

Отчет по лабораторной работе №6

Дисциплина: операционные системы

Королев Федор Константинович

Содержание

Цель работы.....	1
Ход работы.....	1
Контрольные вопросы.....	10
Вывод.....	14

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.

Ход работы

1. Осуществим вход в систему, используя соответствующее имя пользователя
2. Запишем в файл file.txt название файлов, содержащихся в каталоге /etc и в домашнем каталоге(Рис. 1):

```
[liveuser@localhost-live ~]$ ls /etc > file.txt
[liveuser@localhost-live ~]$ ls >> file.txt
[liveuser@localhost-live ~]$ cat file.txt
abrt
adjtime
aliases
alsa
alternatives
anaconda
anthy-unicode.conf
appstream.conf
asound.conf
audit
authselect
avahi
bash_completion.d
bashrc
bindresvport.blacklist
binfmt.d
bluetooth
```

Рис. 1 запись содержимого /etc и домашнего каталога в file.txt

3. Выведем имена всех файлов из file.txt, имеющих расширение .conf, после чего запишем их в новый текстовый файл conf.txt(Рис. 2):

```
[liveuser@localhost-live ~]$ grep -e '\.conf$' file.txt > conf.txt
[liveuser@localhost-live ~]$ ls
conf.txt Desktop Documents Downloads file.txt Music Pictures Public Templates Videos
[liveuser@localhost-live ~]$ cat conf.txt
anthy-unicode.conf
appstream.conf
asound.conf
brltty.conf
chrony.conf
dley-na-renderer-service.conf
dley-na-server-service.conf
dnsmasq.conf
dracut.conf
extlinux.conf
fprintd.conf
fuse.conf
host.conf
idmapd.conf
jwhois.conf
kdump.conf
krb5.conf
ld.so.conf
libaudit.conf
libuser.conf
locale.conf
logrotate.conf
man_db.conf
mke2fs.conf
mtools.conf
nfs.conf
nfsmount.conf
nsswitch.conf
opensc.conf
opensc-x86_64.conf
passwdqc.conf
radvd.conf
request-key.conf
resolv.conf
rsyncd.conf
rygel.conf
sestatus.conf
sudo.conf
swtpm-localca.conf
swtpm_setup.conf
sysctl.conf
tcsd.conf
Trolltech.conf
updatedb.conf
uresourced.conf
```

Рис. 2 запись имен из file.txt с расширением .conf в файл conf.txt

4. С помощью команд

```
ls -a ~ | grep c*
```

```
ls ~/c*
```

Определим, какие файлы в домашнем каталоге начинаются с символа c(Рис. 3):

```
[liveuser@localhost-live ~]$ ls -a ~ | grep c*  
conf.txt  
[liveuser@localhost-live ~]$ ls ~/c*  
/home/liveuser/conf.txt
```

Рис. 3 файлы, начинающиеся с символа c

5. По странично выведем на экран имена файлов и каталога /etc, с помощью команды(Рис. 4 и 5)

```
find /etc -maxdepth 1 -name "h*" | less
```

```
[liveuser@localhost-live ~]$ find /etc -maxdepth 1 -name "h*" | less  
[liveuser@localhost-live ~]$
```

Рис. 4 команда выполнения

```
/etc/hostname  
/etc/hp  
/etc/httpd  
/etc/host.conf  
/etc/hosts  
(END)
```

