

Отчёт по лабораторной работе №4

Дисциплина: Операционные системы

Дупленских Василий Викторович

Содержание

Цель работы:	5
Выполнение лабораторной работы:	6
1. Определяю полное имя моего домашнего каталога и перехожу в этот каталог:	6
2. Перехожу в каталог /tmp, вывожу его содержание, определяю есть ли подкаталог cron в каталоге /var/spool. Перехожу в мой домашний каталог и вывожу его содержимое. Определяю, владелец файлов и подкаталогов:	8
3. В домашнем каталоге создаю новый катлог с именем newdir. В этом каталоге создаю каталог с именем morefun. В домашнем каталоге создаю три новых каталога с именами letters, memos, misk. Затем удаляю эти каталоги одной командой. Пробую удалить ранее созданный каталог ~/newdir командой rm. Проверяю, удален ли каталог. Удаляю каталог ~/newdir/morefun из домашнего каталога. Проверяю, удален ли каталог:	10
4. С помощью команды man определяю, какую опцию команды ls надо использовать для просмотра содержимого не только указанного каталога но и подкаталогов, входящих в него.	11
5. С помощью команды man определяю, какую опцию команды ls надо использовать для сортировки по времени последнего изменения выводимого списка содержимого каталога с развёрнутым описанием файлов.	12
6. Использую команду man для просмотра описания следующих команд: cd, pwd, mkdir, rmdir, rm. Поясняю основные опции этих команд. Cd - команда перемещения по файловой системе. Pwd - показывает директорию в которой находится пользователь. Mkdir - команда, создающая новую директорию. Rmdir - команда, которая удаляет директорию. Rm - команда, которая удаляет файлы.	14
7. Используя информацию, полученную при помощи команды history, выполняю модификацию и исполнение нескольких команд из буфера команд.	18
Вывод:	21

Список иллюстраций

0.1	home	6
0.2	Просмотр всего и внутри и снаружи каталогов!	11
0.3	Сортировка файлов внутри каталогов(ls)	12

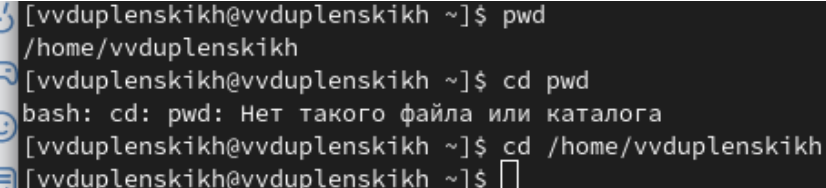
Список таблиц

Цель работы:

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Выполнение лабораторной работы:

1. Определяю полное имя моего домашнего каталога и перехожу в этот каталог:

A terminal window with a dark background and light blue text. The prompt is [vvduplenskikh@vvduplenskikh ~]\$. The first command is pwd, which outputs /home/vvduplenskikh. The second command is cd pwd, which outputs an error message: bash: cd: pwd: Нет такого файла или каталога. The third command is cd /home/vvduplenskikh, which successfully changes the directory. The prompt now shows [vvduplenskikh@vvduplenskikh ~]\$.

```
[vvduplenskikh@vvduplenskikh ~]$ pwd
/home/vvduplenskikh
[vvduplenskikh@vvduplenskikh ~]$ cd pwd
bash: cd: pwd: Нет такого файла или каталога
[vvduplenskikh@vvduplenskikh ~]$ cd /home/vvduplenskikh
[vvduplenskikh@vvduplenskikh ~]$
```

Рис. 0.1: home

2. Перехожу в каталог /tmp, вывожу его содержание, определяю есть ли подкаталог cron в каталоге /var/spool.

