“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної та програмної інженерії

ЗВІТ ПО ВИКОНАННЮ

ЛАБОРАТОРНОЇ РОБОТИ №2

з дисципліни: «Операційні системи»

Тема: “Знайомство з інтерфейсом та можливостями ОС Linux”

Виконали

студенти

групи РПЗ-03

Команда 6: Sichkar Maxim,

Brytyuk Bohdan

Kyiv 2023

**Мета роботи:**

1. Знайомство з інтерфейсами ОС Linux.

2. Отримання практичних навиків роботи в середовищах ОС Linux та мобільної ОС – їх графічною оболонкою, входом і виходом з системи, ознайомлення зі структурою робочого столу, вивчення основних дій та налаштувань при роботі в системі

**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.**

1. Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеликий словник

базових англійських термінів з питань призначення команд та їх параметрів.

2. Вивчіть матеріали онлайн-курсу академії Cisco “NDG Linux Essentials”:

- Chapter 3 - Working in Linux

- Chapter 4 - Open Source Software and Licensing

3. Пройдіть тестування у курсі NDG Linux Essentials за такими темами:

- Chapter 03 Exam

- Chapter 04 Exam

4. Дайте визначення наступним поняттям:

**- CLI-режим**

A command-line interface (CLI) is a type of text-based user interface between a computer and a user, in which instructions can be given to the computer only by entering text strings (commands). It is also known as a console. The command line interface can be contrasted with menu-based program control systems or various graphical interface implementations. The format of information output in the command line interface is not regulated; usually it is simple text output, but it can also be graphical, sound output, etc.

**- Термінал на основі графічного інтерфейсу користувача**

A GUI-based terminal is a program that allows you to execute commands through a graphical user interface. It provides users with the ability to use commands, such as creating folders, moving files, and performing other tasks, through graphical user interfaces displayed on the screen. Such terms are formerly used in operating systems such as Windows and macOS.

**- Віртуальний термінал**

A virtual terminal is an emulation of a text terminal on a computer or server that allows users to work with multiple processes or sessions in a single terminal window, switch between them, and execute commands in each individual virtual terminal.

Standard virtual terminals are used in Unix-based operating systems, such as Linux, to provide multitasking and the ability to interact with many users through a single terminal window. Virtual terminals can also be used to run various shells (shells), isolated environment programs, and other system administration and software development tasks.

5. Підготувати в електронному вигляді початковий варіант звіту:

- Титульний аркуш, тема та мета роботи

- Словник термінів

- Відповіді на п.5 та п.6 з завдань для попередньої підготовки

**Хід роботи.**

1. Робота в графічному режимі в ОС сімейства Linux (робота з інтернет-джерелами):

1.1. Оберіть графічну оболонку для ОС сімейства Linux, яку ви хочете розглянути. Розгляньте

структуру робочого простору користувача, та опишіть основні його компоненти (\*\*\*показано

основні компоненти оболонки Gnome):

- Закладка Applications

- Закладка Places

- Меню System

- Навігаційний простір Activities overview

\*\*\* Якщо ви обрали інший графічний інтерфейс то компоненти меню можуть бути іншими.

1.2. Запуск програм. Дослідіть можливості запуску додатків різними способами (описати спосіб і по-

можливості показати скріншоти):

- Запуск програм через панель швидкого запуску

- Запуск програм через пошук в меню

- Запуск програм через віджет запуску

- Запуск програм через глобальне меню

1.3. Вихід з системи та завершення роботи в Linux. Як виконати в графічному інтерфейсі наступні дії

(наведіть скріни):

- Зміна користувача на root

- Перезавантаження системи

- Вимкнення системи

2. Робота в середовищі мобільної ОС.

2.1. Опишіть головне меню вашої мобільної ОС, який графічний інтерфейс вона використовує?

2.2. Опишіть меню налаштувань компонентів мобільного телефону.

2.3. Використання комбінацій клавіш для виконання спеціальних дій.

2.4. Вхід у систему та завершення роботи пристрою. Особливості налаштувань живлення батареї.

**Контрольні запитання**

1. Наведіть приклади серверних додатків Linux для сервера баз даних, серверів розсилки повідомлень та файлообмінників.

