

“Kyiv Vocation College of Communication”
Cyclical Commission of Computer Engineering

REPORT ON EXECUTION
LABORATORY WORK №6
on the discipline: "Operating Systems"

Topic: "Linux Commands for Data Archiving and Compression.
Working with Text"

Performed by student
group RPZ-13a
Kateryna Hranat
Checked by teacher
V.S. Sushanova

Kyiv 2024

Work objectives:

1. Acquiring practical skills in working with the Bash command shell.
2. Introduction to basic commands for archiving and compressing data.
3. Familiarity with basic operations when working with text in the terminal.

Material Support for Classes:

1. IBM PC type computers.
2. Windows operating system and VirtualBox virtual machine (Oracle).
3. GNU/Linux operating system (any distribution).
4. Cisco Networking Academy website [netacad.com](https://www.netacad.com) and its online courses on Linux.

Tasks for Preliminary Preparation:

1. Read the brief theoretical information for the laboratory work and create a small dictionary of basic English terms related to the purpose of commands and their parameters.

English terms	Ukrainian terms
Lossy	Втратний
Lossless Compression	Безвтратне стиснення
Compression	Стиснення
Archival Basics	Основи архівації
Archival Background	Основи архівування

2. Study the materials of the online course "NDG Linux Essentials" from Cisco:
 - Chapter 09 - Archiving and Compression
 - Chapter 10 - Working With Text

Complete✓

3. Complete testing in the NDG Linux Essentials course on the following topics:
 - Chapter 09 Exam
 - Midterm Exam (Modules 1 - 9)
 - Chapter 10 Exam

Complete✓

4. Based on the material covered, provide answers to the following questions:
 - 4.1. *What is the purpose of the commands tar, xz, zip, bzip, gzip? Provide a brief description of each command and highlight their main parameters. How can they be installed?

Commands for Data Archiving and Compression:

- **tar**: The purpose of this command is to create an archive of files or directories, or to extract such archives. Key parameters: ``c`` (create archive), ``x`` (extract archive), ``z`` (use gzip for compression/extraction), ``j`` (use bzip2 for compression/extraction).

- **xz**: This command is used for compressing files. Key parameter: ``-z`` (compress file).

- **zip**: The zip command creates or extracts ZIP archives. Key parameters: ``-r`` (recursive archive creation), ``-d`` (delete files from archive), ``-u`` (update archive).

To install these commands, you can use the package manager of your operating system, for example, for many Linux distributions, this could be apt, yum, or dnf.

4.2. **Give three examples of implementing data archiving and compression using different commands.

Using tar with gzip compression:

tar -czvf archive.tar.gz directory_to_compress

This command creates a compressed tar archive of the specified directory using gzip compression.

Using zip for compression:

zip -r archive.zip directory_to_compress

This command creates a ZIP archive of the specified directory recursively.

Using xz for compression:

tar -c directory_to_compress | xz > archive.tar.xz

This command creates a tar archive of the specified directory and pipes it to xz for compression, resulting in an xz-compressed tarball.

4.3. *What is the purpose of the commands cat, less, more, head, and tail? Provide a brief description of each command and highlight their main parameters. How can they be installed?

Commands for Working with Text Files:

- **cat:** The purpose of this command is to output the contents of files to the console. Key parameter: none.
- **less:** This command allows viewing the contents of files gradually, allowing scrolling up and down. Key parameter: none.
- **more:** Similar to less, more allows viewing the contents of files gradually but with less functionality. Key parameter: none.
- **head:** This command outputs the first few lines of a file. Key parameter: `-n` (number of lines).
- **tail:** The tail command outputs the last lines of a file. Key parameter: `-n` (number of lines).

To install these commands, you typically don't need to install them separately as they are already included in operating systems such as Linux or macOS.

4.4. **Explain the principles of command shell operation with pipes, streams, and filters.

Principles of Command Shell Operation with Pipes, Streams, and Filters:

The command shell (e.g., bash in Linux) allows using pipes, which transmit the output of one program as input to another program. For example, the command `command1 | command2` will execute command1 and then pass its output as input to command2.

Streams are mechanisms for transmitting data, such as stdin (standard input), stdout (standard output), and stderr (standard error output).

Filters are programs that read data from stdin, process it, and then output the result to stdout. For example, the command `command | grep pattern` will execute command and then pass its output to the grep program, which will filter lines containing a specific pattern.

4.5. *What is the purpose of the grep command?

The `grep` command is designed to search for text strings that match a specific pattern in one or more files. Key parameters: `-r` (recursive

search in subdirectories), `-i` (ignore case), `-v` (output lines that do not match the pattern).

5. Prepare an initial version of the report in electronic form:
 - Title page, topic, and purpose of the work
 - Glossary of terms
 - Answers to points 2.1 - 2.3 from the tasks for preliminary preparation

Complete✓

Progress of Work:

1. Initial work in CLI mode in a Linux operating system of the Linux family:
 - 1.1. Start your Linux-based operating system (if you are using your own PC and have it installed) and open the terminal.
2. Work through all the command examples provided in the lab assignments of the NDG Linux Essentials course - Lab 9: Archiving and Compression and Lab 10: Working With Text. Create a table to describe these commands.

Command	Description
<code>`tar -czvf archive.tar.gz [files]`</code>	Create and compress an archive from specified files using gzip.
<code>`tar -xzvf archive.tar.gz`</code>	Extract an archive created with gzip compression.
<code>`zip -r archive.zip [files]`</code>	Create a ZIP archive from specified files and folders recursively.
<code>`unzip archive.zip`</code>	Extract a ZIP archive.
<code>`gzip [file]`</code>	Compress the specified file using gzip, creating a .gz file.

<code>`gunzip [file.gz]`</code>	Decompress a file compressed with gzip.
<code>`xz [file]`</code>	Compress the specified file using xz, creating a .xz file.
<code>`xz -d [file.xz]`</code>	Decompress a file compressed with xz.
<code>`cat [file1] [file2]`</code>	Display the contents of specified files in the console.
<code>`less [file]`</code>	Allow browsing the content of a file gradually with the ability to scroll up and down.
<code>`more [file]`</code>	Allow browsing the content of a file gradually, but with less functionality compared to less.
<code>`head -n [number] [file]`</code>	Output the first few lines of a file.
<code>`tail -n [number] [file]`</code>	Output the last few lines of a file.
<code>`grep [pattern] [file]`</code>	Search for a specified pattern in a file and output lines containing it.

3. Familiarize yourself with the tar command and use it in the terminal to perform the following actions:

- Create a file with the extension .tar;

```
sysadmin@localhost:~$ tar -cf archive.tar file1 file2 file3
```

- Create a file with the extension .tar consisting of multiple files and directories simultaneously;

```
sysadmin@localhost:~$ tar -cf archive.tar directory1 file1 file2
```

- View the contents of the file;

```
sysadmin@localhost:~$ tar -tf archive.tar
```

- Extract the contents of the tar file;

```
sysadmin@localhost:~$ tar -xf archive.tar
```

- Create a tar archive file compressed using bzip;

```
sysadmin@localhost:~$ tar -cjf archive.tar.bz2 file1 file2 file3
```

- Extract the contents of the tar bzip file;

```
sysadmin@localhost:~$ tar -xjf archive.tar.bz2
```

- Create a tar archive file compressed using gzip;

```
sysadmin@localhost:~$ tar -czf archive.tar.gz file1 file2 file3
```

- Extract the contents of the tar gzip file.

```
sysadmin@localhost:~$ tar -xzf archive.tar.gz
```

4. *How will output stream redirection occur in bash for the following actions with commands (denoted as cmd) and files (denoted as file):

Команда	Що виконує команда?
cmd 1> file	Redirects the standard output (stdout) of `cmd` to `file`, overwriting its contents if it exists.
cmd > file	Same as above, redirects stdout of `cmd` to `file`, overwriting its contents if it exists.
cmd 2> file	Redirects the standard error (stderr) of `cmd` to `file`, overwriting its contents if it exists.
cmd >> file	Redirects stdout of `cmd` to `file`, appending to its contents if it exists.
cmd &> file	Redirects both stdout and stderr of `cmd` to `file`, overwriting its contents if it exists.
cmd > file 2>&1	Redirects stdout of `cmd` to `file` and stderr to the same location as stdout, effectively redirecting both to `file`.
cmd >> file 2>&1	Redirects stdout of `cmd` to `file`, appending to its contents if it exists, and redirects stderr to the same location as stdout.

cmd 2>&1 > /dev/null	Redirects stderr of `cmd` to the same location as stdout (which is `/dev/null`, meaning discard), then redirects stdout to `/dev/null` (discard).
cmd 2> /dev/null	Redirects stderr of `cmd` to `/dev/null`, discarding it.
cmd1 cmd2	Redirects stdout of `cmd1` to stdin of `cmd2`, creating a pipeline where the output of `cmd1` is used as input for `cmd2`.
cmd1 2>&1 cmd2	Redirects stderr of `cmd1` to stdout, then pipes stdout of `cmd1` to stdin of `cmd2`.

