“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”**

Виконав студент

групи БІКС-13

Когут Ігор Святославович

Береза Артем Шихайович

Перевірив викладач

Сушанова В.С.

Київ 2022

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

**The goal of the work:**

**Getting hands-on skills with the Bash shell.**

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

**Getting to know the basic actions when working with text in the terminal.**

**Material support for classes:**

**1. IBM PC type computer.**

**2. OS family Windows and virtual machine Virtual Box (Oracle).**

**3. GNU/Linux OS (any distribution).**

**4. Cisco network academy site netacad.com and its online Linux courses**

**Tasks for preliminary preparation: ( Когут )**

**4)** 1. tar - a utility for creating, unpacking and managing archives of files in tar format.

Main parameters:

-c: create archive

-x: unpack the archive

-f: specifying the name of the archive

-v: output execution details

-z: use gzip for compression or decompression

-j: use bzip2 for compression or decompression

Installation: Usually installed with package management scripts on most Unix-like systems.

xz:

Description: xz is a utility for compressing and decompressing xz format files.

Main parameters:

-c: output to standard output

-d: extract the file

-k: keep the original file

-v: output execution details

Installation: Usually installed with package management scripts on most Unix-like systems.

zip:

Description: zip - a utility for creating, updating and decompressing files in ZIP format.

Main parameters:

-r: recursive compression of folders

-d: delete files from the archive

-u: update files in the archive

-v: output execution details

Installation: Usually installed with the package manager or can be downloaded separately.

2. Examples of implementing data archiving and compression:

Archiving and compression to tar.gz format:

tar -czvf archive.tar.gz directory/

Archiving and zipping:

zip -r archive.zip directory/

Archiving and compression to xz format:

tar -c directory/ | xz -c > archive.tar.xz

3. Commands for working with text files:

cat - prints the contents of a file to standard output.

It is usually present in basic installations of Unix-like systems.

less - a program for viewing text files by pages.

It is usually installed together with the operating system.

more - similar to less, but less functional.

It is usually installed together with the operating system.

head - outputs the first N lines of the file.

-n: specify the number of lines.

Usually installed with base Unix installations.

tail - outputs the last N lines of the file.

-n: specify the number of lines.

Usually installed with base Unix installations.

4. Principles of operation of the command shell with channels, streams and filters:

The command shell in Unix-like systems allows you to redirect the output of one command to the input of another, as well as to use streams and filters to process data. Channels (|) are used to pass the output of one program to another. I/O streams (stdin, stdout, stderr) control how programs interact with the shell. Filters are programs that process input data and output the processed data.

5. The grep command:

grep is a command line utility for searching text in files. It takes a pattern string and searches for it in the given files. Main parameters:

-i: ignore case

-r: recursive search in folders

-v: print strings that do not match the pattern

Installation: Usually present in basic installations of Unix-like systems.