Перехожу в мой домашний каталог и вывожу его содержимое. Определяю, владелец файлов и подкаталогов:

```
[vvduplenskikh@vvduplenskikh ~]$ cd /tmp
[vvduplenskikh@vvduplenskikh tmp]$ ls
nsperfdata_vvduplenskikh
systemd-private-fec33acfaf49476a85b0c4277c9eb436-chronyd.service-3D9Teo
systemd-private-fec33acfaf49476a85b0c4277c9eb436-colord.service-r100bp
systemd-private-fec33acfaf49476a85b0c4277c9eb436-dbus-broker.service-iEIm86
systemd-private-fec33acfaf49476a85b0c4277c9eb436-low-memory-monitor.service-FZAZ
systemd-private-fec33acfaf49476a85b0c4277c9eb436-ModemManager.service-hKwsHw
systemd-private-fec33acfaf49476a85b0c4277c9eb436-power-profiles-daemon.service-S
systemd-private-fec33acfaf49476a85b0c4277c9eb436-rtkit-daemon.service-ou45QF
systemd-private-fec33acfaf49476a85b0c4277c9eb436-switcheroo-control.service-ajbe
systemd-private-fec33acfaf49476a85b0c4277c9eb436-systemd-logind.service-tr7skV
systemd-private-fec33acfaf49476a85b0c4277c9eb436-systemd-oemd.service-0aHpbR
systemd-private-fec33acfaf49476a85b0c4277c9eb436-systemd-resolved.service-d33ULk
systemd-private-fec33acfaf49476a85b0c4277c9eb436-upower.service-PJ6jYC
temp-f4c92a7b-efac-4738-be5e-41fc85aa0448
[vvduplenskikh@vvduplenskikh tmp]$ cd /var/spool
[vvduplenskikh@vvduplenskikh spool]$ ls
abrt abrt-upload cups lpd mail plymouth
```

```
[vvduplenskikh@vvduplenskikh ~]$ ls -la
итого 353136
drwx----- 1 vvduplenskikh vvduplenskikh 1464 июн 21 10:28 .
drwxr-xr-x 1 root root 106 мая 7 22:33 ..
drwxrwxr-x 1 vvduplenskikh vvduplenskikh 0 апр 30 13:33 28
drwxrwxr-x 1 vvduplenskikh vvduplenskikh 46 мая 25 16:50 backup
--wx-wx-wx 1 vvduplenskikh vvduplenskikh 112 мая 21 19:35 backup.sh
-rw-rw-r-- 1 vvduplenskikh vvduplenskikh 0 мая 21 19:32 backup.sh~
-rw----- 1 vvduplenskikh vvduplenskikh 12022 июн 21 10:51 .bash_history
-rw-r--r-- 1 vvduplenskikh vvduplenskikh 18 июл 21 2021 .bash_logout
-rw-r--r-- 1 vvduplenskikh vvduplenskikh 141 июл 21 2021 .bash_profile
-rw-r--r-- 1 vvduplenskikh vvduplenskikh 492 июл 21 2021 .bashrc
drwxr-xr-x 1 vvduplenskikh vvduplenskikh 8 мая 14 20:57 bin
drwxrwxr-x 1 vvduplenskikh vvduplenskikh 38 мая 14 20:42 .cabal
drwx----- 1 vvduplenskikh vvduplenskikh 530 июн 4 21:42 .cache
drwxrwxr-x 1 vvduplenskikh vvduplenskikh 170 июн 4 21:42 calculate
drwxr-xr-x 1 vvduplenskikh vvduplenskikh 420 июн 4 21:40 .config
-rw-rw-r-- 1 vvduplenskikh vvduplenskikh 50 мая 14 15:14 cplus
drwx----- 1 vvduplenskikh vvduplenskikh 28 мая 20 13:27 .emacs.d
-rwxrwxr-x 1 vvduplenskikh vvduplenskikh 236 мая 21 19:50 format.sh
-rw-rw-r-- 1 vvduplenskikh vvduplenskikh 0 мая 21 19:48 format.sh~
-rw-rw-r-- 1 vvduplenskikh vvduplenskikh 469 июн 21 10:28 .gitconfig
drwx----- 1 vvduplenskikh vvduplenskikh 216 июн 21 10:45 .gnupg
-rwxrwxr-x 1 vvduplenskikh vvduplenskikh 173 мая 21 19:36 lab07.sh
-rw-rw-r-- 1 vvduplenskikh vvduplenskikh 97 мая 20 13:30 lab07.sh~
-rw----- 1 vvduplenskikh vvduplenskikh 20 мая 25 17:11 .lessht
drwx----- 1 vvduplenskikh vvduplenskikh 20 апр 22 16:03 .local
--wx-wx-wx 1 vvduplenskikh vvduplenskikh 411 мая 21 19:46 ls.sh
```


