“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»**

Виконав(ла/ли) студент(ка/и)

групи: КСМ-23а

Папснази

Юхимець Д.О.Любежанін М.С

Перевірила викладач

Сушанова В.С.

Київ 2024

**Лабораторна робота №6   
 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.**

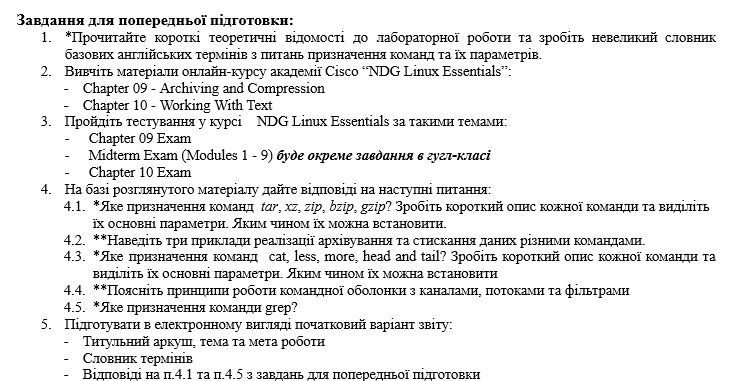
**Матеріальне забезпечення занять:**

**1. ЕОМ типу IBM PC.**

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

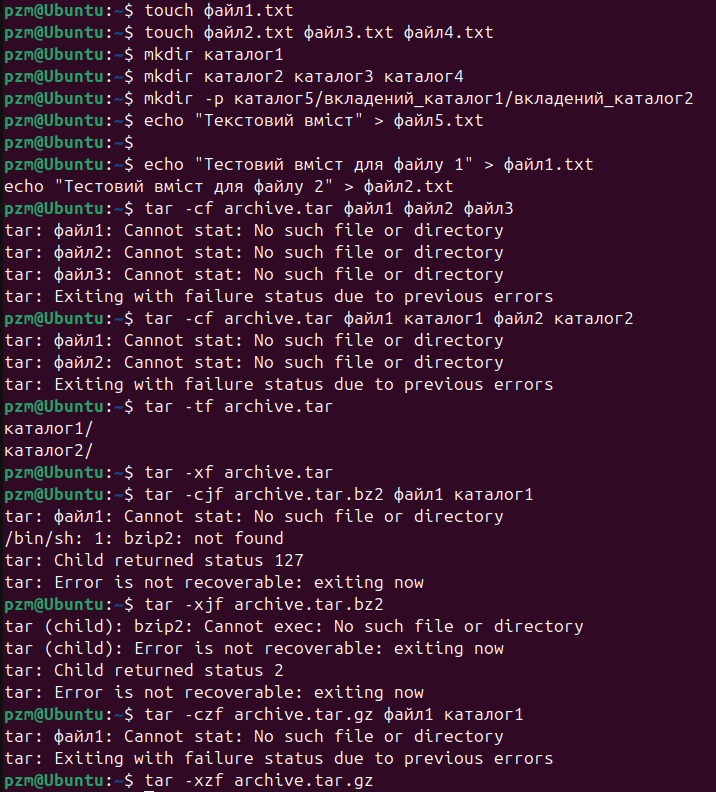
**3. ОС GNU/Linux (будь-який дистрибутив).**

