1a.0': Permission denied
find: '/var/cache/apparmor/26b63962.0': Permission denied
find: '/var/cache/apt/archives/partial': Permission denied
find: '/var/cache/ldconfig': Permission denied
find: '/var/cache/cups': Permission denied
find: '/var/log/private': Permission denied
find: '/var/log/speech-dispatcher': Permission denied
find: '/var/log/gdm3': Permission denied
find: '/var/lib/poikit-1': Permission denied
find: '/var/lib/update-notifier/package-data-downloads/partial': Permission denied
find: '/var/lib/private': Permission denied
find: '/var/lib/snapd/cookie': Permission denied
find: '/var/lib/snapd/vold': Permission denied
find: '/var/lib/AccountsService/users': Permission denied
find: '/var/lib/gdm3/.cache/libweather': Permission denied
find: '/var/lib/gdm3/.config/pulse': Permission denied
find: '/var/lib/gdm3/.config/ibus': Permission denied
find: '/var/lib/gdm3/.local/share/sounds': Permission denied
find: '/var/lib/gdm3/.local/share/xorg': Permission denied
find: '/var/lib/gdm3/.local/share/gnome-shell': Permission denied
find: '/var/lib/gdm3/.local/share/keyrings': Permission denied
find: '/var/lib/NetworkManager': Permission denied
find: '/var/lib/udisks2': Permission denied
find: '/var/lib/apt/lists/partial': Permission denied
find: '/var/lib/sddm': Permission denied
find: '/var/lib/fwupd/gnupg': Permission denied
find: '/var/lib/colord/.cache': Permission denied
find: '/var/spool/rsyslog': Permission denied
find: '/var/spool/cron/crontabs': Permission denied
find: '/var/spool/cups': Permission denied
find: '/etc/poikit-1/localauthority': Permission denied
find: '/etc/ssl/private': Permission denied
find: '/etc/cups/ssl': Permission denied
find: '/sys/kernel/tracing': Permission denied
find: '/sys/kernel/debug': Permission denied
find: '/sys/fs/pstore': Permission denied
find: '/sys/fs/bpf': Permission denied
find: '/run/udisks2': Permission denied
find: '/run/cups/certs': Permission denied
find: '/run/user/1000/inaccessible': Permission denied
find: '/run/sudo': Permission denied
find: '/run/speech-dispatcher': Permission denied
find: '/run/openvpn-server': Permission denied
find: '/run/openvpn-client': Permission denied
find: '/run/systemd/resolve/netif': Permission denied
find: '/run/systemd/unit-root': Permission denied
find: '/run/systemd/inaccessible': Permission denied
find: '/run/inltrams': Permission denied
find: '/snap/core18/2066/etc/ssl/private': Permission denied
find: '/snap/core18/2066/root': Permission denied
find: '/snap/core18/2066/var/cache/ldconfig': Permission denied
find: '/snap/core18/2066/var/lib/private': Permission denied
find: '/snap/core18/1997/etc/ssl/private': Permission denied
```

{ #fig:007 width=70% }

```
Fernando@fernando-VirtualBox:~$ cat logfile
/snap/gnome-3-34-1804/72/usr/lib/x86_64-linux-gnu/perl/5.26.1/bits/long-double.ph
/snap/gnome-3-34-1804/72/usr/share/aclocal/longlong.m4
/snap/gnome-3-34-1804/66/usr/lib/x86_64-linux-gnu/perl/5.26.1/bits/long-double.ph
/snap/gnome-3-34-1804/66/usr/share/aclocal/longlong.m4
/usr/share/doc/libx11-protocol-perl/examples/long-run.pl
/usr/share/doc/git/contrib/long-running-filter
/usr/lib/x86_64-linux-gnu/perl/5.30.0/bits/long-double.ph
/usr/src/linux-hwe-5.8-headers-5.8.0-55/arch/um/include/shared/longjmp.h
/usr/src/linux-hwe-5.8-headers-5.8.0-53/arch/um/include/shared/longjmp.h
fernando@fernando-VirtualBox:~$
```

{ #fig:008 width=70% }

8. я запускаю редактор gedit в фоновом режиме с помощью команды "gedit &" (рис. - @рис.009). После этого на экране появляется окно редактора.

