Laboratory work # 4. Basics of the Unix user interface at the command line level

4.1. Purpose of the work

Acquisition of practical skills of user interaction with the software system by means of the command line.

4.2. Operating instructions

In a Linux-type operating system, the user usually interacts with the system from the command line by entering line-by-line commands. In this case, shell command interpreters are usually used: /bin/sh ; /bin/csh ; /bin/ksh .

Format of the command.

A command in the operating system is a text written according to special rules (possibly with arguments) that indicates the execution of any functions (or actions) in the operating system. Usually, the first word is the command name, and the rest of the text is arguments or options that specify the action. The general format of commands can be represented as follows:

<team_name><separator><arguments>

The man command.

Team man used for viewing (quick help) in dialog mode of the manual for basic commands of a Linux-type operating system.

Command format:

man <command>

Example (displaying information about the team man):

```
man man
```

To control viewing the result of a command execution man you can use next keys:

— Space - move one page forward in the document.

— Enter - move one line forward in the document.

— q - exit the description view mode.

The cd command.

Team cd used to navigate through the file system of a Linux operating system.

Note 1: A Linux OS file system is a hierarchical system of directories, subdirectories, and files that are usually organized and grouped by function . The topmost directory in the hierarchy is called the root directory and is indicated by the symbol / . The root directory contains system files and others directories.

Command format:

cd [path_to_directory]

To navigate to the user's home directory, use the command cd without parameters or cd ~ .

For example, the command

```
cd
```

/afs/dk.sci.pfu.edu.ru/home allows you to go to the folder /afs/dk.sci.pfu.edu.ru/home

(if one exists),
and in order to move up one directory, you should use:
```
cd ..
```
Learn more about the team's options
cd
see the help with the command
man
:
```
man cd
```
The pwd command.
To determine the absolute path to the current directory, use
team
pwd
(print working directory).
Example (absolute name of the current dharma user directory):
```
pwd
```
result:
```
/afs/dk.sci.pfu.edu.ru/home/d/h/dharma
```
Abbreviations of file names.
When working with commands that
use the path to a particular directory or file as arguments, you can use a shortened
path entry. Abbreviations are shown in Table 4.1.
Table 4.1
File name shortening characters

Symbol	Meaning
~	Home Directory
.	Current folder
..	Parent folder

For example, in the command
cd
to navigate through the file system
, you can use the shortened path entry as follows (commands are interleaved with the output
of the command execution result

pwd
):
1
pwd
2
3
/afs/dk.sci.pfu.edu.ru/home/d/h/dharma
4
40
Laboratory work # 4. Basics of the user interaction interface with the system ...
5
cd
..
6
pwd
7
8
/afs/dk.sci.pfu.edu.ru/home/d/h
9
10
cd
../..
11
pwd
12
13
/afs/dk.sci.pfu.edu.ru/home
14
15
cd
~/work
16
pwd
17
18
/afs/dk.sci.pfu.edu.ru/home/d/h/dharma/work
The ls command.
Team
ls
used to view the contents of the folder.
Command format:
ls
[-options]
[path]
Example:
1
cd
2
cd
..
3
pwd
4
5
/afs/dk.sci.pfu.edu.ru/home/d/h
6
7
ls
8
9
dharma
Some files in the operating system are hidden from view and are usually used to configure the working environment. The names of these files start with a dot. To display the names of hidden files, use the command
ls
with the option
a
:
1
ls -a
You can also get information about file types (directory,executable, etc.). link), what the option is used for
F
. When using this option in the name field
displays a character that specifies the file type (see Table 4.2).
Table 4.2
A character that defines the file type
File Type
Symbol
Catalog
/
Executable file
*
Link
@

To display detailed information about files and directories, you must: use this option
l
. The following information will be displayed for each file and directory: information:
— file type
– - access rights

– - number of links
– - owner,
- size
– - date of last revision,
– file or directory name.
Example:

```
cd
/
ls
```