**Database server:**

* MySQL
* PostgreSQL
* MongoDB
* MariaDB
* Oracle Database
* SQLite

**Mail servers:**

* postfix
* Sendmail
* Exim
* Dovecot
* Qmail

**File sharing servers:**

* samba
* NFS (Network File System)
* FTP (File Transfer Protocol) server, such as vsftpd
* SSH (Secure Shell) file transfer, such as scp or sftp
* WebDAV (Web Distributed Authoring and Versioning) server, such as Apache mod\_dav

1. **Порівняйте оболонки Bourne, C, Bourne Again (Bash), the tcsh, Korn shell (Ksh) та zsh.**

**Bourne shell (sh):**

* The original Unix shell, written by Stephen Bourne in the 1970s.
* Basic functionality for command-line interaction, input/output redirection, pipelines, and scripting.
* Limited interactive features, no command-line editing capabilities, and no support for aliases.

**C shell (csh):**

* Developed at UC Berkeley by Bill Joy in the late 1970s.
* Features command-line editing, history, aliases, job control, and support for scripting with a C-like syntax.
* Not as widely used today due to some known limitations, including being less efficient for scripting than the Bourne shell and a non-standard syntax.

**Bourne-Again shell (bash):**

* Written as a free software replacement for the original Bourne shell by Brian Fox for the GNU Project in the 1980s.
* Offers a rich set of interactive features, including command-line editing, history, aliases, and job control.
* Supports shell scripts with a syntax that is largely backward-compatible with the original Bourne shell, as well as many extensions and improvements.

**tcsh:**

* A more recent version of the C shell that adds several new features.
* Includes command-line editing, history, aliases, job control, and support for scripting with a syntax that is similar to C but not identical.
* A popular choice for interactive use among some users, particularly in the scientific computing community.

**Korn shell (ksh):**

* Developed by David Korn in the 1980s as an extension of the Bourne shell.
* Combines the features of both the Bourne and C shells, with support for command-line editing, history, aliases, job control, and scripting with a syntax similar to the C programming language.
* Widely used in commercial Unix systems, including IBM's AIX and HP's HP-UX.

**Z shell (zsh):**

* Developed by Paul Falstad in the early 1990s.
* Similar to bash in terms of interactive features, but also includes advanced completion and globbing capabilities, support for plugins and themes, and a powerful configuration system.
* A popular choice among power users and developers, particularly on macOS.

1. **Для чого потрібен менеджер пакетів. Які менеджери пакетів ви знаєте у Linux?**

A package manager is a software tool used to automate the installation, configuration, and removal of software packages on a computer system. Package managers provide a standardized way to manage software dependencies, enabling users to easily install and update applications and libraries without having to manually download and install each component individually.

In Linux, package managers are used extensively to install and manage software on the system. Some of the most commonly used package managers in Linux include:

**Advanced Package Tool (APT):**

APT is a package manager used by Debian, Ubuntu, and other Debian-based Linux distributions.

APT uses a command-line interface to manage packages, and provides powerful features for dependency resolution, package installation and removal, and package version management.

**Yellowdog Updater Modified (YUM):**

YUM is a package manager used by Red Hat, Fedora, and other Red Hat-based Linux distributions.

YUM uses a command-line interface to manage packages, and provides powerful features for dependency resolution, package installation and removal, and package version management.

**Pacman:**

Pacman is a package manager used by Arch Linux and its derivatives.

Pacman uses a command-line interface to manage packages, and provides powerful features for dependency resolution, package installation and removal, and package version management.

**Zypper:**

Zypper is a package manager used by SUSE Linux and its derivatives.

Zypper uses a command-line interface to manage packages, and provides powerful features for dependency resolution, package installation and removal, and package version management.

**DNF:**

DNF (Dandified YUM) is a modern package manager used by Fedora and its derivatives.

DNF uses a command-line interface to manage packages, and provides powerful features for dependency resolution, package installation and removal, and package version management.

**Snap:**

Snap is a newer package manager that is cross-distribution and allows users to install applications that are built with it.

Snap uses a command-line interface to manage packages and can also be used through a graphical interface.

