

# **Лабораторная работа №4**

**Операционные системы**

Пашаев Юсиф Юнусович

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	15
4	Контрольные вопросы	16
	Список литературы	18

## Список иллюстраций

2.1	каталог tmp . . . . .	6
2.2	каталог . . . . .	7
2.3	каталог . . . . .	7
2.4	новый каталог newdir . . . . .	7
2.5	три новый каталога . . . . .	7
2.6	удаление каталогов . . . . .	8
2.7	удаление каталога newdir . . . . .	8
2.8	команда man . . . . .	10
2.9	команда man . . . . .	11
2.10	команда man . . . . .	12
2.11	команда man . . . . .	13

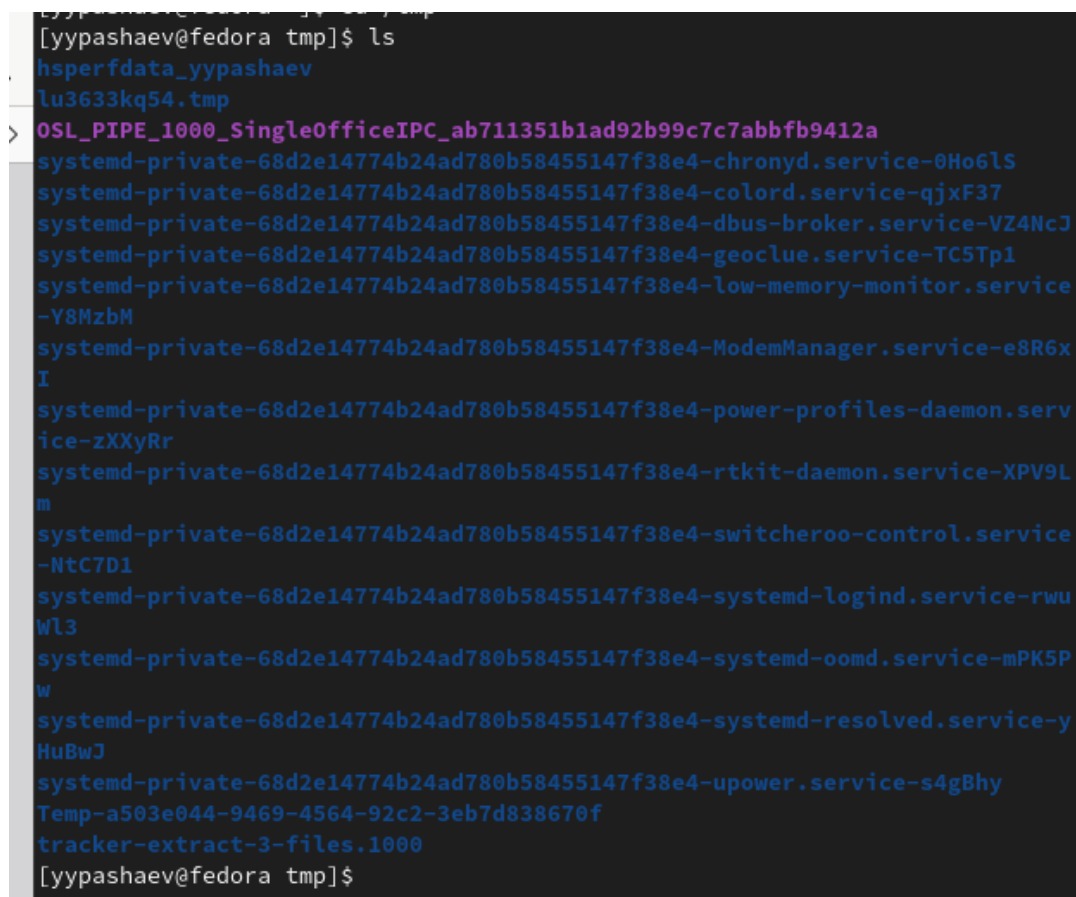
## Список таблиц

# 1 Цель работы

Приобретение практических навыков взаимодействия пользователя с системой посредством командной строки

