"Kyiv Professional College of Communication"

Cycle Commission of Computer Engineering

REPORT ON THE IMPLEMENTATION

LABORATORY WORK №6

in the discipline: "Operating systems"

Topic: "Linux commands for archiving and compressing data. Working with text"

Performed by

student

of BICS-13 group

Zlata Volodymyrivna Los

Checked by the teacher

Sushanova V.S.

Kyiv 2024

**Objectives:**

1. Gaining practical skills in working with the Bash shell.

2. Familiarity with basic commands for archiving and compressing data.

3. Familiarization with basic actions when working with text in the terminal.

**Material support of classes:**

1. IBM PC type computer.

2. Windows operating system and virtual machine Virtual Box (Oracle).

3. GNU/Linux OS (any distribution).

4. Cisco Network Academy website netacad.com and its online courses on Linux

**Tasks for preliminary preparation:**

1. \*Read the brief theoretical information for the lab and make a small glossary of basic English terms on the purpose of commands and their parameters.

Lossy and Lossless Compression - while lossless compression preserves the original data without loss, lossy compression provides a higher compression ratio by losing some information.

gzip (short for GNU zip) - a file compression and recovery (decompression) utility that uses the Deflate algorithm.

bzip2 is a free and open source command line utility (and algorithm) for data compression.

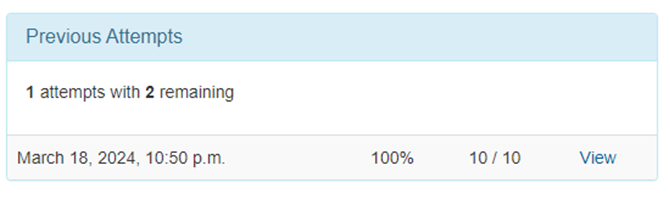
xz is a lossless compression computer program that uses the LZMA2 algorithm and the corresponding file format.

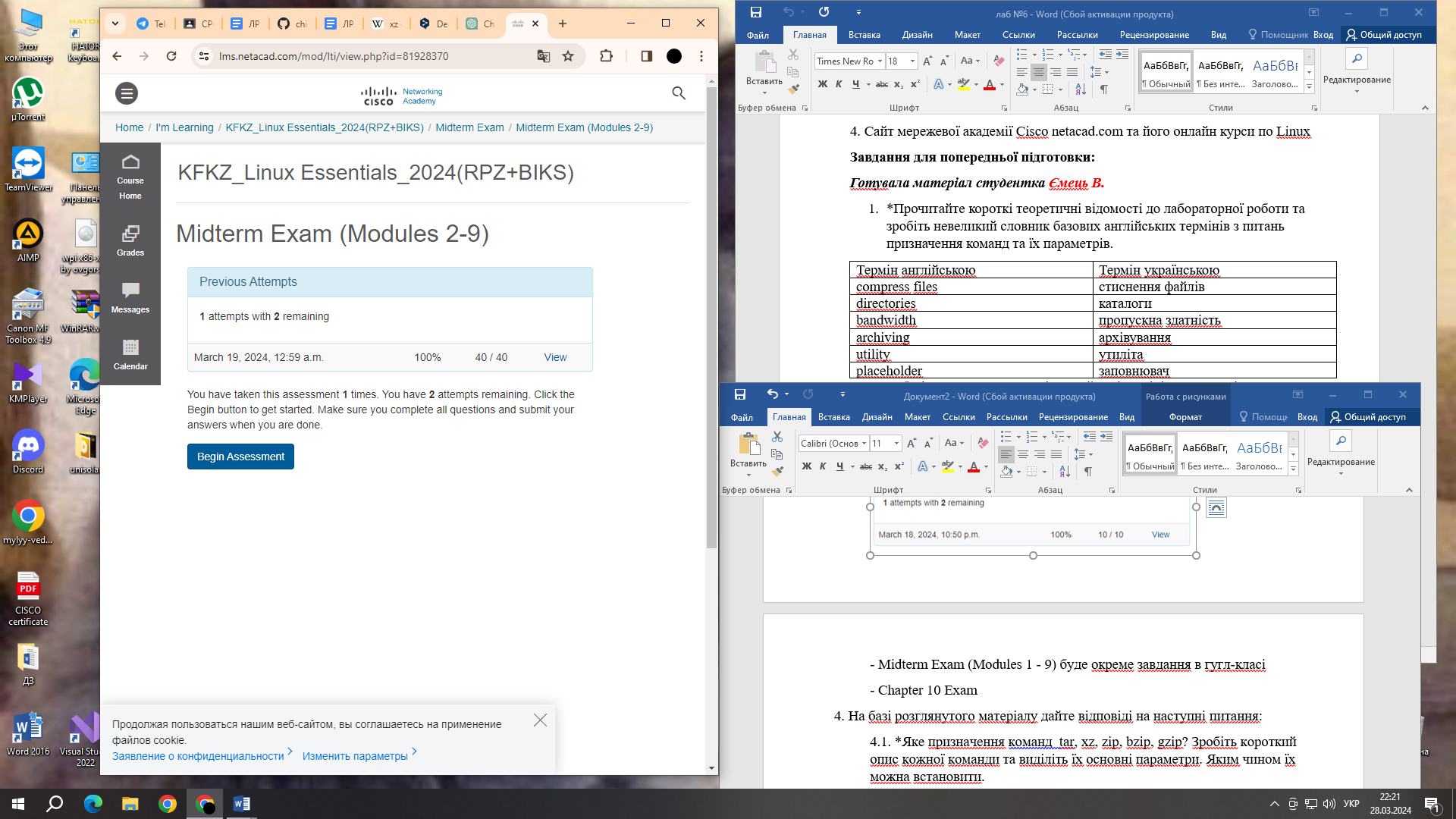
2. Take the NDG Linux Essentials online course from Cisco Academy:

* Chapter 09 - Archiving and Compression
* Chapter 10 - Working With Text

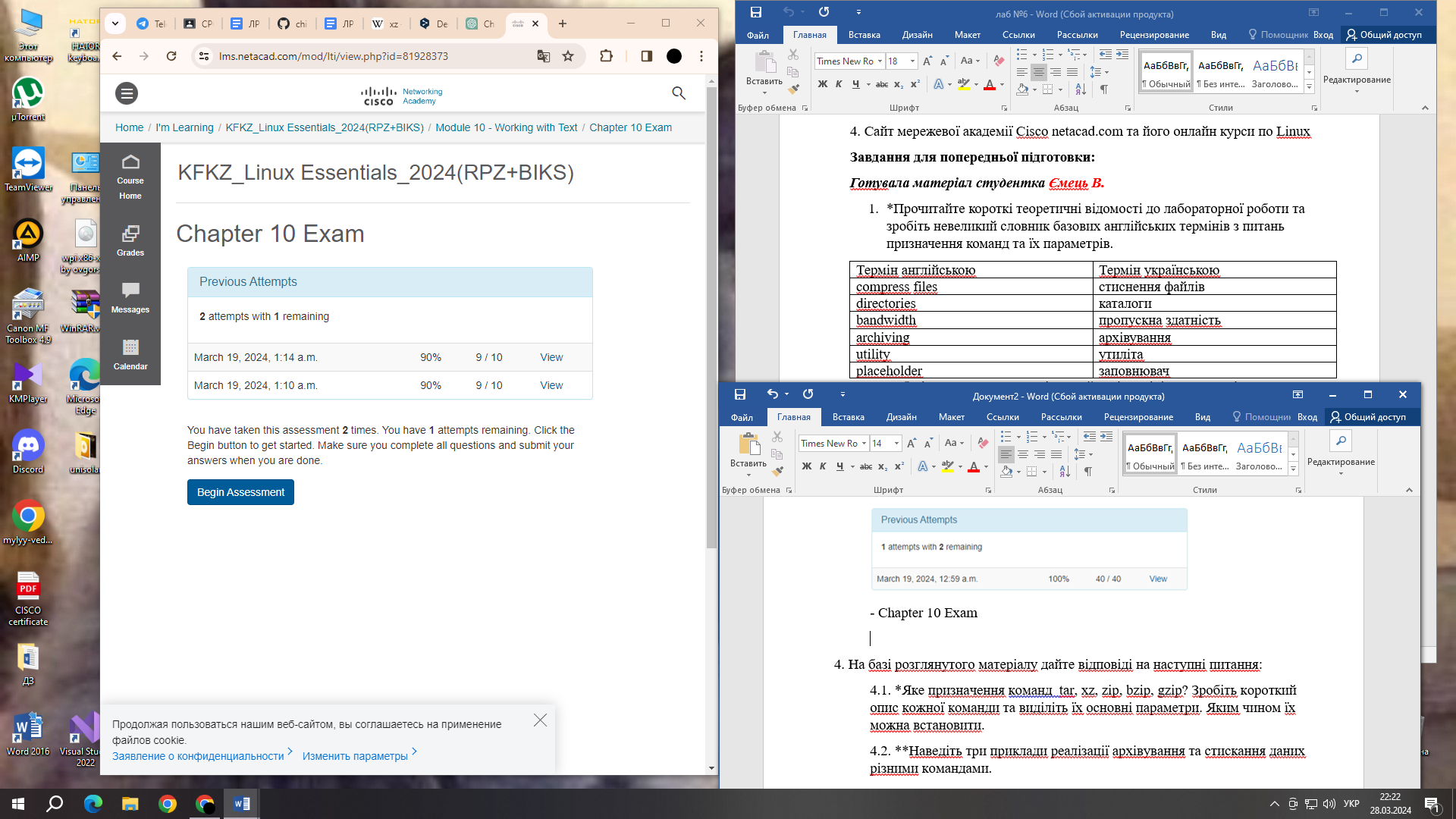
3. Take the NDG Linux Essentials course exams on the following topics:

