“Київський фаховий коледж зв’язку”

Циклова комісія комп’ютерної та програмної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №6**

з дисципліни: «Операційні системи»

**Тема: «Команди Linux для архівування та стиснення даних. Робота з текстом»**

Виконала студентка

групи РПЗ-13а

Балджі В.В.

Перевірив викладач

Сушанова В.С.

Київ 2024

**Тема: “Команди Linux для архівування та стиснення даних. Робота з текстом”**

**Мета роботи:**

1. Отримання практичних навиків роботи з командною оболонкою Bash.
2. Знайомство з базовими командами для архівування та стиснення даних.
3. Знайомство з базовими діями при роботі з текстом у терміналі.

**Матеріальне забезпечення занять:**

1. ЕОМ типу IBM PC.

2. ОС сімейства Windows та віртуальна машина Virtual Box (Oracle).

3. ОС GNU/Linux (будь-який дистрибутив).

4. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

**Завдання для попередньої підготовки:**

1. \*Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеликий словник базових англійських термінів з питань призначення команд та їх параметрів.

|  |  |
| --- | --- |
| **Термін англійською** | **Термін українською** |
| ratio | співвідношення |
| occurrence | виникнення |
| drawback | недолік |
| inherent | притаманний |
| to implement | реалізувати |
| backup | резервне копіювання |

1. Вивчіть матеріали онлайн-курсу академії Cisco “NDG Linux Essentials”:

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

1. Пройдіть тестування у курсі NDG Linux Essentials за такими темами:

* Chapter 09 Exam
* Midterm Exam (Modules 1 - 9) ***буде окреме завдання в гугл-класі***
* Chapter 10 Exam

1. На базі розглянутого матеріалу дайте відповіді на наступні питання:
   1. \*Яке призначення команд *tar*, *xz*, *zip*, *bzip*, *gzip*? Зробіть короткий опис кожної команди та виділіть їх основні параметри. Яким чином їх можна встановити.

These commands are designed to either merge multiple files into a single file or compress a large file into a smaller one. In some cases, the commands will perform both functions.

The tar command is used to merge multiple files into a single file. By default, it does not compress the data.

The -c option tells the tar command to create a tar file. The -v option stands for "verbose", which instructs the tar command to demonstrate what it is doing. The -f option is used to specify the name of the tar file. To create a tar file that is compressed use -z option. To add a file to an existing archive, use the -r option to the tar command.

Use gzip, bzip2, xz  to compress a file.

The zip archive type is more commonly used by other operating systems, such as Windows. If you are planning to share an archive with Windows users, it is usually preferred to use the zip archive type. Unlike gzip and bzip2, when a file is compressed with the zip command, a copy of the original file is compressed and the original remains uncompressed.

With the bzip2, gzip, and zip commands the -r option must be specified in order to perform recursion into subdirectories.

* 1. \*\*Наведіть три приклади реалізації архівування та стискання даних різними командами.

tar –zcvf mybackups/udev.tar.gz /etc/udev

tar cjvf bzipcompressed.tar.bz2 directory2

tar cJvf xzcompressed.tar.xz directory3

* 1. \*Яке призначення команд cat, less, more, head and tail? Зробіть короткий опис кожної команди та виділіть їх основні параметри. Яким чином їх можна встановити

The cat command, short for concatenate, is a simple but useful command whose functions include creating and displaying text files, as well as combining copies of text files.

The less command provides a very advanced paging capability. It is usually the default pager used by commands like the man command. The more command has been around since the early days of UNIX. While it has fewer features than the less command, however, the less command isn't included with all Linux distributions. The more command is always available.

The head and tail commands are used to display only the first few or last few lines of a file, respectively (or, when used with a pipe, the output of a previous command). By default, the head and tail commands display ten lines of the file that is provided as an argument. For example to display the last five lines of the /etc/sysctl.conf file use the -5 option. The -n option can also be used to indicate how many lines to output. Pass a number as an argument to the option. If the -n option is used with a number prefixed by the plus sign, then the tail command recognizes this to mean to display the contents starting at the specified line and continuing all the way to the end. Live file changes can be viewed by using the -f option to the tail command.

