Lab 4: Basic User Interaction Interface with Unix at the Command Line Level

4.1. Objective

Acquisition of practical skills of user interaction with the system using the command line.

4.2. Instructions for work

In an operating system such as Linux, user interaction with the system is usually done through the command line by entering commands line by line . In this case, command interpreters of the shell language are usually used: / bin / sh; /bin/csh; /bin/ksh.

Command format. A command in the operating system is a text written according to special rules (possibly with arguments), which is an indication of the performance of some functions (or actions) in the operating system. Usually the first word is the name of the command, the rest of the text is arguments or options that specify the action.

The general command format can be represented as follows:
<command_name><delimiter><arguments>

man command. The man command is used to view (online help) online manuals on the basic commands of a Linux-type operating system. Command format: man <command>

Example (printing information about the man command):

```
man man
```

To control how the result of a man command is executed, you can use the following keys:

- Space — move one page forward through the document;
- Enter — move forward one line in the document;
- q — exit the description view mode.

cd command. The cd command is used to move around the file system of a Linux-type operating system.

Note 1. A Linux-type OS file system is a hierarchical system of directories, subdirectories, and files, which are usually organized and grouped by function. The topmost directory in the hierarchy is called the root directory and is denoted by the symbol /. The root directory contains system files and other directories.

Command format: cd [path_to_directory]

To change to the user's home directory, use the cd command without options or cd ~. For example, the command

```
cd /afs/dk.sci.pfu.edu.ru/home
```

allows you to go to the directory /afs/dk.sci.pfu.edu.ru/home (if it exists), and in order to go up one directory, you should use:

```
cd ..
```

For more information about the cd command options, see the help with the man command:

```
man cd
```

pwd command. The pwd (print working directory) command is used to determine the absolute path to the current directory . Example (absolute name of user dharma's current directory):

```
pwd _
```

result:

```
/afs/dk.sci.pfu.edu.ru/home/d/h/dharma

```

Filename abbreviations. When working with commands, the arguments of which are the path to a directory or file, you can use the path abbreviation. Abbreviation symbols are given in Table. 4.1.

Table 4.1

Filename abbreviations

Symbol	Meaning
~	home directory
.	Current directory
..	Parent directory

For example, in the cd command to move around the file system, the abbreviated path notation can be used as follows (commands interleaved with the output of the pwd command):

```
pwd _

/afs/dk.sci.pfu.edu.ru/home/d/h/dharma

```

```
cd ..
pwd

/afs/dk.sci.pfu.edu.ru/home/d/h

cd ../..
pwd

/afs/dk.sci.pfu.edu.ru/home

cd ~/work
pwd

/afs/dk.sci.pfu.edu.ru/home/d/h/dharma/work
```

ls command. The ls command is used to view the contents of a directory.

Command format: ls [-options] [path] Example:

```
cd
cd ..
pwd

/afs/dk.sci.pfu.edu.ru/home/d/h

ls

dharma
```

Some files in the operating system are hidden from view and are usually used to set up a working environment. The names of such files start with a dot. In order to display the names of hidden files, you must use the ls command with the a option :

```
ls-a
```

You can also get information about file types (directory, executable file, link), for which the F option is used. When using this option, a symbol is displayed in the name field that determines the file type (see Table 4.2)

Table 4.2

Symbol that specifies the type of file

File type	Symbol
Directory	/
Executable Link	*
	@

To display detailed information about files and directories, you need to use the l option . This will display the following for each file and directory: information:

- file type,

- the right of access,

- number of links

- owner,

– size,

– date of the last revision,

is the name of the file or directory.

Example:

```
cd /
ls
```

Result:

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

In the same directory, the command

```
ls-alF
```

will give the following output:

```
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/
drwxr-xr-x 6 root root 4096 Aug. 5 2009home/
lrwxrwxrwx 1 root root 5 Jan. 12 22:01 lib -> lib64/
drwxr-xr-x 8 root root 4096 Jan. 30 21:41 media/
drwxr-xr-x 5 root root 4096 Jan. 17 2010 mnt/
drwxr-xr-x 25 root root 4096 Jan. 16 09:55 opt/
dr-xr-xr-x 163 root root 0 Feb. 17 13:17 proc/
drwxr-xr-x 31 root root 4096 Feb. 15 23:57 root/
drwxr-xr-x 2 root root 12288 Jan. 18 15:57 sbin/
drwxr-xr-x 12 root root 0 Feb. 17 13:17 sys/
drwxrwxrwt 12 root root 500 Feb. 17 16:35 tmp/
drwxr-xr-x 22 root root 4096 Jan. 18 09:26 usr/
drwxr-xr-x 17 root root 4096 Jan. 14 05:38 var/
```

mkdir command. The mkdir command is used to create directories.