- Chapter 09 Exam



- Midterm Exam (Modules 1 - 9) 

- Chapter 10 Exam



4. Based on the material you have read, answer the following questions:

4.1 \*What is the purpose of the commands tar, xz, zip, bzip, gzip? Make a brief description of each command and highlight their main parameters. How can they be installed.

1) tar: The tar (Tape ARchive) command is used to create archive files and pack directories into a single file. The main parameters are:

* -c: Create a new archive
* -x: Unpack the archive
* -f: Specify the name of the archive

2) xz: The xz command is used to compress files. It provides very efficient compression, but can be a bit slow compared to other methods. Main parameters:

* -z: Compress file

3) zip: The zip command creates archives from files. It is one of the most common on Windows, but is also available on many UNIX-like operating systems. The main options are:

* -r: Recursively add files and folders
* -q: Quiet mode (no message output)

4) bzip2: The bzip2 command is used to compress files. It provides a high compression ratio, but can be a bit slow. Main parameters:

* -z: Compress file

5) gzip: The gzip command is also used to compress files. It is very fast and provides a good level of compression. The main parameter is:

* -r: Compress recursively

On most UNIX-like operating systems, these utilities are installed by default. However, if they are not, you can install them using your operating system's package manager. For example, on Debian or Ubuntu, this can be done by running the command 'sudo apt install <package\_name>', where '<package\_name>' is the name of the package you want to install (e.g. tar, zip, gzip, etc.).

4.2 \*\*Give three examples of how to implement data archiving and compression using different commands.

1) Using 'tar' and 'gzip' to create an archive and compress it:

First, let's create an archive using 'tar', which packs the 'Documents' directory into a file called 'documents.tar': 'tar -cf documents.tar Documents/'

Now let's use 'gzip' to compress the 'documents.tar' archive:

'gzip documents.tar'

After executing these commands, you will have a compressed archive 'documents.tar.gz'.

2) Using 'zip' to create an archive and compress it:

Create an archive 'documents.zip' containing the 'Documents' directory:

'zip -r documents.zip Documents/'

From this command, 'zip' will create an archive 'documents.zip' with all the files and folders contained in the 'Documents' directory.

3) Using 'tar' and 'xz' to create an archive and compress it:

First, let's create the archive using 'tar', which packs the 'Pictures' directory into a file called 'pictures.tar': 'tar -cf pictures.tar Pictures/'

Now let's use 'xz' to compress the 'pictures.tar' archive:

'xz pictures.tar'

After running these commands, you will have a compressed archive 'pictures.tar.xz'.

4.3 \*What is the purpose of the cat, less, more, head and tail commands? Make a brief description of each command and highlight their main parameters. How can they be set?

