**"Kyiv Vocational College of Communication"**

**Cyclic Commission of Computer Engineering**

**EXECUTION REPORT**

**LABORATORY WORK No. 6**

from the discipline: "Operating systems"

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

**Performed by students of the group:**

Барабаш Матвій  
Погребняк Ілля

Целуйко Станіслав

**Checked by the teacher**

Sushanova V.S

**Work of group students КСМ-13Б Team:** **PMC wolf group**

**The goal of the work:**

1. Getting practical skills for working with the Bash command shell.

2. Acquaintance with basic commands for data archiving and compression.

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

Material provision of classes

1. IBM PC type computer.

2. OS family Windows (Windows 7).

3. Virtual machine - Virtual Box (Oracle).

4. GNU/Linux operating system - CentOS.

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

**Progress**

**2)**

|  |  |
| --- | --- |
| tar | Compresses and archives files and directories. |
| tar -cvf archive.tar file1 file2 directory1 | rchives files and directories into archive.tar file. |
| tar -xvf archive.tar | Unzips the archive.tar file. |
| gzip | Compresses files. |
| gzip file.txt | Compresses the file file.txt. |
| gunzip file.txt.gz | Unzips the file file.txt.gz. |
| bzip2 | Compresses files. |
| bzip2 file.txt | Compresses the file file.txt. |
| bunzip2 file.txt.bz2 | Unzips the file file.txt.bz2. sort Sorts text. |
| sort -n file.txt Sorts file.txt | Sorts file.txt by numeric values. |
| sort -r file.txt Sorts file.txt | Sorts file.txt in reverse order |
| uniq | Removes duplicate lines from text. |
| grep | Finds strings that match a specific pattern. |
| grep -n 'pattern' file.txt | Prints the lines matching the pattern along with their numbers. |
|  |  |
| head | Outputs the first lines of the file. |
| head -n 5 file.txt | Prints the first 5 lines of file.txt. |
| tail | Outputs the last lines of the file. |
| tail -n 5 file.txt | Outputs the last 5 lines of the file file.txt. |
| cat | Prints the contents of a file. |
| cat file1 file2 | Prints the contents of files file1 and file2. |

**4)**

|  |  |
| --- | --- |
| **cmd 1>&gt; file** | Redirects the cmd standard output stream to the file file. The contents of the file will be overwritten. |
| **cmd &gt; file** | Redirects the cmd standard output stream to the file file. If the file does not exist, it will be created. |
| **cmd 2>&gt; file** | Redirects the standard error stream of the cmd command to the file file. The contents of the file will be overwritten. |
| **cmd &gt;&gt; file** | Attaches the standard output stream of the cmd command to the file file. The contents of the file will be updated. |
| **cmd &amp;&gt; file** | Attaches the cmd standard error stream to the file file. The contents of the file will be updated. |
| **cmd &gt; file 2>&amp;1** | Redirects the cmd standard output and error stream to the file file. The contents of the file will be overwritten. |
| **cmd &gt;&gt; file 2>&amp;1** | Appends the standard output and error stream of the cmd command to the file file. The contents of the file will be updated. |
| **cmd 2&gt;&amp;1 &gt; /dev/null** | /dev/null Redirects the cmd standard error stream to /dev/null. This file is a special file that does not save the content that comes into it. |
| **cmd 2&gt; /dev/null** | /dev/null Redirects the cmd standard error stream to /dev/null. This file is a special file that does not save the content that comes into it. |
| **cmd1 | cmd2** | executes a pipeline in which the output of cmd1 is fed to the input of cmd2. The output of the cmd2 command is displayed on the screen. |
| **cmd1 2&gt;&amp;1 | cmd2** | executes a pipeline in which the output of the cmd1 command and its standard error stream are fed to the input of the cmd2 command. The output of the cmd2 command is displayed on the screen. |