* 1. \*\*Поясніть принципи роботи командної оболонки з каналами, потоками та фільтрами

Pipes allow you to pass the output of one command as input to another. The symbol used is |. For example, ls | more will pass a list of files to the more command.

Streams. Standard streams include stdin (standard input), stdout (standard output), and stderr (standard error stream).

Filters. Commands that take input, process it, and output the result. For example, grep.

* 1. \*Яке призначення команди grep?

The grep command can be used to filter lines in a file or the output of another command that matches a specified pattern.

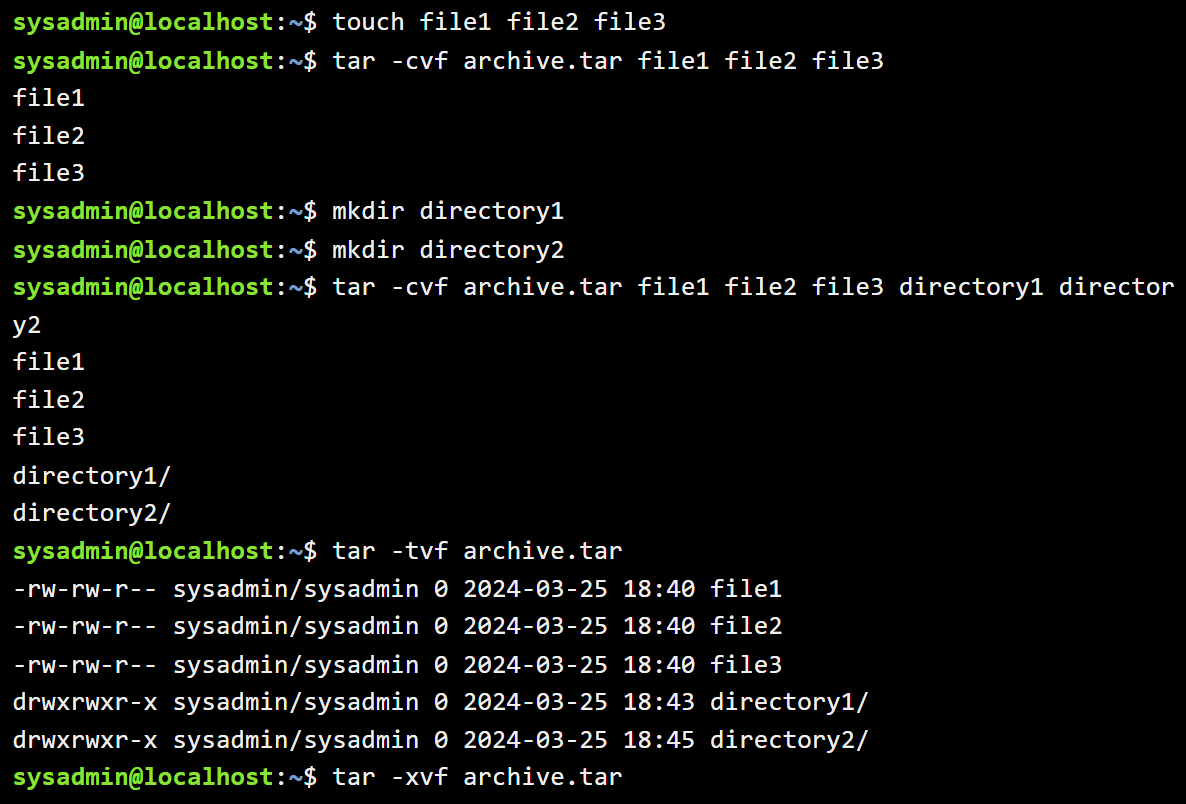
**Хід роботи:**

* 1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:
  2. Запустіть віртуальну машину VirtualBox, оберіть CentOS та запустіть її. Виконайте вхід в систему під користувачем: CentOS, пароль для входу: reverse ***(якщо виконуєте ЛР у 401 ауд.)*** та запустіть термінал.
  3. Запустіть віртуальну машину Ubuntu\_PC ***(якщо виконуєте завдання ЛР через академію netacad)***
  4. Запустіть свою операційну систему сімейства Linux ***(якщо працюєте на власному ПК та її встановили)*** та запустіть термінал.
  5. Опрацюйте всі приклади команд, що представлені у лабораторних роботах курсу ***NDG Linux Essentials - Lab 9: Archiving and Compression*** та ***Lab 10: Working With Text.*** Створіть таблицю для опису цих команд