3. В домашнем каталоге создаю новый катлог с именем newdir. В этом каталоге создаю каталог с именем morefun. В домашнем каталоге создаю три новых каталога с именами letters, memos, misk. Затем удаляю эти каталоги одной командой. Пробую удалить ранее созданный каталог ~/newdir командой rm. Проверяю, удален ли каталог. Удаляю каталог ~/newdir/morefun из домашнего каталога. Проверяю, удален ли каталог:

```
[vvduplenskikh@vvduplenskikh ~]$ cd ~
[vvduplenskikh@vvduplenskikh ~]$ md newdir
bash: md: command not found...
[vvduplenskikh@vvduplenskikh ~]$ mkdir newdir
[vvduplenskikh@vvduplenskikh ~]$ ls
28      '#new3#'      random.sh
backup  '#new4#'      random.sh~
backup.sh newdir        semafor.sh
backup.sh~ prog11.1.sh  semafor.sh~
bin     prog11.1.sh~  text
calculate prog11.2      vvduplenskikh.tar
cplus   prog11.2.c    work
format.sh prog11.2.c~   Видео
format.sh~ prog11.2.sh   Документы
lab07.sh prog11.2.sh~  Загрузки
lab07.sh~ prog11.3.sh   Изображения
ls.sh    prog11.3.sh~  Музыка
ls.sh~   prog11.4.sh   Общедоступные
man.sh   prog11.4.sh~  'Рабочий стол'
man.sh~  prog2.sh      'Снимок экрана от 2022-04-23 20-47-49.png'
'#new2#' prog2.sh~     Шаблоны
[vvduplenskikh@vvduplenskikh ~]$ cd newdir
[vvduplenskikh@vvduplenskikh newdir]$ mkdir morefun
[vvduplenskikh@vvduplenskikh newdir]$ cd ~
[vvduplenskikh@vvduplenskikh ~]$ mkdir letters memos misk
[vvduplenskikh@vvduplenskikh ~]$ ls
28      misk      prog2.sh~
backup  '#new2#'      random.sh
backup.sh '#new3#'      random.sh~
backup.sh~ '#new4#'      semafor.sh
bin       newdir        semafor.sh~
calculate prog11.1.sh  text
cplus     prog11.1.sh~ vvduplenskikh.tar
format.sh prog11.2      work
format.sh~ prog11.2.c    Видео
lab07.sh  prog11.2.c~   Документы
lab07.sh~ prog11.2.sh   Загрузки
letters   prog11.2.sh~  Изображения
ls.sh     prog11.3.sh   Музыка
ls.sh~    prog11.3.sh~  Общедоступные
man.sh    prog11.4.sh   'Рабочий стол'
man.sh~   prog11.4.sh~  'Снимок экрана от 2022-04-23 20-47-49.png'
memos     prog2.sh      Шаблоны
```

4. С помощью команды `man` определяю, какую опцию команды `ls` надо использовать для просмотра содержимого не только указанного каталога но и подкаталогов, входящих в него.

```
[vvduplenskikh@vvduplenskikh ~]$ man ls
[vvduplenskikh@vvduplenskikh ~]$ ls -r
Шаблоны
'Снимок экрана от 2022-04-23 20-47-49.png'
'Рабочий стол'
Общедоступные
Музыка
Изображения
Загрузки
Документы
Видео
work
vvduplenskikh.tar
text
semafor.sh~
semafor.sh
random.sh~
random.sh
prog2.sh~
prog2.sh
prog11.4.sh~
prog11.4.sh
prog11.3.sh~
prog11.3.sh
prog11.2.sh~
prog11.2.sh
prog11.2.c~
prog11.2.c
prog11.2
prog11.1.sh~
prog11.1.sh
newdir
'#new4#'
'#new3#'
'#new2#'
man.sh~
man.sh
ls.sh~
ls.sh
lab07.sh~
lab07.sh
format.sh~
format.sh
cplus
calculate
bin
backup.sh~
backup.sh
backup
28
```

Рис. 0.2: Просмотр всего и внутри и снаружи каталогов!