**5)**

|  |  |  |
| --- | --- | --- |
| **$echo &quot;It is a new story.&quot; &gt; story** | Displays the line "It is a new story." and redirects it to the story file. | Standard output stream |
| **$ date &gt; date.txt** | Displays the current date and time and redirects it to the date.txt file. | Standard output stream |
| **$ cat file1 file2 file3 &gt; bigfile** | Combines the contents of files file1, file2 and file3 into one file bigfile. | Standard output stream |
| **$ls -l &gt;&gt; directory** | Prints a list of files and directories in the current directory and completes the contents of the directory file. | Standard output stream |
| **$ sort &lt; file1\_unsorted &gt; file2\_sorted** | Sorts the contents of the file file1\_unsorted and redirects it to the file file2\_sorted. | Standard output stream |
| **$ find -name &#39;\*.txt&#39; &gt; file.txt 2&gt; /dev/null** | Finds all files with the extension .txt in the current directory and redirects their contents to the file file.txt. Error streams are redirected to /dev/null so that they are not displayed. | Standard output stream, standard error stream |
| **$ cat file1\_unsorted | sort &gt; file2\_sorted** | sort &gt; file2\_sorted` | Sorts the contents of file1\_unsorted and redirects it to file2\_sorted. |
| **$ cat myfile | grep student | wc -l** |  |  |

**Controll question**

**1. Compression and Archiving**

- Compression is the process of reducing the size of a file without loss of information. This is achieved by removing redundant data or using other methods to reduce the file size.

- Archiving is the process of combining one or more files into a single archive file. Archive files are often used for backup, data transfer, or data storage.

**2. Programs for Compression and Archiving in Linux**

- gzip is the standard compression algorithm in Linux. It uses the LZW algorithm, which is effective for compressing text files.

- bzip2 is a compression algorithm that provides a higher compression ratio than gzip. However, it is also slower.

- xz is a new compression algorithm that provides high compression ratio and speed.

- tar is an archiving program that can use different compression algorithms.

- zip is a popular compression algorithm that can also be used in Linux.

**3. Compression Algorithms in Linux**

- LZW is a compression algorithm that uses a table to store repeated data.

- DEFLATE is a compression algorithm that uses the LZW algorithm to compress data.

- LZMA is a compression algorithm that provides high compression ratio.

- XZ is a new compression algorithm that provides high compression ratio and speed.

**4. Programs for Compression and Archiving in Mobile Phones**

- ZArchiver is a popular archiving program that supports different compression algorithms.

- 7-Zip is a port of the 7-Zip program for mobile phones.

- RAR is a port of the WinRAR program for mobile phones.

**5. Programs for Compression and (De)archiving Data in Windows Operating Systems**

- WinRAR is a popular archiving program that supports different compression algorithms.

- 7-Zip is a free archiving program that also supports different compression algorithms.

- WinZip is another popular archiving program.

**6. Using Compression and Archiving for Data Backup**

- Compression and archiving can be used for data backup to reduce the time and cost of storing backups. Archive files can also be used to create backups on multiple media, such as CDs, DVDs, or external hard drives.

- Other System Administration Tasks

- Data Transfer. Archive files can be used to transfer data over the Internet or by email.

- Data Storage. Archive files can be used to store data on hard drives, SSDs, or other storage devices.

- Data Distribution. Archive files can be used to distribute data between users or computers.

**7. Purpose of the file directory /dev/null**

- The /dev/null file directory is a special device that discards any data written to it and reads zero when read from it. It is also known as a "black hole".

\* The /dev/null directory is used for a variety of purposes, including:

- Redirecting the standard output (stdout) or standard error output (stderr) of programs. This can be used to hide program output or to redirect program output to a different file.

- Creating empty files. This can be done using the dd command with the following syntax: dd if=/dev/null of=filename.

- Testing the correctness of programs. If a program is outputting errors, you can redirect the error output to /dev/null to check if the program is still working properly.

**Conclusion**

During the execution of LB, I learned new commands for archiving text and data and how to work with text in the terminal. But due to a problem with the terminal, it didn't work out a bit in practice.

**The work was done by Barabash Matviy**

**Translated by Tselujko Stanislav**

**Ilya Pogrebnyak was looking for material**