"Kyiv Professional College of Communication"

Cycle Commission of Computer Engineering

REPORT ON THE IMPLEMENTATION

LABORATORY WORK №2

in the discipline: "Operating systems"

Topic: "Familiarity with the interface and capabilities of Linux"

Performed by a student

of the group BICS-13

Los Zlata Volodymyrivna

Checked by the teacher

Sushanova V.S.

Kyiv 2024

**Objectives:**

1. Familiarity with the interfaces of the Linux OS.

2. Acquiring practical skills in working in Linux and mobile OS environments - their graphical shell, logging in and out of the system, familiarization with the structure of the desktop, learning the basic actions and settings when working in the system.

**Material support for classes:**

1. IBM PC type computer.

2. OS of the Windows family (Windows 7).

3. Virtual machine - Virtual Box (Oracle).

4. GNU/Linux operating system - CentOS.

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

**Tasks for preliminary preparation.**

**1. Read the brief theoretical information for the lab and make a small glossary of basic English terms on the purpose of commands and their parameters.**

**A command-line interface (CLI) -** is a type of text-based user interface between a computer and a user, in which the computer can be instructed only by entering text strings (commands). It is also known as a console.

**Application server -** is a server that runs some application programs. The term also refers to the software installed on such a server and providing the execution of application software.

**A desktop application -** is a program that requires a desktop computer OS to run. It is installed into the system through a special installer and uses computer resources to operate.

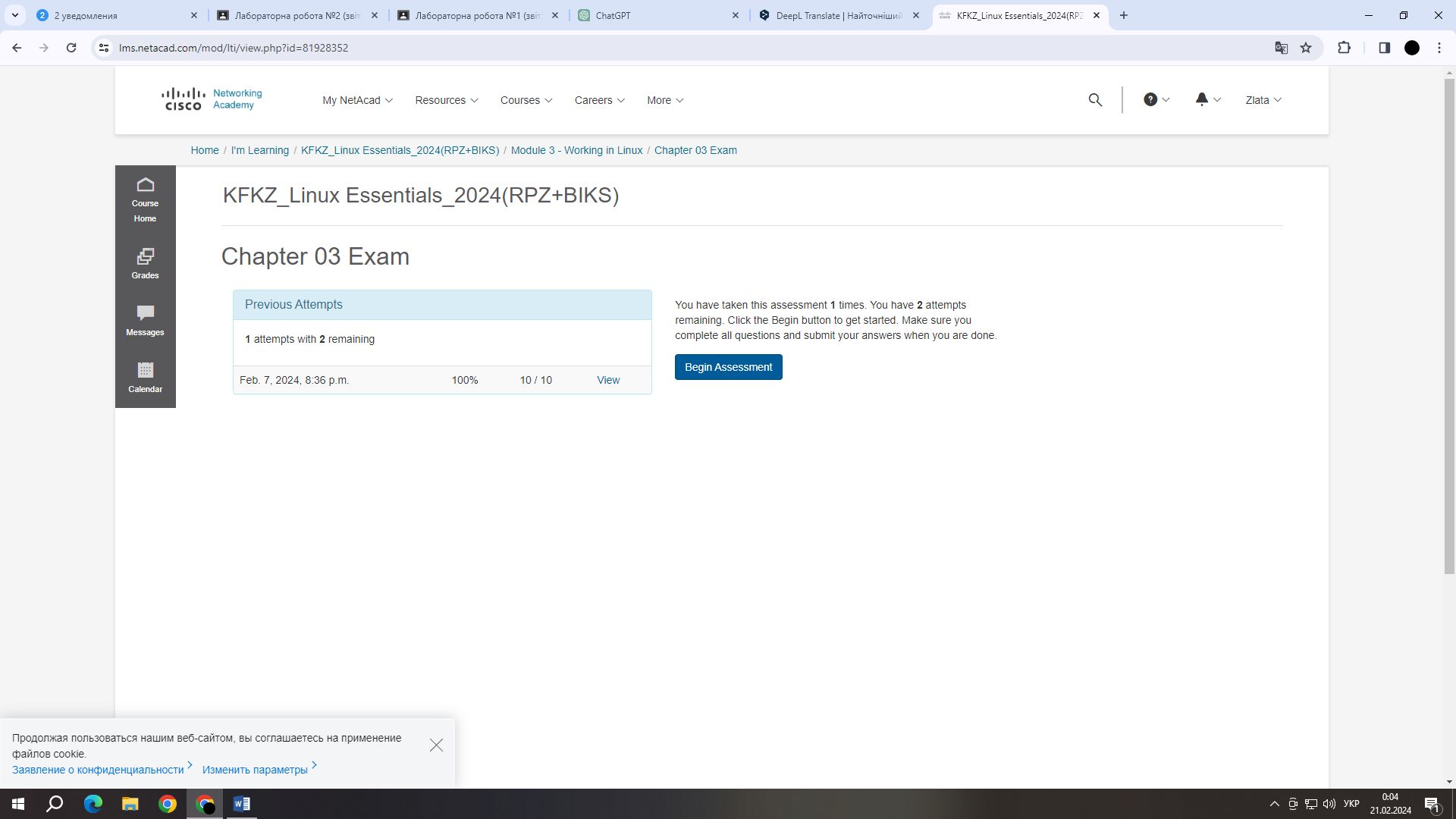
**2. Take the NDG Linux Essentials online course from Cisco Academy:**