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



**Виконав Юхимець Дмитро**

| Team name | Its purpose and functionality |
| --- | --- |
| mkdir mybackups | Creates a new directory mybackups in the user's home directory. |
| tar -cvf mybackups/udev.tar /etc/udev | Archive the contents of /etc/udev into udev.tar in mybackups directory. -c creates, -v is verbose, and -f specifies the archive name. |
| tar -tvf mybackups/udev.tar | Lists the contents of the udev.tar archive in a verbose format. |
| tar -zcvf mybackups/udev.tar.gz /etc/udev | Compresses the /etc/udev directory into a gzip archive udev.tar.gz in mybackups directory. -z enables gzip compression. |
| tar -xvf udev.tar.gz | Extracts the contents of udev.tar.gz to the current directory. |
| tar -rvf udev.tar /etc/hosts | Adds /etc/hosts to the existing udev.tar archive. -r option appends files to an archive. |
| gzip words | Compresses the words file, creating words.gz and removing the original. |
| gunzip words.gz | Decompresses words.gz, restoring the original words file. |
| bzip2 words | Compresses the words file with bzip2, creating words.bz2 and removing the original. |
| bunzip2 words.bz2 | Decompresses words.bz2, restoring the original words file. |
| xz words | Compresses the words file with xz, creating words.xz and removing the original. |
| unxz words.xz | Decompresses words.xz, restoring the original words file. |
| zip words.zip words | Creates a zip archive words.zip containing the words file. |
| zip -r udev.zip /etc/udev | Creates a zip archive udev.zip with the /etc/udev directory, including subdirectories (-r for recursive). |
| unzip -l udev.zip | Lists the contents of udev.zip without extracting. |
| unzip udev.zip | Extracts the contents of udev.zip to the current directory. |
| echo "Hello World" > mymessage | Redirects the output "Hello World" to the file mymessage, overwriting any existing content. |
| find ~ -name "\*bash\*" | Searches recursively for files in the home directory containing "bash" in the filename. |
| find /etc -name hosts 2> err.txt | Searches for "hosts" in /etc and redirects errors to err.txt. |
| tr a-z A-Z < myfile | Converts lowercase letters to uppercase in myfile, displaying the result in the terminal. |
| cut -d: -f1 /etc/passwd | Extracts the first field (usernames) from /etc/passwd, using : as a delimiter. |
| cat /etc/passwd | Displays the entire content of /etc/passwd. |
| more /etc/passwd | Views /etc/passwd with pagination, allowing navigation through the file. |
| less /etc/passwd | Views /etc/passwd with more advanced navigation and search options than more. |
| head /etc/passwd | Displays the first 10 lines of /etc/passwd. |
| grep '^root' passwd | Searches for lines in passwd that start with "root". |

****

| **Команда** | **Що виконує команда?** |
| --- | --- |
| cmd 1> file | **Redirects the standard output (stdout) of the cmd command to the file file, overwriting it if it exists.** |
| cmd > file | **It also redirects stdout to the file file. 1 and have the same effect, overwriting the file.** |
| cmd 2> file | **Redirects the standard error stream (stderr) to the file file, overwriting it if the file exists.** |
| cmd >> file | **Adds the stdout of the cmd command to the end of the file file without overwriting existing data.** |
| cmd &> file | **Redirects both streams, stdout and stderr, to the file file, overwriting it.** |
| cmd > file 2>&1 | **Redirects stdout to the file file and makes sure that stderr is also sent to the same file.** |
| cmd >> file 2>&1 | **Adds stdout to the file file and sends stderr to the same file for writing.** |
| cmd 2>&1 > /dev/null | **First, it sends stderr to stdout, and then redirects stdout to /dev/null (ignored). The result is that stderr is ignored and stdout is not written.** |
| cmd 2> /dev/null | **Redirects stderr to /dev/null, ignoring errors.** |
| cmd1 | cmd2 | **Redirects the stdout of the cmd1 command to the stdin of the cmd2 command..** |
| cmd1 2>&1 | cmd2 | **Redirects the stderr and stdout of the cmd1 command to the stdin of the cmd2 command.** |

| **Команда**  **(контейнер команд)** | **Що виконує команда?** | **Який потік перенаправлення?** |
| --- | --- | --- |
| $echo "It is a new story." > story | **Write the text "It is a new story." to story, overwriting the file if it exists.** | **stdout to file** |
| $ date > date.txt | **Writes the current date and time to date.txt, overwriting it if it exists** | **stdout to file** |
| $ cat file1 file2 file3 > bigfile | **Concatenates the contents of file1, file2, and file3 into bigfile, overwriting bigfile.** | **stdout to file** |
| $ls -l >> directory | **Appends the long listing of files in the current directory to directory.** | **stdout to file (append mode)** |
| $ sort < file1\_unsorted > file2\_sorted | **Sorts the contents of file1\_unsorted and writes the sorted output to file2\_sorted, overwriting it.** | **stdin from file, stdout to file** |
| $ find -name '\*.txt' > file.txt 2> /dev/null | **Writes all found files with the .txt extension to file.txt, while ignoring any error messages.** | **stdout to file, stderr ignored** |
| $ cat file1\_unsorted | sort > file2\_sorted | **Reads the contents of file1\_unsorted, sorts them, and writes the sorted output to file2\_sorted.** | **stdin via pipe to sort, stdout to file.** |
| $ cat myfile | grep student | wc -l | **Reads myfile, searches for lines containing "student," and counts those lines.** | **stdin and stdout via pipe to grep, then pipe to wc -l.** |

**Відповіді на контрольні запитання**

**Відповів Любежанін Максим  
  
  
  
  
  
  
1. Надайте порівняльну характеристику процесам стискання та архівування.  
  
Compression** is the process of reducing the volume of data in a file by removing redundant or duplicated information. Compression can be:

Lossless – original data can be fully recovered after decompression (eg Zip, PNG algorithms).