## 2 Выполнение лабораторной работы

1. Перехожу в каталог /tmp. Вывожу на экран содержимое каталога /tmp. Для этого использую команду ls(рис. 2.1).



```
[yypashaev@fedora tmp]$ ls
hspcrfdata_yypashaev
lu3633kq54.tmp
> OSL_PIPE_1000_SingleOfficeIPC_ab711351b1ad92b99c7c7abfb9412a
systemd-private-68d2e14774b24ad780b58455147f38e4-chronyd.service-0Ho6lS
systemd-private-68d2e14774b24ad780b58455147f38e4-colord.service-qjxF37
systemd-private-68d2e14774b24ad780b58455147f38e4-dbus-broker.service-VZ4NcJ
systemd-private-68d2e14774b24ad780b58455147f38e4-geoclue.service-TC5Tpi
systemd-private-68d2e14774b24ad780b58455147f38e4-low-memory-monitor.service
-Y8MzbM
systemd-private-68d2e14774b24ad780b58455147f38e4-ModemManager.service-e8R6x
I
systemd-private-68d2e14774b24ad780b58455147f38e4-power-profiles-daemon.serv
ice-zXXyRr
systemd-private-68d2e14774b24ad780b58455147f38e4-rtkit-daemon.service-XPV9L
m
systemd-private-68d2e14774b24ad780b58455147f38e4-switcheroo-control.service
-NtC7D1
systemd-private-68d2e14774b24ad780b58455147f38e4-systemd-logind.service-rwu
Wl3
systemd-private-68d2e14774b24ad780b58455147f38e4-systemd-oomd.service-mPK5P
w
systemd-private-68d2e14774b24ad780b58455147f38e4-systemd-resolved.service-y
HuBwJ
systemd-private-68d2e14774b24ad780b58455147f38e4-upower.service-s4gBhy
Temp-a503e044-9469-4564-92c2-3eb7d838670f
tracker-extract-3-files.1000
[yypashaev@fedora tmp]$
```

Рис. 2.1: каталог tmp

2. Определяю, есть ли в каталоге /var/spool подкаталог с именем cron

```
[yypashaev@fedora tmp]$ ls /var/spool
abrt  abrt-upload  cups  lpd  mail  plymouth
[yypashaev@fedora tmp]$
```

Рис. 2.2: каталог

```
[yypashaev@fedora tmp]$ cd ~
[yypashaev@fedora ~]$ ls
w      Видео      Загрузки      Музыка      'Рабочий стол'
work   Документы     Изображения   Общедоступные  Шаблоны
[yypashaev@fedora ~]$
```

Рис. 2.3: каталог

3. Домашнем каталоге создаю новый каталог с именем newdir(рис. 2.4).

```
[yypashaev@fedora ~]$ mkdir newdir
[yypashaev@fedora ~]$ ls
newdir  work   Документы  Изображения  Общедоступные  Шаблоны
w       Видео  Загрузки   Музыка       'Рабочий стол'
[yypashaev@fedora ~]$
```

Рис. 2.4: новый каталог newdir

4. В домашнем каталоге создаю одной командой три новых каталога с именами letters, memos, misk(рис. 2.5).

```
[yypashaev@fedora ~]$ mkdir letter memos misk
[yypashaev@fedora ~]$ ls
letter  newdir  Видео      Изображения  'Рабочий стол'
memos   w       Документы  Музыка       Шаблоны
misk    work    Загрузки   Общедоступные
[yypashaev@fedora ~]$
```

Рис. 2.5: три новых каталога

5. Затем удаляю эти каталоги одной командой (рис. 2.6).

```
rm: невозможно удалить 'misk': Это каталог
[yypashaev@fedora ~]$ rm -r letter memos misk
[yypashaev@fedora ~]$ ls
newdir  work  Документы  Изображения  Общедоступные  Шаблоны
w       Видео  Загрузки  Музыка       'Рабочий стол'
```

Рис. 2.6: удаление каталогов

6. Попробую удалить ранее созданный каталог ~/newdir командой rm. Проверю, был ли каталог удалён (рис. 2.7).

```
w       Видео  Загрузки  Музыка       'Рабочий стол'
[yypashaev@fedora ~]$ rm -r newdir
[yypashaev@fedora ~]$ ls
w       Видео  Загрузки  Музыка       'Рабочий стол'
work  Документы  Изображения  Общедоступные  Шаблоны
[yypashaev@fedora ~]$
```

Рис. 2.7: удаление каталога newdir

7. Использую команду man для просмотра описания следующих команд: cd, pwd, mkdir, rmdir, rm



```
LS(1)                                User Commands                                LS(1)
> NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

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

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
        with -l, print the author of each file

    -b, --escape
Manual page ls(1) line 1 (press h for help or q to quit)
```

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

> NAME
: , . , [ , alias , bg , bind , break , builtin , caller , cd , command ,
comp gen , 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 , readarray , 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 ac-
cepting options interpret arguments beginning with - as invalid
options and require -- to prevent this interpretation.
: [arguments]
    No effect; the command does nothing beyond expanding argu-
ments and performing any specified redirections. The re-
turn status is zero.

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

Рис. 2.8: команда man

```

. 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, but filename does not need to be executable. The file searched for in **PATH** need not be executable. When **bash** is not in posix mode, it searches the current directory 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, **.** inherits any trap on **DEBUG**; if it is not, any **DEBUG** trap string is saved and restored around the call to **.**, and **.** 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 **.** completes. The return status is the status of the last command exited within the script (0 if no commands are executed), and false if filename is not found or cannot be read.

