«Київський фаховий коледж зв’язку»

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №5**

з дисципліни: «Операційні системи»

**Тема: «Знайомство з командами навігації по файловій системі та керування файлами та каталогами»**

Виконали

Студенти групи КСМ-03Б

Команда: Кучмій-Зікеєв

М.О, Зінченко І.О. та

Звєрьков А.Г.

Перевірив викладач

Сушанова В.С.

Київ 2022

**Завдання для попередньої підготовки.**

***Виконав студент Зінченко І.***

**1)** A file system is a way of organizing data, which is used by the operating system to save information in the form of files on storage media. Also, this term refers to a set of files and directories that are located on a logical or physical Device. It organizes files so that it is easier for the operating system to work with them: file system drivers are transferred to the OS with data about file names, their size, attributes, locations. The file system determines the maximum possible length of the file name, its maximum size and other parameters. The file system allows programs to bypass a set of fairly simple operations to perform an action on a somewhat abstract object that represents a file. At the same time, programmers do NOT need to deal with the details of the actual location of data on the disk, data buffering and other low-level problems of data transfer from a long-term memory device. All these functions of the file system are pregnant on their own. The file system allocates disk memory, supports file naming, maps file names to appropriate addresses in external memory, provides data access, supports partitioning, protection, and recovery of files. A variety of files systems can be sued with Linux. Commonly used file systems are ext\* family (ext, ext2, ext3 and ext4) and XFS. Silicon Graphics developed XFS, which is a journaling system with high performance. The ext (extended file system) was developed in early 1990’s. It was the first file system used in Linux operating system. Remy Card developed it by getting inspiration from the UFS (UNIX File System). On Linux, everything is a file. If something is not a file, then it is a process. Programs, audio, video, I/O devices and other devices are considered as files. In Linux, there is no difference between a file and a directory. A directory is simply a file containing names of a set of other files. Special files are a mechanism used for I/O (found in /dev). Sockets (another special file type) provide inter-process communication. Named pipes (much like sockets) are used for inter-process communication without network semantics. Windows uses FAT and NTFS as file systems, while Linux uses a variety of file systems. Unlike Windows, Linux is bootable from a network drive. In contrast to Windows, everything is either a file or a process in Linux. Linux has two kinds of major partitions called data partitions and swap partitions. Because of the existence of swap partitions, you never run out of memory in Linux (like in windows). In terms of recovery tools, only a limited number of tools can be used on Windows, while there is a large number of UNIX based recovery tools available for Linux file systems.

**2)** Filesystem Hierarchy Standard, FHS is a standard adopted to unify the location of general-purpose files and directories in the UNIX OS file system. Today, most UNIX-like systems follow these rules to one degree or another. For example, a typical user database is always stored in the /etc/passwd file. The process of developing a file system hierarchy standard began in August 1993 with attempts to organize the structure of directories and files in the Linux operating system. On February 14, 1994, FSSTND (Filesystem Standard), a Linux-specific file system standard, was released. Further versions were released on October 9, 1994 and March 28, 1995. In early 1996, the BSD development community joined the development of a new version of FSSTND with the goal of developing a standard suitable for all Unix-like operating systems. The name of the standard was changed to Filesystem Hierarchy Standard (FHS). FHS was supported by the Free Standards Group, a non-profit organization that includes powerful software and hardware developers such as HP, Red Hat, IBM, Dell. However, the majority of distribution developers, including those in the Free Standards Group, do not follow the standard 100%. In particular, paths specially created by the group, such as /srv/, are almost never used. Some Linux systems reject FHS and follow their own standard, such as GoboLinux. Since FHS started as an initiative of the Linux community, other UNIX and UNIX-like operating systems have completely ignored it in favor of their own systems, which are sometimes quite widespread. For example, Mac OS X uses names such as /Library/, /Applications/, and /Users/ along with the traditional UNIX hierarchy names.

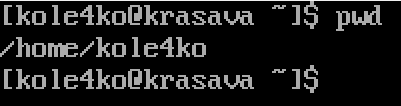
**3)** [pwd](https://dyclassroom.com/reference-linux/linux-commands-working-with-files-and-directories#pwd) - this command displays the present working directory where you are currently in. In the following example I am inside yusufshakeel directory which is inside the home directory. [ls](https://dyclassroom.com/reference-linux/linux-commands-working-with-files-and-directories#ls) - this command will list the content of a directory. In the following example we are listing the content of a directory. ls -la - this command will list all the content of a directory including the hidden files and directories. In the following example we are listing all the content of a directory. [mkdir](https://dyclassroom.com/reference-linux/linux-commands-working-with-files-and-directories#mkdir) - this command will create a new directory, provided it doesn't exists. In the following example we are creating a new directory **example**. [rmdir](https://dyclassroom.com/reference-linux/linux-commands-working-with-files-and-directories#rmdir) - this command will remove/delete an existing directory, provided it is empty. In the following example will are removing/deleting an existing directory **example**. [cd](https://dyclassroom.com/reference-linux/linux-commands-working-with-files-and-directories#cd) - this command is used to change directory. In the following command we are moving to root directory. [touch filename](https://dyclassroom.com/reference-linux/linux-commands-working-with-files-and-directories#touch) - this command will creates a new file. In the following example we are creating a new file **hello.txt**. [rm filename](https://dyclassroom.com/reference-linux/linux-commands-working-with-files-and-directories#rm) - this command will delete a file. In the following example we are deleting a file by the name **hello.txt**. rm -f filename - this command forcefully deletes a file. In the following example we are forcefully deleting a file by the name **hello.txt**. [cp file1 file2](https://dyclassroom.com/reference-linux/linux-commands-working-with-files-and-directories#cp) - this command copies the content of file **file1** into file **file2**. If file **file2** doesn't exists then it is created. If it exists then its content is overwritten. In the following example we are copying the content of file **hello.txt** to **hi.txt**. [mv](https://dyclassroom.com/reference-linux/linux-commands-working-with-files-and-directories#mv) - rename files and directories We can use mv command to rename files and directories. In the following example we are renaming file hello.txt to hi.txt. [cat filename](https://dyclassroom.com/reference-linux/linux-commands-working-with-files-and-directories#cat) - this will print the content of a file. In the following example we will get the content of the file **hello.txt** in the terminal.

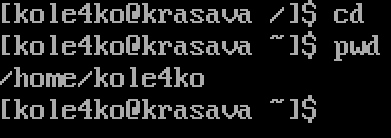
**Хід роботи.**

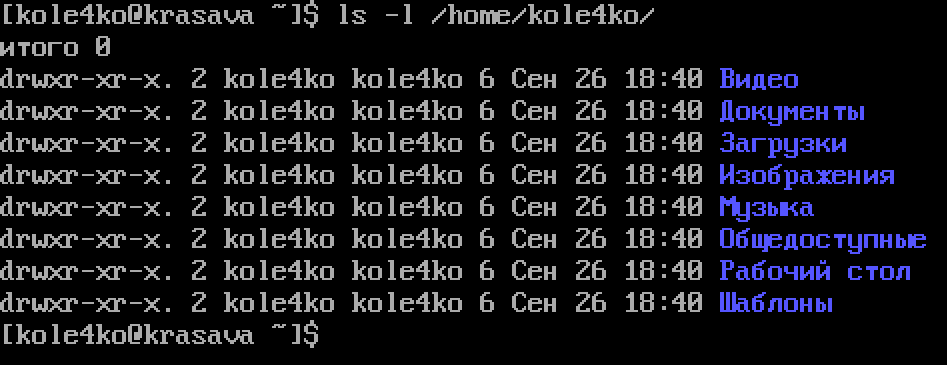
**2. *Виконав студент Кучмій-Зікеєв М.***

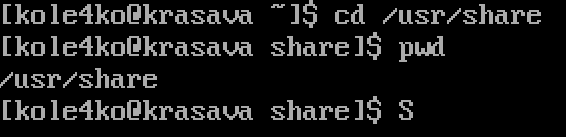
|  |  |
| --- | --- |
| Назва команди | Її призначення та функціональність |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**3. *Виконав студент Звєрьков А.***

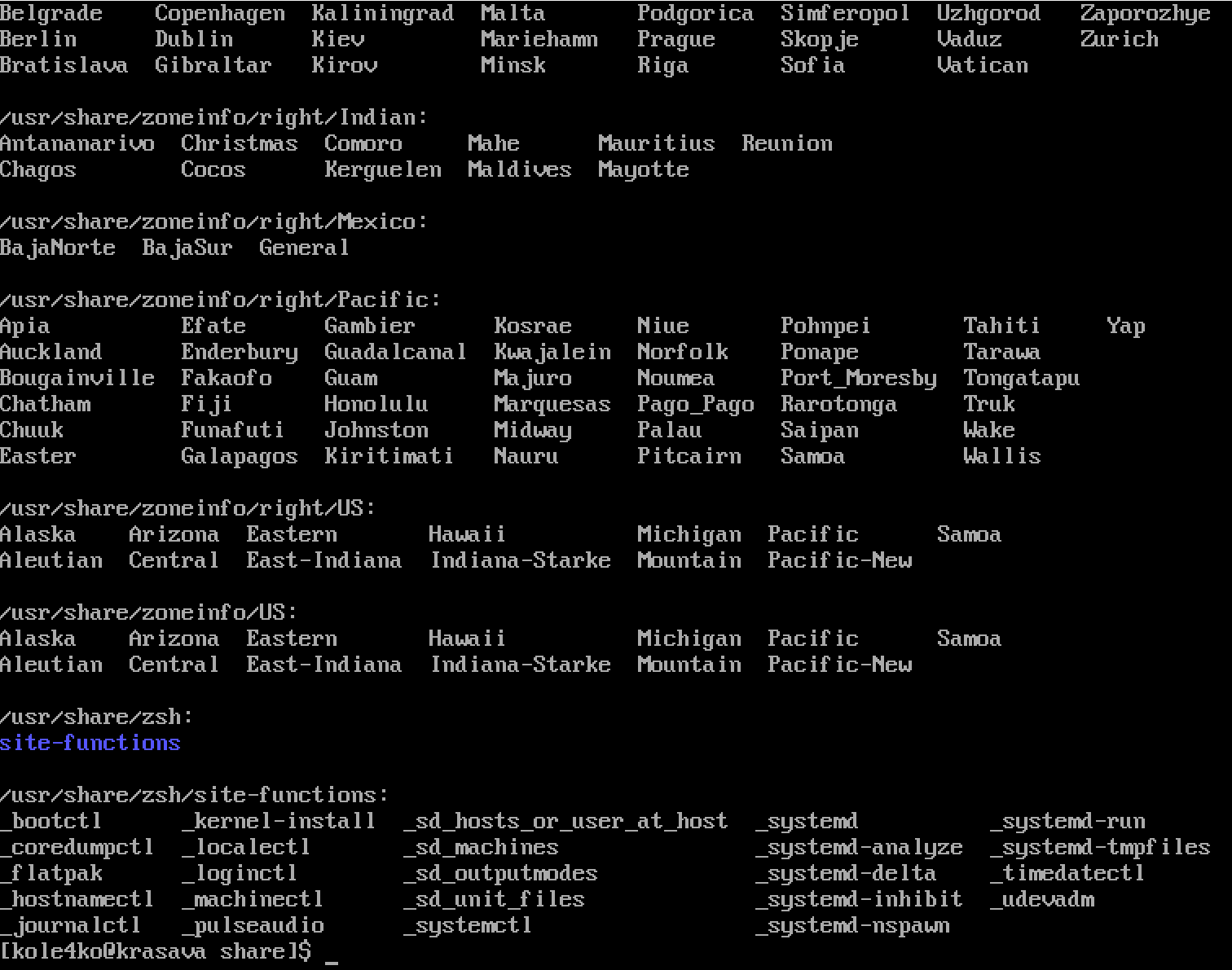
**1)**

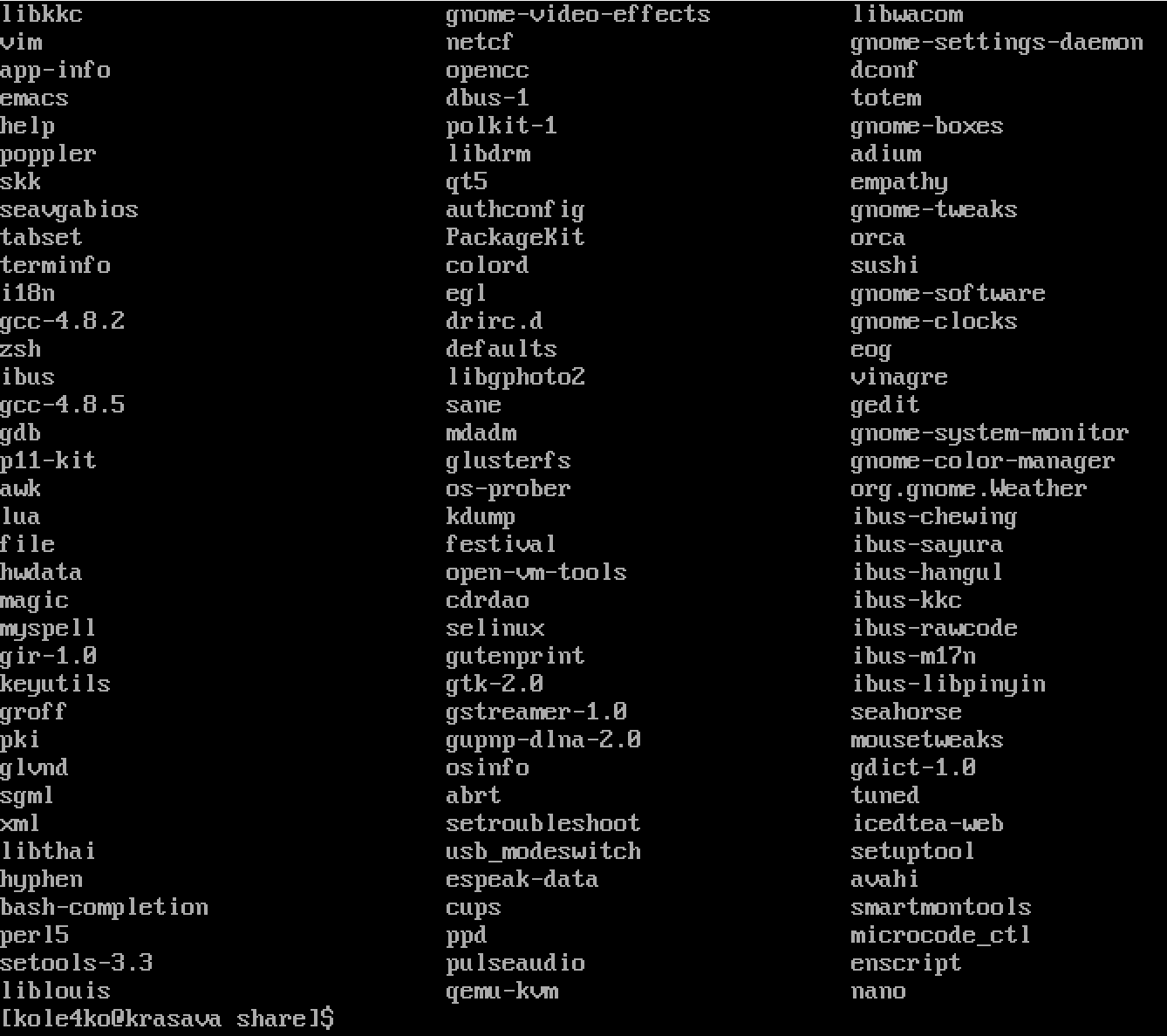
**2)**

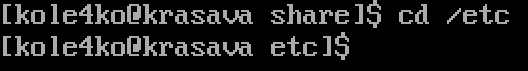
**3)**

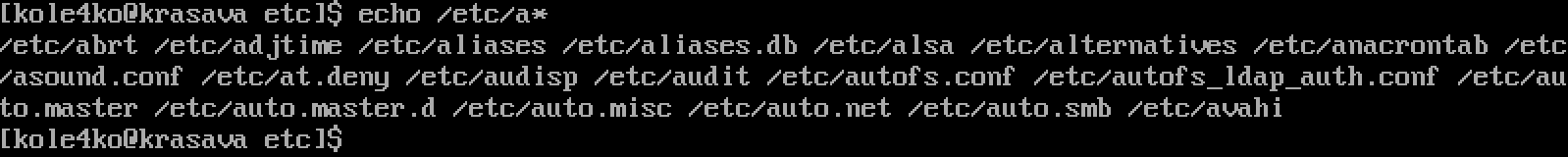
**4)**

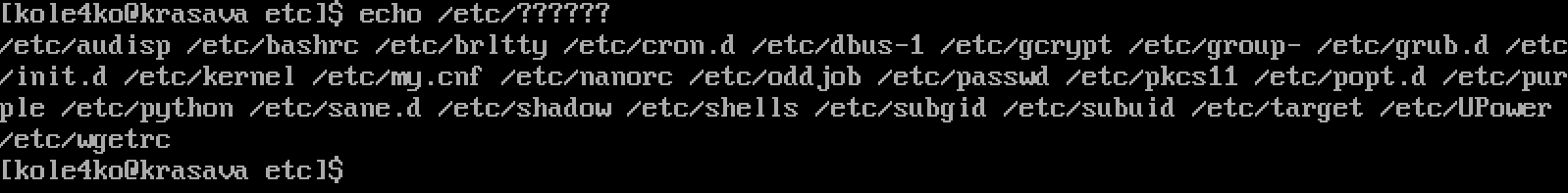
**5)**

****

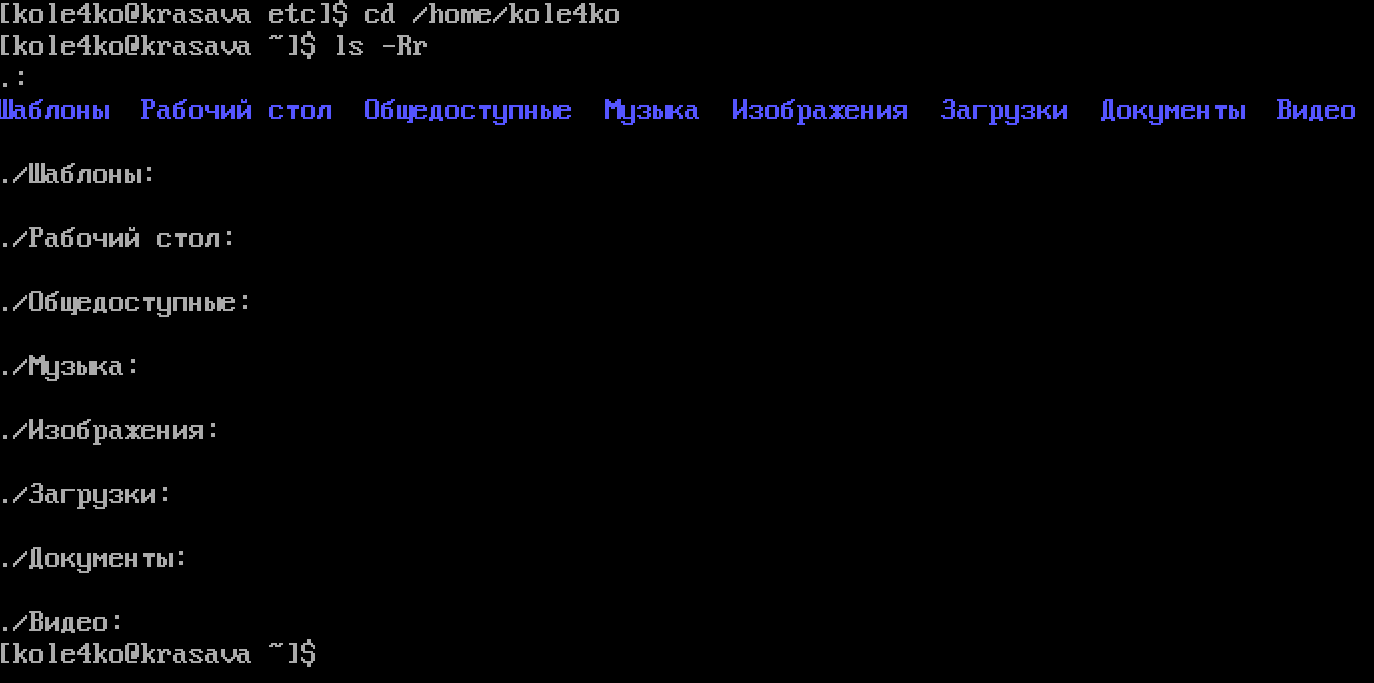
****

**6)**

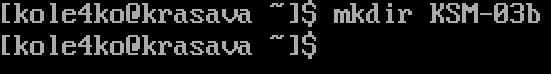
**7)**

**8)**

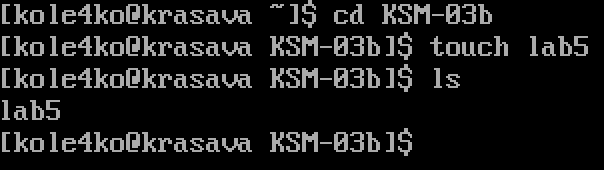
**9)**

****

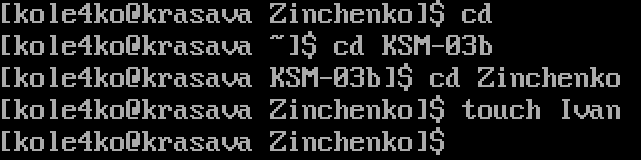
**10)**

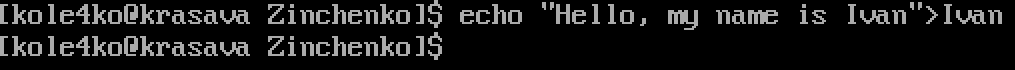
**11)**

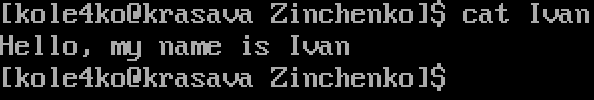
**12)**

**13)**

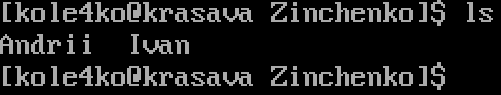
**14)**

**15)**

**16)**

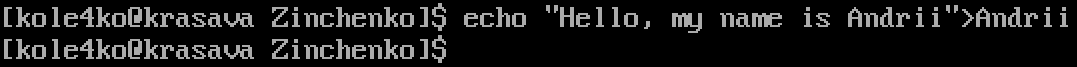
**17)**

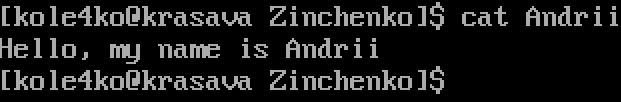
**18)**

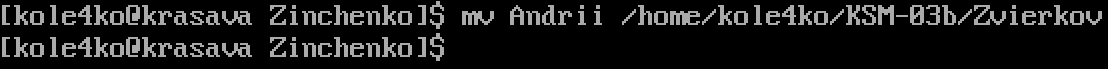
****

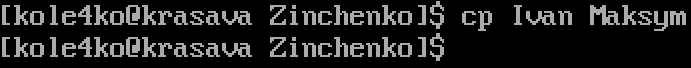
**19)**

**20)**

**21)**

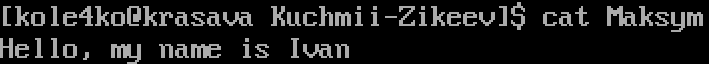
**22)**

**23)**

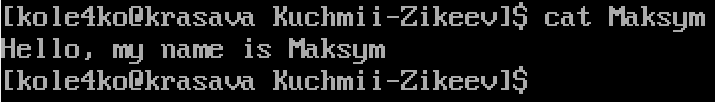
**24)**

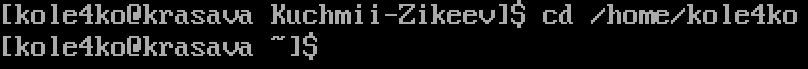
**25)**

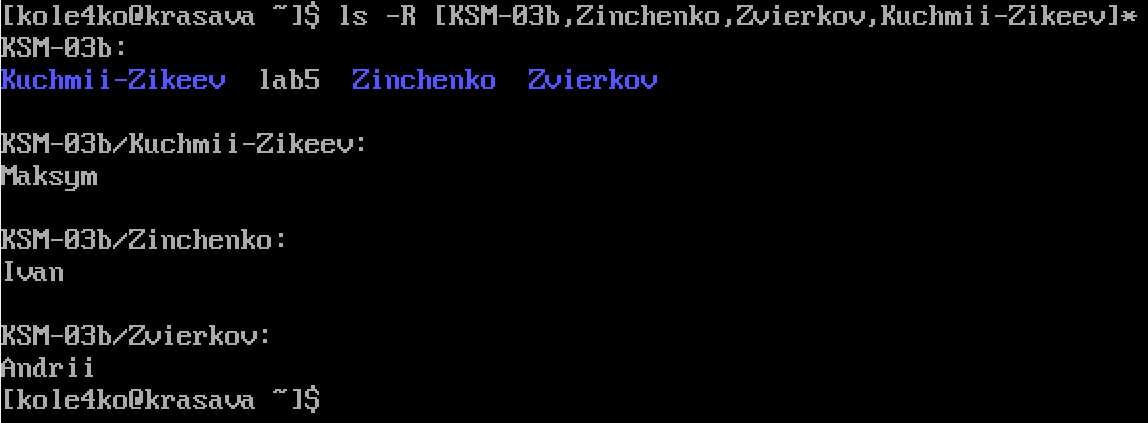
**26)**

**27)**

**28)**

**29)**

**30)**

**31)**

**4. *Виконав студент Звєрьков А.***

**1) cd /** —go to the root directory

**2) cd /home** — go to the home directory

**3) cd ~** — go to the home directory of current user

**4) cd** — go to the home directory of current user

**5) cd ..** — go up to one directory level

**6) cd ../..** — go to the directory that is two level above the current directory

**7) cd -** — go to the previous directory

**Відповіді на контрольні запитання**

***Виконав студент Зінченко І.***

**1)** The best way to find a user's home directory in Linux is to use the echo $HOME command. To do this, open a terminal and enter the following command: echo $HOME. This will print the value of the HOME environment variable, which is usually the home directory for the current user.

$HOME is a shell environment variable that contains the full path to the user's directory. The $HOME variable is automatically set by the system during installation and usually has the value /home/username. However, you can set the $HOME variable to any custom path as needed. To see the full path to your home directory, run the following echo $HOME command from your shell terminal.

Another way to find a user's home directory in Linux is to use the ~ symbol. This is a shortcut that represents the current user's home directory. To use it, just type ~ followed by the name of the user you want to find the home directory for. For example, if we want to find the home directory for the user 'jane', we would type echo ~jane.

The final way to find a user's home directory in Linux is to look in the /etc/passwd file. This file contains information about all users on the system, including their home directories. To view this file, type the following command: cat /etc/passwd. This will print the contents of the file. Scroll through it until you find the entry for the user you're looking for.

You can also grep username /etc/passwd to get information about a user, which includes that user's home directory.

**2)** With the help of the dir command, it is easy to view the contents of not only the root directory, but also any directory. For example, to find out about the contents of the TUTORIAL directory, you need to type the command:

D:\>dir TUTORIAL

That is, to view the contents of a nested directory, you need to specify the path to it from the current directory in the dir command. Note that a space should be placed after the dir command.

If you want to view one of the directories, for example, 8-A, when the root directory of the C: drive is active, you need to write the command:

C:\>dir d:\TUTORIAL\8-А

If the TUTORIAL directory is active, then to view the contents of the 8-A directory, you need to write a command

D:\TUTORIAL>dir 8-A

Team

D:\TUTORIAL>dir 8-A\\*.pas - will display the names of all files with the pas extension located in the 8-A directory.

**3)** Introduction – Empty file means files with zero bytes and in a plain text format — no data stored in a file. When you create a blank open office file, it contains extra data structure. The best way to create an empty file on Linux is to use the touch command. Data can be displayed on a screen (stdout) or in a concatenated file (under Linux or Unix) using cat as the command line system. The cat command can be used to change the name of a file or to add data to a file with text. You can find a blank file inside your terminal window by typing blank text. Cat commands are primarily used to read anderating files, but they can also be used to create new ones. Ensure that a new file has been created by following the redirection operator/x with the cat command. In this case, the cat redirection option is used to append a file, whereas the cat redirect option is used to redirect the file. To append text to a file, use echo.  
It is now time to say “Hello, World!” To test the file, go to the test.txt page.  
When you add this, a greeting will be sent. To get to the file test.txt, open it.

**4)** Using rsync:

rsync -av --delete source/ target

This would delete all the contents of the directory target that does not match the contents of the directory source, and would additionally copy the contents of source there.

The trailing / at the end of source/ is significant as without it, you would get a directory at target/source instead of making target a copy of source.

The -a (or --archive) option makes rsync copy timestamps and other metadata, and the -v (or --verbose) option makes rsync operate verbosely. Without --delete, no existing contents in target would be deleted (unless it had the same name as things in source in which case it would be updated).

**5)**

*- mv /work/tech/comp.png. /Desktop* — moving file to another directory

*- mv /work/tech/comp.png. /work/tech/my\_car.png* — renaming a file

*- mv /work/tech/comp.png. /Desktop/computer.png* — moving file to another directory and renaming it.

**Висновок**

Виконуючи лабораторну роботу, ми отримали практичні навички роботи з оболонкою Bash, ознайомилися з базовими командами навігації по фійловій систем, ознайомилися з базовими командами для керування файлами та каталогами.