“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної та програмної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №5**

з дисципліни: «Операційні системи»

**Тема:** **“Команди Linux для архівування та стиснення даних. Робота з текстом”**

Виконали студенти

групи КСМ-13а

MathematicainsLivesMatter: Дзизиль Д. Є., Чех І. В.

Перевірив викладач

Сушанова В. С.

Київ 2023

**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 (Windows 7).

3. Віртуальна машина – Virtual Box (Oracle).

4. Операційна система GNU/Linux – CentOS.

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

**Завдання для попередньої підготовки**

***Готував матеріал студент Дзизиль Д.Є.***

1. На базі розглянутого матеріалу дайте відповіді на наступні питання:
   1. Яке призначення команд *tar*, *xz*, *zip*, *bzip*, *gzip*? Зробіть короткий опис кожної команди та виділіть їх основні параметри. Яким чином їх можна встановити.

* Gzip is used to compress and decompress compressed files. It has the following parameters: -l — provides information about the file (how much space the uncompressed file takes up, but how much is compressed), -d — unpack the compressed file. sudo yum install gzip.
* Tar is used for archiving and unzipping files, viewing archive contents without unzipping. Has such parameters as: -c — create an archive, -f ARCHIVE — operate on this archive, -z — compress (or decompress) an archive using gzip, -j — compress (or decompress) an archive using bzip, -t — expand a list of files in the archive, -x — unpack files from the archive, -v — a detailed list of processed files. sudo yum install tar.
* ZIP is used to compress and archive files. It has the following options: -r — add files to the archive, -d — delete a file from the archive. sudo yum install zip.
* Xz is used to compress files. Has such options as: -c — compress the file, -d — unpack the compressed file. sudo yum install xz.
* Bzip is used to compress files. It has the following parameters: -z — compress the file, -d — unpack the compressed file. sudo yum install bzip2.
  1. Наведіть три приклади реалізації архівування та стискання даних різними командами.
* tar -cf alpha\_files.tar alpha\*
* gzip -l alpha\_files.tar.gz
* zip alpha\_files.zip alpha\*
  1. Яке призначення команд cat, less, more, head and tail? Зробіть короткий опис кожної команди та виділіть їх основні параметри. Яким чином їх можна встановити.
* Cat is used to create and display text files, and to merge copies of text files. It has such parameters as: -n — number the lines, file — the name of the file or files to be output. sudo yum install cat.
* Less is used to view large text files page by page. It has the following parameters: file — the name of the file to view, /pattern — search for a text pattern in the file. sudo yum install less.
* More is used to view files, has fewer functions than less. Parameters: file — the name of the file to view. sudo yum install more.
* Head is used to output the first few lines from a file. file — the name of the file from which the first lines should be output, -n — the number of lines to output. sudo yum install head.
* Tail is used to output the last few lines from a file. –f — view changes in the file in real time, -n — number of lines to output. sudo yum install tail.
  1. Поясніть принципи роботи командної оболонки з каналами, потоками та фільтрами.
* Pipes:

Principle of operation: A channel is a mechanism that allows you to transfer the output of one command to the input of another command without the need to save the intermediate result in a file. Channels are created using a vertical bar | and are used to combine several commands into one data processing chain.

* Streams:

How it works: In Linux, there are three main data streams: standard input (stdin), standard output (stdout), and standard error (stderr). The command shell allows you to redirect these input and output streams for commands. For example, you can redirect the output of a command to a file or pass a file as input to a command.

* Filters:

How it works: A filter is a command that processes input data and outputs the changed data to standard output. Filters are often used in conjunction with channels to manipulate data. Examples of filters include grep for searching text, sed for editing text, and awk for processing text data.

* 1. Яке призначення команди grep?

The grep command is a powerful tool for searching and filtering data in Linux and Unix-like operating systems. It enables users to search for text strings in a file or filter output from other commands using pattern strings. The grep command has several essential purposes, including searching for strings by pattern, output filtering, and others.

**Хід роботи**

1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:

1.1. Запустіть віртуальну машину VirtualBox, оберіть CentOS та запустіть її. Виконайте вхід в систему під користувачем: CentOS, пароль для входу: reverse (якщо виконуєте ЛР у 401 ауд.) та запустіть термінал.