1. **Які засоби безпеки використовуються в Linux?**

* User and group permissions: Linux uses a permission-based security model, where every file, directory, and resource is owned by a specific user and group, and permissions are granted based on the user and group ownership.
* File system security: Linux supports a range of filesystem security mechanisms, such as Access Control Lists (ACLs) and SELinux, which enable administrators to set fine-grained access control policies to restrict access to files and directories.
* Firewall: Linux comes with a built-in firewall called iptables or nftables, which allows users to filter incoming and outgoing traffic based on port and protocol.
* Encryption: Linux supports encryption at different levels, such as encryption of individual files, encryption of entire partitions, and encrypted communication channels using protocols like SSL/TLS and SSH.
* Auditing and monitoring: Linux has built-in auditing and monitoring capabilities that enable administrators to track system activities, log events, and analyze system performance.
* Package signing: Linux package managers such as APT and YUM support package signing, which ensures that packages are authentic and have not been tampered with.
* Root account restrictions: Linux has restrictions on the use of the root account, which is the system administrator account with full privileges. By default, the root account is disabled, and users are required to use the sudo command to execute administrative tasks.
* Security-focused distributions: There are several security-focused Linux distributions, such as Kali Linux and Parrot Security OS, that are specifically designed for penetration testing, digital forensics, and security auditing.

In summary, Linux provides a range of built-in security features and utilities that can be used to secure the system and data. However, as with any operating system, it is important to keep the system up to date with security patches and to follow best practices for security, such as using strong passwords, restricting access to sensitive files and directories, and keeping software and packages up to dates

1. **Чому використання віртуалізації зараз стало таким актуальним?**

* The use of virtualization has become increasingly relevant in recent years for several reasons:
* Cost savings: Virtualization allows multiple virtual machines to run on a single physical server, which can lead to significant cost savings by reducing the need for additional physical servers, associated hardware, and infrastructure.
* Improved resource utilization: Virtualization enables more efficient use of resources by allowing multiple virtual machines to share resources such as CPU, memory, and storage, leading to improved utilization of physical hardware.
* Increased flexibility: Virtualization provides a high degree of flexibility and agility in terms of deploying and managing software applications and services, enabling faster provisioning of resources and greater scalability.
* Simplified management: Virtualization provides a centralized and simplified management interface for managing multiple virtual machines and associated resources, making it easier to monitor and maintain the system.
* Disaster recovery: Virtualization enables faster and more efficient disaster recovery solutions by enabling the quick restoration of virtual machines in the event of hardware failure or system downtime.
* Security: Virtualization can provide a more secure environment by enabling the isolation of applications and services within separate virtual machines, reducing the risk of security breaches and data loss.
* Cloud computing: Virtualization is a key technology behind cloud computing, which has become increasingly popular as a means of delivering scalable and cost-effective IT services to businesses and organizations.

In summary, the use of virtualization has become increasingly relevant due to the benefits it provides in terms of cost savings, improved resource utilization, increased flexibility, simplified management, disaster recovery, security, and cloud computing. As the demand for more efficient and scalable IT services continues to grow, virtualization is likely to become even more important in the future.

1. **Як ви розумієте поняття контейнеризації?**

Containerization is a method of deploying and running applications in a lightweight, self-contained environment that provides a consistent and isolated runtime environment for the application. Containerization uses container images to package and distribute applications, along with their dependencies and configurations, in a standardized and portable format.

The main idea behind containerization is to provide a way to package an application along with its dependencies in a single container, which can then be deployed on any platform that supports the container format. Containers provide a lightweight and isolated runtime environment that enables applications to run consistently across different environments, such as development, testing, and production, without requiring significant changes to the underlying infrastructure.

* Portability: Containers are portable and can run on any platform that supports the container format, making it easier to deploy applications across different environments.
* Isolation: Containers provide a high degree of isolation between the application and the host system, reducing the risk of conflicts and ensuring that the application runs consistently and reliably.
* Lightweight: Containers are lightweight and require less overhead than traditional virtual machines, making them easier to deploy and scale.
* Efficiency: Containerization can improve resource utilization and reduce infrastructure costs by enabling more efficient use of resources and improving application performance.
* Flexibility: Containerization provides a high degree of flexibility in terms of deploying and managing applications, enabling faster provisioning of resources and greater scalability.
* Security: Containers can provide a more secure runtime environment by isolating applications from each other and from the host system, reducing the risk of security breaches and data loss.

In summary, containerization provides a way to package and deploy applications in a consistent and isolated runtime environment, making it easier to deploy and manage applications across different environments and improving resource utilization and efficiency.