- Chapter 3 - Working in Linux

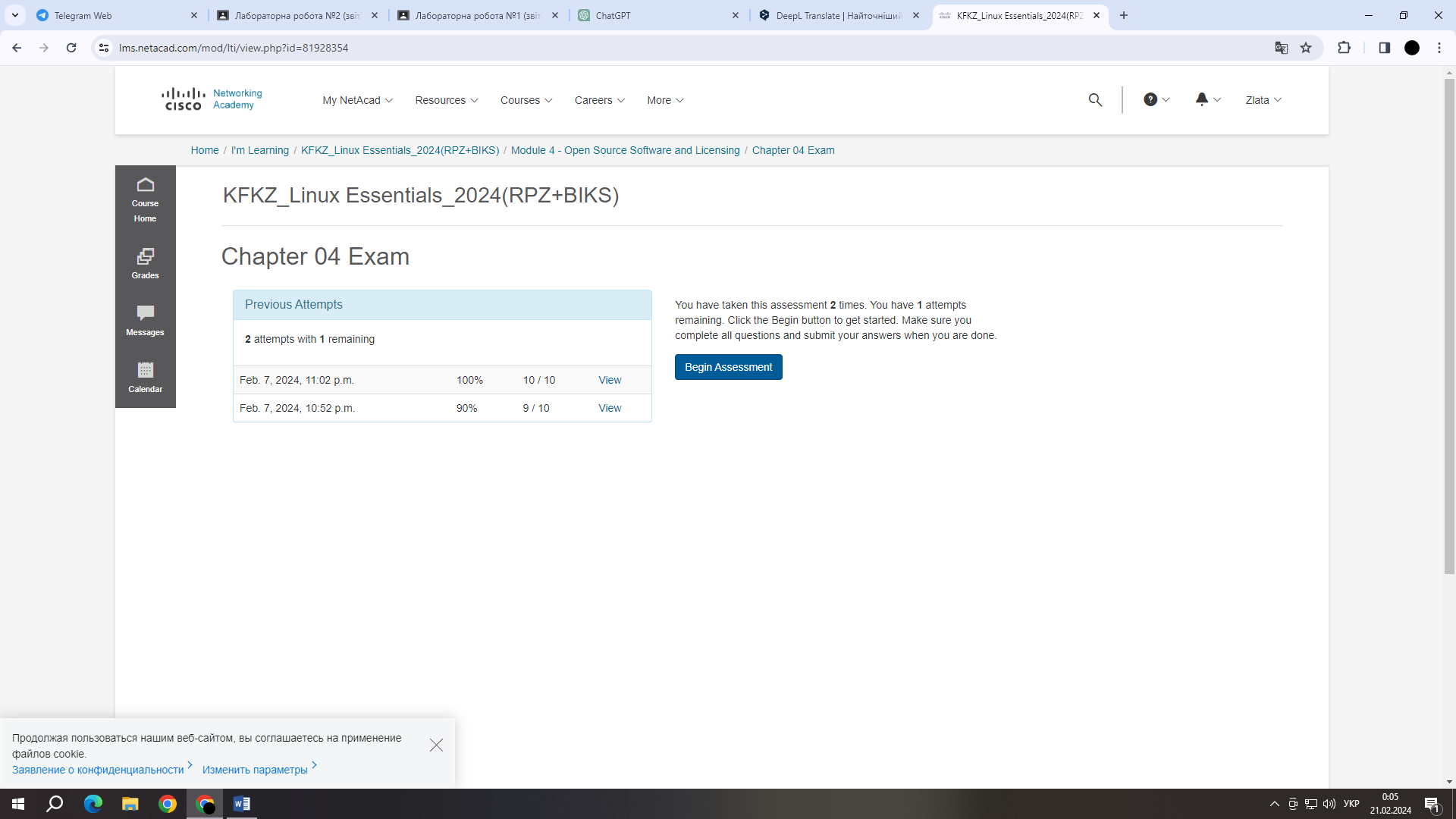
- Chapter 4 - Open Source Software and Licensing