```

alias [-p] [name[=value] ...]

```

**Alias** with no arguments or with the **-p** option prints the

Manual page cd(1) line 28 (press h for help or q to quit)

Рис. 2.9: команда man

```
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

**bind** readline-command-line

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 a **readline** initialization file such as .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 sub-

Manual page cd(1) line 61 (press h for help or q to quit)

Рис. 2.10: команда man

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

break [n]
Exit from within a for, while, until, or select loop. If n
is specified, break n levels. n must be  $\geq 1$ . If n is
greater than the number of enclosing loops, all enclosing
loops are exited. The return value is 0 unless n is not
greater than or equal to 1.

builtin shell-builtin [arguments]
Execute the specified shell builtin, passing it arguments,
and return its exit status. This is useful when defining a
function whose name is the same as a shell builtin, retain-
ing the functionality of the builtin within the function.
The cd builtin is commonly redefined this way. The return
status is false if shell-builtin is not a shell builtin
command.

caller [expr]
Returns the context of any active subroutine call (a shell
function or a script executed with the . or source
builtins). Without expr, caller displays the line number
and source filename of the current subroutine call. If a
non-negative integer is supplied as expr, caller displays
the line number, subroutine name, and source file corre-
Manual page cd(1) line 130 (press h for help or q to quit)
```

Рис. 2.11: команда man

8. Используя информацию, полученную при помощи команды **history**, выпол-  
ню мо- дификацию и исполнение нескольких команд из буфера команд(рис.  
??).

```
31 pip install pandoc-fignos pandoc-eqnos pandoc-tablenos pandoc-secnos
--user
32 dnf install git
33 sudo -i
34 gh auth login
35 mkdir -p ~/work/study/2022-2023/"Операционные системы"
36 cd ~/work/study/2022-2023/"Операционные системы"
37 gh repo create study_2022-2023_os-intro --template=yamadharm/course
-directory-student-template --public
38 git clone --recursive https://github.com/yusif123321/study_2022-2023
_os-intro.git os-intro
39 cd ~/work/study/2022-2023/"Операционные системы"/os-intro
40 rm package.json
41 echo os-intro > COURSE
42 make
43 git add .
44 git commit -am 'feat(main): make course structure'
45 git push
46 make
47 man ls
48 man cd
49 man pwd
50 man mkdir
51 man rmdir
52 man rm
53 history
[yypashaev@fedora ~]$
```

## **3 Выводы**

Приобрел практических навыков взаимодействия пользователя с системой посредством командной строки

## 4 Контрольные вопросы

1. Что такое командная строка? Ответ: текстовый интерфейс взаимодействия пользователя с системой
2. При помощи какой команды можно определить абсолютный путь текущего каталога? Приведите пример. Ответ: команда `pwd`, пример: `• cd /var/www • pwd • /var/www/`
3. При помощи какой команды и каких опций можно определить только тип файлов и их имена в текущем каталоге? Приведите примеры. Ответ: команда `ls` с опцией `-F`.
4. Какие файлы считаются скрытыми? Как получить информацию о скрытых файлах? Приведите примеры. Ответ: Некоторые файлы в операционной системе скрыты от просмотра и обычно используются для настройки рабочей среды. Имена таких файлов начинаются с точки. информацию о них можно получить с помощью команды `ls` с опцией `-a`.
5. При помощи каких команд можно удалить файл и каталог? Можно ли это сделать одной и той же командой? Ответ: С помощью команды `rm` можно удалить как отдельный файл так и целый каталог, в случае каталога необходимо указать опцию `-r`.
6. Как определить, какие команды выполнил пользователь в сеансе работы? Ответ: с помощью команды `history`.



7. Каким образом можно исправить и запустить на выполнение команду, которую пользователь уже использовал в сеансе работы? Приведите примеры  
Ответ: узнать порядковый номер этой команды с помощью history затем изменить её сл. образом: `!:s//`
8. Можно ли в одной строке записать несколько команд? Если да, то как? Приведите примеры  
Ответ: да, можно, необходимо разделить команды символом точки с запятой в таком случае они будут выполняться последовательно в том порядке, в котором они записаны пример: `cd /tmp/; ls -l; pwd`
9. Что такое символ экранирования? Приведите примеры использования этого символа.  
Ответ: символ экранирования (обратный слэш) - символ, экранирующие управляющие конструкции и символы в названии файлов и папок  
Пример: `ls /etc/nginx`
10. Какая информация выводится на экран о файлах и каталогах, если используется опция `l` в команде `ls`?  
Ответ: тип файла, право доступа, число ссылок, владелец, размер, дата последней ревизии, имя файла или каталога.
11. Что такое относительный путь к файлу? Приведите примеры использования относительного и абсолютного пути при выполнении какой-либо команды.  
Ответ: относительный путь - путь к тому или иному файлу или директории относительной текущей рабочей директории, пример: папка `/www/` в директории `/var/` абсолютный путь: `/var/www/` относительный путь(если рабочая директория - `/var/`): `/www/`
12. Как получить информацию об интересующей вас команде? Ответ: можно попробовать найти информацию по использованию с помощью утилиты `man`, или попробовать ввести опцию `-help`.
13. Какая клавиша или комбинация клавиш служит для автоматического дополнения вводимых команд? Ответ: клавиша `Tab`

## **Список литературы**