Kyiv College of Communications

Cycle Commission of Computer Engineering

**PERFORMANCE REPORT**

**LABORATORY WORK №5**

*in discipline: "Operating systems"*

*Topic: "Linux commands for archiving and compressing data"*

Performed by students

RPZ-93A group

Makarenko Danylo

Nemechkin Maxim

The teacher checked

Povkhlib V.S.

Kyiv 2021

**The goal of the work:**

1. Gaining practical skills in working with the Bash command shell.

2. Familiarity with the basic actions when working with help.

3. Familiarity with the basic actions when working with files and directories.

**Material support of classes**

1. Computer type IBM PC.

2. Windows family of operating systems (Windows 7).

3. Virtual machine - Virtual Box (Oracle).

4. GNU / Linux operating system - CentOS.

5. Cisco Network Academy website netacad.com and its online Linux courses

**Tasks for preliminary preparation.**

**The material was prepared by student Nemechkin M.D.**

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

| echo | prints text, and values of variables |
| --- | --- |
| hostname | displaysnameofcomputer |
| / | represents higher directory in the hierarchy |
| history [last #] | more like lessons than manuals |
| clear | clearsthescreen |
| --help | viewsbasiccommandusage |
| find | locates "newer" files |
| info | more like lessons than manuals |

2. On the basis of the considered material give answers to the following questions:

2.1. What is the directory structure of a Unix-like file system? What is the purpose of basic directories?

Each file in the UNIX system has a unique index. The index contains the information that any process needs to access a file, such as file ownership, file access rights, file size, and the location of file data in the file system. Processes access files using a well-defined set of system calls and identifying the file with a string of characters that act as a compound file name. Each composite name uniquely identifies a file, so the system kernel converts that name to a file index.

Indexes are stored on disk in static form, and the kernel reads them into memory before working with them. Disk indexes contain the following main fields:

- file owner ID. Proprietary rights are divided between the individual owner and the "group", thus helping to determine the range of users who have access to the file. The privileged user has the right to access all files in the system;

- file type. The file can be a regular file (-), a directory (d), a special file corresponding to I / O devices in blocks (b) or characters (c), a symbolic link (l), a socket (s) (s), and abstract channel file (pipe) (p), which organizes the service of requests in the order of receipt, "first come - first served". The parentheses indicate the types of files that the ls command displays when outputting;

- file access rights. The system differentiates file access rights for three classes of users: individual file owner, group owner and other users; each class is assigned certain rights to read, write and execute the file, which are set individually. Because directories as files cannot be executed, permission to execute in this case is interpreted as the right to search the directory by file name;

- calendar information that characterizes the work with the file: the time of the last changes in the file, the time of the last access to the file, the time of the last changes in the index;

- the number of pointers to the file, which means the number of names used when searching for the file in the directory hierarchy;

- file size. The data in the file is addressed by offset in bytes relative to the beginning of the file, starting with an offset of 0, so the file size in bytes is 1 larger than the maximum offset. For example, if a user creates a file and writes only 1 byte of information from an address with an offset of 1000 from the beginning of the file, the file size will be 1001 bytes. Note that the index does not have the compound file name needed to access the file;

- table of addresses on the disk in which the file information is located. Although users think of the information obtained from the file as a logical stream of bytes, the kernel generally places this data in inconsistent disk blocks. Disk blocks that contain file information are listed in the index.

2.2. Explain the concept of FHS. How is this standard used in the context of file systems?

Filesystem Hierarchy Standard, FHS (FHS) is a standard adopted to unify the location of general-purpose files and directories in the UNIX file system. Today, most UNIX-like systems follow these rules to some degree. For example, the default user database is always stored in the / etc / passwd file.

2.3. Define the mounting process. What is the approach to its use in Linux. Give examples.

As I said, Linux has a single root file system that connects all devices and other resources. In fact, something similar happens in Windows, only it's all hidden from the user.

In fact, the mounted partition becomes part of the root file system and the system tries to make working with all partitions, regardless of their file systems, transparent. This means that if you mount a section of RAM or a remote network folder, you can work with it in the file manager as well as with the local disk.

For example, you want to mount a flash drive. You instruct the system to mount it to the / run / media / username / UUID\_flash / folder. The system determines the file system of the device, and then, using the kernel driver, connects it to the specified folder. Then you have to work with that folder as with any other. You don't have to think about anything else. When you decide to remove the flash drive, you need to unmount it.

2.4. List the basic commands for working with files and directories in Linux: create, move, copy, view content, delete.

1. The pwd command

Use the pwd command to find the path to the current working directory (folder) where you are from the root directory. The command returns the absolute (complete) path, which is actually the path of all directories, starting with a slash (/). An example of an absolute path is / home / username.

2. The cd command

Use the cd command to navigate Linux files and directories. It requires the full path or directory name, depending on the current working directory you are in.

Let's say you're in / home / username / Documents and want to go to Photos, the Documents subdirectory. To do this, enter the following command: cd Photos.

Another scenario is if you want to move to a new directory, such as / home / username / Movies. In this case, you need to enter cd and then the absolute path to the directory: cd / home / username / Movies.

There are also several shortcodes for faster navigation:

cd .. (with two dots) to move up one directory

cd to go directly to your home folder

cd- (with a hyphen) to go to the previous directory

It is also worth noting that the Linux shell is case sensitive. It is important to enter directory names.

3. Team ls

The ls command is used to view the contents of a directory. By default, this command displays the contents of your current working directory.

If you want to view the contents of other directories, enter ls and then the path to the directory. For example, type ls / home / username / Documents to view the contents of Documents.

Options for using the Linux ls command:

ls -R will also list all files in subdirectories

ls -a will show hidden files

ls -al displays a list of files and directories with detailed information such as resolution, size, owner, etc.