Рис. 5 постраничный вывод

6. Запустим в фоновом режиме процесс, который будет записывать в файл logfile файлы, имена которых начинаются с log, с помощью команды(Рис. 6)

```
find / -name "log*" > logfile&
```

```
[liveuser@localhost-live ~]$ find / -name "log*" > logfile&
[1] 2784
find: '/var/spool/abrt-upload': Permission denied
find: '/var/spool/abrt': Permission denied
find: '/var/spool/cups': Permission denied
find: '/var/log/samba': Permission denied
find: '/var/log/libvirt': Permission denied
find: '/var/log/swtpm/libvirt/qemu': Permission denied
find: '/var/log/private': Permission denied
find: '/var/log/gdm': Permission denied
find: '/var/log/httpd': Permission denied
find: '/var/log/ppp': Permission denied
find: '/var/log/speech-dispatcher': Permission denied
find: '/var/log/audit': Permission denied
find: [liveuser@localhost-live ~]$ '/var/log/sss': Permission denied
find: '/var/log/chrony': Permission denied
find: '/var/lib/swtpm-localca': Permission denied
find: '/var/lib/samba/private': Permission denied
find: '/var/lib/openssh': Permission denied
find: '/var/lib/libvirt/swtpm': Permission denied
find: '/var/lib/libvirt/images': Permission denied
find: '/var/lib/libvirt/network': Permission denied
find: '/var/lib/libvirt/filesystems': Permission denied
find: '/var/lib/libvirt/boot': Permission denied
find: '/var/lib/libvirt/qemu': Permission denied
find: '/var/lib/private': Permission denied
find: '/var/lib/gdm': Permission denied
find: '/var/lib/polkit-1': Permission denied
find: '/var/lib/NetworkManager': Permission denied
find: '/var/lib/gssproxy/rcache': Permission denied
find: '/var/lib/gssproxy/clients': Permission denied
find: '/var/lib/colord/.cache': Permission denied
find: '/var/lib/AccountsService/users': Permission denied
find: '/var/lib/machines': Permission denied
find: '/var/lib/httpd': Permission denied
find: '/var/lib/sss/desktopfile': Permission denied
find: '/var/lib/sss/secrets': Permission denied
find: '/var/lib/sss/pipes/private': Permission denied
find: '/var/lib/sss/keytabs': Permission denied
find: '/var/lib/sss/db': Permission denied
find: '/var/lib/selinux/targeted/active': Permission denied
find: '/var/lib/selinux/final': Permission denied
find: '/var/lib/portables': Permission denied
find: '/var/lib/mlocate': Permission denied
find: '/var/lib/tpm': Permission denied
find: '/var/lib/nfs/statd': Permission denied
find: '/var/lib/fprint': Permission denied
find: '/var/lib/udisks2': Permission denied
find: '/var/lib/chrony': Permission denied
```

Рис. 6 запуск и исполнение программы в фоновом режиме

7. Удалим файл logfile(Рис. 7):

```
[liveuser@localhost-live ~]$ rm logfile
[liveuser@localhost-live ~]$ ls
conf.txt  Desktop  Documents  Downloads  file.txt  Music  Pictures  Public  Templates  Videos
```

Рис. 7 удаление файла logfile

8. Запустим редактор gedit в фоновом режиме, с помощью команды(Рис. 8): gedit&

```
[liveuser@localhost-live ~]$ gedit&
[1] 2847
```

Рис. 8 запуск *gedit* в фоновом режиме

9. Определим идентификатор процесса с помощью команды (Рис. 9)
`ps | grep -i "gedit"`

Видно, что он равен 2847. Также, идентификатор процесса можно определить с помощью команд (Рис. 9)

```
pgrep gedit
```

```
pidof gedit
```

```
[liveuser@localhost-live ~]$ ps | grep -i "gedit"
2847 pts/0    00:00:00 gedit
[liveuser@localhost-live ~]$ pgrep gedit
2847
[liveuser@localhost-live ~]$ pidof gedit
2847
```

Рис. 9 определение идентификатора *gedit*

10. Прочитаем справку по команде `kill` (с помощью `man`, Рис. 10), далее завершим процесс `gedit` (Рис. 11):

```
kill(1)                                User Commands                                kill(1)
NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|--signal=-p] [-q value] [--timeout milliseconds signal] [--] pid|name...
    kill -l [number] | -t

DESCRIPTION
    The command kill sends the specified signal to the specified processes or process groups.

    If no signal is specified, the TERM signal is sent. The default action for this signal is to terminate the process. This signal should be used in preference to the KILL signal (number 9), since a process may install a handler for the TERM signal in order to perform clean-up steps before terminating in an orderly fashion. If a process does not terminate after a TERM signal has been sent, then the KILL signal may be used; be aware that the latter signal cannot be caught, and so does not give the target process the opportunity to perform any clean-up before terminating.

    Most modern shells have a builtin kill command, with a usage rather similar to that of the command described here. The --all, --pid, and --queue options, and the possibility to specify processes by command name, are local extensions.

    If signal is 0, then no actual signal is sent, but error checking is still performed.

ARGUMENTS
    The list of processes to be signaled can be a mixture of names and PIDs.

    pid
        Each pid can be expressed in one of the following ways:

        0
            where g is larger than 0. The process with PID g is signaled.

        0
            All processes in the current process group are signaled.

        -1
            All processes with a PID larger than 1 are signaled.

        -n
            where g is larger than 1. All processes in process group g are signaled. When an argument of the form -n is given, and it is meant to denote a process group, either a signal must be specified first, or the argument must be preceded by a -- option, otherwise it will be taken as the signal to send.

    name
        All processes invoked using this name will be signaled.

OPTIONS
    -s, --signal signal
        The signal to send. It may be given as a name or a number.
```

