“Kyiv specialized College of Communications”

Commission of computer engineering

**REPORT ON THE IMPLEMENTATION**

**LABORATORY WORK №6**

From the discipline: "Operating systems"

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

The students

performed Groups RPZ-03

Team 3:

Kanavets K.S.,

Kryvenko A.I.,

Kulikovska M.V.

Checked by the teacher

Sushanova V.S.

Кyiv 2023

Tasks for preliminary preparation.

**The material was prepared by student Andrew Kryvenko (@AndrewKryvenko)**

**1. Read the brief theoretical information for the lab and make a small glossary**

**of basic English terms related to the purpose of commands and their parameters.**

significant data - важливі дані

lossless compression - стиснення без втрат

archival background - архівний вигляд

.gz extension - розширення .gz.

memory requirements - вимоги до пам'яті

**4. Based on the material reviewed, answer the following questions:**

**1)What is the purpose of the tar, xz, zip, bzip, gzip commands? Make a brief description of each command and highlight their main parameters.**

tar: The tar command is used to create and manipulate archives (also known as tarballs) of files and directories. It can be used to create compressed or uncompressed archives, add or extract files from an existing archive, and more. Its main parameters are:

c: create a new archive

x: extract files from an archive

v: verbose mode (display progress and list of files being processed)

f: specify the name of the archive file

xz: The xz command is used to compress and decompress files using the LZMA algorithm. Its main parameters are:

c: create a compressed file

d: decompress a compressed file

k: keep the original file after compression/decompression

zip: The zip command is used to create, modify and extract archives in the ZIP format. Its main parameters are:

c: create a new archive

x: extract files from an archive

v: verbose mode (display progress and list of files being processed)

r: add files recursively

bzip: The bzip command is used to compress and decompress files using the Burrows-Wheeler algorithm. Its main parameters are:

c: create a compressed file

d: decompress a compressed file

k: keep the original file after compression/decompression

gzip: The gzip command is used to compress and decompress files using the Lempel-Ziv algorithm. Its main parameters are:

c: create a compressed file

d: decompress a compressed file

k: keep the original file after compression/decompression

**2)Give three examples of how to implement archiving and data compression using different commands.**

Three examples of how to implement archiving and data compression using different commands:

To create a compressed tarball of a directory named "mydir", use: tar -czvf mydir.tar.gz mydir

To compress a file named "myfile.txt" using bzip, use: bzip -c myfile.txt > myfile.txt.bz2

To extract a zip archive named "myarchive.zip", use: unzip myarchive.zip

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

cat: The cat command is used to display the contents of one or more files on the terminal. Its main parameters are:

n: display line numbers

b: display line numbers only for non-blank lines

E: display an '$' symbol at the end of each line

less: The less command is used to view the contents of a file one page at a time, allowing you to scroll through the file. Its main parameters are:

N: display line numbers

F: keep the file open and display new content as it's added

more: The more command is similar to less, but with fewer features. It's used to display the contents of a file one page at a time. Its main parameter is:

N: display line numbers

head: The head command is used to display the first few lines of a file. Its main parameter is:

n: specify the number of lines to display (default is 10)

tail: The tail command is used to display the last few lines of a file. Its main parameters are:

n: specify the number of lines to display (default is 10)

f: keep the file open and display new content as it's added

These commands are typically built into most Unix-based systems and do not require installation. To set parameters or options for these commands, you can include them in the command line after the command name. For example, to display the contents of a file with line numbers using the cat command, you can run:

cat -n myfile.txt

Similarly, to display the first 20 lines of a file using the head command, you can run:

head -n 20 myfile.txt

And to display the last 5 lines of a file, with new content added as it's added, using the tail command, you can run:

tail -f -n 5 myfile.txt

**4)Explain how the command shell works with channels, streams, and filters**

The command shell is a program that provides an interface for users to interact with their operating system. It accepts commands from the user, interprets them, and executes them. Channels, streams, and filters are important concepts in the command shell that help to manipulate and redirect input and output.

Channels: A channel is a connection between two processes, typically used for passing data between them. In the command shell, channels are used to redirect the input and output of commands. There are three types of channels:

STDIN (standard input): The channel through which commands receive input, typically from the keyboard or from a file using redirection.