5. С помощью команды `man` определяю, какую опцию команды `ls` надо использовать для сортировки по времени последнего изменения выводимого списка содержимого каталога с развёрнутым описанием файлов.

```
[vvduplenskikh@vvduplenskikh ~]$ ls -t
Изображения      prog11.2          '#new3#'
newdir            prog11.2.sh       '#new4#'
Загрузки          prog11.2.c        '#new2#'
calculate         prog11.2.c~       lab07.sh~
random.sh         prog11.2.sh~      work
random.sh~        prog11.1.sh       bin
man.sh            prog11.1.sh~      cplus
man.sh~           format.sh         text
semafor.sh        format.sh~        Шаблоны
semafor.sh~       ls.sh             28
backup            ls.sh~            'Снимок экрана от 2022-04-23 20-47-49.png'
vvduplenskikh.tar prog2.sh          Видео
prog11.4.sh       prog2.sh~         Документы
prog11.4.sh~      lab07.sh          Музыка
prog11.3.sh       backup.sh         Общедоступные
prog11.3.sh~      backup.sh~        'Рабочий стол'
```

Рис. 0.3: Сортировка файликов внутри каталогов(`ls`)

6. Использую команду `man` для просмотра описания следующих команд: `cd`, `pwd`, `mkdir`, `rmdir`, `rm`. Поясняю основные опции этих команд. `Cd` - команда перемещения по файловой системе. `Pwd` - показывает директорию в которой находится пользователь. `Mkdir` - команда, создающая новую директорию. `Rmdir` - команда, которая удаляет директорию. `Rm` - команда, которая удаляет файлы.

```
BASH_BUILTINS(1)                                General Commands Manual                                BASH_BUILTINS(1)

NAME
bash, :, ., [, alias, bg, bind, break, builtin, caller, cd, command, compgen, complete, compopt, continue, declare, dirs, disown,
echo, enable, eval, exec, exit, export, false, fc, fg, getopts, hash, help, history, jobs, kill, let, local, logout, mapfile,
popd, printf, pushd, pwd, read, readonly, return, set, shift, shopt, source, suspend, test, times, trap, true, type, typeset,
ulimit, umask, unalias, unset, wait - bash built-in commands, see bash(1)

BASH BUILTIN COMMANDS
Unless otherwise noted, each builtin command documented in this section as accepting options preceded by - accepts -- to signify
the end of the options. The :, true, false, and test/[ builtins do not accept options and do not treat -- specially. The exit,
logout, return, break, continue, let, and shift builtins accept and process arguments beginning with - without requiring --.
Other builtins that accept arguments but are not specified as accepting options interpret arguments beginning with - as invalid
options and require -- to prevent this interpretation.
: [arguments]
    No effect; the command does nothing beyond expanding arguments and performing any specified redirections. The return sta-
    tus is zero.

. filename [arguments]
source filename [arguments]
    Read and execute commands from filename in the current shell environment and return the exit status of the last command
    executed from filename. If filename does not contain a slash, filenames in PATH are used to find the directory containing
    filename. The file searched for in PATH need not be executable. When bash is not in posix mode, the current directory is
    searched if no file is found in PATH. If the sourcepath option to the shopt builtin command is turned off, the PATH is
    not searched. If any arguments are supplied, they become the positional parameters when filename is executed. Otherwise
    the positional parameters are unchanged. If the -T option is enabled, source inherits any trap on DEBUG; if it is not,
    any DEBUG trap string is saved and restored around the call to source, and source unsets the DEBUG trap while it executes.
    If -T is not set, and the sourced file changes the DEBUG trap, the new value is retained when source completes. The re-
    turn status is the status of the last command exited within the script (0 if no commands are executed), and false if file-
    name is not found or cannot be read.

alias [-p] [name[=value] ...]
    Alias with no arguments or with the -p option prints the list of aliases in the form alias name=value on standard output.
    When arguments are supplied, an alias is defined for each name whose value is given. A trailing space in value causes the
    next word to be checked for alias substitution when the alias is expanded. For each name in the argument list for which
    no value is supplied, the name and value of the alias is printed. Alias returns true unless a name is given for which no
    alias has been defined.

bg [jobspec ...]
    Resume each suspended job jobspec in the background, as if it had been started with &. If jobspec is not present, the
    shell's notion of the current job is used. bg jobspec returns 0 unless run when job control is disabled or, when run with
    job control enabled, any specified jobspec was not found or was started without job control.

bind [-m keymap] [-lpsvPSVX]
bind [-m keymap] [-q function] [-u function] [-r keyseq]
bind [-m keymap] -f filename
bind [-m keymap] -x keyseq:shell-command
bind [-m keymap] keyseq:function-name
bind [-m keymap] keyseq:readline-command
    Display current readline key and function bindings, bind a key sequence to a readline function or macro, or set a readline
    variable. Each non-option argument is a command as it would appear in .inputrc, but each binding or command must be
    passed as a separate argument; e.g., '"\C-x\C-r": re-read-init-file'. Options, if supplied, have the following meanings:
    -m keymap
        Use keymap as the keymap to be affected by the subsequent bindings. Acceptable keymap names are emacs, emacs-stan-
        dard, emacs-meta, emacs-ctlx, vi, vi-move, vi-command, and vi-insert. vi is equivalent to vi-command (vi-move is
        also a synonym); emacs is equivalent to emacs-standard.
    -l
        List the names of all readline functions.
    -p
        Display readline function names and bindings in such a way that they can be re-read.
    -P
        List current readline function names and bindings.
    -s
        Display readline key sequences bound to macros and the strings they output in such a way that they can be re-read.
    -S
        Display readline key sequences bound to macros and the strings they output.

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