1) cat: The cat (concatenate) command is used to display the contents of one or more files to standard output. Basic parameters:

* Has no basic parameters, but can take a list of file names as arguments.

2) less: The less command allows you to scroll up and down the textual content of a file to view it conveniently. Main parameters:

* Open the file for viewing: less <file\_name>.
* Navigate up and down: Use the cursor keys or the scroll keys.

3) more: The more command is also used to gradually display the contents of a file on the screen, but is less flexible than less. Basic options:

* Open the file for viewing: more <file\_name>.
* Navigate up and down: Use the Space key to scroll down one screen or the Enter key to scroll down one line.

4) head: The head command displays the first few lines of a file. Basic parameters:

* Print the first N lines: head -n N <file\_name>.

5) tail: The tail command displays the last few lines of the file. Basic parameters:

* Print the last N lines: tail -n N <file\_name>.

These commands are standard on most UNIX-like operating systems, so they are usually installed by default. If they are not, you can install them using your operating system's package manager. For example, on Debian or Ubuntu, this can be done with the command 'sudo apt install <package\_name>', where '<package\_name>' is the name of the package you want to install (e.g. 'coreutils' for 'cat', 'less' for 'less', 'more', 'head', and 'tail').

4.4 \*\*Explain how the shell works with channels, streams, and filters

The basic principles of their operation:

1) Input, output, and error streams:

* Every command in UNIX has three streams: the standard input (stdin), the standard output (stdout), and the standard error stream (stderr).
* The stdin stream is usually associated with the keyboard, and stdout and stderr are usually associated with the screen. However, they can be redirected.

2) Pipes:

* Channels (|) allow you to redirect the output of one command to some other commands.
* For example, command1 | command2 will take the output from command1 and pass it as input to command2.

3) Filters:

* Filters are programs that process or modify input data and output the result.
* Commands such as grep, sed, awk, sort, uniq are examples of filters.
* They can be used for searching, replacing, sorting, extracting, etc.

4.5 \*What is the purpose of the grep command?

The main purpose of the grep command is to find strings that match a specified pattern in text input or files. The user can specify the pattern in a regular expression format for more flexible and precise searching.

5. Prepare an initial version of the report in electronic form:

- Cover sheet, topic and purpose of the work

- Glossary of terms

- Answers to p.4.1 and p.4.5 from the preliminary preparation tasks

**Progress:**

1. Initial work in CLI mode in the Linux family of operating systems:

1.1. Start the VirtualBox virtual machine, select CentOS, and start it. Log in to the system as a user: CentOS, login password: reverse (if you are performing the LP in 401) and launch the terminal.

Start the virtual machine Ubuntu\_PC (if you perform the task of the PL 1.2. through the netacad academy)

1.3. Start your Linux operating system (if you are working on your own PC and have installed it) and launch the terminal.

2. Work through all the sample commands presented in the labs of the NDG Linux Essentials course - Lab 9: Archiving and Compression and Lab 10: Working With Text. Create a table to describe these commands.

|  |  |
| --- | --- |
| **Name of the commands** | **Its purpose and functionality** |
| mkdir mybackups | Creating a new mybackups directory in the user's home directory |
| tar -cvf mybackups/udev.tar /etc/udev | The tar command is used to combine several files into a single file. In this case, the contents of the /etc/udev directory will be saved to the udev.tar archive in the mybackups directory. The -c option tells the tar command to create a tar file. The -v option stands for "verbose," which tells the tar command to show what it is doing. The -f option is used to specify the name of the tar file. |
| tar -xvf udev.tar.gz | Unpacking the archive contained in the file named udev.tar.gz |
| tar –tvf mybackups/udev.tar | Displays the contents of the udev.tar archive, which is located in the mybackups directory |
| tar -rvf udev.tar /etc/hosts | Add (or update) the /etc/hosts file to an existing udev.tar archive |
| tar –tvf udev.tar | Display the contents of the udev.tar archive without unpacking it |
| ls -l words | Display detailed information about a file or directory named words |
| gzip words | Compresses files into gzip format, which reduces their size and reduces the amount of disk space they take up |
| gunzip words.gz | Unpack gzip-compressed files to restore their original contents |
| zip -r udev.zip /etc/udev | Create ZIP archives that contain one or more files or directories |
| unzip -l udev.zip | Unpack files that have been compressed in a ZIP archive |
| rm -r etc | Delete files or directories from the file system |
| cat | Print the contents of a text file to standard output |
| find ~ -name "\*bash\*" | Search for files and directories in the file system based on various criteria |
| find /etc -name hosts | Search for files and directories according to specified criteria in a given directory and its subdirectories |
| ls -l /etc | more | Displays the contents of a directory (a list of files and subdirectories) or the properties of specific files, depending on the arguments passed |
| cut -d: -f1 /etc/passwd | Select and display specific parts of lines from a given data stream or file |
| head | Print the first lines of the specified file or data stream |
| tail /etc/passwd | Display the last lines of the specified file or input stream |
| ls /etc | tail -5 | Display the last five lines of output from the ls /etc command, that is, display a list of files and subdirectories in the /etc directory, and then display the last five lines of the list |
| grep sshd passwd | Search for lines of text in one or more files or in standard input that match a specified pattern |
| cd /etc | The cd command changes the current working directory to /etc |
| egrep 'no(b|n)' passwd | It is an extended version of the grep command and is used to search for lines in the entered text stream or in files, using regular expressions to define search patterns |