STDOUT (standard output): The channel through which commands send their output, typically to the terminal or to a file using redirection.

STDERR (standard error): The channel through which commands send error messages, typically to the terminal or to a file using redirection.

Streams: A stream is a sequence of data, typically used for input or output. In the command shell, streams are used to redirect and manipulate data being sent between commands. There are two types of streams:

Input stream: A stream of data being sent to a command, typically from a file or from the output of another command using a pipe.

Output stream: A stream of data being sent from a command, typically to the terminal or to a file using redirection or to another command using a pipe.

Filters: A filter is a command that reads input data from STDIN, processes it, and then writes output data to STDOUT. Filters are often used in combination with pipes to create complex command chains.

**5)What is the purpose of the grep command?**

The grep command is a powerful tool used to search for a specific pattern or string of characters within one or more files.

**The material was prepared by student Andrew Kryvenko (@AndrewKryvenko)**

*2.Work through all of the sample commands presented in the NDG Linux Essentials labs - Lab 9: Archiving and Compression and Lab 10: Working With Text. Create a table to describe these commands\*\*\*.*

|  |  |
| --- | --- |
| Name of command | Its purpose and functionality |
| mkdir mybackups | Create a new mybackups directory in the user's home directory |
| tar -cvf mybackups/udev.tar /etc/udev | The tar command is used to combine several files into a single file. In this case, the contents of the /etc/udev directory will be saved to the udev.tar archive in the mybackups directory. The -c option tells the tar command to create a tar file. The -v option stands for "verbose," which tells the tar command to show what it is doing. The -f option is used to specify the name of the tar file. |
| Tar -tvf mybackups/udev.tar | The command "tar -tvf mybackups/udev.tar" is used to list the contents of a tar archive named "udev.tar" that is located in the "mybackups" directory. |
| ls -lh mybackups/ | The command "ls -lh mybackups/" is used to list the contents of the "mybackups" directory with detailed information about each file. |
| ls etc/udev | The command "ls etc/udev" is used to list the contents of the "udev" directory located in the "/etc" directory. |
| ls etc/udev/rules.d | The command "ls etc/udev/rules.d" is used to list the contents of the "rules.d" directory located within the "udev" directory in the "/etc" directory. |
| tar -rvf udev.tar /etc/hosts | The command "tar -rvf udev.tar /etc/hosts" is used to add the "/etc/hosts" file to an existing tar archive called "udev.tar" |
| zip -r udev.zip /etc/udev | The command "zip -r udev.zip /etc/udev" is used to create a zip archive called "udev.zip" that contains the entire "udev" directory located in the "/etc" directory. |
| find ~ -name "\*bash\*" | The command "find ~ -name "bash"" is used to search for files and directories within the home directory that contain the string "bash" in their name. |
| find /etc -name hosts | The command "find /etc -name hosts" is used to search for files and directories within the "/etc" directory that have the exact name "hosts". |
| find /etc -name hosts 2> err.txt | The command "find /etc -name hosts 2> err.txt" is used to search for files and directories within the "/etc" directory that have the exact name "hosts", and any error messages produced by the "find" command will be redirected to a file called "err.txt" |
| find /etc -name hosts > std.out 2> std.err | The command "find /etc -name hosts > std.out 2> std.err" is used to search for files and directories within the "/etc" directory that have the exact name "hosts", and any standard output and error messages produced by the "find" command will be redirected to separate files. |
| ls /etc | tail -5 | The command "ls /etc | tail -5" is used to list the contents of the "/etc" directory and display only the last 5 lines of the output. |
| head -n -20 /etc/passwd | The command "head -n -20 /etc/passwd" is used to display all lines of the "/etc/passwd" file except for the last 20 lines |
| /etc$ grep 'sync$' passwd | The command "grep 'sync$' passwd" searches for all lines in the "passwd" file located in the "/etc" directory that end with the word "sync". |
| /etc$ grep '.y' passwd | The command "grep '.y' passwd" searches for all lines in the "passwd" file located in the "/etc" directory that contain the letter "y" preceded by any other character. |
| /etc$ grep 'sshd|root|operator' passwd | The command "grep 'sshd|root|operator' passwd" searches for all lines in the "passwd" file located in the "/etc" directory that contain any of the three strings "sshd", "root", or "operator". |
| /etc$ grep -E 'sshd|root|operator' passwd | The command "grep -E 'sshd|root|operator' passwd" is similar to the previous command you asked about, but it uses the "-E" option to enable extended regular expressions |
| /etc$ head passwd | grep '[0-9]' | The command "head passwd | grep '[0-9]'" will print the first 10 lines of the "passwd" file (since we didn't specify a line count with the "head" command), and then search those lines for any digits using the regular expression "[0-9]". |