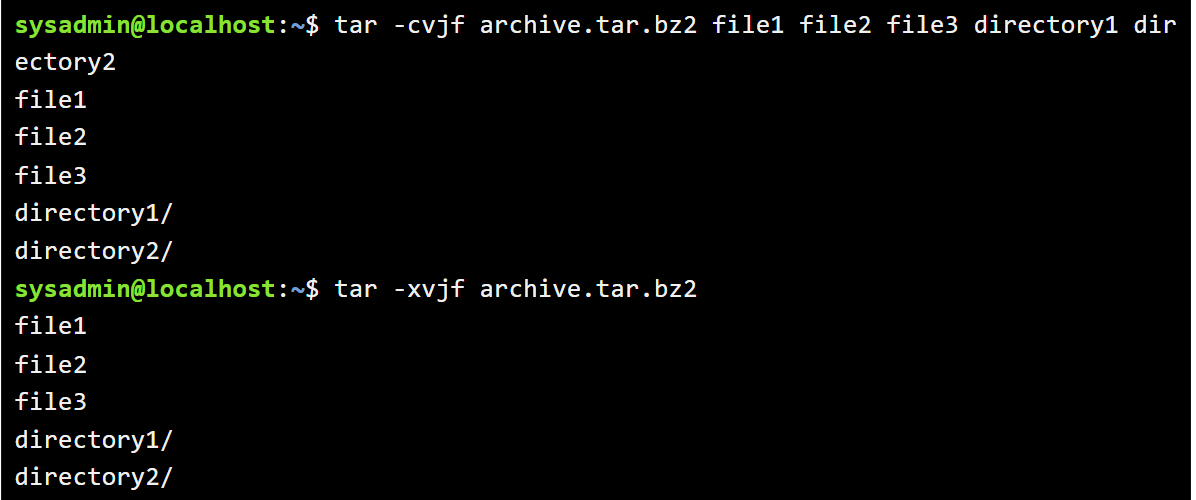
|  |  |
| --- | --- |
| **Назва команди** | **Її призначення та функціональність** |
| mkdir mybackups | Створення нової директорії **mybackups** у домашньому каталозі користувача |
| tar -cvf mybackups/udev.tar /etc/udev | Команда **tar** використовується для об’єднання кількох файлів в один файл. В даному випадку вміст директорії **/etc/udev** буде збережено в архів **udev.tar** у директорії **mybackups**. Параметр **-c** повідомляє команді tar створити файл tar. Параметр **-v** означає "verbose", який наказує команді tar продемонструвати, що вона робить. Параметр **-f** використовується для вказівки назви файлу tar. |
| tar –tvf mybackups/udev.tar | Displays the contents of a tar file by using the available options (t = list contents, v = verbose, f = filename). |
| tar –zcvf mybackups/udev.tar.gz /etc/udev | Creates a tar file that is compressed use -z option. The -z option makes use of the gzip utility to perform compression. |
| tar –xvf udev.tar.gz | Extracts the contents of an archive. |
| tar -rvf udev.tar /etc/hosts | Adds a file to an existing archive. |
| cp /usr/share/dict/words | Copies the file words. |
| gzip words | Compresses the file. |
| gunzip words.gz | Uncompresses the file. |
| bzip2 words | Compresses the file. |
| bunzip2 words.bz2 | Uncompresses the file. |
| xz words | Compresses the file. |
| unxz words.xz | Uncompresses the file. |
| zip words.zip words | Compresses the file. |
| zip -r udev.zip /etc/udev | Compresses the /etc/udev directory and its contents. |
| unzip -l udev.zip | To view the contents of a zip archive, use with the -l option with the unzip command. |
| unzip udev.zip | To extract the zip archive, use the unzip command without any options. |
| echo "Hello World"  echo "Hello World" > mymessage  cat mymessage | cat command can be used to display file contents. |
| echo "Greetings" > mymessage  cat mymessage | When you use the > symbol to redirect stdout, the contents of the file are first destroyed. This example shows it. |
| cat mymessage  echo "How are you?" >> mymessage  cat mymessage | You can avoid clobbering a file by using >> instead of >. By using >> you append to a file. |
| find ~ -name "\*bash\*" | The find command will begin the search in the directory specified and recursively search all of the subdirectories. |
| find /etc -name hosts | Run the following command and observe the output. |
| find /etc -name hosts 2> err.txt  cat err.txt | To redirect stderr (error messages) to a file, issue this command. |
| find /etc -name hosts > std.out 2> std.err  cat std.err  cat std.out | You can also redirect stdout and stderr into two separate files. |
| find /etc -name hosts > find.out 2>&1  cat find.out | To redirect both standard output (stdout) and standard error (stderr) to one file, first redirect stdout to a file and then redirect stderr to that same file by using the notation 2>&1. |
| tr a-z A-Z  this is interesting  how do I stop this?  ^D | Standard input (stdin) can also be redirected. Normally stdin comes from the keyboard, but sometimes you want it to come from a file instead. For example, the tr command translates characters, but it only accepts data from stdin, never from a file name given as an argument. This is great when you want to do something like capitalize data that is inputted from the keyboard (Note: Press **Control**+**d**, to signal the tr command to stop processing standard input). |
| tr A-Z a-z > myfile  Wow, I SEE NOW  This WORKS! | The tr command accepts keyboard input (stdin), translates the characters and then redirects the output to stdout. To create a file of all lower-case characters, execute this command. |
| cat myfile | Press the **Enter** key to make sure your cursor is on the line below "This works!", then use **Control**+**d** to stop input. To verify you created the file, execute this command. |
| cat myfile  tr a-z A-Z < myfile | Execute these commands to use the tr command by redirecting stdin from a file. |
| ls -l /etc | more | Execute this command to take the output of the ls command and send it into the more command, which displays one page of data at a time. |
| cut -d: -f1 /etc/passwd | In this example, you will use a command called cut to extract all of the usernames from a database called /etc/passwd (a file that contains user account information). |
| cut -d: -f1 /etc/passwd | sort | In this step you are going to take the output of the cut command and send it into the sort command to provide some order to the output. |
| cut -d: -f1 /etc/passwd | sort | more | Now the output is sorted, but it still scrolls off the screen. Send the output of the sort command to the more command to solve this problem. |
| cat /etc/passwd | The /etc/passwd is likely too large to be displayed on the screen without scrolling the screen. To see a demonstration of this, use the cat command to display the entire contents of the /etc/passwdfile. |
| more /etc/passwd | Use the more command to display the entire contents of the /etc/passwd file. |
| h | While you are in the more command, you can view the help screen by pressing the **h** key. |
| <SPACE> | ress the **Spacebar** to view the rest of the document. |
| less /etc/passwd  /bin  nnnNNNq | Use the less command to display the entire contents of the /etc/passwd file. Then search for the word bin, use **n** to move forward, and **N** to move backwards. Finally, quit the less pager by typing the letter **q.** |
| head /etc/passwd | You can use the head command to display the top part of a file. By default, the head command will display the first ten lines of the file. |
| tail /etc/passwd | Use the tail command to display the last ten lines of the /etc/passwd file. |
| head -2 /etc/passwd | Use the head command to display the first two lines of the /etc/passwd file. |
| ls /etc | tail -5 | Execute this command line to pipe the output of the ls command to the tail command, displaying the last five file names in the /etc directory. |
| head -n -20 /etc/passwd | Another way to specify how many lines to output with the head command is to use the option -n -#, where # is the number of lines counted from the bottom of the output to exclude. Notice the minus symbol - in front of the #. For example, if the /etc/passwd contains 27 lines, this command will display lines 1-7, excluding the last twenty lines. |
| cd /etc  grep sshd passwd | The grep command will print the entire line containing the match. |
| grep root passwd | Regular expressions are "greedy" in the sense that they will match every single instance of the specified pattern. |
| grep '^root' passwd | To limit the output, you can use regular expressions to specify a more precise pattern. For example, the caret ^ character can be used to match a pattern at the beginning of a line; so, when you execute the following command line, only lines that begin with root should be matched and displayed. |
| grep 'sync' passwd | Match the pattern sync anywhere on a line. |
| grep 'sync$' passwd | Use the $ symbol to match the pattern sync at the end of a line. |
| grep '.y' passwd | Use the period character . to match any single character. For example, execute this command to match any character followed by a 'y'. |
| grep 'sshd|root|operator' passwd | The pipe character, |, or "alternation operator", acts as an "or" operator. For example, execute this to attempt to match either sshd, root or operator. |
| grep -E 'sshd|root|operator' passwd | Use the -E switch to allow grep to operate in extended mode in order to recognize the alternation operator. |
| egrep 'no(b|n)' passwd | Use another extended regular expression, this time with egrep with alternation in a group to match a pattern. The strings nob and non will match. |
| head passwd | grep '[0-9]' | The [ ] characters can also be used to match a single character. However, unlike the period character ., the [ ] characters are used to specify exactly what character you want to match. For example, if you want to match a numeric character, you can specify [0-9]. Execute this command for a demonstration. |
| grep -E '[0-9]{3}' passwd | Suppose you want to search for a pattern containing a sequence of three digits. You can use { } characters with a number to express that you want to repeat a pattern a specific number of times; for example: {3}. The use of the numeric qualifier requires the extended mode of grep. |