**3. Take the NDG Linux Essentials exam on the following topics:**

- Chapter 03 Exam



- Chapter 04 Exam



**4. Define the following terms:**

- **CLI mode** - is a type of text-based user interface between a computer and a user, in which the computer can be instructed only by entering text strings (commands). Also known as a console.

- **A GUI terminal** - is a computer terminal or program that uses graphical symbols and images to interact with the user. A GUI allows you to use a mouse, keyboard, and other visual elements to enter data and control your computer.

- **A virtual terminal** - is a software terminal emulator that allows a user to interact with an operating system or other programs through a text-based command line interface. Virtual terminals are commonly used in UNIX and Linux operating systems.

**5. Prepare an initial version of the report in electronic form:**

- Cover sheet, topic and purpose of the work

- Glossary of terms

- Answers to p.5 and p.6 of the preliminary preparation tasks

**Work progress**

**1. Working in graphical mode in Linux operating systems (working with Internet sources):**

**1.1.** Choose a graphical shell for the Linux family of operating systems that you want to consider. Consider the structure of the user workspace and describe its main components (Gnome shell):

- **Applications tab** - this tab contains a list of installed programs and applications on the computer. The user can search using filters or manually browse the available programs alphabetically or by category. Usually, icons for launching programs are located here.

- **Places tab** - in this tab, the user has access to various sections of the file system and other resources. For example, you can quickly access your home folder, file storage, network shares, and other frequently used locations.

- **System menu** - The System menu contains system settings and other useful options. Here you can find access to system settings, user account settings, the ability to shut down or restart your computer, and other system options.

- **The Activities overview** is a GNOME element that provides a view of all open windows and workspaces, and allows you to use a search tool to quickly find programs or activities. Usually, it is activated by pressing the "Super" key (with a partial image of the GNOME logo) or by clicking the mouse in the upper left of the screen. Activities overview allows the user to efficiently manage open applications and use workspaces.

**1.2.** Running applications. Explore the different ways to launch apps:

- *Launch programs from the Quick Launcher* - click the program icon in the Quick Launcher.

- *Launch programs by searching the menu* - enter the name of the program in the search field and select a result.

- *Launch programs through the launcher widget* - use the widget to launch or switch between open programs

- *Launch programs through the global menu* - access program options at the top of the panel instead of in the program window itself

**1.3.** Log out and shut down Linux. How to perform the following actions in the graphical interface:

- Change the user to root: enter the terminal and use the command 'su' or 'sudo -i'.