4. The cat command

cat (short for concatenate) is one of the most commonly used commands in Linux. Used to display the contents of a file on the command line (sdout). To run this command, enter cat, then the file name and file extension. For example: cat file.txt.

Here are other ways to use the Linux cat command:

cat> filename creates a new file

cat filename1 filename2> filename3 merges two files (1 and 2) and saves their contents in a new file (3)

To convert a file to uppercase or lowercase, cat filename | tr a-z A-Z> output.txt

5. Team cp

Use the cp command to copy files from the current directory to another directory. For example, the cp scenery.jpg / home / username / Pictures command will create a copy of scenery.jpg (from your current directory) to the Pictures directory.

6. Team mv

The main purpose of the mv command is to move files, although you can also use them to rename them.

The mv arguments are similar to the cp arguments. You need to enter mv, file name and destination directory. For example: mv file.txt / home / username / Documents.

The Linux command for renaming files will look like this: mv starojeimia.ext novojeimia.ext.

7. The mkdir command

Use the mkdir command to create a new directory. If you enter mkdir Music, the command will create a directory named Music.

Additional mkdir commands:

To create a new directory inside another directory, use this basic command mkdir Music / Newfile;

Use the p (parents) option to create a directory between two existing directories. For example, mkdir -p Music / 2020 / Newfile will create a new "2020" file.

8. The rmdir command

If you want to delete a directory, use the rmdir command. However, rmdir only allows you to delete empty directories.

9. The rm command

The rm command is used to delete files. If you want to delete the directory with all its contents, use rm with the -r option as an alternative to rmdir.

Note: Be very careful with this command and always check which directory you are in. It deletes everything and cannot be undone.

10. Touch command

The touch command allows you to create a new empty file via the Linux command line. As an example, enter touch /home/username/Documents/Web.html to create an HTML file called Web in the Documents directory.

11. The locate command

Use this command to find the file you want. It works as a search command in Windows. Moreover, the -i argument will make the command case-insensitive, so you can search for files even if you can't remember their exact names.

Use a star (\*) to find a file that contains two or more words. For example, the locate -i school \* note command will search for any file that contains the words "school" and "note", whether they are uppercase or lowercase.

12. Find command

Like the locate command, find also searches for files and directories. The difference is that the find command is used to search for files in the current directory.

For example, the find / home / -name notes.txt command will search for a file named notes.txt in the home directory and its subdirectories.

Other uses for the Linux find command:

Use find to search for files in the current directory. -name notes.txt

To search

**Progress**

**The material was prepared by student Makarenko D.A.**

Task 3. Create a table of commands studied in paragraph 2 of the work

| **Command name** | **Its purpose and functionality** |
| --- | --- |
| date | displays today's date |
| man | access the team manual page |
| less | displays a page of the manual |
| apropos | how to view the keyword guide page. |
| whatis | provides an explanation for the entered value |
| info | provides information about the team |
| locate | easy way to search for a file |
| find | performs a search on a live file system, not a static database |
| whereis | Indicates where the commands or manual pages are located |
| pwd | Displays the path to the current directory |
| cd | Changes the current directory |
| echo | Allows you to display the required information |
| ls | provides file and directory names in the specified (or current) directory |
| ascii | Allows you to see the ASCII table |
| rm | used to delete the file |
| mkdir | Creates a new directory |
| touch | Creates an empty file |
| mv | Cuts and pastes the specified file into the specified directory |

Test questions

**The material was prepared by student Makarenko D.A.**

1. List the main features of the cat command.

You can transfer several files to the utility, and then their contents will be displayed in turn, without delimiters. Options allow you to greatly modify the conclusion and do exactly what you need. Consider the main options:

-b - number only non-empty lines;

-E - show the character $ at the end of each line;

-n - number all lines;

-s - delete duplicate blank lines;

-T - display tabs as ^ I;

-h - display help;

-v - version of the utility.

2. How can I add information to a file in the terminal? What will be the difference if it is necessary not to add, but to rewrite its contents?

With the help of special. commands you can add information to the file. If overwritten, the contents of the file will take up a different memory space.

3. How to copy and delete an existing directory? Will there be a difference in the commands if the directory is not empty?

File commands

ls - list of files and directories

ls - al - formatted list with hidden directories and files

cd dir - change the directory to dir

cd - change to home directory

pwd - show current directory

mkdir dir - create a dir directory

rm file - delete file

rm - r dir - delete directory dir

rm - f file - delete the forced file

rm - rf dir - delete forced directory dir \*

cp file1 file2 - copy file1 to file2

cp - r dir1 dir2 - copy dir1 to dir2; create a dir2 directory if it does not exist

mv file1 file2 - rename or move file1 to file2. If file2 is an existing directory, move file1 to file2

ln - s file link - create a symbolic link to the file

touch file - create a file

cat> file - send standard input to file

more file - output the contents of the file

head file - print the first 10 lines of the file

tail file - print the last 10 lines of the file

tail - f file - display the contents of the file as it grows, starting with the last 10 lines

FOR TEAMS, ONLY THE ESSENCE OF THE TEAM AND WHETHER THE FILE CONTENT IS IMPORTANT.

4. In which of the following examples is the file moving? rename it? both actions at the same time?

**- mv /work/tech/comp.png. /Desktop**

**- mv /work/tech/comp.png. /work/tech/my\_car.png**

**- mv /work/tech/comp.png. /Desktop/computer.png**

**Conclusion**

**The material was prepared by student Nemechkin M.D.**

Gained practical skills in working with the Bash command shell. Get acquainted with the basic actions when working with help. Get acquainted with the basic actions when working with files and directories.