```
fernando@fernando-VirtualBox:~$ gedit &
[1] 11512
fernando@fernando-VirtualBox:~$
```

{ #fig:009 width=70% }

9. Чтобы определить идентификатор процесса gedit, используйте команду "ps / grep -l "gedit"". Наш процесс имеет PID 518. Вы также можете узнать

идентификатор процесса с помощью команды "pgrep gedit" или "pidof gedit" (рис. - @рис.010).

```
fernando@fernando-VirtualBox:~$ ps | grep -i "gedit"
[1]+  Done                  gedit
fernando@fernando-VirtualBox:~$ pgrep gedit
fernando@fernando-VirtualBox:~$ pidof gedit
fernando@fernando-VirtualBox:~$
```

{ #fig:009 width=70% }

10. Прочитав информацию из команды kill с помощью команды "man kill", я использую ее для завершения процесса gedit (команда "kill 518") (рис. - @рис.: 011) (рис. - @рис.012).

```
fernando@fernando-VirtualBox:~$ man kill
fernando@fernando-VirtualBox:~$ kill 518
bash: kill: (518) - No such process
fernando@fernando-VirtualBox:~$ man df
fernando@fernando-VirtualBox:~$ man du
fernando@fernando-VirtualBox:~$
```

{ #fig:0011 width=70% }

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

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

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

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

  --signal <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.

  -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 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 -l 11
    Translate number 11 into a signal name.

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

  kill 123 543 2341 3453
    Send the default signal, SIGTERM, to all those processes.

SEE ALSO
  kill(2), killall(1), niced(1), pkill(1), renice(1), signal(7), skill(1)

STANDARDS
  This command meets appropriate standards. The -L flag is Linux-specific.

AUTHOR
  Albert Cahalan (albert@users.sf.net) wrote kill in 1999 to replace a bsdutils one that was not standards compliant. The util-linux one might also work correctly.

REPORTING BUGS
  Please send bug reports to (procps@freelists.org)

procps-ng Manual page kill(1) line 1/56 (END) (press h for help or q to quit) 2018-05-31 KILL(1)
```

{ #fig:0012 width=70% }

11. С помощью команд "man df" (рис. - @рис.: 013) и "man du" (рис. - @рис.: 014) я узнаю информацию о необходимых командах и использую их дальше (рис. - @рис.015).

df-это утилита, которая отображает список всех файловых систем по имени устройства, сообщает их размер, занимаемое и свободное пространство и точки монтирования.

Синтаксис: df параметры устройства

du-это утилита, предназначенная для отображения информации о том, сколько дискового пространства занимает файлы и каталоги. Он принимает путь к элементу файловой системы и отображает информацию о количестве

байтов дискового пространства или дисковых блоков, используемых для его хранения.

Синтаксис: du параметры directorофайл

```
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)
    -t, --inodes
        list inode information instead of block usage
    -k
        like --block-size=1K
    -l, --local
        limit listing to local file systems
    --no-sync
        do not invoke sync before getting usage info (default)
    --output=FIELD_LIST
        use the output format defined by FIELD_LIST, or print all fields if FIELD_LIST is omitted.
    -P, --portability
        use the POSIX output format
    --sync
        invoke sync before getting usage info
    --total
        elide all entries insignificant to available space, and produce a grand total
    -t, --type=TYPE
        limit listing to file systems of type TYPE
    -T, --print-type
        print file system type
Manual page df(1) line 1 (press h for help or q to quit)
```

{ #fig:0013 width=70% }

```
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
        print sizes in human readable format (e.g., 1K 234M 2G)
    --inodes
        list inode usage information instead of block usage
    -k
        like --block-size=1K
    -L, --dereference
        dereference all symbolic links
    -l, --count-links
        count sizes many times if hard linked
    -m
        like --block-size=1M
    -P, --no-dereference
Manual page du(1) line 1 (press h for help or q to quit)
```

{ #fig:0014 width=70% }

```

fernando@fernando-VirtualBox:~$ man df
fernando@fernando-VirtualBox:~$ man du
fernando@fernando-VirtualBox:~$ df