Рис. 10 `man kill`

```
[liveuser@localhost-live ~]$ kill %gedit
[1]+  Terminated                  gedit
```

Рис. 11 завершение процесса *gedit*

11. Выполним команды `df` и `du`, предварительно получив более подробную с помощью `man` (Рис. 12, 13, 14):

```

00(1) User Commands 00(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

    --direct
        show statistics for a file instead of mount point

    -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

    -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.

```

Puc. 12 man df

```

00(1) User Commands 00(1)
NAME
    du - estimate file space usage

SYNOPSIS
    du [OPTION]... [FILE]...
    du [OPTION]... --files-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.

    -B, --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

    --files-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

```

Puc. 13 man du

```

[liveuser@localhost-live ~]$ df
Filesystem            1K-blocks      Used Available Use% Mounted on
devtmpfs               1372492         0    1372492   0% /dev
tmpfs                  1406000         0    1406000   0% /dev/shm
tmpfs                   562404        8348    554056   2% /run
/dev/sr0               1962240 1962240         0 100% /run/initramfs/live
/dev/mapper/live-rw    7640588 5463932    2160272  72% /
tmpfs                  1406004         72    1405932   1% /tmp
vartmp                 1406000         0    1406000   0% /var/tmp
tmpfs                   281200        128    281072   1% /run/user/1000
[liveuser@localhost-live ~]$ du
4      ./mozilla/extensions
4      ./mozilla/plugins
12     ./mozilla
8      ./config/gtk-3.0
4      ./config/enchant
8      ./config/gedit
8      ./config/dconf
8      ./config/evolution/sources
12     ./config/evolution
4      ./config/goa-1.0
8      ./config/autostart
4      ./config/abrt
8      ./config/ibus/bus
12     ./config/ibus
8      ./config/pulse
88     ./config
40     ./local/share/gvfs-metadata
4      ./local/share/icc
4      ./local/share/keyrings
4      ./local/share/flatpak/db
8      ./local/share/flatpak
4      ./local/share/sounds
4      ./local/share/applications
4      ./local/share/evolution/addressbook/system/photos
92     ./local/share/evolution/addressbook/system
4      ./local/share/evolution/addressbook/trash
100    ./local/share/evolution/addressbook
4      ./local/share/evolution/tasks/system
4      ./local/share/evolution/tasks/trash
12     ./local/share/evolution/tasks
4      ./local/share/evolution/calendar/system
4      ./local/share/evolution/calendar/trash
12     ./local/share/evolution/calendar
4      ./local/share/evolution/memos/trash
8      ./local/share/evolution/memos
4      ./local/share/evolution/mail/trash
8      ./local/share/evolution/mail

```

Рис. 14 выполнение df и часть du

12. Выведе имена всех директорий, имеющих в домашнем каталоге (Рис. 16-18), предварительно посмотрев информацию о find, с помощью команды `find ~ -type d`

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		
<p>This manual page documents the GNU version of find. GNU find searches the directory tree rooted at each given starting-point by evaluating the given expression from left to right, according to the rules of precedence (see section OPERATORS), until the outcome is known (the left hand side is false for and operations, true for or), at which point find moves on to the next file name. If no starting-point is specified, '.' is assumed.</p> <p>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.</p>		
OPTIONS		
<p>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).</p> <p>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 -- could theoretically be used to signal that any remaining arguments are not options, but this does not really work due to the way find determines the end of the following path arguments: it does that by reading until an expression argument comes (which also starts with a '-'). Now, if a path argument would start with a '-', then find would treat it as expression argument instead. Thus, to ensure that all start points are taken as such, and especially to prevent that wildcard patterns expanded by the calling shell are not mistakenly treated as expression arguments, it is generally safer to prefix wildcards or dubious path names with either './' or to use absolute path names starting with '/'. </p> <p>-P Never follow symbolic links. This is the default behaviour. When find examines or prints information about files, and the file is a symbolic link, the information used shall be taken from the properties of the symbolic link itself.</p> <p>-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.</p> <p>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 -flname predicates always to return false.</p> <p>-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).</p> <p>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.</p> <p>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 me 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</p>		