1. **Які переваги/недоліки використання програмного забезпечення з відкритим кодом?**

Open source software is software whose source code is publicly available and can be modified and distributed by anyone under an open source license. There are several advantages and disadvantages of using open source software, as discussed below:

**Advantage:**

* Cost: Open source software is usually free to download, use, and distribute, which can save organizations a significant amount of money compared to proprietary software.
* Customization: The source code of open source software can be modified and customized to meet specific needs, allowing greater flexibility and control over the software.
* Community support: Open source software is often supported by a large community of developers, users, and contributors who provide support and contribute to the development of the software.
* Security: Since the source code of open source software is publicly available, security vulnerabilities can be identified and fixed quickly by the community, making it more secure than proprietary software.
* Interoperability: Open source software is often designed to be compatible with other open source software, enabling greater interoperability and flexibility in the software ecosystem.

**Disadvantages:**

* Support: Since open source software is often developed and supported by a community of volunteers, commercial support may be limited or not available, which can be a disadvantage for organizations that require professional support.
* Compatibility: Open source software may not always be fully compatible with proprietary software, which can cause compatibility issues and difficulties in integration with other software.
* Complexity: Open source software can be more complex to install and configure compared to proprietary software, requiring a higher level of technical expertise.
* User interface: Open source software may not always have a user interface that is as polished or user-friendly as proprietary software, which can make it less appealing to some users.
* Documentation: The documentation for open source software may not be as comprehensive or well-maintained as that of proprietary software, which can make it more difficult for users to learn and use the software.
* In summary, open source software has several advantages, such as cost savings, customization, and community support, but also has some disadvantages, such as limited commercial support, compatibility issues, and complexity. The choice of whether to use open source or proprietary software depends on the specific needs and requirements of the organization.

1. **\*\*\*Скільки активних віртуальних консолей (терміналів) може бути у процесі роботи Linux по замовчуванню. Як їх викликати та між ними перемикатися? Наведіть приклади?**

By default, a Linux system can have up to six active virtual consoles (terminals) available. These virtual consoles can be accessed by pressing the "Ctrl+Alt+F1" to "Ctrl+Alt+F6" keys on the keyboard.

To switch between the virtual consoles, you can use the "Ctrl+Alt+Fn" key combination, where "n" is the number of the virtual console you want to switch to (e.g., "Ctrl+Alt+F2" to switch to the second virtual console).

Here are some examples of how to access and switch between virtual consoles:

* To access the first virtual console, press "Ctrl+Alt+F1".
* To switch to the second virtual console, press "Ctrl+Alt+F2".
* To switch back to the first virtual console, press "Ctrl+Alt+F1" again.
* To access the sixth virtual console, press "Ctrl+Alt+F6".
* To switch to the third virtual console, press "Ctrl+Alt+F3".

Virtual consoles are useful for a variety of tasks, such as accessing the command line interface, debugging system issues, and running multiple terminal sessions simultaneously. They provide a convenient way to switch between different sessions without needing to log in and out of the system repeatedly.

1. **\*\*\*Яка віртуальна консоль (термінал) виконує функцію графічної оболонки?**

Typically, the graphical shell runs on the seventh virtual console (tty7) in Linux. It is here that the X server, which is responsible for handling graphical displays, is started and runs. Users interact with the graphical shell by logging in through a graphical login screen, which is provided by the display manager.

When a user logs in through the graphical login screen, the display manager launches a session manager, which in turn starts the user's preferred desktop environment or window manager. This environment provides a graphical user interface (GUI) for the user to interact with and run applications.

So, in summary, the seventh virtual console (tty7) is the one that serves as the graphical shell in Linux.

1. **\*\*\*Чи можлива реєстрація в системі Linux декілька разів під одним і тим же системним ім’ям? Які переваги це може надати?**

It is technically possible to register multiple users with the same system name in a Linux system, but it is generally not recommended. Each user account in Linux has a unique User ID (UID) that is used to identify and manage the user's access to system resources. If multiple users have the same system name but different UIDs, it can cause confusion and potential security issues.

However, there are some cases where creating multiple users with the same system name may be beneficial, such as:

* Service accounts: In some cases, a service may require its own user account with a specific system name, such as "apache" or "mysql". Multiple instances of the same service may require separate user accounts with the same system name to manage permissions and access to resources.
* Virtual users: Some systems, such as FTP servers, use virtual users to authenticate and manage access to files and directories. Virtual users can be created with the same system name but different UIDs to provide a common login name for multiple users.

In general, it is best to avoid creating multiple users with the same system name, as it can cause confusion and potential security issues. Instead, it is recommended to create unique user names and UIDs for each user account to ensure proper management of system resources and access.