**The material was prepared by student Kanavets Kateryna (@kanavetsk)**

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

- create a file with the .tar extension;

The tar (or Tape Archive) command is a utility for creating and working with archive files that stores many files in one file and can also compress data.

To create an archive file with the .tar extension:



- create a file with the .tar extension that contains multiple files and directories at once;

To create an archive file with the .tar extension that contains multiple files and directories at once:



- view the contents of the file;

To view the contents of the archive file:



Щоб витягти вміст архівного файлу:

- extract the contents of a tar file;

To extract the contents of an archive file:



- create a bzip compressed tar archive file;

To create a bzip compressed tar archive file:



- extract the contents of a bzip tar file;

To extract the contents of a tar archive file with bzip compression:



- create a gzip compressed archive tar file;

To create an archive tar file compressed with gzip:



- extract the contents of a gzip tar file.

To extract the contents of a tar archive file with gzip compression:



**The material was prepared by student Kulikovska Maria (@Smith5004)**

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

|  |  |
| --- | --- |
| Command | What does the command do? |
| cmd 1> file | The thread of the standard cmd command output will be redirected to a file named file. If the file does not exist, it will be created. If the file already exists, its contents will be overwritten by the output from the cmd command. |
| cmd > file | Directs the standard output thread to a file. If the file does not exist it will be created, if it exists it will be overwritten from above. |
| cmd 2> file | Directs the standard error thread to a file. If the file does not exist it will be created, if it exists it will be overwritten from above. |
| cmd >> file | Directs the standard output thread to a file. If the file does not exist it will be created, if it does, the data will be appended to it at the end. |
| cmd &> file | This command redirects both the output stream and the error stream to the file file. This means that the results of the cmd command will be written to the file file, and any errors that occur during the execution of the cmd command will also be written to that file. |
| cmd > file 2>&1 | This command redirects the output stream to the file file and the error stream is redirected to the same file as the output stream (stdout). This means that the results of the cmd command will be written to the file file and any errors that occur during the execution of the cmd command will also be written to that file. |
| cmd >> file 2>&1 | This command is similar to cmd > file 2>&1 except that the output stream and error stream will be added to the end of the file instead of overwritten. Thus, the results of the cmd command and any errors that occur during the execution of the cmd command will be added to the end of the file. |
| cmd 2 > &1> /dev/null | This command redirects the error stream of the "cmd" command into the same stream as the standard output stream and then redirects the standard output stream into /dev/null, which ignores all data sent to it. In this way, no data will be output to the screen or files. |
| cmd 2 > /dev/null | This command redirects the error stream of the "cmd" command to the /dev/null device, ignoring all errors. In this way no errors will be shown on the screen or in the |
| cmd1 | cmd2 | This command sends the standard output of one process to the standard output of another process. |
| cmd1 2>&1 | cmd2 | This command redirects both the standard output stream and the error stream of command "cmd1" to the standard input stream of command "cmd2". The result of "cmd1", including the error stream, will be passed as input to "cmd2". |

