

# **Лабораторная работа №6**

**Дисциплина: Операционные системы**

Савостин Олег

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>

# Список иллюстраций

4.1	Полное название домашнего каталога . . . . .	8
4.2	Файлы в соответствующем каталоге . . . . .	8
4.3	Cron . . . . .	9
4.4	Домашний каталог . . . . .	9
4.5	Новые каталоги . . . . .	9
4.6	Каталоги . . . . .	9
4.7	Ошибка . . . . .	9
4.8	Удаление newdir . . . . .	10
4.9	-R . . . . .	10
4.10	-t . . . . .	10
4.11	cd . . . . .	11
4.12	pwd . . . . .	11
4.13	mkdir . . . . .	12
4.14	rmdir . . . . .	12
4.15	rm . . . . .	13
4.16	История . . . . .	13
4.17	Модификация . . . . .	14

## **Список таблиц**

# 1 Цель работы

Приобретение практических навыков взаимодействия пользователя с системой посредством командной строки

## 2 Задание

1. Базовые команды в терминале.

### 3 Теоретическое введение

В операционной системе типа Linux взаимодействие пользователя с системой обычно осуществляется с помощью командной строки посредством построчного ввода команд. При этом обычно используются командные интерпретаторы языка shell: /bin/sh; /bin/csh; /bin/ksh. Формат команды. Командой в операционной системе называется записанный по специальным правилам текст (возможно с аргументами), представляющий собой указание на выполнение какой-либо функций (или действий) в операционной системе. Обычно первым словом идёт имя команды, остальной текст — аргументы или опции, конкретизирующие действие. Общий формат команд можно представить следующим образом: Команда `man`. Команда `man` используется для просмотра (оперативная помощь) в диалоговом режиме руководства (`manual`) по основным командам операционной системы типа Linux. Формат команды: `man` Пример (вывод информации о команде `man`):  
1 `man man` Для управления просмотром результата выполнения команды `man` можно использовать следующие клавиши: — Space — перемещение по документу на одну страницу вперёд; — Enter — перемещение по документу на одну строку вперёд; — q — выход из режима просмотра описания. Команда `cd`. Команда `cd` используется для перемещения по файловой системе операционной системы типа Linux. Замечание 1. Файловая система ОС типа Linux — иерархическая система каталогов, подкаталогов и файлов, которые обычно организованы и сгруппированы по функциональному признаку. Самый верхний каталог в иерархии называется корневым и обозначается символом `/`. Корневой каталог содержит системные файлы и другие каталоги

## 4 Выполнение лабораторной работы

Проверяю полное название домашнего каталога(рис. 4.1).

```
root@vbox:~# pwd
/root
root@vbox:~#
```

Рис. 4.1: Полное название домашнего каталога

Перехожу в каталог tmp и смотрю файлы в нем. (рис. 4.2).

```
root@vbox:~# ls
afs  boot  dev  home  lib64  media  opt  root  sbin  sys  usr  work
bin  code.txt  etc  lib  lost+found  mnt  proc  run  srv  tmp  var
root@vbox:~# cd tmp
root@vbox:~/tmp# ls
plasma-csd-generator.PLnTrX
sddm-auth-9a667c8e-3217-4992-908d-eb74a29d0927
sddm-woqKJE
systemd-private-5111752c82544ae7828cb7e099b4079f-abrt.service-THuz2q
systemd-private-5111752c82544ae7828cb7e099b4079f-chronyd.service-U8bgz1
systemd-private-5111752c82544ae7828cb7e099b4079f-dbus-broker.service-qVY3Wc
systemd-private-5111752c82544ae7828cb7e099b4079f-fprintd.service-pt6lWj
systemd-private-5111752c82544ae7828cb7e099b4079f-irqbalance.service-ZsQ7H0
systemd-private-5111752c82544ae7828cb7e099b4079f-ModemManager.service-kuNMx7
systemd-private-5111752c82544ae7828cb7e099b4079f-polkit.service-MM1kUN
systemd-private-5111752c82544ae7828cb7e099b4079f-rtkit-daemon.service-f9Izua
systemd-private-5111752c82544ae7828cb7e099b4079f-switcheroo-control.service-4l6IIZ
systemd-private-5111752c82544ae7828cb7e099b4079f-systemd-hostnamed.service-7dmKW6
systemd-private-5111752c82544ae7828cb7e099b4079f-systemd-logind.service-7T2LnV
systemd-private-5111752c82544ae7828cb7e099b4079f-systemd-oomd.service-0l4KzB
systemd-private-5111752c82544ae7828cb7e099b4079f-systemd-resolved.service-x56yPe
systemd-private-5111752c82544ae7828cb7e099b4079f-upower.service-YpNwhW
root@vbox:~/tmp#
```

Рис. 4.2: Файлы в соответствующем каталоге

Определяю если есть подкаталог cron (рис. 4.3).



```

root@vbox:~# cd var
root@vbox:/var# ls
account  cache  db      ftp      kerberos  local  log  nis  preserve  spool  yp
adm      crash  empty  games  lib       lock   mail  opt  run      tmp
root@vbox:/var# cd spool
root@vbox:/var/spool# ls
abrt  abrt-upload  anacron  at  cron  cups  lpd  mail  plymouth
root@vbox:/var/spool#

```

Рис. 4.3: Cron

Перехожу в домашний каталог и вывожу на экран его содержимую Владелец - root, супер-пользователь (рис. 4.4).

```

root@vbox:~# ls
anaconda-ks.cfg  git-extended  LICENSE  pass2.txt  password.txt  texput.log
bin              LABS          package.json  pass.txt  quit          work
root@vbox:~# id siadmin
id: «siadmin»: такого пользователя нет
root@vbox:~# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