1.2. Запустіть віртуальну машину Ubuntu\_PC (якщо виконуєте завдання ЛР через академію netacad)

1.3. Запустіть свою операційну систему сімейства Linux (якщо працюєте на власному ПК та її встановили) та запустіть термінал.

***Готував матеріал студент Дзизиль Д.Є.***

1. Опрацюйте всі приклади команд, що представлені у лабораторних роботах курсу ***NDG Linux Essentials - Lab 9: Archiving and Compression*** та ***Lab 10: Working With Text.***

|  |  |
| --- | --- |
| Назва команди | Її призначення та функціональність |
| mkdir mybackups | Creating a new mybackups directory in the user's home directory |
| tar -cvf mybackups/udev.tar /etc/udev | The tar command is used to combine multiple files into a single file. 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 stands for "verbose", which tells the tar command to demonstrate what it is doing. The -f option is used to specify the name of the tar file. |
| ls mybackups | Lists the files and directories in the "mybackups" directory. |
| tar -tvf mybackups/udev.tar | Used to list the files and directories contained in the `udev.tar' archive located in the "mybackups" directory. |
| Tar -zcvf mybackups/udev.tar.gz /etc/udev | Creates an archive of files and directories from /etc/udev and saves it as a compressed tar.gz archive. |
| ls -lh mybackups | Lists the files and directories in the "mybackups" directory, along with size and permission information. |
| cd mybackups | Changes the current working directory to "mybackups" |
| ls | This command lists the files and directories in a directory. |
| tar -xvf udev.tar.gz | This command uses tar to extract the udev.tar.gz archive, which is located in the current directory ("mybackups"). |
| ls etc | These files are extracted from the dev.tar.gz archive, which is located in the current directory ("mibaccups"). |
| ls etc/udev | This command lists the files and directories that are in the "etc/udev" directory, which also appeared after the archive was unzipped. |
| ls etc/udev/rules.d | This command lists the files and directories that are in the "rules.d" subdirectory, which also appeared after the archive was unzipped. |
| tar -rvf udev.tar /etc/hosts | This command adds (or updates) the /etc/hosts file to the existing udev.tar archive. |
| cp /usr/share/dict/words . | Copies the file /usr/share/dict/words to the current working directory. A dot (.) indicates the current directory as the copy destination. |
| ls -l words | Displays information about the words file, including size and permissions. The words file has just been copied with the word dictionary. |
| gzip words | Compresses the words file using the gzip compression program. After executing this command, you get a compressed words.gz file. |
| ls -l words.gz | Displays information about the compressed words.gz file, including size and permissions. |
| gunzip words.gz | Extracts the compressed words.gz file using gunzip. After executing this command, you get the words file in its standard, uncompressed form. |
| ls -l words | Prints information about the extracted words file, including size and permissions. |
| bzip2 words | Uses the Bzip2 compression program to compress the words file. After executing this command, the words file is compressed and gets the extension .bz2. |
| ls -l words.bz2 | Prints information about the compressed words.bz2 file, including its size and permissions after compression. |
| bunzip2 words.bz2 | Uses the Bzip2 decompressor to decompress the words.bz2 compressed file. After executing this command, the words.bz2 file is unzipped and returned to its normal state as a words file. |
| xz words | Uses the XZ compression program to compress the words file. After executing this command, the words file is compressed and gets a .xz extension. |
| ls -l words.xz | Prints information about the compressed words.xz file, including its size and permissions, before unzipping. |
| unxz words.xz | Uses the XZ decompressor to decompress the compressed words.xz file. After executing this command, the words.xz file is unzipped and returned to its normal state as a words file. |
| zip words.zip words | Creates a compressed words.zip file that contains the contents of the words file. You use the zip utility to create a ZIP archive. A words file is added to this archive. |
| ls -l words.zip | Prints information about the compressed words.zip file, including its size and permissions. |
| zip -r udev.zip /etc/udev | Creates a compressed udev.zip archive that contains the entire contents of the /etc/udev directory and its subdirectories. The -r option indicates recursive compression, that is, the entire contents of the directory and subdirectories are included in the archive. |
| ls -l udev.zip | Prints information about the compressed udev.zip archive, including its size and permissions. |
| unzip -l udev.zip | Displays information about the contents of the udev.zip archive, namely a list of files and directories contained in the archive, without unpacking them. |
| rm -r etc | Deletes the etc directory and all its subdirectories and files. The -r option indicates recursive deletion, meaning all subdirectories and files will also be deleted. |
| unzip udev.zip | Extracts the udev.zip archive and restores the contents you deleted in the previous command. As a result, you will have the etc directory and its subdirectories and the files that were in the udev.zip archive. |
| echo "Hello World" | This command prints the string "Hello World" to standard output (usually the screen). |
| echo "Hello World" > mymessage | This command outputs the string "Hello World" and redirects it to a file called "mymessage". As a result, this file "mymessage" will be created or overwritten, and the contents of this file will be the string "Hello World". |
| cat mymessage | This command prints the contents of the file "mymessage" to standard output. |
| echo "Greetings" > mymessage | This command outputs the string "Greetings" and redirects it to a file called "mymessage". |