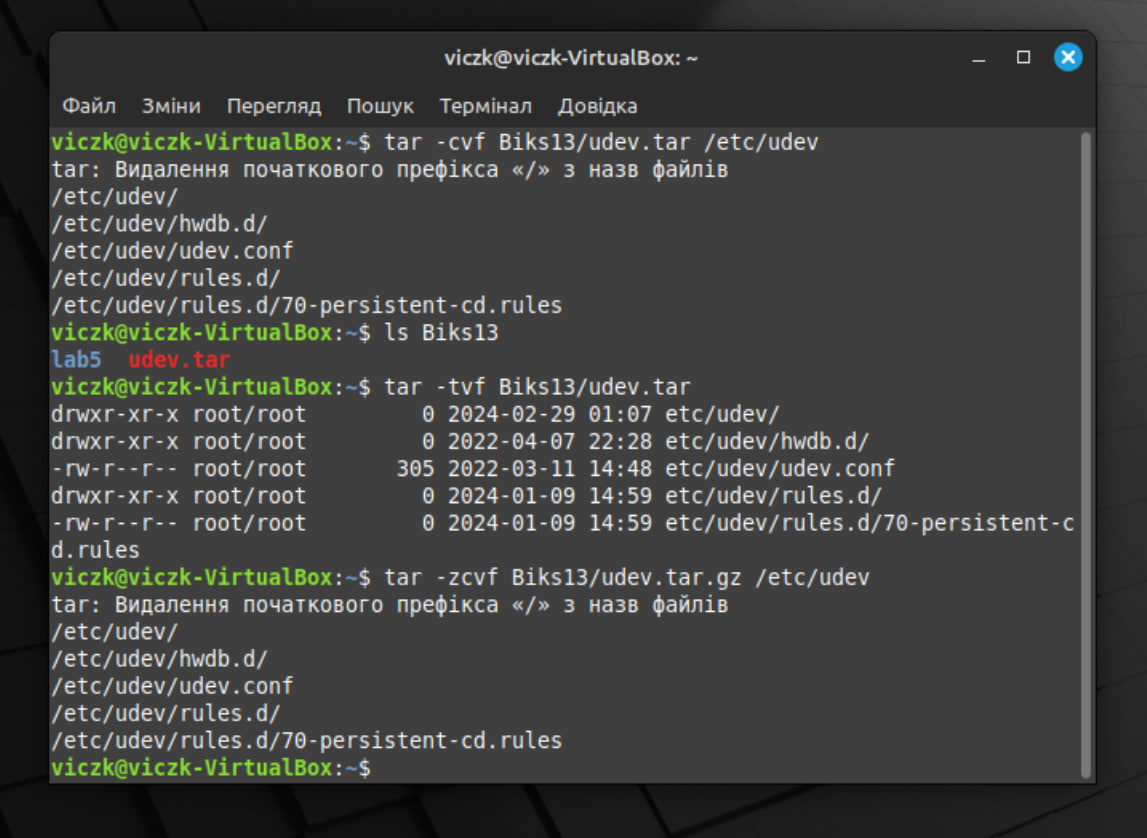
|  |  |
| --- | --- |
| **Термін англійською** | **Термін українською** |
| Compression | Стиснення |
| Lossy Compression | Втратне стиснення |
| Lossless Compression | Безвтратне стиснення |
| gzip Compression | Стиснення gzip |
| bzip2 Compression | Стиснення bzip2 |
| xz Compression | Стиснення xz |
| tar | Архіватор tar |
| -z, -j, -J | Параметри команд |
| -c | Параметр команди tar |
| -t -x | Параметр команди tar |