Puc. 15 man find


```
[liveuser@localhost-live ~]$ man find
[liveuser@localhost-live ~]$ find ~ -type d
/home/liveuser
/home/liveuser/.mozilla
/home/liveuser/.mozilla/extensions
/home/liveuser/.mozilla/plugins
/home/liveuser/.config
/home/liveuser/.config/gtk-3.0
/home/liveuser/.config/enchant
/home/liveuser/.config/gedit
/home/liveuser/.config/dconf
/home/liveuser/.config/evolution
/home/liveuser/.config/evolution/sources
/home/liveuser/.config/goa-1.0
/home/liveuser/.config/autostart
/home/liveuser/.config/abrt
/home/liveuser/.config/ibus
/home/liveuser/.config/ibus/bus
/home/liveuser/.config/pulse
/home/liveuser/.local
/home/liveuser/.local/share
/home/liveuser/.local/share/gvfs-metadata
/home/liveuser/.local/share/icc
/home/liveuser/.local/share/keyrings
/home/liveuser/.local/share/flatpak
/home/liveuser/.local/share/flatpak/db
/home/liveuser/.local/share/sounds
/home/liveuser/.local/share/applications
/home/liveuser/.local/share/evolution
/home/liveuser/.local/share/evolution/addressbook
/home/liveuser/.local/share/evolution/addressbook/system
/home/liveuser/.local/share/evolution/addressbook/system/photos
/home/liveuser/.local/share/evolution/addressbook/trash
/home/liveuser/.local/share/evolution/tasks
/home/liveuser/.local/share/evolution/tasks/system
/home/liveuser/.local/share/evolution/tasks/trash
/home/liveuser/.local/share/evolution/calendar
/home/liveuser/.local/share/evolution/calendar/system
/home/liveuser/.local/share/evolution/calendar/trash
/home/liveuser/.local/share/evolution/memos
/home/liveuser/.local/share/evolution/memos/trash
/home/liveuser/.local/share/evolution/mail
/home/liveuser/.local/share/evolution/mail/trash
/home/liveuser/.local/share/gnome-settings-daemon
/home/liveuser/.local/share/pki
/home/liveuser/.local/share/pki/nssdb
/home/liveuser/.local/share/gnome-shell
/home/liveuser/.local/state
/home/liveuser/.local/state/wireplumber
```

```

/home/liveuser/Music
/home/liveuser/Templates
/home/liveuser/Videos
/home/liveuser/.cache
/home/liveuser/.cache/mesa_shader_cache
/home/liveuser/.cache/mesa_shader_cache/ff
/home/liveuser/.cache/mesa_shader_cache/19
/home/liveuser/.cache/mesa_shader_cache/d6
/home/liveuser/.cache/mesa_shader_cache/73
/home/liveuser/.cache/mesa_shader_cache/84
/home/liveuser/.cache/mesa_shader_cache/68
/home/liveuser/.cache/mesa_shader_cache/f7
/home/liveuser/.cache/mesa_shader_cache/a8
/home/liveuser/.cache/mesa_shader_cache/82
/home/liveuser/.cache/mesa_shader_cache/8e
/home/liveuser/.cache/mesa_shader_cache/44
/home/liveuser/.cache/mesa_shader_cache/ed
/home/liveuser/.cache/mesa_shader_cache/29
/home/liveuser/.cache/mesa_shader_cache/63
/home/liveuser/.cache/mesa_shader_cache/9a
/home/liveuser/.cache/mesa_shader_cache/90
/home/liveuser/.cache/mesa_shader_cache/81
/home/liveuser/.cache/mesa_shader_cache/fa
/home/liveuser/.cache/mesa_shader_cache/88
/home/liveuser/.cache/mesa_shader_cache/91
/home/liveuser/.cache/mesa_shader_cache/a1
/home/liveuser/.cache/mesa_shader_cache/ca
/home/liveuser/.cache/mesa_shader_cache/6e
/home/liveuser/.cache/mesa_shader_cache/2f
/home/liveuser/.cache/mesa_shader_cache/0d