```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
udev	2435156	0	2435156	0%	/dev
tmpfs	492944	1332	491612	1%	/run
/dev/sda5	30313412	10803432	17947100	38%	/
tmpfs	2464708	0	2464708	0%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
tmpfs	2464708	0	2464708	0%	/sys/fs/cgroup
/dev/loop1	56832	56832	0	100%	/snap/core18/1997
/dev/loop0	56832	56832	0	100%	/snap/core18/2066
/dev/loop2	224256	224256	0	100%	/snap/gnome-3-34-1804/66
/dev/loop3	224256	224256	0	100%	/snap/gnome-3-34-1804/72
/dev/loop4	63616	63616	0	100%	/snap/gtk-common-themes/1506
/dev/loop5	66688	66688	0	100%	/snap/gtk-common-themes/1515
/dev/loop6	52352	52352	0	100%	/snap/snap-store/518
/dev/loop7	52224	52224	0	100%	/snap/snap-store/542
/dev/loop8	32896	32896	0	100%	/snap/snapd/12057
/dev/loop9	32896	32896	0	100%	/snap/snapd/11841
/dev/sda1	523248	4	523244	1%	/boot/efi
tmpfs	492940	16	492924	1%	/run/user/1000
/dev/sr0	59588	59588	0	100%	/media/fernando/VBox_GAs_6.1.20

```

fernando@fernando-VirtualBox:~$ du
4      ./docs
4      ./monthly
4      ./monthly.00/monthly
8      ./monthly.00
4      ./Desktop/etc/vbox
8      ./Desktop/etc
13808 ./Desktop/usr/share/virtualbox/nls
168   ./Desktop/usr/share/virtualbox/rdesktop-vrdp-keymaps
24    ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/linux
176   ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/common/string
180   ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/common
4     ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include/iprt/linux
4     ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include/iprt/nocrt
188   ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include/iprt
12    ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include/internal
4     ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include/VBox/vmm
8     ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include/VBox
212   ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include
4     ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/r0drv/linux
8     ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/r0drv
4     ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/math/gcc
8     ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/math
456   ./Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp
52    ./Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/linux
68    ./Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/common/time
52    ./Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/common/checksum

```

{ #fig:0015 width=70% }

12. Мы получаем информацию с помощью команды " Man find "(рис. - @рис.: 016) и отображаем имена всех доступных каталогов в домашнем каталоге с помощью команды" find ~ -type d " (рис. - @рис.017).

FIND(1)

General Commands Manual

FIND(1)

NAME

find - search for files in a directory hierarchy

SYNOPSIS

find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [expression]

DESCRIPTION

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, the current directory is used. If no expression is given, the expression **print** is used (but you should probably consider using **print0** instead, anyway).

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

OPTIONS

The **-H**, **-L** and **-P** options control the treatment of symbolic links. Command-line arguments following these are taken to be names of files or directories to be examined, up to the first argument that begins with **-**, or the argument **(** or **{**. That argument and any following arguments are taken to be the expression describing what is to be searched for. If no paths are given, the current directory is used. If no expression is given, the expression **print** is used (but you should probably consider using **print0** instead, anyway).

This manual page talks about 'options' within the expression list. These options control the behaviour of **find** but are specified immediately after the last path name. The five 'real' options **-H**, **-L**, **-P**, **-D** and **-O** must appear before the first path name, if at all. A double dash **--** can also be used to signal that any remaining arguments are not options (though ensuring that all start points begin with either **./** or **/** is generally safer if you use wildcards in the list of start points).

-P Never follow symbolic links. This is the default behaviour. When **find** examines or prints information a file, and the file is a symbolic link, the information used shall be taken from the properties of the symbolic link itself.

-L Follow symbolic links. When **find** examines or prints information about files, the information used shall be taken from the properties of the file to which the link points, not from the link itself (unless it is a broken symbolic link or **find** is unable to examine the file to which the link points). Use of this option implies **-noleaf**. If you later use the **-P** option, **-noleaf** will still be in effect. If **-L** is in effect and **find** discovers a symbolic link to a subdirectory during its search, the subdirectory pointed to by the symbolic link will be searched.

