“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №6**

з дисципліни: «Операційні системи»

**Тема: «Робота з текстом в CLI-режимі Linux та створення скриптових сценаріїв»**

Виконали

студенти

групи РПЗ-93а

Команда: Процевич В.І, Христинченко І.А, та Куліда Д.М

Перевірив викладач

Сушанова В.С.

Київ 2022

**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 text in the terminal.

3. Familiarity with the 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.**

**The material was prepared by student Христинченко І.А**

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

| formatting features | особливості форматування |
| --- | --- |
| Concatenate | Конкатенувати |
| redirecting file | файл перенаправлення |
| Capability | Здатність |
| Respectively | Відповідно |
| initial command | початкова команда |
| specified pattern | зазначений візерунок |
| Recursively | Рекурсивно |
| delimter | роздільник |

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

**Cat** – it allows you to create, merge, and output the contents of files on the command line or in another file.

The utility does not require additional installations, as it is pre-installed in the coreutils package on any Debian or Red Hat based system.

With the cat command, you can quickly create a file and put text into it. To do this, use the > redirect operator to redirect text to a file.

**Less**- a command line utility that displays the contents of a file or the output of a command, one page at a time. This is similar to more but has more advanced features and allows you to move forward and backward through the file. The less command is mainly used to open large files.

The utility supports a large number of parameters, and almost all of these parameters are highly specialized and are unlikely to be useful to most users, and the most popular of them are the -N parameter, which allows you to display line numbers, which is very convenient when reading program code, the -X parameter, which allows you not to clear the contents of the terminal after the utility terminates, as well as the -I option, which allows you to search in the text in a case-insensitive manner.

**More**- 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.

**Head**- allows you to use the utility of the same name to display the first few lines from a text file or the output of another utility. If you're running a command that outputs a lot of data and you're only interested in the first few lines of output, this command is exactly what you need.

The head utility can print either the leading lines of the output of another utility (in which case no filenames are specified) or the leading lines of one or more text files with the specified names. In the event that the initial lines of several files are displayed, their names are mentioned in the output. The utility supports a number of useful options, namely, the -n option, which allows you to set the number of output lines (10 lines by default), the -c option, which allows you to set the number of bytes to display, the -q option, which allows you not to add information about file names to the output with their leading lines, the -v option to always display information about file names, and the -z option to print line terminators instead of newlines (this is useful for using the utility from other utilities).

**Tail**- allow the user to read file commands from the end. Also, with the help of this utility, you can track new information updated in real time in a particular file. This

makes it very easy to check the latest entries in the system. Also, the tail command in Linux allows you to view multiple files at the same time.

Used to output the Nth number of last lines of a file. It usually displays the last 10 lines of the file on standard output. If we run it on a single filename, the data from each file is processed by filename with a header.

**The material was prepared by student Процевич В.І**

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

Channels  
Multithreading is a natural extension of multitasking, just as virtual machines that allow you to run multiple operating systems on a single computer are a logical extension of the concept of resource sharing. In an informal but simple definition, a thread is the execution of a sequence of machine instructions. A multi-threaded application has multiple threads running at the same time. Some authors avoid the term "thread" and use the term "thread" instead, probably to avoid confusion between program threads and I/O threads. To denote the sequential execution of a chain of instructions, I personally prefer the term "thread", which I will use. I hope that readers of Linux Format will not get confused in contexts and, having met the word thread, they will always understand whether it is about program threads, input output streams, or stormy flood streams. If there are several threads in a process, the application (process) becomes multi-threaded. In the Linux world, things look different. In Linux, every thread is a process, and in order to create a new thread, you need to create a new process. What, then, is the advantage of Linux multithreading over multiprocessing? Multithreaded Linux applications use a special type of process to create additional threads. These processes are normal child processes of the main process, but they share address space, file descriptors, and signal handlers with the main process. To designate processes of this type, a special term is used - light processes (lightweight processes). The adjective "easy" in the name of the processes-threads is quite justified. Since these processes do not need to create their own copy of the address space (and other resources) of their parent process, creating a new lightweight process is much less expensive than creating a full-fledged child process. Since Linux threads are actually processes, in the Linux world one cannot say that one process contains multiple threads. If you say this, you will immediately be suspected of an enemy infiltrator!

Streams

You can think of a pipe as a small ring buffer in the operating system kernel. From the point of view of processes, the pipe looks like a pair of open file descriptors - one for reading and one for writing (more is possible, but inconvenient). We can write to the pipe as long as there is space in the buffer, if the space in the buffer runs out, the process will be blocked on writing. We can read from the channel while there is data in the buffer, if there is no data, the process will be blocked on reading. If you close the handle responsible for writing, then an attempt to read will show the end of the file. If the handle responsible for reading is closed, then an attempt to write will result in the delivery of a SIGPIPE signal and an EPIPE error.

When using a pipe in shell programming read/write locks ensure that the speed of execution of two programs is synchronized and that they end at the same time.

There is no concept of read/write position for channels, so writing is always done to the tail of the buffer, and reading from the head. For the i386 architecture, the size of the buffer associated with the channel is set to a multiple of the page size (4096 bytes). On Linux, versions before 2.6.11 used one page (4 KB), after - 16 pages (65 KB), with the ability to change via fcntl. POSIX defines a value for PIPE\_BUF that specifies the maximum size of an atomic record. On Linux PIPE\_BUF is 4096 bytes.

Filters

One of the founding principles of Linux is that each element should serve only one function and that all elements can be easily combined with each other. These elements are building blocks that can be collected and thus build whatever we want. In this and the following articles, we will talk about specific building blocks. The first one is filters.

Filters are commands that transform the passed text data in a certain way. With the help of filters, we can take the raw data that is stored in files and manage it - bring it into a form convenient for us.

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

**grep** is a command line interface utility that finds and outputs strings that match a given regular expression. The name of the utility is a sequence of commands to search for regular expressions in the editor ed - g / re / p. This sequence of commands can be described by the English phrase "search globally for lines matching the regular expression, and print them".

It was originally created for the UNIX operating system.

There are modifications of grep:

egrep (with extended regular expression processing),

fgrep (which interprets the characters $ \* [] ^ | () \ literally),

rgrep (with recursive search enabled).

*Syntax*

grep "regular expression" "file"

or

cat "file" | grep "regular expression"

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

*3.1 Describe the concept of scripting in the command shell.*

A shell script is a list of commands in a computer program that is run by the Unix shell which is a command line interpreter. A shell script usually has comments that describe the steps. The different operations performed by shell scripts are program execution, file manipulation and text printing. A wrapper is also a kind of shell script that creates the program environment, runs the program etc.

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

Scripts are created by combining individual blocks: either sequentially or by placing the block in a specific location of another block (control structure, functions, etc.). One sprite can have several scripts that run independently of the user's action (by pressing a key or mouse button), timer, or receiving a message from another sprite. The script consists of stacks.

Stack is a set of multi-colored graphics blocks connected in series within a single event. A block is a minimal piece of a program in Scratch: a variable, an operator, a function, or a control structure.