Result:

```
bin
boot
dev
etc
home
lib media
mnt
opt
proc
root
sbin
sys
tmp
usr
var
```

In the same folder, the command

```
ls -alF
```

will give approximately the following result:

```
drwxr-xr-x
21
root root
4096
Jan.
17 09
:00 ./
drwxr-xr-x
21
root root
4096
Jan.
17 09
:00 ../
drwxr-xr-x
2
root root
4096
Jan.
18 15
:57 bin/
drwxr-xr-x
2
root root
4096
Apr.
14
2008
boot/
drwxr-xr-x
20
root root
14120
Feb.
17 10
:48 dev/
drwxr-xr-x
```

170
root root
12288
Feb.
17 09
:19 etc/
7
drwxr-xr-x
6
root root
4096
Aug.
5
2009
home/
8
lrwxrwxrwx
1
root root
5
Jan.
12 22
:01 lib -> lib64/
9
drwxr-xr-x
8
root root
4096
Jan.
30 21
:41 media/
10
drwxr-xr-x
5
root root
4096
Jan.
17
2010
mnt/
11
drwxr-xr-x
25
root root
4096
Jan.
16 09
:55 opt/
12
dr-xr-xr-x
163
root root
0
Feb.
17 13
:17 proc/
13
drwxr-xr-x
31
root root
4096
Feb.
15 23
:57 root/
14
drwxr-xr-x
2
root root
12288
Jan.
18 15
:57 sbin/
15
drwxr-xr-x
12
root root
0
Feb.
17 13
:17 sys/
16
drwxrwxrwt
12
root root
500
Feb.
17 16
:35 tmp/
17
drwxr-xr-x
22

root root
4096
Jan.
18 09
:26 usr/
18
drwxr-xr-x
17
root root
4096
Jan.
14 17
:38 var/

The mkdir command.

Team
mkdir
used for creating directories.

Command format:

mkdir catalog_name 1 [catalog_name 2...]

Example of creating a folder in the current folder:


```
1
cd
2
pwd
3
4
/afs/dk.sci.pfu.edu.ru/home/d/h/dharma
5
6
ls
7
8
Desktop
public
tmp
9
GNUstep
public_html
work
10
11
mkdir abc
12
ls
13
14
abc
GNUstep
public_html
work
15
Desktop
public
tmp
```

Note 2. In order to create a directory in a specific location on the file system,
access rights must be set correctly.

You can also create a subdirectory in an existing subdirectory:

```
1
mkdir parentdir
2
mkdir parentdir/dir
```

Multiple arguments create multiple directories:

```
1
cd
parentdir
2
mkdir dir1 dir2 dir3
```

You can use a dimension:

```
1
mkdir parentdir/
{
dir1,dir2,dir3
}
```

If you want to create a subdirectory in a directory other than the current one, then the path to it is
you must specify it explicitly:

```
1
mkdir ../dir1/dir2
```

or

```
1
mkdir ~/dir1/dir2
```

The following options are interesting:

--mode
(or
-m
)- setting access attributes.
--parents
(or
-p
)- creating a directory together with its parent ones in relation to it
directories.