- Reboot the system: use the options in the shutdown menu or the 'reboot' command in the terminal.

- Shut down the system: select the shutdown option from the menu or enter the command 'shutdown now' in the terminal.

**2. Work in a mobile OS environment.**

**2.1.** Describe the main menu of your mobile OS, what kind of graphical interface does it use?

iOS uses a graphical interface with large icons located on the Home screen. The main menu, known as "Springboard," contains application icons and can be organized into different screens. By default, the screens are organized from left to right, but the user can create folders to group applications by topic.

**2.2.** Describe the settings menu for the components of a mobile phone.

The Settings menu in iOS is divided into different sections where the user can customize system, app, security, network, and other settings. Some of the main sections include:

* **General:** Configure basic settings and features.
* **Display & Brightness:** Adjust the screen and brightness settings.
* **Sounds & Haptics:** Control sounds and vibrations.
* **Privacy:** Control how applications access personal information.
* **Battery:** Monitor battery health and power consumption.

**2.3.** Use keyboard shortcuts to perform special actions.

**Open Multitasking (view open apps):**

- On an iPhone without a Home button: Swipe up from the bottom of the screen and stop your finger in the middle of the screen.

- On iPhone with Home button: Double-click the Home button.

**Screenshot:**

- Press and hold the Power button (side or top) and press the Volume Up button (or "+" on volume) at the same time.

**Call Siri:**

- Press and hold the Power button (side or top) or say "Hey Siri" if this feature is activated.

**2.4.** Logging in and shutting down the device. Features of the battery power settings.

**Log in to the system:**

On iOS devices, Face ID authentication is typically used, along with a passcode or PIN. This is what it might look like:

**Face ID (facial recognition):**

- Hold the device up to your face.

- If your face is recognized, your device unlocks.

**Enter a passcode or PIN:**

- Enter a numeric code or passcode, if necessary.

**Shut down the device:**

Typically, you can shut down your iOS device using the Sleep/Wake button (on the side or top, depending on the model). Hold down the button until the slider to turn off the device appears. Slide the slider to confirm the shutdown.

**Set the battery power:**

In the Battery/Battery section, the user can view the battery status, set the power saving mode, and view the power usage statistics by application.

**Control questions**

*1. Provide examples of Linux server applications for a database server, mailing servers, and file sharing servers.*

**1. Database Servers:**

- **MySQL:** An open source relational database management system (RDBMS) that provides efficient data storage and management.

- **PostgreSQL:** Another powerful relational database management system with an emphasis on extensibility and advanced features.

- **MongoDB:** A document-oriented database that uses JSON-like documents and provides flexibility and performance.

**2. Message distribution servers:**

**- Postfix:** A high-performance mail transfer server that is widely used for sending and receiving email.

**- Exim:** Another popular MTA (Mail Transfer Agent) with extensive configuration options and flexible settings.

**- Sendmail:** A classic MTA that runs on many Unix-like systems and is used to process mail messages.

**3. File Sharing:**

**- vsftpd (Very Secure FTP Daemon):** A lightweight and fast FTP server that strives to be secure and efficient.

**- ProFTPD:** Another powerful FTP server with many customization options and support for many protocols.

**- Samba:** Provides the ability to share files between Windows systems and Linux servers using SMB/CIFS protocols.

*2. Compare the Bourne, C, Bourne Again (Bash), the tcsh, Korn shell (Ksh), and zsh shells.*

*3. Why do you need a package manager. What package managers do you know in Linux?*

**Linux package managers** are tools that allow users to efficiently install, update, remove, and manage software packages on their systems. The main functions of a package manager include:

**-** Installing programs: Allows you to easily install new programs or packages from official repositories.

- Updates: Provides the ability to update installed programs and their dependencies to the latest versions.

- Uninstall: Allows users to uninstall apps and packages that they no longer need.

- Dependency management: Automatically resolves dependencies, meaning that it installs the necessary libraries and other components for programs to work correctly.

- Search and information: Provides tools for searching for packages, learning about them, and viewing a list of installed packages.

**Some popular package managers in Linux:**

**1) Advanced Package Tool (APT):**

**-** Used in Debian and derivatives (Ubuntu, Linux Mint).

- Commands: **'apt-get', 'apt', 'apt-cache'.**

**2) Yellowdog Updater, Modified (YUM):**

**-** Used in Red Hat, CentOS, Fedora and derivatives.

- Commands: **'yum'.**

**3) Pacman:**

**-** Used in Arch Linux and derivatives.

- Commands: **'pacman'.**

**4) zypper:**

**-** Used in openSUSE and SUSE Linux Enterprise.

- Commands: **'zypper'.**

**5) dnf:**

**-** A newer package manager that replaces YUM in Fedora and derivatives.

- Commands: **'dnf'.**

**6) Portage (emerge):**

**-** Used in Gentoo Linux.

- Commands: **'emerge'.**

**7) Pkg:**

**-** Used in FreeBSD and other systems that use ports.

- Commands: **'pkg'.**

*4. What security features are used in Linux?*

**Permissions:** Linux uses a system of file and directory permissions (rwx) that restricts user access to resources.

**Firewall:** Tools such as iptables or the higher-level firewalld and ufw are used to control network traffic and restrict access to the system.

**SELinux and AppArmor:** Both of these mechanisms control the access of processes to system resources and can limit the capabilities of vulnerable applications.

**Security Auditing:** Tools such as Auditd allow you to record events and audit user actions to detect anomalous behavior or abuse.

**SSH (Secure Shell):** For secure remote access, SSH is used to encrypt communications between the client and server.

**Tunneling:** Use of virtual private networks (VPNs) or other tunneling techniques to secure communications on a network.

**Resource Limiting:** Use cgroups to limit the resources that processes can use, such as CPU, memory, and I/O.

**Updates and patches:** Regular system updates and patching are an important part of security because they fix vulnerabilities.

**File system encryption:** Use encryption to protect data at the file system level, for example, using LUKS (Linux Unified Key Setup) to encrypt disks.

**Anti-virus software:** Antivirus software can be used to deal with files that have potential threats, although this is less common than with other operating systems.

**Logging:** Logging allows you to detect and respond to potential threats in a timely manner.

*5. Why has the use of virtualization become so relevant now?*

The use of virtualization has become very relevant for several reasons that reflect important technological and business trends:

**Efficient use of resources:** Virtualization allows you to create virtual environments that isolate different applications and operating systems from physical hardware. This allows for efficient use of server resources, reducing the amount of unused resources.

**Easy to manage and deploy:** Virtual machines can be quickly created, copied, and moved between physical servers without the need for extensive hardware reconfiguration. This makes the management and deployment process more flexible and faster.

**Isolation and security:** Virtualization allows you to isolate different environments from each other, which helps to avoid interference between applications and operating systems. This increases the security and resilience of systems.

**Fast recovery and backup:** With the ability to create and restore virtual machines from backups, virtualization provides fast recovery in the event of data leaks or other disasters.

**Migration and load balancing:** The ability to move virtual machines between physical servers allows for load balancing and optimized resource utilization.

**Testing and development:** Virtualization provides isolated software testing and development without impacting the real production environment.

**Open standards and cloud services:** Virtualization is supported by open standards, which facilitates interoperability between different vendors. In addition, virtualization technology is widely used in cloud services to ensure scalability and efficient use of resources.

*6. How do you understand the concept of containerization?*

**Containerization** is a methodology for deploying and managing software applications and their dependencies based on the use of containers. A container is an isolated unit that includes an application and all of its dependencies, including libraries, configuration files, and more. Containers isolate applications from each other and from the operating system, providing a consistent and reliable environment for deploying and running applications.

*7. What are the advantages/disadvantages of using open source software?*

**Advantages of using open source software:**

- Open access to the source code: Users can view, modify, and improve the source code of the software, which provides greater transparency and auditability.

- Collaboration and community: A large community of developers and users can collaborate on improvements and bug fixes, resulting in rapid product development and improvement.

- Cost: Open source software is often free or comparable in cost to closed source software, reducing license costs.

- Flexibility and adaptability: With the ability to modify the source code, users can adapt the software to their needs and requirements.

- Security: The community can quickly fix identified vulnerabilities and provide updates, which contributes to an increased level of security.

- Open standards: The use of open standards promotes compatibility and integration with other software products.

- Vendor independence: Users are not dependent on a specific vendor or manufacturer as they can interact with different implementations of the source code.

**Disadvantages of using open source software:**

- Lack of guaranteed support: Some projects can be developed within an open community without guaranteed support from an official vendor.

- Unstable updates: A large number of changes made can lead to unstable software versions.

- Limited functionality: Depending on the project, some programs may have limited functionality compared to commercial counterparts.

- Cost of support: While the software itself may be free, support and training may be costly.

- Specialization of specialists: Developing and maintaining open source software may require the availability of specialists with open source experience.

- Lack of management standards: The open nature of development can lead to a lack of governance and project management standards.

- Less commercial support: In some cases, enterprise users may feel that they lack the guaranteed commercial support that closed solutions can provide.

*8. \*\*\*How many active virtual consoles (terminals) can be present in the Linux process by default. How to call them and switch between them? What are some examples?*

In Linux systems, there are usually several virtual consoles (terminals), and the number of them can be configured in the configuration files. Usually, by default, the system has 6 virtual consoles available through the keyboard shortcuts **'Ctrl + Alt + F1'** to **'Ctrl + Alt + F6'.**

To switch between virtual consoles, use the keyboard shortcut **'Ctrl + Alt + F[console number]'**. For example:

- **'Ctrl + Alt + F1'**: Switch to the first console.

- **'Ctrl + Alt + F2'**: Switch to the second console.

- and so on until **'F6'**.

In addition, in graphical environments such as GNOME or KDE, you can use terminal emulators to work with virtual consoles inside the GUI.

For example, if you leave the GUI and need to use a text terminal, you can call a new terminal, for example, with **'gnome-terminal'** (GNOME) or **'konsole'** (KDE), depending on your environment.

*9. \*\*\*Which virtual console (terminal) serves as a graphical shell?*

In Linux systems that use a graphical environment, the graphical user interface (GUI) usually runs on one of the virtual consoles, usually labeled **'tty7'** or **':0'**. This virtual console is designed for graphical display and user interaction.

*10. \*\*\*Is it possible to log in to a Linux system multiple times under the same system name? What advantages can this provide?*

In a Linux system, it is possible to register several sessions (logins) under the same system name, and this functionality is called terminal multiplexing. For this purpose, virtual consoles or terminal sessions are used.

**The main advantages of terminal multiplexing include:**

**- Work with multiple tasks:** The user can open multiple terminal windows or virtual consoles, each of which can perform different tasks.

**- Switching between tasks:** The user can easily switch between different terminal sessions without leaving the graphical environment.

**- Access to the terminal from many locations:** If a user connects to the system remotely via SSH or other means, he or she can open multiple terminal sessions under one account.

**- Saving sessions:** Some multiplexing tools, such as tmux or screen, allow you to save sessions and resume them later.

**- Parallel work:** Different terminal sessions can perform tasks in parallel, which improves user productivity.

To use terminal multiplexers, you can use tools such as **'tmux'** or **'screen'**. These tools allow you to create, switch, and manage different terminal sessions on one or more virtual consoles.

**Conclusion:** I got acquainted with the interfaces of the Linux OS, gained practical skills in working in the Linux and mobile OS environments - their graphical shell, logging in and out of the system, familiarized myself with the structure of the desktop, learned the basic actions and settings when working in the system.