Lossy – some data is lost during compression, resulting in a loss of quality but significantly reducing the file size (eg JPEG for images, MP3 for audio).

**Archiving** is the process of combining multiple files and folders into a single file for easy storage or transfer. Archiving does not necessarily reduce the size of files - its main purpose is to organize and preserve the integrity of the file structure.

Compression optimizes file size by reducing file bulk, while archiving organizes files for easier storage and transfer

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

### Functionality: Supports the proprietary ARC format, providing a high level of compression.

### Advantages: High speed, modern compression algorithms, support for scripts for automation.

### Operating systems: Windows, Linux.

### 2. Gzip and Bzip2.

### Functionality: Standard compression utilities for UNIX/Linux systems; Gzip is good for fast compression, and Bzip2 is good for high quality compression.

### Advantages: Excellent compression level for text files, easy to use in a terminal.

### Operating systems: UNIX/Linux.

### 3. XZ Utilities.

### Functionality: A high compression tool specifically designed for UNIX systems.

### Advantages: One of the best compression formats for large files, particularly text files.

### Operating systems: Linux, UNIX.

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

**1. Gzip (Deflate algorithm)**

### Speed: Medium, compression and decompression is quite fast.

### Compression ratio: Moderate, better for text files, but not for binary files.

### Advantages: A well-balanced algorithm between speed and compression size.

### Usage: Ideal for compressing files with text content (e.g., log files).

### Command: gzip

### 2. Bzip2 (Burrows-Wheeler Transform algorithm from Huffman Coding)

### Speed: Slow compression, but decompression is relatively fast.

### Compression ratio: High, better than gzip for most file types.

### Advantages: Good compression for text and some binary files.

### Usage: Suitable for archives that need to be stored in a compressed form.

### The command : bzip2

### 3. Xz (LZMA algorithm)

### Speed: Very slow compression, especially at high compression levels; decompression is faster but not instantaneous.

### Compression ratio: Very high, significantly superior to gzip and bzip2.

### Advantages: One of the best options for creating small archives when speed is not critical.

### Usage: Ideal for archives that need the highest possible compression and are rarely unzipped.

### The command: xz

The fastest is LZ4, but at the expense of a lower compression ratio. Ideal for tasks where speed is important (for example, compressing data streams in real time).

The most efficient in terms of the balance between speed and compression ratio is Zstandard (zstd). This makes it a universal option for many tasks, including archiving and backup.

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

### WinZip (for Android and iOS)

### Functionality: Supports ZIP, 7z, RAR, and ZIPX. Can create and unpack ZIP archives, work with cloud services (Google Drive, Dropbox).

### Features: Can encrypt files, send files by email or instant messengers directly from the program, integrates with cloud storage.

### Operating system: Android, iOS.

### Interface: User-friendly, provides quick access to files in the cloud.

4 .**iZip (для iOS)**

**Functionality**: Supports ZIP, RAR, 7z, ZIPX. It can create and unpack ZIP archives. Can also open archives in cloud storage.

**Features**: Allows you to view, edit and share archives through other apps and services.

**Operating system**: iOS.

**Interface**: Clear and minimalistic, with an emphasis on ease of use.

### B1 Archiver (for Android and iOS)

### Functionality:

### Supports over 40 archive formats including ZIP, RAR, 7z, TAR, ISO.

### Features:

### Can work as a file manager, supports unpacking protected archives, working with multi-volume archives.

### Operating system:

### Android, iOS.

### Interface:

### Simple and functional, with ample opportunities to work with archives.

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

### Formats: RAR, ZIP, supports unpacking 7z, ISO, TAR, GZ and others.

### Functionality: Creates and decompresses RAR and ZIP archives, works with multi-volume and password-protected archives.

### Advantages: High level of compression in RAR format.

### Supports self-extracting archives (SFX).

### Intuitive interface and built-in integration into Windows Explorer.

### Disadvantages: The program is shareware, with notifications to purchase a license after the trial period.

### Best for: Use at home and in business when you need the RAR format and high compression.

### 2. 7-Zip

### Formats: 7z (proprietary high compression format), supports ZIP, RAR, TAR, GZ, BZ2, XZ, etc.

### Functionality: Create and decompress 7z, support for multi-volume archives, AES-256 encryption capability.

### Advantages: Free and open source.

### High compression level for 7z format that surpasses many others.

### Lightweight, easy-to-use interface.

### Disadvantages: The interface is somewhat simplistic and may seem less user-friendly for beginners.

### Best for: Users who need maximum compression for free or the 7z format.

**Висновки**