“Kyiv Professional College of Communications”

Computer Engineering Cycle Commission

**PERFORMANCE REPORT**

**LABORATORY WORK №6**

in the discipline: "Operating Systems"

**Topic: "Working with text in CLI-mode Linux and creating scripting scripts"**

Performed by students

RPZ-93B group

Team:

Бушовська О.В,

Білобровенко О.С.,

Скворцов Д.Є.

The teacher checked:

Сушанова В.С.

Київ 2022

The purpose of the work:

Gaining practical skills in working with the Bash shell.

Familiarity with the basic actions when working with text in the terminal.

Familiarity with basic actions when working with scripting scripts.

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. *Готувала матеріал студентка Бушовська Ольга***

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.

|  |  |
| --- | --- |
| Англійська | Українська |
| Input/Output (I/O) redirection | Перенаправлення введення/виводу (I/O). |
| specified pattern | зазначений візерунок |
| particular phrase | конкретна фраза |
| means ignore case distinctions | означає ігнорувати регістр |
| specifies the delimter | визначає роздільник |

***Готувала матеріал студентка Білобровенко Олександра***

1. On the basis of the considered material give answers to the following questions:
   1. What is the purpose of *cat, less, more, head and tail commands* ? Make a brief description of each team and highlight their main parameters. How to install them.

The cat command is one of the most commonly used Linux commands. It is often used by experienced users while working with the terminal. With this command, you can very easily view the contents of a small file, merge several files, and much more.

Although the utility is very simple and does only one task in the best Unix style, it will be very useful. And knowing about its additional features will definitely not hurt you. This article will cover the cat linux command, its syntax, options, and capabilities.

The name of the command is an abbreviation for the word catate. In fact, the task of the cat command is very simple - it reads data from a file or standard input and displays it on the screen. That's all the utility does. But you can do a lot with its options and output redirection operators.

$ cat options file1 file2 ...

We have already written about the utility and the more command, which is designed for page-by-page viewing of large text files . And today we’ll talk about a more functional less command - it allows you to rewind the text not only forward, but also back, search in both directions, go immediately to the end or to the beginning of the file.

The peculiarity of less is that the command does not read the text completely, but loads it in small fragments.

The less command entry in the terminal looks like this:

file option command

A list of all options and internal commands can be viewed in the terminal by running the command

$ man less

The more utility is designed to page through files in the Linux terminal. It owes its name to the inscription more (in the Russian version - further), appearing at the bottom of each page.

The more linux command is one of the most primitive commands for working with text. Its closest relative, the less command, has a much larger set of options and additional features. Let's understand its syntax and usage examples.

In the Linux terminal emulator, the command is written like this:

$ more options file

The head command prints the leading lines (10 by default) from one or more documents. It can also show data that another utility sends to the output.

Today we will explain how this command works for Linux and show how the most popular head options are used in practice.

The syntax for the head command is:

$ head options file

Here:

Options is a setting that allows you to customize how the command works so that the result matches the specific needs of the user.

File is the name of the document (or the names of the documents, if there are more than one). If this value is not set, or if it is replaced by a "-" sign, the command will take data from the standard output.

Everyone knows about the cat command, which is used to view the contents of files. But in some cases you don't need to look at the whole file, sometimes it's enough to look only at the end of the file. For example, when you want to see the contents of a log file, you don't need what it starts with, the latest error messages will suffice.

To do this, you can use the tail command, it allows you to output a specified number of lines from the end of the file, as well as output new lines in interactive mode. This article will cover the Linux tail command.

Before we look at tail linux examples, let's look at its syntax and options. And the syntax is very simple:

$ tail options file

* 1. Explain how the shell works with channels, streams, and filters.

Possibility of shells - redirection of input-output streams. As a rule, most commands (utilities) take information from the keyboard, or from a file, if specified as a parameter, and displays the results. However, they actually work with so-called standard I / O streams that are associated with certain files. A UNIX file is treated as a byte stream. Because UNIX devices are treated as files, and I / O operations are read and written to the appropriate files, it allows you to easily translate input and output streams from files to devices or vice versa.

The shell allows you to redirect streams to a specified file. The <<symbol redirects the incoming stream. This character is followed by the name of the file or device from which the input stream will be taken.

Shell tools can be used to organize a separate redirection of threads due to clever programming techniques.

I / O redirection is widely used in two cases. The first is to run utilities in the background. To ensure that their work does not interfere with the user's work with the terminal, I / O streams should be redirected so that they work only with files. The second is the use of special commands-utilities, which are designed specifically to receive certain information from one file, process it, and write the result to another file. Such utilities are called filters. The cat utility, which has been discussed above with stream redirection, is a simpler filter. It does not process information, can only link several files into one. Other useful filters: cut, grep, sort.

It is possible to redirect the output stream of one utility directly to the input stream of another, without the use of temporary files. These are the so-called pipelines. In the UNIX system, all utilities connected to the pipeline run in parallel and process information as it arrives. The pipeline is formed with the symbol ‘|’ as follows:

util1 | util2 | util3

When creating a pipeline, it is not possible to redirect input and output streams separately in intermediate stages - this will either be perceived as a syntactic error, or the result may be unpredictable.

Example of a conveyor:

ps –al | grep root | the sea

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

Grep is a powerful search utility built into every popular GNU/Linux OS distribution . However, even in the unlikely event that this application is not included in your build, it can be installed without too much difficulty. And the built-in package manager is capable of this ( apt-get on Debian and Ubuntu distributions, yum on RHEL, Fedora and CentOS distributions).

GREP This is a command to help us find the text in the file we specify. Its name comes from g / re / p, a command that works for something similar in a Unix / Linux text editor. Like many other commands, GREP has many options available, which we will add in the form of letters and each of them will perform a different task. By combining these parameters, we will be able to perform complex searches in one or more files. Here we will show you everything you need to know.

1. Learn Cisco Academy Online Course Materials:

* NDG Linux Essentials (Chapter 10-12 all Topics)

***Готував матеріал студент Скворцов Дмитро***

1. On the basis of the considered material in online courses give answers to the following questions:

4.1 Describe the concept of scripting in the command shell.

The custom shell bash can work in two modes - interactive and, accordingly, non-interactive. You can open the shell in Ubuntu by pressing **Ctrl + Alt + F1**, the usual graphical interface will disappear, and you will see one of the seven virtual terminals available in the Ubuntu distribution. If the shell issues an invitation, then you are working in interactive mode:

**user @ host: ~ $**

Here you can enter a variety of unix-commands (such as: ls, grep, cd, mkdir, rm) and see the result of their execution. This shell is called interactive because it interacts with the user directly. Desktop environments (graphical interface), in the family of Debian systems (including Ubuntu), are usually placed in the seventh virtual terminal, so to return to the usual desktop environment, type **Ctrl + Alt + F7.**

In non-interactive mode, the shell reads commands from a file and executes them sequentially. When the interpreter reaches the end of the file, the shell work will end automatically. You can run the shell in non-interactive mode with the following commands:

**sh script**

**bash script**

Where a script is a path to a file that contains commands to execute. Such a file is a plain text document that can be created using any text document. However, you can simplify the script call just by making it executable. To do this, grant the appropriate access rights to this file using the chmod command:

**chmod + x script**

In addition, the first line of the script must specify which shell should run this script. This can be done by placing the appropriate #! / Bin / sh (for the sh shell) or #! / Bin / bash (for bash) at the beginning. After that, the file can be called for execution by contacting it in the terminal:

**./script**

4.2 How do I create and edit scripts, what do I need to do to run the script?

We open terminal with Ctrl+Alt+T, start key and write terminal or with the shell icon I always have in the Ubuntu launcher, come on, on the left sidebar.

To run it, change to the directory where the file is located. imagine we have ok.sh file in /scripts/ folder

We enter scripts using (you need to follow the path where you have it)

cd scripts

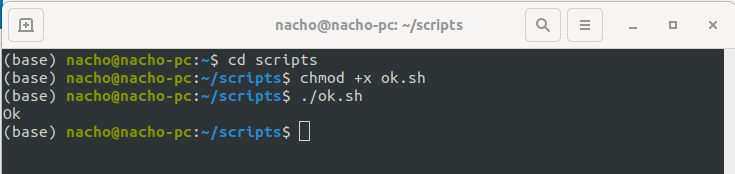
If this is the first time we run it, we must grant file permissions

sudo chmod +x ok.sh

And then we run it

./approx.sh

And here is the sequence



Picture 4.2.1

In our case, “OK” appears because we entered what this script does.

The most important thing and what people make the most mistakes is the path, the path, the lack of access to the folder where the executable is located.

1. Take the NDG Linux Essentials course on the following topics:

* Chapter 10 Exam
* Chapter 11 Exam

1. Prepare the initial version of the report in electronic form:

* Title page, topic and purpose of the work
* Glossary of terms
* Answers to items 2.1-2.3 and items 4.1-4.2 of the tasks for preliminary training

**Progress.**

* 1. Initial work in CLI mode in Linux family:
  2. Start the VirtualBox virtual machine, select CentOS, and start it. Log in under the user: CentOS, login password: reverse ***(if you are performing LR in room 401)*** and lower the terminal.
  3. Start the Ubuntu\_PC virtual machine ***(if you are performing LR tasks through the netacad academy)***
  4. Start your Linux operating system ***(if you are running your own PC and have it installed)*** and start the terminal.
  5. Study all the examples of commands presented in the laboratory work of the ***NDG Linux Essentials course:***
* ***Lab 10: Working With Text***
* ***Lab 11: Basic Scripting***
* ***Lab 12: Understanding Computer Hardware***

***Готувала матеріал студентка Білобровенко Олександра***

* 1. Create a table of commands studied in paragraph 2 of the work in the following form:

|  |  |
| --- | --- |
| Command name | Its purpose and functionality |
| echo | outputs a line of text to the terminal |
| cat | allows you to create, merge, and also output the contents of files on the command line or in another file |
| find | command to search files and directories based on special conditions |
| tr | used to replace or remove characters |
| ls | to display information about files and directories |
| more | designed to page through files in the Linux terminal |
| cut | to extract fields from files |
| sort | to output text strings in a specific order |
| less | allows you to rewind the text not only forward, but also backward, search in both directions, jump immediately to the end or to the beginning of the file. |
| head | outputs leading lines from one or more documents |
| tail | allows you to output a specified number of lines from the end of the file |
| grep | to search for a specific string of characters in the data stream |
| fgrep | used to match literal characters, ignoring the special meaning of regular expression characters |
| egrep | use of extended regular expressions |
| vi | it is a powerful text editor with a certain learning curve |
| nano | is the standard console editor for many Linux distributions. |

* 1. Create scripting scripts with text messages for the user:
* the script should display greetings to the current user indicating the current date and information about the current system;
* the script should display information about the hardware configuration of the current system (use the commands discussed in Lab 12).

**Test questions *Готувала матеріал студентка Бушовська Ольга***

1. How can I redirect streams in a shell? Demonstrate examples of when I / O / error messages are redirected.

I / O redirection (I / O) allows you to transfer command line information to different streams.

STDIN

Standard input, or STDIN, is information that is usually entered by the user using the keyboard.

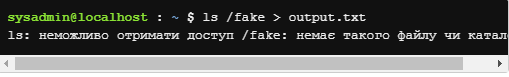
**STDOUT**

Standard output, or STDOUT, is the usual output of commands. When a command works correctly (without errors), the output it creates is called STDOUT. STDOUT can be directed to files. To begin, look at the result of the following echo command, which is displayed on the screen:



**STDERR**

A standard error, or STDERR, is an error message generated by commands.

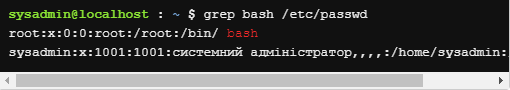


I / O redirection allows the user to redirect STDIN so that the data comes from the file, and STDOUT / STDERR so that the output data comes to the file. Redirection is performed using <> arrow characters.

1. What are the filters commands used for? Give some application tasks where their use is necessary.

The grep command can be used to filter lines in a file or output another command that matches a given pattern. This template can be as simple as the exact text you want to find, or it can be much expanded by using regular expressions.

For example, to find all users who can log in using the BASH shell, grep can use the command to filter lines from the / etc / passwd file for lines that contain a bash template:



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

Redirection to / dev / null is most often used to suppress standard output (output stream) and / or display error messages (diagnostic flow) of the program redirecting them to / dev / null, such suppression is most often used in shell scripts to suppress unwanted output on the console.

***Готувала матеріал студентка Білобровенко Олександра***

**Conclusion:** I gained practical skills in working with the Bash command shell, got acquainted with the basic actions when working with text in the terminal and learned the basic actions when working with scripting scripts.