“Київський фаховий коледж зв’язку”

Циклова комісія комп’ютерної та програмної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №4**

з дисципліни: «Операційні системи»

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

Виконав(ла/ли) студент(ка/и)

групи: КСМ-23а

The Awkward Turtles:

Когут Б.М.,

Михайленко О.О.,

Трощинський Я.П.

Перевірила викладач

Сушанова В.С.

Київ 2024

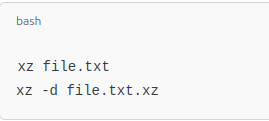
*О.Михайленко*

***1*.**

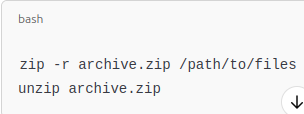
The tar utility is used to combine multiple files into a single archive. It's often used alongside compression, such as with gzip or bzip2.



The xz utility is used to compress files using the LZMA algorithm, which provides a high compression ratio.



The zip utility allows creating ZIP archives, a popular format on Windows.



The bzip2 utility is used for file compression, similar to gzip, but with a better compression ratio, though it’s slower.



Gzip is one of the most widely used tools for file compression. It’s faster than bzip2 but has a slightly lower compression ratio.

**

***2.***

*1.Archiving and Compressing with tar and gzip*

This method creates a .tar.gz file where tar is used for archiving, and gzip is used for compression.



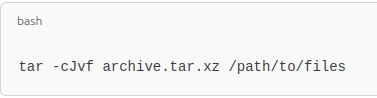
*2.Archiving and Compressing with zip*

The zip command is convenient for creating ZIP archives, which are widely supported across different operating systems.



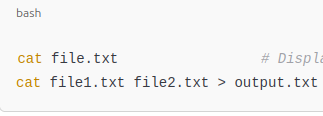
*3.Archiving with tar and Compressing with xz*

The xz command provides a high compression ratio, reducing file size even more than gzip or bzip2.

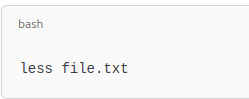
**

***3.***

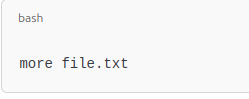
*The cat command displays the contents of a file in the terminal. It can also concatenate multiple files and display the output or save it to a new file.*

**

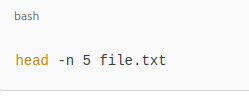
*The less command allows you to view the contents of a file page-by-page. Unlike cat, it supports scrolling and is more suitable for large files, as it doesn’t load the entire file at once.*

**

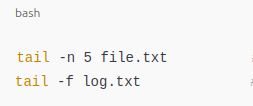
*The more command is also used to view file contents page-by-page. It’s similar to less but offers more basic navigation features.*

**

*The head command displays the first few lines of a file. By default, it shows the first 10 lines.*

**

*The tail command displays the last few lines of a file. By default, it shows the last 10 lines. It can also be used for live monitoring of files, such as logs.*

**

***4.***

*1.* ***Command Shell Overview***

A command shell (e.g., Bash on Linux, Command Prompt or PowerShell on Windows) is a program that allows users to interact with the operating system by typing commands. The shell interprets these commands and executes the corresponding actions. It can also handle inputs and outputs, which is where streams, pipes, and filters come into play.

2. **Streams**

In the context of a shell, a stream refers to a flow of data that can be read from or written to by a command. There are three standard streams:

* **Standard Input (stdin)**: This is the default input stream. It provides data that a program or command reads.
* **Standard Output (stdout)**: This is the default output stream. It is where a command sends its output.
* **Standard Error (stderr)**: This is the stream used for error messages or diagnostics.

Commands can read from stdin and write to stdout and stderr. The shell provides mechanisms for redirecting or manipulating these streams.

3. **Pipes (**|**)**

A pipe is a mechanism that allows the output of one command to be used as the input of another. This is done using the pipe operator |.

* The pipe allows you to chain commands together so that the output of one command is "piped" into the next command.
* It enables the creation of powerful workflows by combining simple commands into complex ones.



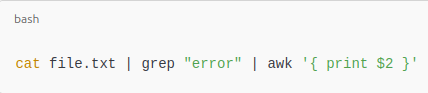
### ***4.* ***Filters*****

A filter is a program or command that processes input data and produces output, often by transforming it in some way. Filters are commonly used in combination with pipes to manipulate or process streams of data.

Common examples of filters:

* grep: Searches for patterns in text.
* sort: Sorts lines of text.
* awk: Processes and formats text.
* sed: Stream editor for modifying text in a stream.

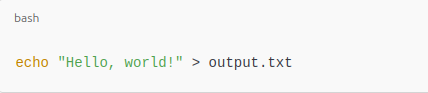
Filters are typically used in the middle of a pipeline to perform some transformation on the data.

**

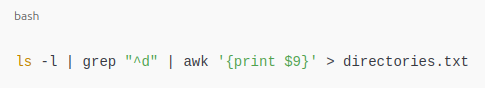
### *5.* ***Redirection***

In addition to pipes, the shell allows redirection of streams using operators like >, >>, <, and 2>:

* >: Redirects stdout to a file, overwriting it.
* >>: Redirects stdout to a file, appending to it.
* <: Redirects a file to stdin.
* 2>: Redirects stderr to a file.