```

Рис. 4.4: Домашний каталог

Создаю в домашнем каталоге новый каталог newdir и morefun(рис. 4.5).

```

root@vbox:~# mkdir newdir
root@vbox:~# cd newdir
root@vbox:~/newdir# mkdir morefun
root@vbox:~/newdir# cd -
/root
root@vbox:~# ls ~/newdir/morefun
root@vbox:~# ls ~/newdir
ls: невозможно получить доступ к '/root/newdir': Нет такого файла или каталога
root@vbox:~# ls ~/newdir
morefun

```

Рис. 4.5: Новые каталоги

Создаю одной командой три каталога и одной командой их удаляю (рис. 4.6).

```

root@vbox:~# mkdir letters memos misk
root@vbox:~# ls
anaconda-ks.cfg  git-extended  letters  memos  newdir  pass2.txt  password.txt  texput.log
bin              LABS          LICENSE  misk   package.json  pass.txt  quit          work
root@vbox:~# rmdir letters memos misk
root@vbox:~# ls
anaconda-ks.cfg  git-extended  LICENSE  package.json  pass.txt  quit          work
bin              LABS          newdir   pass2.txt    password.txt  texput.log
root@vbox:~#

```

Рис. 4.6: Каталоги

Пробую удалить newdir, но команда неверная, ибо это каталог, а не конкретный файл(рис. 4.7).

```

root@vbox:~# rm newdir
rm: невозможно удалить 'newdir': Это каталог

```

Рис. 4.7: Ошибка

Удаляю ненужный каталог (рис. 4.8).

```
root@vbox:~# ls
anaconda-ks.cfg  git-extended  LICENSE  package.json  pass.txt  quit  work
bin              LAB5          newdir   pass2.txt     password.txt  texput.log
root@vbox:~# ls newdir
morefun
root@vbox:~# rmdir -p newdir/morefun
root@vbox:~# ls
anaconda-ks.cfg  git-extended  LICENSE  pass2.txt  password.txt  texput.log
bin              LAB5          package.json  pass.txt  quit  work
```

Рис. 4.8: Удаление newdir

Определяю какую опцию нужно использовать чтобы просмотреть содержимое(рис. 4.9).

```
-R, --recursive
    list subdirectories recursively
```

Рис. 4.9: -R

Смотрю, какая опция подойдет, чтобы отсортировать по времени изменения выводимый список каталога (рис. 4.10).

```
-t    sort by time, newest first; see --time
```

Рис. 4.10: -t

Использую команду man для просмотра описания команд (рис. 4.11) (рис. 4.12) (рис. 4.13) (рис. 4.14) (рис. 4.15).

```

-m keymap
    Use keymap as the keymap to be affected by the subsequent bindings. Acceptable keymap
    names are emacs, emacs-standard, emacs-meta, emacs-ctix, 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.
-v
    Display readline variable names and values in such a way that they can be re-read.
-V
    List current readline variable names and values.
-f filename
    Read key bindings from filename.
-q function
    Query about which keys invoke the named function.
-u function
    Unbind all keys bound to the named function.
-r keyseq
    Remove any current binding for keyseq.
-x keyseq:shell-command
    Cause shell-command to be executed whenever keyseq is entered. When shell-command is exe-
    cuted, the shell sets the READLINE_LINE variable to the contents of the readline line
    buffer and the READLINE_POINT and READLINE_MARK variables to the current location of the
    insertion point and the saved insertion point (the mark), respectively. The shell assigns
    any numeric argument the user supplied to the READLINE_ARGUMENT variable. If there was no
    argument, that variable is not set. If the executed command changes the value of any of
    READLINE_LINE, READLINE_POINT, or READLINE_MARK, those new values will be reflected in the
    editing state.
-X
    List all key sequences bound to shell commands and the associated commands in a format that
    can be reused as input.

The return value is 0 unless an unrecognized option is given or an error occurred.

```

Рис. 4.11: cd

```

NAME
    pwd - print name of current/working directory

SYNOPSIS
    pwd [OPTION]...

DESCRIPTION
    Print the full filename of the current working directory.

    -L, --logical
        use PWD from environment, even if it contains symlinks

    -P, --physical
        avoid all symlinks

    --help display this help and exit

    --version
        output version information and exit

    If no option is specified, -P is assumed.

    Your shell may have its own version of pwd, which usually supersedes the version described here. Please
    refer to your shell's documentation for details about the options it supports.

```

Рис. 4.12: pwd

```

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, with their file modes unaffected by any
        -m option.

    -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

```

Рис. 4.13: mkdir

```

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 to remove a non-empty directory

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

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

    --help display this help and exit

    --version
        output version information and exit

```

Рис. 4.14: rmdir

```

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

```

Рис. 4.15: rm

Использую команду history и исполню модификацию (рис. 4.16) (рис. 4.17).

```

332 cd newdir
333 mkdir morefun
334 cd -
335 ls ~/newdir/morefun
336 ls ~/newidr
337 ls ~/newdir
338 mkdir letters memos misk
339 ls
340 rmdir letters memos misk
341 ls
342 rm newdir
343 rmdir newdir
344 rmdir -r newdir
345 rmdir --help
346 ls
347 ls newdir
348 rmdir -p newdir/morefun
349 ls
350 man ls
351 man cd
352 man pwd
353 man mkdir
354 man rmdir
355 man rm
356 history
root@vbox:~#

```

Рис. 4.16: История

```
root@vbox:~# !334:s/-/  
cd  
root@vbox:~#
```

Рис. 4.17: Модификация

## **5 Выводы**

В ходе работы, я приобрел практических навыков взаимодействия пользователя с системой посредством командной строки

# Список литературы

Лабораторная работа 6, ТУИС ::: {#refs} :::