3. Familiarize yourself with the tar command and use it to perform the following actions in the terminal:

create a file with the .tar extension

- tar -cf archive.tar file1 file2 file3

create a file with the .tar extension that consists of several files and directories at the same time

- tar -cf archive.tar file1 directory1

view the contents of the file

- tar -tf archive.tar

extract the contents of the tar file

- tar -xf archive.tar

create an archive tar file compressed with bzip

- tar -cjf archive.tar.bz2 file1 file2 directory1

extract the contents of the tar bzip file

- tar -xjf archive.tar.bz2

create an archive tar file compressed with gzip

- tar -czf archive.tar.gz file1 file2 directory1

extract the contents of the tar gzip file

- tar -xzf archive.tar.gz\ 4.

4. \*How the output streams will be redirected to bash for the following actions with commands (denoted as cmd) and files (denoted as file):

|  |  |
| --- | --- |
| **Command** | **What does the command do?** |
| cmd 1> file | The standard output of the cmd command will be redirected to the file. Errors (STDERR) will remain in place. |
| cmd > file | The same as the previous one. Shortened version of the record, 1 is optional. |
| cmd 2> file | The error output (STDERR) of the cmd command will be redirected to the file. The standard output (STDOUT) remains in place. |
| cmd >> file | The standard output of the cmd command is appended to the end of the file. Errors (STDERR) will remain in place. |
| cmd &> file | If you want to redirect both the standard output and errors to the file, this is the option for you. This is equivalent to cmd > file 2>&1. |
| cmd > file 2>&1 | Same as cmd &> file. |
| cmd >> file 2>&1 | The output (STDERR and STDOUT) of the cmd command will be appended to the end of the file. |
| cmd 2>&1 > /dev/null | STDOUT and STDERR will be redirected to the same location (/dev/null, which rejects input), so both outputs will be lost. |
| cmd 2> /dev/null | Errors (STDERR) will be redirected to /dev/null, i.e. lost, and the standard pin will remain in place. |
| cmd1 | cmd2 | The output of cmd1 will be passed as input to cmd2. This is the so-called "pipeline" or "pipe". |
| cmd1 2>&1 | cmd2 | The output (STDERR and STDOUT) of cmd1 will be passed as input to cmd2 |

5. \*\*Review the following examples and explain what these commands do and what type of flow redirection they use:

|  |  |  |
| --- | --- | --- |
| **Command(command container)** | **What does the command do?** | **What is the redirection flow?** |
| $echo "It is a new story." > story | This command uses the echo command to print the text "This is a new story." to the standard output (STDOUT). Using redirect >, this output is redirected to a file named story. Thus, this line will be written to the story file. | This is using the output redirection (STDOUT). The output of the echo command ("It is a new story.") will be sent to a file named story. |
| $ date > date.txt | The date command returns the current date and time. Using redirect >, this date and time are written to a file named date.txt. So the contents of the date.txt file will be the current date and time. | This is also the use of output redirection (STDOUT). The result of the date command (the current date and time) will be written to a file named date.txt. |
| $ cat file1 file2 file3 > bigfile | The cat command is used to display the contents of file1, file2, and file3. With the > redirect, this content will be written to the bigfile file. So, the contents of file1, file2, and file3 will be merged into the bigfile. | This is also an output redirection (STDOUT). The contents of file1, file2, and file3 will be combined and written to a file named bigfile. |
| $ls -l >> directory | The ls -l command displays a list of files in the current directory along with detailed information about each file. With the >> redirect, this output is appended to the end of a file named directory. So the file information will be appended to the directory file. | This is a redirection of output with append (STDOUT). The output of the ls -l command (a list of files in the current directory with detailed information) is appended to the end of a file named directory. |
| $ sort < file1\_unsorted > file2\_sorted | The sort command is used to sort the contents of the file file1\_unsorted. Using the < redirection, the contents of this file will be used as input to the sort command. The sorting result will be written to the file file2\_sorted. | This is the use of input and output redirection (STDIN and STDOUT). The contents of file1\_unsorted will be used as input to the sort command, and the sorted result will be written to file2\_sorted. |
| $ find -name '\*.txt' > file.txt 2> /dev/null | The find -name '\*.txt' command is used to search for files with the .txt extension. Using redirect >, the list of such files will be written to file.txt. However, with the redirect 2> /dev/null, all error messages will be discarded and not printed. | This is a combined use of output redirection and errors (STDOUT and STDERR). The result of the find -name '\*.txt' command (a list of files with the .txt extension) will be written to a file named file.txt, and any error messages will be discarded. |
| $ cat file1\_unsorted | sort > file2\_sorted | This command uses a pipeline. The contents of the file file1\_unsorted are passed as input to the sort command, which sorts it. The result of the sort is written to the file file2\_sorted. | This is the use of a pipeline. The contents of the file file1\_unsorted will be passed to the sort command, and the result of the sort will be written to the file file2\_sorted. |
| $ cat myfile | grep student | wc -l | This command also uses a pipeline. First, the contents of the file myfile are passed as input to the grep student command, which searches for all lines containing the word "student". The result of this search is then passed to the wc -l command, which counts the number of lines. Thus, this command counts the number of lines in the myfile file that contain the word "student". | This is also the use of a pipe. The contents of the myfile file will be passed to the grep student command, which will filter out lines containing the word "student". The filtering result will be passed to the wc -l command, which will count the number of lines. |

**Control questions:**

1. Provide a comparative description of compression and archiving.

**Compression:**

- Objective: Compression is aimed at reducing the size of files by removing redundant information or applying special compression algorithms.

- Examples of algorithms: Gzip, Bzip2, LZMA, Deflate (used in ZIP and gzip formats).

- Usage: Typically used to compress a single file, not necessarily a group of files, but can be applied to multiple files in the case of archiving.

- Advantages: Faster execution, removes unnecessary information, allows you to compress a file into a smaller volume.

- Disadvantages: Usually applies to a single file, so it does not allow you to create structured archives with many files.

**Archiving:**

- Purpose: Archiving is aimed at creating structured packages or archives that can contain one or more files and directories.

- Examples of archive formats: ZIP, TAR (and TAR.GZ, TAR.BZ2), RAR, 7z.

- Usage: Typically used to create an archive containing many files and directories for easy storage, transportation, or backup.

- Advantages: Allows you to combine many files and directories into one archive, preserves the directory structure, can contain metadata and other information about files.

- Disadvantages: Usually takes more time to create an archive compared to compression, but this time can vary depending on the size and number of files.

2. What programs, other than those given in the work, can be used to compress and archive files and directories in Linux? Give examples and a brief description.

**XZ Utils:**

- XZ Utils provides tools for compressing and decompressing files using the LZMA2 compression format. It usually has better compression compared to other compression algorithms.

- Example of use: xz, unxz.

**Ark:**

- Ark is a graphical archive management tool based on KDE. It supports a variety of archive formats and provides options for creating, opening, viewing, editing, and extracting archives.

- Example of use: A graphical user interface is available.

**P7zip:**

- P7zip is an implementation of the 7-Zip program for Unix-like operating systems. It supports compressing and decompressing files in the 7z format, which usually provides very good compression.

- Example usage: 7z, 7za, 7zr.

3. \*Compare the compression algorithms used in the commands (programs) used in Linux. Which of the algorithms can be considered the fastest and most efficient?

- **Gzip (DEFLATE):** Gzip uses the DEFLATE compression algorithm, which is a combination of the LZ77 and Huffman coding algorithms. It is widely used because of its efficiency and compression speed.

- **Bzip2:** Bzip2 uses the Burrows-Wheeler transform (BWT) compression algorithm combined with the move-to-front transform (MTF) compression algorithm and the Huffman coding algorithm. Although it usually provides better compression than Gzip, it is usually slower.

- **XZ (LZMA/LZMA2):** XZ uses the LZMA compression algorithm or its modification LZMA2. It provides high compression and is the only program that usually provides better compression than Bzip2. However, it can be slower than Gzip or Bzip2.

4. \*Describe the compression and archiving software that can be used on your mobile phone.

- Archiving: iOS has a built-in file archiving feature that allows you to create ZIP archives from files and directories. You can compile a list of files and directories that you want to archive, and then create a ZIP archive to save them together. This feature is available in the standard file manager on iPhone or in other apps that support archiving.

- Files by Google: This app is available for iPhone and provides tools for managing files on the device. It also has an archiving feature that allows you to create ZIP archives from files and directories right on your phone.

5. \* Describe and compare the software tools for compressing and (de)archiving data in Windows operating systems.

**WinZip:**

- WinZip is one of the most popular file compression and archiving software in the Windows environment. It supports a variety of archive formats, including ZIP, ZIPX, RAR, 7Z, TAR, and many others. WinZip also has the capabilities to password protect archives and encrypt files.

- Advantages: Easy to use, wide range of supported formats, high compression rate.

- Disadvantages: Paid version has limitations, free version has limited functionality and features.

**7-Zip:**

- 7-Zip is a free and open-source file compression and archiving software for Windows. It supports various archive formats, including 7z, ZIP, TAR, GZIP, BZIP2, and many others. 7-Zip has a high level of compression and can handle very large files.

- Advantages: Free, open source, high compression level, large list of supported formats.

- Disadvantages: Less intuitive user interface compared to commercial counterparts.

**Windows Built-in ZIP Utility:**

- The Windows operating system has a built-in tool for working with ZIP archives. This tool allows you to create, open, and edit ZIP archives without the need to install additional software.

- Advantages: Built into the operating system, no additional installation required, easy to use.

- Disadvantages: It has limited functionality and features compared to other programs.

6. \*\*Explain how data compression and archiving can be used for data backup. What other system administration tasks can it be used for?

Data compression and archiving are important tools for backing up and managing data efficiently in computer and network administration systems. Here are some ways in which they can be used:

**1. Data backup:**

- Data compression and archiving can reduce the amount of copies of data stored on backup media. This saves space on backup media and reduces the cost of purchasing it.

- Using compression also speeds up the backup process because less data takes less time to transfer and store.

**2. Data transfer over the network:**

- When transmitting large amounts of data over a network, data compression and archiving can reduce the transfer time and amount of network bandwidth used.

- The reduced amount of data can also reduce the cost of transferring data over paid network channels.

**3. Optimization of data storage:**

- Data compression and archiving can optimize storage space on servers and storage devices. Less data takes up less disk space.

- This can be important for system administration, as it reduces the need for large storage devices and increases resources that can be used for other tasks.

**4. Archiving old data:**

- Old data that is no longer actively used can be compressed and archived to reduce the space it takes up on servers.

- This allows you to store old data for possible recovery or auditing without taking up a lot of space on active systems.

**5. Data protection:**

- Some archiving software allows you to protect your data with a password or encryption. This helps to maintain the confidentiality of the archived data and protect it from unauthorized access.

7. \*\*What is the purpose of the /dev/null file directory?

The main purpose of /dev/null is to provide a mechanism for removing unnecessary output or unnecessary data from the system when it is no longer needed. For example, if you want to send the output of a program to a "black hole" so that it doesn't take up disk space or display unnecessary data on the screen, you can redirect the output to /dev/null.

**Conclusion:** I have gained practical skills in working with the Bash shell. I got acquainted with the basic commands for archiving and compressing data. I got acquainted with the basic actions when working with text in the terminal.