**Примітка:** **Скріншоти** виконання команд в терміналі можна **не представляти**, достатньо **коротко описати команди в таблиці**.

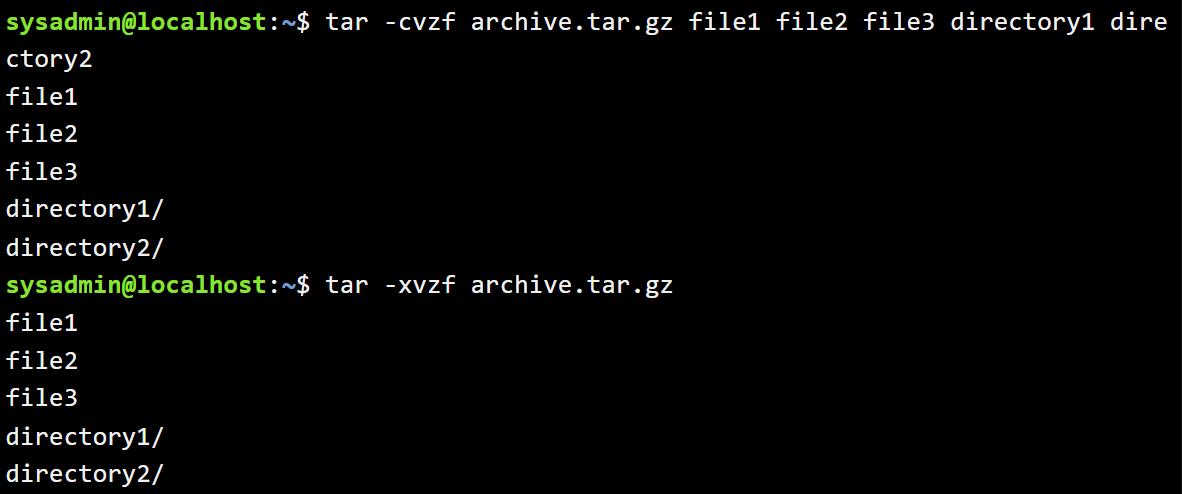
* 1. Ознайомтесь з командою tar та за її допомогою виконати у терміналі наступні дії:
* створити файл з розширенням .tar;
* створити файл з розширенням .tar, що складається з декількох файлів і каталогів одночасно;
* перегляду вмісту файлу;
* витягти вміст файлу tar;