**

*6.* ***Practical Example***

**

*ls -l: Lists files and directories in long format.*

* grep "^d": Filters out only directories (since directories start with "d" in the ls -l output).
* awk '{print $9}': Extracts the 9th field from each line (which is the name of the directory).
* > directories.txt: Redirects the output to a file called directories.txt.

***5.***

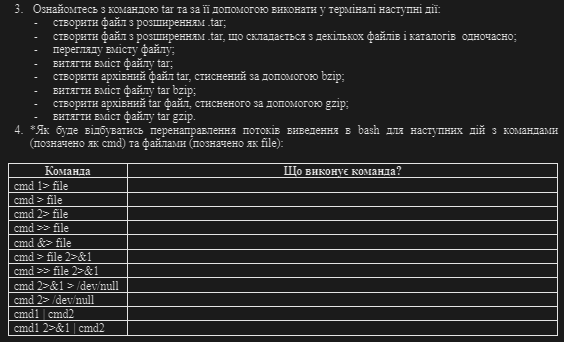
*Purpose of the grep command*

The grep command in Unix-like operating systems (such as Linux, macOS) is used to search for text strings that match a given pattern or regular expression in files or the output of other commands. The name grep comes from the phrase "global regular expression print," which means "global print by regular expression."

### Main purpose of grep:

* Search for lines containing a specified pattern.
* Print the lines that match the given criteria.

*Я.Трощинський*

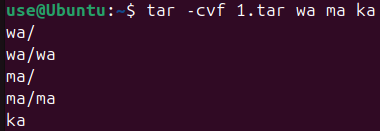


3. Read the TAR command manual and use the following steps in the terminal:

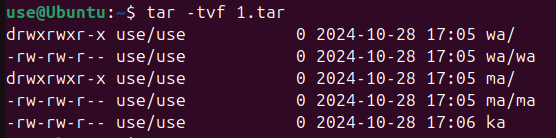
* create a .tar file;



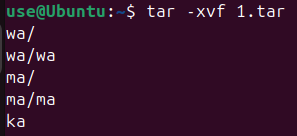
* create a .tar file consisting of several files and directories at a time;



* viewing the content of the file;



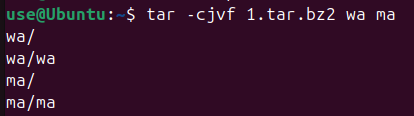
* extract the contents of the TAR file;



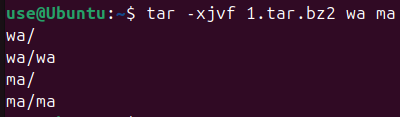
* Create a TAR archive file compressed with BZIP;

To gain ability to use a bzip2 compression I needed firstly to download it via

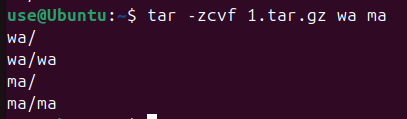
sudo apt install bzip2



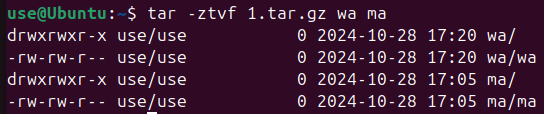
* extract the contents of the Tar Bzip file;



* Create an archive tar file compressed with Gzip;



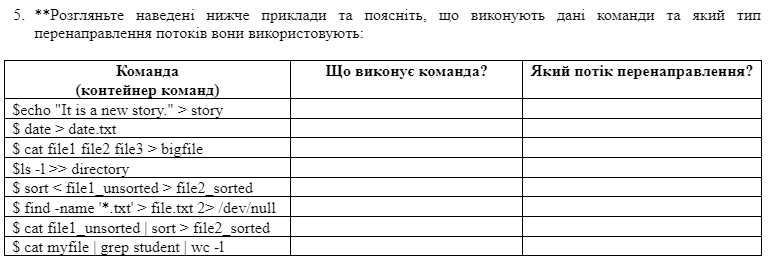
* TAR GZIP file content.



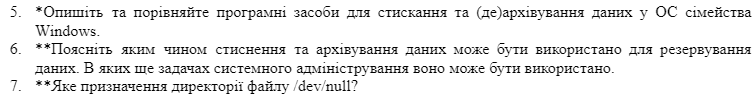
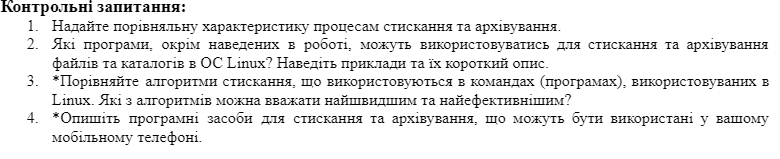
4. How the output streams will be redirected to bash for the following actions with commands (marked as cmd) and files (marked as file):