5. **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	Writes the string "It is a new story." to the file 'story'.	Standard output redirection.
\$ date > date.txt	Writes the current date and time to the file 'date.txt'.	Standard output redirection.
\$ cat file1 file2 file3 > bigfile	Concatenates the contents of file1, file2, and file3 into a single file named 'bigfile'.	Standard output redirection.
\$ls -l >> directory	Appends the long listing of the current directory to the file 'directory'.	Standard output redirection (appending).
\$ sort < file1_unsorted > file2_sorted	Reads the contents of 'file1_unsorted', sorts them, and writes the sorted output to 'file2_sorted'.	Standard input redirection.
\$ find -name '*.txt' > file.txt 2> /dev/null	Searches for files with the '.txt' extension and writes the output to 'file.txt'. Errors (if any) are redirected to /dev/null.	Standard output and error redirection.
\$ cat file1_unsorted sort > file2_sorted	Pipes the contents of 'file1_unsorted' to the sort command, and then writes	Standard input/output redirection.

	the sorted output to 'file2_sorted'.	
\$ cat myfile grep student wc -l	Reads the contents of 'myfile', filters lines containing 'student' using grep, and counts the number of lines using wc -l.	Standard input/output redirection.

Control questions

1. Provide a comparative characteristic of compression and archiving processes.

Comparative characteristic of compression and archiving processes:

- **Compression:** Compression is the process of reducing the size of a file or data by encoding information using fewer bits than the original representation. This reduces storage space and transmission time. Compression can be lossless (original data can be perfectly reconstructed) or lossy (some data is discarded, but the overall meaning is preserved).
- **Archiving:** Archiving is the process of combining multiple files and directories into a single file (archive). It's often used for data organization, backup, and transfer. Archiving doesn't necessarily reduce the size of the files; its main purpose is to consolidate multiple files into one for easier management.

2. Besides the ones mentioned in the work, what programs can be used for file and directory compression and archiving in Linux OS? Provide examples and brief descriptions.

Other programs for file and directory compression and archiving in Linux:

- **7z:** A compression tool that supports various formats like 7z, ZIP, GZIP, BZIP2, XZ, TAR, and WIM. It offers high compression ratios and strong encryption.

- **rar**: A proprietary compression format and tool for creating and extracting RAR files. It offers a good compression ratio and supports splitting archives into multiple volumes.

- **zstd**: A fast compression algorithm providing both compression and decompression.

- **pigz**: A parallel implementation of gzip for faster compression on multi-core processors.

3. *Compare compression algorithms used in commands (programs) used in Linux. Which algorithms can be considered the fastest and most efficient?

Comparison of compression algorithms in Linux:

- **gzip**: Uses the DEFLATE algorithm, offering good compression ratios and reasonable speed.

- **bzip2**: Employs the Burrows-Wheeler block sorting text compression algorithm. It provides better compression ratios than gzip but is slower.

- **xz**: Utilizes the LZMA (Lempel-Ziv-Markov chain algorithm) compression, providing excellent compression ratios but slower compression and decompression speeds.

- **zstd**: Offers fast compression and decompression speeds with good compression ratios, making it suitable for real-time compression needs.

Among these, 'zstd' is often considered one of the fastest and most efficient compression algorithms, balancing compression ratio and speed effectively.

4. *Describe software tools for compression and archiving that can be used on your mobile phone.

Software tools for compression and archiving on mobile phones:

- **WinRAR**: Available for Android, it allows creating and extracting RAR and ZIP archives.

- **ZArchiver**: An Android app supporting multiple archive formats like ZIP, RAR, 7z, TAR, and others, with features for creating, extracting, and managing archives.

5. *Describe and compare software tools for data compression and (de)archiving in Windows OS family.

Software tools for data compression and (de)archiving in Windows OS:

- **WinRAR:** A popular file archiver for Windows supporting various formats like RAR, ZIP, CAB, ARJ, LZH, TAR, GZIP, UUE, ISO, BZIP2, Z, and 7-Zip.

- **7-Zip:** A free, open-source file archiver supporting high compression ratios in 7z format, as well as decompressing many other formats like ZIP, TAR, WIM, and more.

6. **Explain how data compression and archiving can be used for data backup. In what other system administration tasks can it be used?

Data compression and archiving for data backup and system administration:

- **Data Backup:** Compression and archiving help in creating backups of files and directories, reducing storage space and facilitating transfer and storage.

- **System Administration:** They are used for packaging and distributing software, creating system snapshots for recovery, transferring large amounts of data efficiently, and maintaining system configuration files in a structured manner.

7. **What is the purpose of the /dev/null file directory?

Purpose of the /dev/null file directory:

- `/dev/null` is a special device file in Unix-like operating systems that discards all data written to it. It acts as a black hole in the file system, where any data sent to it is immediately discarded. This file is often used to suppress output (stdout or stderr) from commands or scripts when their output is not needed. It's commonly used for sending unwanted output to oblivion.

Conclusions:

During the course of the laboratory work, I gained practical skills in working with the Bash command shell. Introduction to basic commands for archiving and compressing data. Familiarity with basic operations when working with text in the terminal. I improved my English language skills. I prepared for the work, completed the assigned tasks, and answered control questions.