```
PWD(1)                                User Commands                                PWD(1)

NAME
pwd - print name of current/working directory

SYNOPSIS
pwd [OPTION]...

DESCRIPTION
Print the full filename of the current working directory.
```

```

MKDIR(1)                                     User Commands                               MKDIR(1)

NAME
    mkdir - make directories

SYNOPSIS
    mkdir [OPTION]... DIRECTORY...

DESCRIPTION
    Create the DIRECTORY(ies), if they do not already exist.

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

    -m, --mode=MODE
        set file mode (as in chmod), not a=rwx - umask

    -p, --parents
        no error if existing, make parent directories as needed

    -v, --verbose
        print a message for each created directory

    -Z
        set SELinux security context of each created directory to the default type

    --context[=CTX]
        like -Z, or if CTX is specified then set the SELinux or SMACK security context to CTX

    --help
        display this help and exit

    --version
        output version information and exit

AUTHOR
    Written by David MacKenzie.

REPORTING BUGS
    GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
    Report any translation bugs to <https://translationproject.org/team/>

COPYRIGHT
    Copyright © 2020 Free Software Foundation, Inc.  License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
    This is free software: you are free to change and redistribute it.  There is NO WARRANTY, to the extent permitted by law.

SEE ALSO
    mkdir(2)

    Full documentation <https://www.gnu.org/software/coreutils/mkdir>
    or available locally via: info '(coreutils) mkdir invocation'

GNU coreutils 8.32                               July 2021                               MKDIR(1)
Manual page mkdir(1) line 1/50 (END) (press h for help or q to quit)

```

```

RMDIR(1)                                User Commands                                RMDIR(1)

NAME
    rmdir - remove empty directories

SYNOPSIS
    rmdir [OPTION]... DIRECTORY...

DESCRIPTION
    Remove the DIRECTORY(ies), if they are empty.

    --ignore-fail-on-non-empty
        ignore each failure that is solely because a directory
        is non-empty

    -p, --parents
        remove DIRECTORY and its ancestors; e.g., 'rmdir -p a/b/c' is similar to 'rmdir a/b/c a/b a'

    -v, --verbose
        output a diagnostic for every directory processed

    --help
        display this help and exit

    --version
        output version information and exit

AUTHOR
    Written by David MacKenzie.

REPORTING BUGS
    GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
    Report any translation bugs to <https://translationproject.org/team/>

COPYRIGHT
    Copyright © 2020 Free Software Foundation, Inc.  License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
    This is free software: you are free to change and redistribute it.  There is NO WARRANTY, to the extent permitted by law.

SEE ALSO
    rmdir(2)

    Full documentation <https://www.gnu.org/software/coreutils/rmdir>
    or available locally via: info '(coreutils) rmdir invocation'

GNU coreutils 8.32                        July 2021                                RMDIR(1)
Manual page rmdir(1) line 1/46 (END) (press h for help or q to quit)

```



```

RM(1)                                     User Commands                                     RM(1)

NAME
  rm - remove files or directories

SYNOPSIS
  rm [OPTION]... [FILE]...

DESCRIPTION
  This manual page documents the GNU version of rm. rm removes each specified file. By default, it does not remove directories.

  If the -i or --interactive=once option is given, and there are more than three files or the -r, -R, or --recursive are given, then rm prompts the user for whether to proceed with the entire operation. If the response is not affirmative, the entire command is aborted.

  Otherwise, if a file is unwritable, standard input is a terminal, and the -f or --force option is not given, or the -i or --interactive=always option is given, rm prompts the user for whether to remove the file. If the response is not affirmative, the file is skipped.

OPTIONS
  Remove (unlink) the FILE(s).

  -f, --force
    ignore nonexistent files and arguments, never prompt

  -i
    prompt before every removal

  -I
    prompt once before removing more than three files, or when removing recursively; less intrusive than -i, while still giving protection against most mistakes

  --interactive[=WHEN]
    prompt according to WHEN: never, once (-I), or always (-i); without WHEN, prompt always

  --one-file-system
    when removing a hierarchy recursively, skip any directory that is on a file system different from that of the corresponding command line argument

  --no-preserve-root
    do not treat '/' specially

  --preserve-root[=all]
    do not remove '/' (default); with 'all', reject any command line argument on a separate device from its parent

  -r, -R, --recursive
    remove directories and their contents recursively

  -d, --dir
    remove empty directories

  -v, --verbose
    explain what is being done

  --help
    display this help and exit

  --version
    output version information and exit

  By default, rm does not remove directories. Use the --recursive (-r or -R) option to remove each listed directory, too, along with all of its contents.

  To remove a file whose name starts with a '-', for example '-foo', use one of these commands:

```

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

7. Используя информацию, полученную при помощи команды `history`, выполняю модификацию и исполнение нескольких команд из буфера команд.

```
[vduplenski@vduplenski ~]$ history
1 su -
2 sudo su-
3 sudo su -
4 sudo su -
5 ls
6 cd "Операционные системы"
7 ls
8 cd os-intro
9 git add .
10 git status
11 git commit -am "image3"
12 git commit -am "image"
13 git push
14 git status
15 git push
16 git commit -am "image"
17 git add .
18 git status
19 git commit -am "r"
20 gh repo create study_2021-2022_os-intro --template=yamadharma/course-directory-student-template --public
21 git add .
22 git status
23 git commit -am "r"
24 git push
25 clear
26 cd
27 ls
28 clear
29 sudo dnf install gh
30 git config --global user.name "Vasily Duplenski"
31 git config --global user.email "1032212268@pfur.ru"
32 gpg --full-generate-key
33 gpg --list-secret-keys --keyid-format LONG
34 gpg --armor --export <PGP Fingerprint> | xclip -sel clip
35 gpg --armor | xclip -sel clip
36 gpg --armor --export 2FFA42F559462D20 | xclip -sel clip
37 git config --global user.signinkey -----BEGIN PGP PUBLIC KEY BLOCK-----
38 mINBGJR8RoBEACnpKYk2UMYOL4CTmBoryBD7EG+MsBWqd/90hg3jDlhQ80uk8BY
39 ww90l0iZssjMtDL3INjV9ezRb+j5Ebt7GtGh8ShwoKhn4ICLEaed+pfXPuZrXuJ5
40 7F349qn+nW2azsPPLpzEQafgc9Z/z2C606Ij0nsJ0VymVIsqmqvBiT+eFsaQr8n9d
41 g6Q3vij6G/bz5qKsNTIkeL9c8QvttVesAwy9FEQX/h7hEY3YvbJ6RyYL3orFeG9t
42 i+qxDb8c/p9fSnq/IpmwmtQIvEfDjwqSghFOWRvGy8fBI561U1NBDW/epIYj770fF
43 JVDTeJIMzkfQj6zEHgNvEFnw7shVRAojVdBE5/s8UWpTI531EWm60jHwsKgTQY
44 5QWk3haYMMtBLENC8Up4vFFtPy3g9RGdar4bu/vBKJWYEGGyqFW0qWPdx9c5W18
45 XS0lMaU7wTc9v0jdQbtG7Zb63AaawotRv/E1h4HAi7Q3+m4fk3HGp3/kLhXFKJ1
46 L8qAYnXWYwBzNlsku1eBI9Pwtd1jAq3I3lWRlNHWRGw5KxhuCT0xQ3xntTVLXb
47 Uu/fVJ3GQ2nN4p/SuyJZRpdXeutBtRhEmKQKY0gxW0GU1Wrso7KU8tdG95R56C+F
48 Wn/g35ZjwknN2srUHKUnuAdCWUnNdtY6ht00K76f5Tj+nSszRFVxUSBwARAQAB
49 tBtWYXNpbHkgPDEwMzIyNTIyNjEAcGZlc3Y5dGt6JAlIEEwEIAADwIQS2eahgdbDi
50 D472BuAv+kL1WUYtIAUCYmvxGgIbAwULCQhAgMiAgEGFQoJCAcCBYCAwECHgcC
51 F4AAcGkYpC9VLGLSAdsAAI50C3DX0qiTLWhP3Pvj33d4+F0y4Mv3ybjRlqnuC
52 Z4eR16fU8U7brXNvxzk3KHEVCiZCFj+0L7QjFUAHz9gbTDEKohHhfrlYJ3bbA8V
53 FywITKfz/ecn7eqPXau1j+CfaPJD+x8hARluGs2rGUE7615PTd9Sww2jTmKkita
54 bh2whEypG27kH3gmxp+wbeAi00imZA9y9BS6G06IPg034x2u6bTcyMeRchcPfZP