|  |  |
| --- | --- |
| **Command** | **What does the team perform?** |
| **cmd 1> file** | **Redirects only stdout to the file** |
| **cmd > file** | **Same as cmd 1> file** |
| **cmd 2> file** | **Redirects only stderr to the file** |
| **cmd >> file** | **Adds both, stdout and stderr to the file** |
| **cmd &> file** | **Redirects both, stdout and stderr to the file** |
| **cmd > file 2>&1** | **Redirects stdout to the file, and then redirects stderr to the file descriptor** |
| **cmd >> file 2>&1** | **Same as previous but adds stdout, not writes from scratch** |
| **cmd 2>&1 > /dev/null** | **Redirects both, stdout and stderr to the /dev/null, discarding them** |
| **cmd 2> /dev/null** | **Redirects stderr to the /dev/null, discarding it** |
| **cmd1 | cmd2** | **Pipes the stdout of cmd1 to the stdin of cmd2** |
| **cmd1 2>&1 | cmd2** | **Pipes both stdout and stderr to the stdin of cmd2** |

*Б.Когут*



|  |  |  |
| --- | --- | --- |
| **Command (command container)** | **What does the command do?** | **What type of redirection?** |
| $ echo "It is a new story." > story | The echo command outputs the text "It is a new story." and writes it to the file story. If the file story doesn't exist, it will be created; if it exists, it is overwritten. | > — redirects the standard output (stdout) to a file. |
| $ date > date.txt | The date command outputs the current date and time, and this result is written to the file date.txt. If the file doesn’t exist, it will be created, and if it does exist, its content will be overwritten. | > — redirects the standard output (stdout) to a file. |
| $ cat file1 file2 file3 > bigfile | The cat command combines the contents of file1, file2, and file3 and writes them to the file bigfile. If bigfile doesn’t exist, it will be created; if it exists, its content will be overwritten. | > — redirects the standard output (stdout) to a file. |
| $ ls -l >> directory | The ls -l command lists the files in the current directory in long format. The result is appended to the end of the file directory. If directory doesn’t exist, it will be created, and if it exists, its content will not be overwritten. | >> — appends the standard output (stdout) to a file. |
| $ sort < file1\_unsorted > file2\_sorted | The sort command reads the content of file1\_unsorted, sorts it, and writes the result to file2\_sorted. If file2\_sorted doesn't exist, it will be created; if it exists, it is overwritten. | < — redirects the standard input (stdin) from a file. |
| $ find -name '.txt' > file.txt 2> /dev/null | The find command searches for files with the .txt extension in the current directory and its subdirectories. The result is written to file.txt. All error messages are redirected to /dev/null, meaning they are discarded. | > — redirects the standard output (stdout) to a file.  2> — redirects the standard error (stderr) to a file, or in this case, to /dev/null. |
| $ cat file1\_unsorted | sort > file2\_sorted | The cat command reads the content of file1\_unsorted and passes it to the sort command via a pipeline (|). The sort command then sorts the input and writes the result to file2\_sorted. If file2\_sorted doesn’t exist, it will be created; if it exists, its content will be overwritten. | | — pipeline, passes the standard output of one command as the input to the next.  > — redirects the standard output (stdout) to a file. |
| $ cat myfile | grep student | wc -l | The cat command reads the content of myfile and passes it to the grep command via a pipeline. The grep command finds lines containing the word "student" and passes them to the wc -l command, which counts the number of such lines. | | — pipeline, passes the standard output of one command as the input to the next. |



1. **Compression vs. Archiving**: Compression reduces file size without losing data. Archiving bundles multiple files into a single file for easier management but doesn't reduce their size.

2. **Linux Compression/Archiving Tools**: Beyond common tools like tar and gzip, Linux also supports:

bzip2: Offers higher compression than gzip but is slower.

xz: Provides efficient compression and is suited for large files.

7zip (p7zip): Known for high compression ratios across different formats.

3. **Linux Compression Algorithms**:

gzip uses the DEFLATE algorithm, balancing speed and compression.

bzip2 uses the Burrows-Wheeler algorithm, achieving better compression but slower speed.

xz uses LZMA, which provides high compression for large files but can be slow.

The fastest is often gzip, while xz is efficient for storage.

4. **Mobile Compression/Archiving Apps**:

Apps like RAR, ZArchiver, and WinZip allow compressing and extracting multiple formats on mobile phones, offering both easy file management and compatibility with desktop formats.

5. **Windows Compression/Archiving Tools:**

Windows offers WinRAR, 7-Zip, and WinZip. WinRAR and 7-Zip support multiple formats (like .zip and .rar), while 7-Zip is open-source and offers high compression.

6. **Compression/Archiving for Backup and Admin Tasks:**

Compression and archiving help save storage space and organize files, making them easier to manage and restore during backups. They're also used in system administration for log management and packaging software.

7. **Purpose of /dev/null:**

/dev/null is a special directory in Linux used to discard unwanted output. Writing to it is like sending data into "a black hole," freeing up system resources without saving the output.

**Conclusion**

Through this lab work, we gained practical skills in working with the Bash command shell, became familiar with basic commands for archiving and compressing data, and learned basic text manipulation in the terminal.