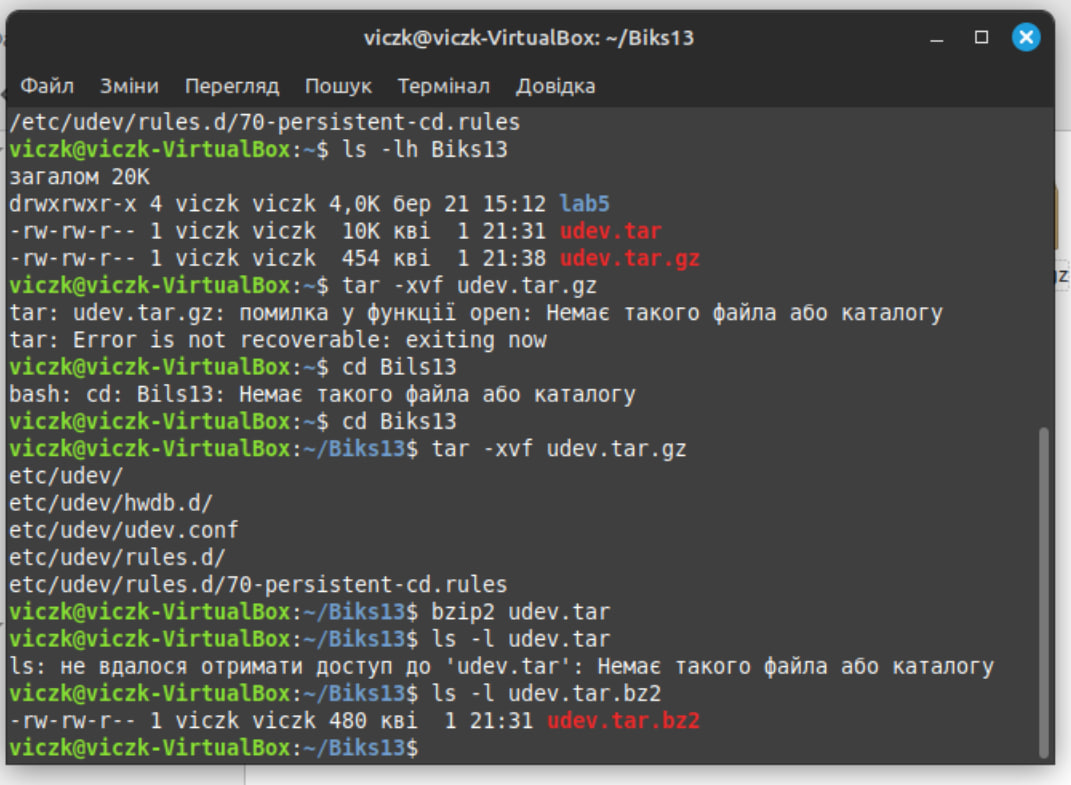
**Progress work**

**2. ( Когут 9 лекція, Береза 10 лекція )**

|  |  |
| --- | --- |
| Command name | Its purpose and functionality |
| mkdir mybackups | Create a new mybackups directory in your home directory user |
| tar -cvf mybackups/udev.tar | The tar command is used to combine multiple files into a single file |
| /etc/udev | In this case, the contents of the /etc/udev directory will be  saved in the udev.tar archive in the mybackups directory. The -c option  tells the tar command to create a tar file. The -v option means  &quot;verbose&quot;, which tells the tar command to demonstrate that it  does The -f option is used to specify the name of the tar file. |
| tar -cvf archive.tar files... | Creates and packages files and directories into a tar archive |
| tar -tvf archive.tar | Lists the contents of the tar archive. |
| tar -zcvf archive.tar.gz files... | Creates and compresses files and directories into a gzip archive.. |
| tar -jcvf archive.tar.bz2 files... | Creates and compresses files and directories into a bzip2 archive.. |
| tar -Jcvf archive.tar.xz files... | Creates and compresses files and directories into an xz archive. |
| zip archive.zip files... | Creates a ZIP archive from files and directories (with compression) |
| unzip -l archive.zip | Displays a list of files and directories contained in the ZIP archive. |
| unzip archive.zip | Extracts the ZIP archive, restoring files and directories. |
| gzip file | Compresses the file, replacing the original with the compressed file. |
| gunzip file.gz | Extracts a gzip-compressed file, restoring the original. |
| tr A-Z a-z > myfile | Convert text from uppercase to lowercase. Changes the case of letters from uppercase to lowercase and writes the result to the file myfile. |
| cat myfile | Display the contents of the file on the screen. Displays the contents of the file myfile on the screen. |
| `cat /etc/passwd | more` Display the contents of the file in pages. |
| head /etc/passwd | Output the first 10 lines of the file. Prints the first 10 lines of the /etc/passwd file. |
| tail /etc/passwd | Output the last 10 lines of the file. Prints the last 10 lines of the /etc/passwd file. |
| head -2 /etc/passwd | Output the first two lines of the file. Prints the first two lines of the /etc/passwd file. |
| `ls /etc | tail -5` Display the last five files in the directory. |
| grep sshd passwd | Search for strings with the substring "sshd". Finds and outputs lines with the substring "sshd" in the passwd file. |
| grep '^root' passwd | Search for strings starting with "root". Finds and outputs lines starting with "root" in the passwd file. |
| grep 'sync$' passwd | Search for strings that end with "sync". Finds and outputs lines ending with "sync" in the passwd file. |
| grep '.y' passwd | Search for strings where "y" is preceded by any character. Finds and outputs lines where "y" is preceded by any character in the passwd file. |
| `egrep 'no(b | Search for strings that contain "nob" or "non". |
| `head passwd | Search for strings that contain at least one digit in the first 10 lines of the file. |
| grep -E '[0-9]{3}' passwd | Search for strings that contain a sequence of three digits. Finds and outputs lines that contain a sequence of three digits in the passwd file. |