/home/liveuser/.cache/mesa_shader_cache/ac
/home/liveuser/.cache/mesa_shader_cache/f2
/home/liveuser/.cache/mesa_shader_cache/69
/home/liveuser/.cache/mesa_shader_cache/a7
/home/liveuser/.cache/mesa_shader_cache/a5
/home/liveuser/.cache/mesa_shader_cache/74
/home/liveuser/.cache/mesa_shader_cache/83
/home/liveuser/.cache/mesa_shader_cache/26
/home/liveuser/.cache/mesa_shader_cache/bc
/home/liveuser/.cache/mesa_shader_cache/1c
/home/liveuser/.cache/mesa_shader_cache/52
/home/liveuser/.cache/mesa_shader_cache/1b
/home/liveuser/.cache/mesa_shader_cache/dc
/home/liveuser/.cache/mesa_shader_cache/ba
/home/liveuser/.cache/mesa_shader_cache/21
/home/liveuser/.cache/mesa_shader_cache/15
/home/liveuser/.cache/mesa_shader_cache/d2
/home/liveuser/.cache/mesa_shader_cache/1d
/home/liveuser/.cache/mesa_shader_cache/77
/home/liveuser/.cache/mesa_shader_cache/e7
/home/liveuser/.cache/mesa_shader_cache/e8
/home/liveuser/.cache/mesa_shader_cache/72
/home/liveuser/.cache/mesa_shader_cache/d9
/home/liveuser/.cache/mesa_shader_cache/6c
/home/liveuser/.cache/mesa_shader_cache/f9
/home/liveuser/.cache/mesa_shader_cache/67
/home/liveuser/.cache/mesa_shader_cache/32
/home/liveuser/.cache/mesa_shader_cache/df
/home/liveuser/.cache/mesa_shader_cache/20
/home/liveuser/.cache/mesa_shader_cache/4f
/home/liveuser/.cache/mesa_shader_cache/8f
/home/liveuser/.cache/mesa_shader_cache/1f
/home/liveuser/.cache/mesa_shader_cache/ef
/home/liveuser/.cache/mesa_shader_cache/3c
/home/liveuser/.cache/mesa_shader_cache/64
/home/liveuser/.cache/mesa_shader_cache/10
/home/liveuser/.cache/mesa_shader_cache/03
/home/liveuser/.cache/mesa_shader_cache/e3
/home/liveuser/.cache/mesa_shader_cache/37
/home/liveuser/.cache/mesa_shader_cache/c0
/home/liveuser/.cache/mesa_shader_cache/ad
/home/liveuser/.cache/mesa_shader_cache/dd
/home/liveuser/.cache/mesa_shader_cache/6b
/home/liveuser/.cache/evolution
/home/liveuser/.cache/evolution/addressbook
/home/liveuser/.cache/evolution/addressbook/trash
/home/liveuser/.cache/evolution/sources
/home/liveuser/.cache/evolution/sources/trash
/home/liveuser/.cache/evolution/tasks
/home/liveuser/.cache/evolution/tasks/trash
/home/liveuser/.cache/evolution/calendar
/home/liveuser/.cache/evolution/calendar/trash
/home/liveuser/.cache/evolution/memos
/home/liveuser/.cache/evolution/memos/trash
/home/liveuser/.cache/evolution/mail
/home/liveuser/.cache/evolution/mail/trash
/home/liveuser/.cache/abrt
/home/liveuser/.cache/ibus
/home/liveuser/.cache/gstreamer-1.0
/home/liveuser/Downloads
/home/liveuser/Documents
/home/liveuser/Desktop
/home/liveuser/Public
/home/liveuser/Pictures

```

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:

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+/\;
- 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 (вканале) и будет выполнять поиск по результатам команды ps.

Утилита `ckill` – 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.

Вывод

В ходе данной лабораторной работы я научился перенаправлять ввод, работать с конвейером, искать файлы, фильтровать текст, а также проверять использование диска и управлять процессами, получать о них информацию.