**The material was prepared by student Kanavets Kateryna (@kanavetsk)**

*5.Consider the following examples and explain what these commands do and what type of flow redirection they use:*

|  |  |  |
| --- | --- | --- |
| **Command**  **(command container)** | **What does the command do?** | **What is the redirection thread?** |
| $echo "It is a new story." > story | This command writes a line of text "This is a new story." to a file named story. | The redirect > operation indicates that the contents of the echo command output will be written to the story file. If the file exists, the contents of the file will be overwritten. |
| $ date > date.txt | This command writes the current date and time to a file named date.txt. | The redirect > operation indicates that the contents of the date command output will be written to the date.txt file. If the file exists, the contents of the file will be overwritten. |
| $ cat file1 file2 file3 > bigfile | This command reads the contents of files file1, file2, and file3 and writes them to a file named bigfile. | The redirect > operation indicates that the contents of these files will be written to the bigfile. If the bigfile file exists, the contents of the file will be overwritten. |
| $ls -l >> directory | This command executes the ls -l command for the current directory and appends the result to the end of the file named directory. | The >> redirect operation indicates that the contents of the ls -l command output will be appended to the end of the directory file. If the file exists, the contents of the file will not be overwritten, but appended to the end of the file. |
| $ sort < file1\_unsorted > file2\_sorted | This command reads the contents of the file file1\_unsorted and sorts its lines, writing the result to a file named file2\_sorted. | The < redirect operation indicates that the input stream for the sort command will be taken from the file1\_unsorted file. The > redirect operation indicates that the output of the sort command will be written to file2\_sorted. |
| $ find -name '\*.txt' > file.txt 2> /dev/null | Searches the current directory for all files with the .txt extension. | Redirects the list of found files to the file.txt file. |
| $ cat file1\_unsorted | sort > file2\_sorted | Takes the contents of file1\_unsorted, sorts it, and redirects the result to file2\_sorted. | In the cat file1\_unsorted | sort > file2\_sorted command, the standard output of the sort command is redirected to the file2\_sorted file. |
| $ cat myfile | grep student | wc -l | Reads the contents of the myfile file, filters the lines containing the word "student" using the grep command, and counts the number of such lines using the wc -l command. | In the cat myfile | grep student | wc -l command, the standard output of the wc -l command (the number of lines containing the text you are looking for) is redirected to the standard terminal output. |

**The material was prepared by student Kulikovska Maria (@Smith5004)**

Control questions

1.Give a comparative description of compression and archiving.

Compression reduces the amount of data needed to store or transfer a file, saving it in such a way that the file can be recovered. In a file with legible text, frequently used words can be replaced with something smaller, or an image with a solid background can represent sections of that color using code. A compressed version of a file is usually not viewed and used, instead it is unpacked before use.

Archiving is compressing one or more files to save memory and placing the compressed data into a single archive file. Data archiving is reducing the physical size of the files in which the data is stored without significant information loss.

2.What programs, other than those given in the work, can be used to compress and archive files and directories in Linux? Give examples and a brief description.

Shar allows you to create self-extracting archives. It is a shell script and needs Bash or another Bourne Shell compatible shell to unpack it. Shar has several advantages but it is also potentially insecure because the archive is an executable file.

Examples of how to use shar to archive a linux folder:

Create a shar archive: shar file\_name.extension > filename.shar

Unpack the shar archive: ./filename.shar

ar is a utility to create and manage archives. It is mostly used to archive static libraries, but can be used to create any kind of archives. Used quite often in the past, but was superseded by tar. Now it is used only for creating and updating static library files.

Examples of use.

Create a static library, libmath.a, from the object files substraction.o and division.o:

ar cr libmath.a substraction.o division.o

Extract the files from the archive:

ar x libmath.a

You can unpack any static library in this way.

cpio stands for Copy in and out. This is another standard archiver for Linux. It is actively used in the Red Hat package manager and also to create initramfs. It does not perform archiving in Linux for normal files.

Example usage.

Creating a cpio archive:ls file1.o file2.o file3.o

ls | cpio -ov > /path/to/output\_folder/obj.cpio

Unpack the archive:

cpio -idv < /path/to folder/obj.cpio

Kgb is a lesser known but effective archiving tool, capable of compressing multiple files into a small archive with high compression. It supports all common Linux file formats such as zip, gzip, 7zip and rar.

File Roller is an archive drawer in the GUI for Linux. It can create or modify archives and display the contents of the archive. It is a good alternative to 7zip or WinRar. Supports almost all file formats such as .iso, .lsa, 7zip, gzip, zip, zz, etc. Embedded in the Mint and Ubuntu distributions.