| echo "How are you?" >> mymessage | This command adds the line "How are you?" to the contents of the file "mymessage" without overwriting the previous contents. |
| find ~ -name "\*bash\*" | This command searches for all files and directories in the user's home directory that have "bash" in their name. Found objects will be displayed on the screen. |
| find /etc -name hosts | This command looks for a file(s) named "hosts" in the /etc directory. The well-known /etc/hosts file is a file that usually contains information about network addresses and hostnames on the system. The found file will be displayed on the screen. |
| find /etc -name hosts 2> err.txt | This command searches for a file named "hosts" in the /etc directory, and any error messages encountered during the search will be redirected to the err.txt file. Option 2> indicates redirection of the standard error stream (stderr) to the err.txt file. |
| cat err.txt | This command prints the contents of the err.txt file to standard output. |
| find /etc -name hosts > std.out 2> std.err | This command searches for a file named "hosts" in the /etc directory. Search results will be redirected to std.out, and any error messages encountered during the search will be redirected to std.err. The > option indicates redirection of standard output (stdout), and the 2> option redirects the standard error stream (stderr). |
| cat std.err | This command prints the contents of std.err to standard output. |
| cat std.out | This command prints the contents of std.out to standard output. |
| find /etc -name hosts > find.out 2>&1 | This command searches for a file named "hosts" in the /etc directory. The search results (if a "hosts" file is found) will be redirected to the find.out file. The important part is the 2>&1 part, which redirects the standard error stream (stderr) to the standard output (stdout). This means that if any errors occur during the search, they will also be written to the find.out file. |
| cat find.out | This command outputs the contents of the find.out file to standard output. This content may contain the search results (if the "hosts" file is found) and any error messages encountered when running the find command. You can view both search results and error messages in this file. |
| tr a-z A-Z | The command indicates the conversion of all letters from lowercase (a-z) to uppercase (A-Z). |
| tr A-Z a-z > myfile | This command converts all uppercase letters (A-Z) in the content coming from standard input to lowercase letters (a-z) and writes this changed text to a file named "myfile". |
| cat myfile | This command prints the contents of the file "myfile" to standard output. |
| ls -l /etc | more | Uses two commands to display information about files and directories in the /etc directory through the more filter program. |
| cut -d: -f1 /etc/passwd | This command uses the cut command to truncate and output the first field from each line of the /etc/passwd file. |
| cut -d: -f1 /etc/passwd | sort | This command uses the cut command to cut , sort the strings fed to it through the pipe. In this case, usernames will be sorted in ascending alphabetical order and output the first field from each line of the /etc/passwd file. |
| cut -d: -f1 /etc/passwd | sort | more | This command uses the cut command to truncate , sorts the lines fed to it through a pipe, outputs large amounts of text page by page, allowing you to scroll through the contents page by page, and outputs the first field of each line in the /etc/passwd file. |
| cat /etc/passwd | Will display the contents of the /etc/passwd file on the screen. |
| head /etc/passwd | Outputs the first few lines from the /etc/passwd file. |
| tail /etc/passwd | Outputs the last few lines from the /etc/passwd file. Typically, the /etc/passwd file contains information about system users, and the latest entries reflect new users that have been added to the system. |
| head -2 /etc/passwd | Outputs the first two lines from the /etc/passwd file. |
| ls /etc | tail -5 | Lists the last five items (files or directories) in the /etc directory. |
| head -n -20 /etc/passwd | Displays lines 1-7, excluding the last twenty lines. |
| cd /etc | You become active in the /etc directory. |
| grep sshd passwd | Will print the entire line containing the match. |
| grep root passwd | Finds all lines in a file named "passwd" that contain the word "root". |
| grep '^root' passwd | Uses grep to find lines in a file named "passwd" that begin with the word "root" (since the "^" character marks the beginning of a line). |
| grep 'sync' passwd | Uses grep to find lines in a file named "passwd" that contain the word "sync". |
| grep 'sync$' passwd | Uses grep to find lines in a file named "passwd" that end with the word "sync" (since the "$" character marks the end of a line). |
| grep '.y' passwd | Uses grep to find lines in a file named "passwd" that contain any character followed by "y". |
| grep 'sshd|root|operator' passwd | Uses grep to find lines in a file named "passwd" where the matching lines contain one of three words: "sshd", "root", or "operator". |
| grep -E 'sshd|root|operator' passwd | Uses grep with extended regular expression mode (-E) to search for lines in a file named "passwd" that contain one of three words: "sshd", "root", or "operator". |
| egrep 'no(b|n)' passwd | Uses egrep with extended regular expression mode to find lines in the "passwd" file where a word that begins with "no" and can be followed by either "b" or "n" occurs. |
| head passwd | grep '[0-9]' | This command starts by outputting the first few lines from the "passwd" file using the head command. The output strings are then filtered through the grep command using the regular expression [0-9], which looks for any string that contains at least one digit in the "passwd" file. |
| grep -E '[0-9]{3}' passwd | Uses grep with extended regular expression mode to find lines in the "passwd" file where there are sequences of three digits in a row (total of 3 digits). |