Command format:

mkdir dirname1 [dirname2...]

An example of creating a directory in the current directory:

```
cd
pwd

/afs/dk.sci.pfu.edu.ru/home/d/h/dharma

ls

Desktop public tmp
GNUstep public_html work

mkdir abc ls


abc GNUstep public_html work
Desktop public tmp
```

Note 2: In order to create a directory at a specific location in the file system, permissions must be set correctly.

You can also create a subdirectory within an existing subdirectory:

```
mkdir parentdir
mkdir parentdir/dir
```

When given multiple arguments, multiple directories are created:

```
cd parentdir
mkdir dir1 dir2 dir3
```

Grouping can be used:

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

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

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

or

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

The following options are interesting:
- --mode (or -m) - setting access attributes;
- --parents (or -p) - create a directory along with its parents in relation to it directories.

Attributes are specified in numerical or symbolic notation:

```
mkdir --mode=777 dir
```

or

```
mkdir -m a+rwx dir
```

The --parents option (short form -p) allows you to create a hierarchical chain subdirectories, creating all intermediate directories:

```
mkdir -p ~/dir1/dir2/dir3
```

rm command. The rm command is used to remove files and/or directories.

Command format:

rm [-options] [file]

If you want to be prompted for confirmation to delete a file, you need to we can use the i option.

To remove a directory containing files, use the r option. No indication this option, the command will not be executed.

Example:

```
cd
mkdir abs
rm abc

  rm: abc is a directory

rm -r abc
```

If the directory is empty, then you can use the rmdir command. If removed directory contains files, then the command will not be executed - you need to use rm -r directory_name.

history command. The history command is used to display a list of previously executed commands . The commands displayed on the screen are numbered in the list. To any a command from the list displayed on the screen can be accessed by its number in the list, using the !<command_number> construct.

Example:

```
history
  1pwd _
  2ls _
  3ls -a
  4 ls -l
  5 cd /
  6 history

  !5
cd /
```

You can modify a command from the displayed list by using the following building structure:

!<command_number>:s/<what_to_change>/<what_to_change>

Example:

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

Note 3. If there are special characters in the given context (such as ".", "/", "*" , etc.), they must be preceded by an escape character \ (backslash).

Use of the symbol ";". If you need to perform several consecutive commands written on one line, then a semicolon character is used for this

Example:

```
cd; ls
```

4.3. Work sequence

1. Determine the full name of your home directory. More about this cat log will be performed in subsequent exercises.
2. Do the following:
   2.1. Change to the /tmp directory.
   2.2. Display the contents of the /tmp directory. To do this, use the ls command with various options. Explain the difference in the information displayed on the screen.
   2.3. Determine if there is a subdirectory named cron in the /var/spool directory?
   2.4. Change to your home directory and display its contents on the screen. Determine who owns files and subdirectories?
3. Do the following:
   3.1. In your home directory, create a new directory named newdir.
   3.2. In the ~/newdir directory, create a new directory named morefun.
   3.3. In your home directory, create three new directories with the names letters, memos, misk. Then remove those directories with one command.
   3.4. Try removing the previously created ~/newdir directory with the rm command . Check whether the directory has been deleted.
   3.5. Remove the ~/newdir/morefun directory from your home directory. Check if there was directory has been removed.
4. Use the man command to determine which ls command option to use . to view the contents of not only the specified directory, but also subdirectories, included in it.
5. Use the man command to define a set of options for the ls command to sort by last modified time the output listing of directory contents. with detailed file descriptions.
6. Use the man command to view descriptions of the following commands: cd, pwd, mkdir, rmdir, rm. Explain the main options of these commands.
7. Using the information obtained from the history command, run my modification and execution of several commands from the command buffer.

4.4. Report content

1. Title page indicating the number of laboratory work and the name of the student. 2. Formulation of the purpose of the work. 3. Description of the results of the task: **-** screenshots (screen shots) that record 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 with the purpose of the work. 5. Answers to control questions.

4.5. test questions

1. What is the command line? 2. What command can be used to determine the absolute path of the current directory? Give an example. 3. Which command and which options can be used to determine only the type of files and their names in the current directory? Give examples. 4. How to display information about hidden files? Give examples. 5. What commands can be used to delete a file and directory? Can it be done

by the same team? Give examples.

6. How can you display information about the last commands executed by the user? work? 7. How to use the history of commands for their modified execution? At

lead examples.

8. Give examples of running multiple commands on one line. 9. Define and give examples of escape characters. 10. Describe the output of information on the screen after executing the ls command with the option l.

11. What is a relative path to a file? Give examples of using relative and absolute paths when executing a command.

12. How to get information about the team you are interested in? 13. What key or key combination is used to automatically complete the entered commands?