* створити архівний файл tar, стиснений за допомогою bzip;
* витягти вміст файлу tar bzip;



* створити архівний tar файл, стисненого за допомогою gzip;
* витягти вміст файлу tar gzip.



* 1. \*Як буде відбуватись перенаправлення потоків виведення в bash для наступних дій з командами (позначено як cmd) та файлами (позначено як file):

|  |  |
| --- | --- |
| **Команда** | **Що виконує команда?** |
| cmd 1> file | Redirects the standard output (stdout) of the cmd command to the file file, replacing its contents. |
| cmd > file | Redirects the standard output (stdout) of the cmd command to the file file, replacing its contents. |
| cmd 2> file | Redirects the standard error stream (stderr) of the cmd command to the file file, replacing its contents. |
| cmd >> file | Appends the standard output of the cmd command to the end of the file, preserving its previous contents. |
| cmd &> file | Redirects both the standard output and the standard error stream of the cmd command to the file, replacing its contents. |
| cmd > file 2>&1 | Redirects both the standard output and the standard error stream of the cmd command to the file, replacing its contents (the error stream is redirected to the location where the standard output has already been redirected). |
| cmd >> file 2>&1 | Appends both the standard output and the standard cmd error stream to the end of the file, preserving its previous contents. |
| cmd 2>&1 > /dev/null | Redirects the standard cmd command error stream to the standard output, and then both streams are redirected to /dev/null, i.e. ignored. |
| cmd 2> /dev/null | Redirects only the standard cmd error stream to /dev/null, ignoring it. |
| cmd1 | cmd2 | Passes the standard output of the cmd1 command as input to the cmd2 command. |
| cmd1 2>&1 | cmd2 | Redirects both the standard output and the standard error stream of the cmd1 command to the cmd2 command. |

* 1. \*\*Розгляньте наведені нижче приклади та поясніть, що виконують дані команди та який тип перенаправлення потоків вони використовують:

|  |  |  |
| --- | --- | --- |
| **Команда**  **(контейнер команд)** | **Що виконує команда?** | **Який потік перенаправлення?** |
| $echo "It is a new story." > story | Prints the string "It is a new story." and writes it to the story file, replacing the previous contents of the file. | Standard output (stdout). |
| $ date > date.txt | Prints the current date and time and writes them to the date.txt file, overwriting the previous contents of the file. | Standard output (stdout). |
| $ cat file1 file2 file3 > bigfile | Combines the contents of files file1, file2, and file3 and writes the result to the bigfile file, overwriting the previous file contents. | Standard output (stdout). |
| $ls -l >> directory | Displays a detailed list of files in the current directory and adds them to the directory file, preserving the previous contents of the file. | Adding to the standard output (stdout). |
| $ sort < file1\_unsorted > file2\_sorted | Sorts the contents of file1\_unsorted and writes the sorted result to file2\_sorted, overwriting the previous contents of the file. | Standard input (stdin) and standard output (stdout). |
| $ find -name '\*.txt' > file.txt 2> /dev/null | It searches for all files with the .txt extension and writes their names to the file.txt file, and redirects errors to /dev/null (ignores them). | Standard output (stdout) and standard error stream (stderr). |
| $ cat file1\_unsorted | sort > file2\_sorted | Outputs the contents of the file file1\_unsorted, passes it to the sort command, and writes the sorted result to the file file2\_sorted. | Pipe and standard output (stdout). |
| $ cat myfile | grep student | wc -l | Prints the contents of the file, filters the output, leaving only those lines that contain the word "student", counts the number of lines remaining after filtering. | Pipe. |