55 hzVeAlBzo7rajkJkD0sUrBvhNuK0UpoyTESRVDHoF+Dpzena53/dLw/0tcc+yJu
56 Qq/K6R6uGEHndvXp3JLDIaqeZ3l0J14hyi+kDMORQ9lg2fC7vojYr+8mQTKita
57 TAYF/hdS28AWc+k1p4mU0hBYq5IS6VwuKTknFJR6eGxbe1afkvRAVALXuBtKLZ5
58 tWADy6vNF/4lylVmQkrPZVndgbXVKeM+02EK2tFyAvY1gSuvFwjLDocNvdiGssjZ
59 PPNR6GwKeRadoI 9eezR3a3wsBwFBPEX6Xp8e63Vev0kZYm3zrDdFeIhVTKGaSalc1
```

Ответы на вопросы: 1.Интерфейс командной строки - управление программами

с помощью команд. Команды состоят из букв, цифр, символов, набираются построчно, выполняются после нажатия клавиши Enter. Основным инструментом здесь является клавиатура. Данный интерфейс встроен в ядро системы, он будет доступен, даже если графический интерфейс не запустится. Добраться до командной строки можно двумя способами: через консоль или терминал.

2. При помощи команды `realpath` можно определить абсолютный путь текущего каталога. Например, если ввести `realpath /var` на экран выведется `/home//var`.

3. При помощи команды `ls -F` можно определить только тип файлов и их имена в текущем каталоге.
4. Файл (или директория) считается скрытым, если его название начинается с символа точка «.». Например, «.myfile». Обычно такие файлы используются приложениями для хранения настроек, конфигураций и другой информации, которую нужно скрыть от пользователя. Зачастую пользователю требуется отредактировать соответствующий конфигурационный скрытый файл, чтобы настроить какую-нибудь программу, и пользователи сталкиваются с тем, что не знают, как их вообще просмотреть. По умолчанию файловые менеджеры обычно не отображают такие файлы. Для просмотра списка файлов в командной строке используется команда `ls`. Чтобы по команде `ls` также выводились скрытые файлы, существует опция `-a`.
5. При помощи команд `rm` и `rmdir` можно удалить файл и каталог. Это нельзя сделать одной и той же командой. `rmdir` используется, чтобы удалить файлы, которые должны быть пустые. `rm` используется, чтобы удалить непустые файлы или целые деревья каталогов.
6. Определить какие команды выполнил пользователь в сеансе работы можно с помощью команды `history`.
7. Исправить и запустить на выполнение команду, которую пользователь уже использовал в сеансе работы, можно с помощью команды: `!!` Например,

history . . 3 ls -a . . !3:s/a/F ls -F

8. В одной строке можно записать несколько команд. Если требуется выполнить последовательно несколько команд, записанный в одной строке, то для этого используется символ точка с запятой. Пример: `cd; ls`. 9. Экранирование — это способ заключения в кавычки одиночного символа. Экранирующий символ (`\`) сообщает интерпретатору, что следующий за ним символ должен восприниматься как обычный символ. Пример: `echo "Привет" # Привет echo "Он сказал: "Привет"." # Он сказал: "Привет"`.
9. Если используется опция `l` в команде `ls`, то на экран выводится подробный список, в котором будет отображаться владелец, группа, дата создания, размер и другая информация о файлах и каталогах.
10. Относительный путь — это путь к файлу относительно текущей папки. При использовании команды `pwd` на экран выведется относительный путь текущей директории, а при использовании команды `realpath` на экран выведется абсолютный путь текущей директории.
11. Получить информацию об интересующей вас команде можно с помощью команды `man`. Например, команда `man ls` выведет все опции команды `ls`.
12. Сочетание клавиш `Ctrl+C` прерывает текущий процесс, запущенный в терминале.

Вывод:

Мы приобрели практические навыки взаимодействия пользователя с системой посредством командной строки.