**3. ( Береза )**





**4. ( Береза )**

|  |  |
| --- | --- |
| **Команда** | **Що виконує команда?** |
| cmd 1> file | Printing the output of the cmd command to the file file, overwriting the contents of the file file. |
| cmd > file | The same action as in the previous case. |
| cmd 2> file | Outputting errors (stderr) of the cmd command to the file file, overwriting the contents of the file file. |
| cmd >> file | Adding the output of the cmd command to the end of the file file. |
| cmd &> file | Output both output and errors (stdout and stderr) of the cmd command to the file file. |
| cmd > file 2>&1 | Output of the output and errors of the cmd command to the file file. 2>&1 indicates the flow direction of stderr (2) to stdout (1). |
| cmd >> file 2>&1 | Adding both output and errors (stdout and stderr) of the cmd command to the end of file. |
| cmd 2>&1 > /dev/null | Sending cmd output and errors to /dev/null, a "black hole" that discards all data. |
| cmd 2> /dev/null | Sending errors (stderr) of the cmd command to /dev/null. |
| cmd1 | cmd2 | Forwarding the output of the cmd1 command to the input of the cmd2 command. |
| cmd1 2>&1 | cmd2 | Forwarding the output and errors of the cmd1 command to the input of the cmd2 command. |

**5. ( Когут )**

|  |  |  |
| --- | --- | --- |
| **Команда**  **(контейнер команд)** | **Що виконує команда?** | **Який потік перенаправлення?** |
| $echo "It is a new story." > story | Writes a string to the story file. | Output (stdout) redirection to file |
| $ date > date.txt | Writes the current date and time to the date.txt file. | Output (stdout) redirection to file. |
| $ cat file1 file2 file3 > bigfile | Reads the contents of file1, file2, and file3 and writes them to bigfile. | Output (stdout) redirection to file. |
| $ls -l >> directory | The output of the ls -l command is added to the directory file (the data is not overwritten). | Writing output (stdout) to a file. |
| $ sort < file1\_unsorted > file2\_sorted | Sorts the contents of the file file1\_unsorted and writes the result to the file file2\_sorted. | Input from file, output to file. |
| $ find -name '\*.txt' > file.txt 2> /dev/null | Executes the find command to search for files with the extension .txt and outputs the result to the file file.txt. All error messages are either sent to /dev/null (discarded). | Output (stdout) redirect to file, errors (stderr) are discarded. |
| $ cat file1\_unsorted | sort > file2\_sorted | sort > filet2\_sorted | Reads the contents of file1\_unsorted, sorts it, and writes the contents to file2\_sorted |
| $ cat myfile | grep student | wc -l | grep student | wc -l` |

**Control questions: ( Когут )**

1. Compression is the process of reducing the size of a file or data stream by encoding information using fewer bits than the original representation. It's primarily used to save storage space and reduce data transfer time.

Archiving involves bundling multiple files and directories into a single file, often known as an archive. This process helps in organizing and storing related files together, simplifying their management and transfer.

2. 7-Zip: A popular open-source compression tool that supports various formats like 7z, ZIP, GZIP, etc. It provides high compression ratios and strong encryption.

A proprietary archive manager that supports the RAR file format. It offers multiple volume archives, self-extracting archives, and recovery records for data protection.

Another open-source archive manager that supports a wide range of archive formats, including 7z, ZIP, TAR, etc. It features a user-friendly interface and supports encryption and file splitting.

3. gzip: Uses the DEFLATE algorithm, which offers a good balance between compression ratio and speed. It's efficient for general-purpose compression tasks.

bzip2: Utilizes the Burrows-Wheeler transform and Huffman coding. It typically provides better compression ratios than gzip but at the cost of slower compression and decompression speeds.

xz: Implements the LZMA2 compression algorithm, which offers high compression ratios and is suitable for compressing large files. However, it tends to have slower compression and decompression speeds compared to gzip and bzip2.

Comparison: The fastest algorithm among these is usually gzip, followed by bzip2, with xz being the slowest but providing the best compression ratios.

4. ZArchiver (Android): ZArchiver is a popular app for managing archives on Android devices. It supports various formats like ZIP, RAR, 7z, etc., and allows users to create, extract, and view archive contents directly on their mobile devices.

iZip (iOS): iZip is a file compression and extraction app for iOS devices. It supports ZIP, RAR, 7z, and other archive formats, enabling users to compress and decompress files on their iPhones or iPads.

**Conclusion**

Linux commands offer robust capabilities for data archiving and compression, as well as efficient text manipulation. The combination of utilities like tar, gzip, xz, and zip provides users with versatile tools for organizing, compressing, and transferring files and directories efficiently. These commands offer various options and parameters to cater to different needs, whether it's creating archives, compressing files using different algorithms, or extracting specific content from archives.