**Контрольні запитання:**

1. Надайте порівняльну характеристику процесам стискання та архівування.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Purpose** | **Tools** | **Result** | **Resize** |
| **Archiving** | Combine multiple files or directories into a single file for easy storage and transfer. | Tar on Linux, zip on Windows. | One archive file that contains the structure and contents of the original files. | Usually does not reduce the size of files, unless used in combination with compression. |
| **Compression** | Reduce the size of files using compression algorithms. | gzip, bzip2, xz on Linux, WinZip, WinRAR on Windows. | A smaller file that needs to be unzipped before use. | Reduces the size of files, which effectively saves disk space and reduces file transfer times. |

1. Які програми, окрім наведених в роботі, можуть використовуватись для стискання та архівування файлів та каталогів в ОС Linux? Наведіть приклади та їх короткий опис.

7-Zip (p7zip): a version of 7-Zip for Unix-like systems that supports 7z, XZ, BZIP2, GZIP, TAR, ZIP, and other formats. lrzip (Long Range ZIP): a compression tool used to compress large files, providing a high level of compression by using multiple algorithms. star: a tool similar to tar, but with additional features such as better support for ACLs and other metadata. cpio: this tool is used to create file and directory archives via stdin/stdout and supports both cpio and tar formats.

1. \*Порівняйте алгоритми стискання, що використовуються в командах (програмах), використовуваних в Linux. Які з алгоритмів можна вважати найшвидшим та найефективнішим?

gzip: Uses the DEFLATE algorithm, provides a good balance between speed and compression level.

bzip2: Uses the Burroughs-Wheeler algorithm, which usually gives a better compression rate than gzip, but is slower.

xz: Uses the LZMA algorithm, which provides a high level of compression, but can be slower to compress and decompress.

If you need the highest level of compression, xz with the -e option may be the best choice, although it will be slower. But the choice of algorithm depends on your specific needs and usage conditions.

1. \*Опишіть програмні засоби для стискання та архівування, що можуть бути використані у вашому мобільному телефоні.

The built-in Files app on my iOS 16.1.1 mobile device allows me to archive files to ZIP format. To do this, just select the files and the appropriate option.

1. \*Опишіть та порівняйте програмні засоби для стискання та (де)архівування даних у ОС сімейства Windows.

WinZip: This is one of the most well-known tools that supports many file formats and offers features such as password management and virus detection.

WinRAR: This is a powerful tool that supports compression in RAR and ZIP format and is known for its high compression rate.

7-Zip: A free and open-source tool that offers high compression rates, especially in the 7z format, and supports AES-256 encryption.

PeaZip: Another free tool that supports many formats and offers flexible compression and encryption options.

Each of these tools has its own features and can be chosen depending on the user's needs. WinZip and WinRAR are commercial products with additional features, while 7-Zip and PeaZip offer free solutions with many options for compressing and archiving files.

1. \*\*Поясніть яким чином стиснення та архівування даних може бути використано для резервування даних. В яких ще задачах системного адміністрування воно може бути використано.

Compressing data before backing it up saves storage space and reduces the time it takes to transfer data to a storage location. Archived and compressed backups can be restored faster because less data needs to be transferred. In addition to backup, compression and archiving can be used for the following system administration tasks: data transfer, disk space optimization, log files, and security.

1. \*\*Яке призначення директорії файлу /dev/null?

This is a special file used to discard any output that is redirected to it without saving it.

**Висновки**

В ході виконання лабораторної роботи мною було досліджено, команди Linux для архівування та стиснення даних, більш детально теоретично досліджено питання роботи з текстом та інструментів архівування і стиснення даних у Linux та Windows. Отримано практичні навички роботи з командами архівування та стиснення даних у терміналі.