When the **-L** option is in effect, the **-type** predicate will always match against the type of the file that a symbolic link points to rather than the link itself (unless the symbolic link is broken). Actions that can cause symbolic links to become broken while **find** is executing (for example **-delete**) can give rise to confusing behaviour. Using **-L** causes the **-lname** and **-ilname** predicates always to return false.

-H Do not follow symbolic links, except while processing the command line arguments. When **find** examines or prints information about files, the information used shall be taken from the properties of the symbolic link itself. The only exception to this behaviour is when a file specified on the command line is a symbolic link, and the link can be resolved. For that situation, the information used is taken from whatever the link points to (that is, the link is followed). The information about the link itself is used as a fallback if the file pointed to by the symbolic link cannot be examined. If **-H** is in effect and one of the paths specified on the command line is a symbolic link to a directory, the contents of that directory will be examined (though of course **-maxdepth 0** would prevent this).

If more than one of **-H**, **-L** and **-P** is specified, each overrides the others; the last one appearing on the command line takes effect. Since it is the default, the **-P** option should be considered to be in effect unless either **-H** or **-L** is specified.

GNU **find** frequently stats files during the processing of the command line itself, before any searching has begun. These options also affect how those arguments are processed. Specifically, there are a number of tests that compare files listed on the command line against a file we are currently considering. In each case, the file specified on the command line will have been examined and some of its properties will have been saved. If the named file is in fact a symbolic link, and the **-P** option is in effect (or if neither **-H** nor **-L** were specified), the information used for the comparison will be taken from the properties of the symbolic link. Otherwise, it will be taken from the properties of the file the link points to. If **find** cannot follow the link (for example because it has insufficient privileges or the link points to a nonexistent file) the properties of the link itself will be used.

When the **-H** or **-L** options are in effect, any symbolic links listed as the argument of **-newer** will be dereferenced, and the timestamp will be taken from the file to which the symbolic link points. The same consideration applies to **-newerXY**, **-anewer** and **-cnewer**.

The **-follow** option has a similar effect to **-L**, though it takes effect at the point where it appears (that is, if **-L** is not used but **-follow** is, any symbolic links appearing after **-follow** on the command line will be dereferenced, and those before it will not).

-D debugopts

Print diagnostic information; this can be helpful to diagnose problems with why **find** is not doing what you want. The list of debug options should be comma separated. Compatibility

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

{ **#fig:0015** width=70% }

<pre> fernando@fernando-VirtualBox:~\$ man find fernando@fernando-VirtualBox:~\$ find ~ -type d /home/fernando /home/fernando/docs /home/fernando/monthly /home/fernando/monthly.00 /home/fernando/monthly.00/monthly /home/fernando/Desktop /home/fernando/Desktop/etc /home/fernando/Desktop/etc/vbox /home/fernando/Desktop/usr /home/fernando/Desktop/usr/share /home/fernando/Desktop/usr/share/virtualbox /home/fernando/Desktop/usr/share/virtualbox/nls /home/fernando/Desktop/usr/share/virtualbox/rdesktop-vrdp-keymaps /home/fernando/Desktop/usr/share/virtualbox/src /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/linux /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/common /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/common/string /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include/iprt /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include/iprt/linux /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include/iprt/nocrt /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include/internal /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include/VBox /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/include/VBox/vmm /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/r0drv /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/r0drv/linux /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/math /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxnetadp/math/gcc /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/linux /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/common /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/common/checksum /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/common/table /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/common/string /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/common/log /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/common/misc /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/common/err /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/common/alloc /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/generic /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/include /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/include/iprt /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/include/iprt/linux /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/include/iprt/nocrt /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/include/internal /home/fernando/Desktop/usr/share/virtualbox/src/vboxhost/vboxdrv/include/VBox </pre>

{ **#fig:0015** width=70% }

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.
Программа 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. 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.

7. `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 дамп памяти. Сочетание клавиш `Ctrl+/\`;
 - `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 `killall` – 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 утилитами. 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.

Выводы

Во время этой лабораторной работы я изучил инструменты поиска файлов и фильтрации текстовых данных, такие как практические навыки: управление процессами (и задания), проверка использования диска и обслуживание файловой системы.