Attributes are set in numeric or symbolic notation:

```
1
mkdir --mode
=777
```

dir

or
```
1
mkdir -m a+rwx dir
```
Option
--parents
(short form
-p
) allows you to create a hierarchical chain
subdirectories, creating all intermediate directories:
```
1
mkdir -p ~/dir1/dir2/dir3
```
The rm command.
Team
rm
used to delete files and / or directories.
Command format:
rm [- options] [file]
If you want a confirmation request to delete a file, then you must:
dimo use the option
i
.
To delete a folder containing files, use the option
r
. Without specifying
this option will not execute the command.
Example:
```
1
cd
2
mkdir abs
3
rm abc
4
5
rm: abc is a directory
6
7
rm -r abc
```
If the folder is empty, you can use the command
rmdir
. If the user is being deleted
the folder contains files, then the command will not be executed — you need to use
rm -
r catalog_name
.
The history command.
To display a list of previously executed commands, use
the command is being used
history
. The commands displayed on the screen are numbered in the list. Any
command from the list displayed on the screen can be accessed by its number in the list,
using the construction
!<team_number>
.
Example:
```
1
history
2
1
pwd
3
2
ls
4
3
ls -a
5
4
ls -l
6
5
cd
/
7
6
history
8
9
!5
10
cd
/
```
You can modify a command from the list displayed on the screen using the following:
using the design:

!<command number>:s/<what to change>/<what to change>
Example:

```
1
!3:s/a/F
2
ls -F
```

Note 3: If special characters occur in the specified context (such as "
.
»,
«
/
», «
*
", etc.), you need to put an escape character in front of them
\
(backslash).

Using the"; " symbol.

If you want to perform several tasks in sequence
if the number of commands is written in a single line, then the semicolon character is used for this purpose
Example:

```
1
cd; ls
```

4.3. Sequence of work execution

1. Determine the full name of your home directory. Further, regarding this kata-
subsequent exercises will be performed in the log.
2. Follow these steps::
2.1. Go to the catalog
/tmp
.
2.2. Display the contents of the folder
/tmp
. To do this, use the command
ls
with various options. Explain the difference in the information displayed on the screen.
2.3. Determine whether the catalog contains
/var/spool
a subdirectory named
cron
?
2.4. Go to your home directory and display its contents. Opre-
share who is the owner of files and subdirectories?
3. Follow these steps::
3.1. In the home directory, create a new directory with the name
newdir
.
3.2. In the catalog
~/newdir
create a new folder named
morefun
.
3.3. In the home directory, use one command to create three new directories with the names
letters
,
memos
,
misk
. Then delete these directories with one command.
3.4. Try deleting the previously created folder
~/newdir
as a team
rm
. Check it out,
whether the folder was deleted.
3.5. Delete the folder
~/newdir/morefun
from the home directory. Check if there was
the folder was deleted.
4. Using the command
man
determine which option of the command
ls
you need to use
the contents of not only the specified directory, but also its subdirectories to view
it.
5. Using the command
man
define a set of command options
ls
, which allows you
to sort the displayed list of folder contents
with a detailed description of files by the time of the last change.
6. Use the command
man

to view the description of the following commands:
cd
,
pwd
,
mkdir
,
rmdir
,
rm
. Explain the main options for these commands.
7. Using the information obtained with the command
history
, run the mod-
multiple commands can be selected and executed from the command buffer.

4.4. Report content

1. Title page with the number of the laboratory work and full name of the student.
2. Statement of the work goal.
3. Description of the task results:

— screenshots (screenshots) that capture the performance of laboratory work–
- listings (source code) of programs (if any)–
- results of program execution (text or screenshot, depending on
tasks).

4. Conclusions agreed upon for the purpose of the work.
5. Answers to security questions.

4.5. Security issues

1. What is the command line?
2. What command can I use to determine the absolute path of the current directory?
Give us an example.
3. Which command and options can be used to determine only the file type
and their names in the current directory? Please provide some examples.
4. How do I display information about hidden files? Please provide some examples.
5. What commands can I use to delete a file and directory? Can this be done?
by the same command? Please provide some examples.
6. How can I display information about the most recently completed users?
lem commands? jobs?
7. How can I use the command history for modified execution? When-
give examples.
8. Give examples of running multiple commands on the same line.
9. Give a definition and examples of escape characters.
10. Describe the output of information on the screen after the command is executed
ls
with the option
l
.
11. What is a relative file path? Please provide examples of how to use the following methods:
a specific and absolute path when executing a command.
12. How do I get information about the team you are interested in?
13. Which key or key combination is used for automatic padding
input commands?