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

1. gzip is a popular compression algorithm that is widely used in Linux. It is fast and provides good compression, although it can be slightly slower than some other compression algorithms.

2) lz4 is a lossless compression algorithm that is faster than gzip and provides good compression. This makes it ideal for real-time data compression.

3) bzip2 is a compression algorithm that provides better compression than gzip, but is slower. It uses a sophisticated compression algorithm that revisits the data in multiple passes to achieve better compression.

4) xz is a compression algorithm that uses more advanced data compression techniques and provides better compression than gzip and bzip2.

4.Describe the compression and archiving software that can be used on your mobile phone.

PAP is a free, simple and fast file compression application for Android. It's very similar to the WinRAR app you use on your desktop computer, but it's designed for mobile devices. With RAR, you can create compressed files in RAR and ZIP format. In addition, you can also decompress RAR, ZIP, TAR, GZ, BZ2, XZ, ISO archives.

B1 Archiver is a free tool for creating and extracting zip files. The application supports 37 different compression formats. You can use the app to compress or extract zip files.

ZArchiver is a more advanced version of the aforementioned RAR application. The only thing that makes ZArchiver different from RAR is file support. With ZArchiver you can create ZIP, BZIP2, GZIP, XZ, TAR, ZST and some other archives. In addition, it allows you to create and uncompress password-protected compressed files, edit archives, etc.

The zipper got the ability to extract and compress almost every major compression file format. The app also brings a task manager, flash viewer, file manager, image viewer, etc. Apart from that, the app also allows users to transfer zip files via WiFi. What makes Zipper more useful is that it also offers file management and remote file management features via FTP/HTTP.

5.Describe and compare software tools for data compression and (de)archiving in Windows operating systems.

WinZip is one of the most well-known and widely used software tools for data compression and (de)archiving. It is a commercial software that supports most popular compression formats such as ZIP, RAR, 7Z and others.

7-Zip is a free and open source software. It also supports a large number of compression formats, including ZIP, RAR, 7Z and others. One of the key features of 7-Zip is the high level of compression that can be achieved when using its compression algorithms.

WinRAR is another commercial data compression and (de)archiving tool that supports a variety of formats, including ZIP, RAR, 7Z, and others. It also has a user-friendly interface and numerous additional features, such as password protection, splitting archives into parts, etc.

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 can be very useful for backing up data because they reduce the file size and the amount of data stored. This can help you save space on your hard drives, as well as reduce the time it takes to transfer files over the Internet or local network.

Some ways to use data compression and archiving for backups:

Compressing files before saving them to an external storage device, such as an external hard drive, flash drive, or cloud storage. Compression reduces file size, which speeds up the process of copying to an external storage device and saves space on the external storage device.

Using archivers to pack multiple files into a single archive. This reduces the number of files that need to be copied, and makes it easy to save and transfer the files in their entirety.

Creating backups using specialized backup programs that can automatically compress and archive data. These programs often allow you to set settings for compression, such as the degree of compression and the type of compression algorithm used.

Using command-line tools to compress and archive files, such as gzip, tar, and 7-Zip.

In addition to data backup, data compression and archiving can be used for other system administration tasks, such as:

Compressing log files: Log files can be compressed to save disk space and make them easier to transfer for analysis.

Archiving old data. Old data can be archived to save space in the production file system while keeping the data available for future use.

Deploying software. Software packages can be compressed and archived to simplify their deployment across multiple systems.

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

In Linux "/dev/null " is a virtual file with a unique way of handling files. Anything written to /dev/null will disappear and cannot be recovered.

It acts as a black hole for the data and is often used to remove unwanted command output or to display a file by moving it to /dev/null. Its purpose is to provide a place for the data to be discarded.

Conclusion.

In this lab, we learned how to create archives in Linux, which is a convenient way to protect and store files. We looked at the main ways to archive a folder using various utilities and options, and we also looked at compressing files with the tar, zip, gzip, bzip2, and xz extensions. Each utility has its own syntax and commands for performing operations such as creating, unpacking, and viewing archives.