\*\*\***Скріншоти** виконання команд в терміналі можна **не представляти**, достатньо **коротко описати команди в таблиці**.

***Готував матеріал студент Чех І.В.***

1. Ознайомтесь з командою tar та за її допомогою виконати у терміналі наступні дії:

* створити файл з розширенням .tar;
* створити файл з розширенням .tar, що складається з декількох файлів і каталогів одночасно;
* перегляду вмісту файлу;
* витягти вміст файлу tar;
* створити архівний файл tar, стиснений за допомогою bzip;
* витягти вміст файлу tar bzip;
* створити архівний tar файл, стисненого за допомогою gzip;
* витягти вміст файлу tar gzip.

***Готував матеріал студент Дзизиль Д.Є.***

1. Як буде відбуватись перенаправлення потоків виведення в bash для наступних дій з командами (позначено як cmd) та файлами (позначено як file):

|  |  |
| --- | --- |
| **Команда** | **Що виконує команда?** |
| cmd 1> file | Output stdout of the cmd command to the file file. |
| cmd > file | Output stdout of the cmd command to the file file. |
| cmd 2> file | Stderr output of the cmd command to the file file. |
| cmd >> file | Adding the stdout of the cmd command to the contents of file. |
| cmd &> file | Output both stdout and stderr of the cmd command to the file file. |
| cmd > file 2>&1 | Output both stdout and stderr of the cmd command to the file file. 2>&1 redirects stderr to stdout. |
| cmd >> file 2>&1 | Adding both stdout and stderr of the cmd command to the contents of the file file (compensation). 2>&1 redirects stderr to stdout. |
| cmd 2>&1 > /dev/null | cmd stdout output to /dev/null (discard). |
| cmd 2> /dev/null | stderr output of the cmd command to /dev/null (discard). |
| cmd1 | cmd2 | Redirection of cmd1 command stdout to cmd2 command stdin. |
| cmd1 2>&1 | cmd2 | Redirection of both stdout and stderr of cmd1 to stdin of cmd2 (pipelining commands). |

***Готував матеріал студент Чех І.В.***

1. Розгляньте наведені нижче приклади та поясніть, що виконують дані команди та який тип перенаправлення потоків вони використовують:

***Готував матеріал студент Чех І.В.*:**

**Контрольні запитання**

***Готував матеріал студент Чех І.В.*:**

Висновок: