"Kyiv Professional College of Communications" Computer Engineering Cycle Commission

**Laboratory work №1**

**Theme:** “Introduction to the virtual machine environment and features of the Linux operating system”

in the discipline: "Operating Systems"

Performed by students

prepared by students:

KSM-33 group

Team:

Koval Ivan Volodymyrovych,

Kuzmenko Denys Maksymovych

The teacher checked:

Sushanova V.S.

**Objectives:**

1. Introduction to various types of hypervisors, virtualization when working with operating systems.

2. Introduction to the main types of modern OS, a brief overview of their capabilities.

**Material support for classes:**

1. IBM PC type computer.

2. Windows family OS and Virtual Box virtual machine (Oracle).

3. GNU/Linux OS (any distribution).

4. Cisco network academy website netacad.com and its online Linux courses

### **Словник базових англійських термінів**

* **Virtual Machine** : Програмна імітація комп'ютерної системи, що дозволяє запускати операційну систему та програми так, ніби це реальне обладнання.
* **Hypervisor**: Програмне забезпечення, яке створює, запускає та керує віртуальними машинами.
* **Type 1 Hypervisor** : Працює безпосередньо на залізі комп'ютера. Він сам є операційною системою. Приклади: VMware ESXi, Microsoft Hyper-V, Xen.
* **Type 2 Hypervisor**: Працює як звичайна програма всередині існуючої операційної системи. Приклади: Oracle VirtualBox, VMware Workstation.
* **Host Operating System**: Основна ОС, встановлена на фізичному комп'ютері, в якій запускається гіпервізор 2-го типу.
* **Guest Operating System**: ОС, що працює всередині віртуальної машини.

### **Поняття та типи гіпервізорів**

**Гіпервізор** — це програмний або апаратний шар, який забезпечує віртуалізацію, дозволяючи на одному фізичному комп'ютері одночасно запускати одну або декілька ізольованих віртуальних машин. Його головне завдання — керувати ресурсами хоста та розподіляти їх між гостьовими операційними системами.

Існує два основних типи гіпервізорів:

* **Гіпервізори 1-го типу**  Вони встановлюються безпосередньо на фізичне обладнання комп'ютера, минаючи хостову операційну систему. Такий підхід забезпечує максимальну продуктивність та безпеку, оскільки немає зайвого програмного шару.
* **Гіпервізори 2-го типу** Вони встановлюються та працюють як звичайна програма всередині повноцінної хостової операційної системи (наприклад, Windows, macOS або Linux). Вони простіші в установці та керуванні, що робить їх популярними для розробки, тестування програмного забезпечення та для настільного використання.

**Хід роботи**

* 1. Подивіться ознайомчі відео та демонстраційні матеріали з наступних напрямків:
  2. Linux - Кращі дистрибутиви 2023 <https://youtu.be/PahmJBU9HKA?si=maxRf0nZlqs2hFGU>
  3. ТОП 5 ПРИЧИН ЧОМУ АЙТІШНИКУ ВАРТО ПЕРЕЙТИ НА ЛІНУКС

Доступ: <https://youtu.be/bP3_mZKezvM?si=sM3Mpc9JQ_0bY9Yd>

* 1. Як встановити Linux разом з Windows спосіб #1 Microsoft Store

Доступ: <https://youtu.be/eEdGl6HvSdM?si=WDbwa71i034D2rQj>

* 1. Як встановити Linux разом з Windows спосіб #2 Dual Boot

Доступ: <https://youtu.be/Hfky8TEyXss?si=ilduY167LS-vKl9y>

* 1. Як встановлювати програми на Linux. Linux українською #1

Доступ: <https://youtu.be/M8XHJME6cxI?si=L0Koom59jTRnPXnU>

* 1. Як зробити панель завдань Linux як у Windows. Linux українською #2

Доступ: <https://youtu.be/9szAz-A4gaM?si=LxaVueluI3tKRb1r>

* 1. Як встановити Ubuntu на VirtualBox <https://youtu.be/ADOaHm1VZII?si=hG5kDRsajFn7se8d>
  2. The Shell (Linux) <https://drive.google.com/open?id=0B0PV0_SM0LoDSVNPWUVRdUxaN2s>
  3. Linux Desktop Environments: XFCE vs GNOME vs KDE

Доступ: <https://youtu.be/2JBGQfPR5xQ?si=euswD7IHrODd-6JH>

**Відповіді на питання**

2.1. **Етапи розгортання операційної системи на базі віртуальної машини VirtualBox**

1. Створення нової віртуальної машини: Запуск VirtualBox, натискання кнопки "Створити", введення назви, вибір типу та версії операційної системи.
2. Налаштування апаратних ресурсів: Виділення необхідного обсягу оперативної пам'яті, створення або вибір існуючого віртуального жорсткого диска та налаштування його розміру.
3. Вибір образу операційної системи: Підключення файлу-образу (ISO) операційної системи, з якого буде відбуватися встановлення.
4. Запуск віртуальної машини: Натискання кнопки "Запустити" для початку процесу встановлення.
5. Встановлення ОС: Виконання кроків встановлення операційної системи, що включають вибір мови, розкладки клавіатури, часового поясу, створення користувача та розділів диска.
6. Встановлення Guest Additions (Доповнень гостьової ОС): Після встановлення ОС, інсталяція спеціальних доповнень, які покращують взаємодію між гостьовою ОС та VirtualBox (наприклад, динамічна зміна розміру екрана, спільні папки тощо).

2.2. **Апаратні обмеження для 32- та 64-бітних ОС**

### **1. Обмеження 32-бітних ОС**

1. **Обсяг оперативної пам’яті (RAM)**
   * 32-бітні процесори використовують 32-бітну адресну шину.
   * Це означає, що вони можуть адресувати максимум **2³² комірок пам’яті = 4 ГБ**.
   * Реально доступно ще менше (приблизно 3–3,5 ГБ), бо частина адресного простору резервується для відеопам’яті, драйверів та пристроїв.
2. **Робота з програмами**
   * Кожен процес зазвичай отримує максимум **2 ГБ** адресного простору (інколи 3 ГБ, якщо в системі використовується спеціальний параметр /3GB).
   * Тому важкі програми (CAD, відеоредактори, ігри) можуть працювати з обмеженнями.
3. **Процесори та драйвери**
   * 32-бітна ОС не зможе повноцінно використати багатоядерні 64-бітні процесори.
   * Нові драйвери для сучасного обладнання часто не випускаються у 32-бітному варіанті.
4. **Програмне забезпечення**
   * Сучасні програми рідко підтримують 32-бітні системи, що обмежує можливості користувача.

### **2. Обмеження 64-бітних ОС**

1. **Оперативна пам’ять**
   * 64-бітна адресна шина теоретично дозволяє адресувати **2⁶⁴ комірок = 16 ексабайт** RAM.
   * Реально обсяг обмежується архітектурою процесора і версією ОС.
   * Наприклад, Windows 10 Home підтримує максимум **128 ГБ**, тоді як Windows 10 Pro — **2 ТБ**.
2. **Програми та сумісність**
   * 64-бітна ОС може запускати як 64-бітні, так і більшість 32-бітних програм (через спеціальні підсистеми).
   * Але 16-бітні програми у ній зазвичай не підтримуються.
3. **Драйвери**
   * Для 64-бітної ОС потрібні спеціально розроблені 64-бітні драйвери.
   * Старе обладнання може не мати таких драйверів, тому інколи його неможливо підключити.
4. **Використання ресурсів**
   * 64-бітні програми зазвичай займають більше пам’яті (через більші розміри вказівників), але натомість можуть працювати з великими обсягами даних і ефективніше використовувати багатоядерність.

2.3. **Основні етапи встановлення Linux в текстовому режимі**

1. Вибір мови та клавіатурної розкладки: Використовуючи стрілки на клавіатурі, виберіть потрібні налаштування.
2. Налаштування мережі: Призначення IP-адреси, маски підмережі та DNS-серверів.
3. Розбиття диска: Створення, редагування або видалення розділів диска вручну (наприклад, для кореневого розділу /, домашнього каталогу /home, та розділу підкачки swap).
4. Створення користувача: Введення імені користувача, пароля та, за потреби, налаштування прав адміністратора (sudo).
5. Встановлення програмного забезпечення: Вибір необхідних пакетів і компонентів для встановлення, включно з базовою системою, ядром та утилітами.
6. Налаштування завантажувача: Встановлення GRUB або іншого завантажувача, щоб система могла коректно запускатися після перезавантаження.
7. Завершення встановлення: Перезавантаження системи для переходу до першого запуску.

2.4. **Встановлення графічних оболонок Gnome та KDE в Linux, що встановлена в текстовому режимі**

Для встановлення Gnome та KDE в текстовому режимі потрібно скористатися менеджером пакетів, який залежить від вашого дистрибутива (наприклад, apt для Debian/Ubuntu, dnf для Fedora, pacman для Arch).

## 

## 

## **Установка GNOME**

### **Для Debian / Ubuntu:**

* *sudo apt update*
* *sudo apt install ubuntu-gnome-desktop* (встановлює GNOME та необхідні пакети)

Якщо хочете мінімальну версію GNOME (без зайвих програм):

* *sudo apt install gnome-core*

Після встановлення можна переключитись на графічний режим:

* *sudo systemctl set-default graphical.target* (робочий режим за замовчуванням – GUI)
* *sudo systemctl start graphical.target*  (запустити графічний режим зараз)

### **Для Fedora:**

* *sudo dnf groupinstal*
* *sudo systemctl set-default graphical.target*
* *sudo systemctl start graphical.target*

## **Установка KDE Plasma**

### **Для Debian / Ubuntu:**

* *sudo apt update*
* *sudo apt install kde-plasma-desktop* (мінімальний KDE Plasma)
* *sudo apt install kde-standard*  (стандартний пакет з додатковими утилітами)

Перехід у GUI:

* *sudo systemctl set-default graphical.target*
* *sudo systemctl start graphical.target*

### **Для Fedora:**

* *sudo dnf groupinstall "KDE Plasma Workspaces"*
* *sudo systemctl set-default graphical.target*
* *sudo systemctl start graphical.target*

## **Перемикання між графічними оболонками** (тільки для Debian/Ubuntu)

* При вході в GUI можна обрати потрібну оболонку (GNOME або KDE) у диспетчері входу (GDM, SDDM).
* Якщо потрібно змінити оболонку для конкретного користувача:
  + *sudo update-alternatives --config x-session-manager*

1. **Важливі пакети**

**Gnome:**

* gnome-core
* ubuntu-gnome-desktop

**KDE Plasma:**

* kde-plasma-desktop
* kde-standard

2.5. **Характеристика графічної оболонки**

**GNOME (GNU Network Object Model Environment):**

Це одна з найпопулярніших і найсучасніших графічних оболонок для Linux. Має мінімалістичний і елегантний дизайн, орієнтований на простоту використання. Активно використовує концепцію "робочих просторів" (workspaces) та "огляду активностей" (Activities Overview) для управління вікнами та додатками. Рекомендується для користувачів, які цінують сучасний вигляд, зручність та інтеграцію з хмарними сервісами.Потребує значних системних ресурсів порівняно з іншими середовищами.

**JWM (Joe's Window Manager):**

Легковаговий менеджер вікон, призначений для швидкої та ефективної роботи.Ідеально підходить для старих комп'ютерів або систем з обмеженими ресурсами (наприклад, Raspberry Pi).Має класичний вигляд робочого столу з кнопкою "Пуск", панеллю завдань і системним треєм.Налаштовується за допомогою XML-файлу конфігурації. Не має вбудованих інструментів для управління файлами чи робочим столом, тому часто використовується разом з іншими програмами (наприклад, Thunar для файлів, nitrogen для шпалер).

**Контрольні питання**

### 1. Порівняння гіпервізорів типу 1 та типу 2. Відмінність та сфера застосування.

**Гіпервізор типу 1 (bare-metal)** — це програмне забезпечення, яке працює безпосередньо на фізичному обладнанні комп'ютера . Він управляє ресурсами системи та надає їх віртуальним машинам . Гіпервізор типу 1 є більш ефективним, оскільки він має доступ до апаратних ресурсів без необхідності взаємодіяти з операційною системою, яка могла б накладати додаткові обмеження.  
**Приклад**: VMware ESXi, Microsoft Hyper-V, Xen.  
**Сфера застосування**: Серверні системи, дата-центри, віртуалізація на рівні підприємства.

**Гіпервізор типу 2 (hosted)** — це програмне забезпечення, яке працює поверх існуючої операційної системи. Гіпервізор типу 2 використовує ресурси хост-операційної системи для управління віртуальними машинами.  
**Приклад**: VMware Workstation, Oracle VirtualBox.  
**Сфера застосування:** Домашні користувачі, розробка та тестування, легка віртуалізація.

### 2. **GNU GPL** — це ліцензія для програмного забезпечення, що передбачає, що програма з відкритим кодом може бути вільно використана, змінена і розповсюджена, але з певними умовами. Основною концепцією є свобода користувачів на кожному етапі використання програмного забезпечення.

**Основна концепція GNU GPL**

* Свобода використання
* Свобода вивчення та зміни коду
* Свобода поширення копій
* Свобода поширення модифікацій

### **3. Суть програмного забезпечення з відкритим кодом.**

**Суть програмного забезпечення з відкритим кодом** полягає в тому, що його вихідний код є відкритим і доступним для перегляду, вивчення, змін та поширення будь-яким користувачем. Це принципово відрізняє його від пропрієтарного (закритого) ПЗ, де код прихований і контрольований лише власником.

### **Основні риси програмного забезпечення з відкритим кодом:**

1. Доступність вихідного коду
2. Можливість модифікації
3. Вільне поширення
4. Спільнотна розробка
5. Прозорість та безпека

### **4. Що таке дистрибутив?**

Дистрибутив — це конкретна збірка операційної системи, яка включає ядро, програмне забезпечення, бібліотеки, утиліти і засоби для управління системою. Він може бути налаштований під різні потреби користувача чи організації. У випадку Linux, дистрибутиви можуть варіюватися за складом програмного забезпечення і навіть інтерфейсом користувача.

### **5. Задачі системного адміністрування на базі ОС Linux.**

ОС Linux є потужним інструментом для системного адміністрування, і через свою гнучкість дозволяє виконувати різноманітні задачі:

* Налаштування та управління користувачами і групами.
* Управління пакетами і програмним забезпеченням.
* Моніторинг системи.
* Захист і безпека системи.
* Налаштування серверів.
* Автоматизація задач за допомогою скриптів.

### **6. Зв'язок ОС Android та Linux.**

ОС Android побудована на ядрі Linux. Це означає, що Android використовує ті ж основи, що й Linux, зокрема ядро, драйвери та базові механізми керування ресурсами. Однак Android додає спеціалізовані шари, такі як бібліотеки, інтерфейси користувача та середовище для мобільних застосунків. Таким чином, Android є специфічним дистрибутивом Linux, орієнтованим на мобільні пристрої.

### **7. Основні можливості та сфера використання Embedded Linux.**

Embedded Linux — це адаптоване до специфічних потреб програмне забезпечення на основі Linux, яке використовується для вбудованих систем. Ці системи часто мають обмежені ресурси, тому для них потрібна оптимізація.

**Основні можливості**:

* Легка настройка для специфічного апаратного забезпечення.
* Підтримка реального часу.
* Низький рівень споживання ресурсів.

**Сфера використання**: Мобільні пристрої, автомобільні системи, розумні пристрої, побутова електроніка, IoT.

### **8. Зміна типу завантаження Linux: текстовий (3 рівень) або графічний (5 рівень).**

У Linux режими роботи системи визначаються **рівнями виконання (runlevels)** або їх сучасним аналогом у **systemd** — **target-и**.

1. **Традиційна схема (SysVinit, runlevels):**
   * **Runlevel 3** – багатокористувацький режим з мережею, але **без графічного інтерфейсу** (CLI).
   * **Runlevel 5** – багатокористувацький режим з мережею **і графічним середовищем** (GUI).

Щоб змінити режим:

* + **Постійно:** у файлі **/etc/inittab** вказати потрібний рівень
  + **Тимчасово (до перезавантаження):**

**init 3**  (перейти в текстовий режим)

**init 5**  (перейти в графічний режим)

1. **Сучасна схема (systemd):**

У більшості сучасних дистрибутивів (Ubuntu, Fedora, Arch, Debian) використовується **systemd**.  
 Тут замість runlevels є targets:

* **multi-user.target** – аналог runlevel 3 (CLI, текстовий режим).
* **graphical.target** – аналог runlevel 5 (GUI).

**Команди:**

* Подивитись поточний режим:
  + **systemctl get-default**
* Задати режим за замовчуванням:
  + **systemctl set-default multi-user.target**  (завантаження в текстовий режим
  + **systemctl set-default graphical.target** (завантаження в графічний режим
* Тимчасово змінити режим:
  + **systemctl isolate multi-user.target** (перейти в CLI)
  + **systemctl isolate graphical.target**  (перейти в GUI)

## **Відмінності між CLI та GUI**

### **1. CLI (Command-Line Interface, текстовий режим)**

* Користувач працює через термінал, вводячи команди.
* Менше ресурсів споживає (не потрібно запускати графічний сервер).
* Швидше завантажується.
* Гнучкіше для адміністрування, автоматизації та роботи на серверах.  
  Складніше для новачків, бо треба пам’ятати команди.

### **2. GUI (Graphical User Interface, графічний режим)**

* Користувач взаємодіє через вікна, меню, кнопки, іконки.
* Зручніше для новачків, легше освоювати.
* Потребує більше ресурсів (процесор, оперативна пам’ять, відеокарта).
* У серверних системах зазвичай не використовується, бо збільшує навантаження.

**Висновок**

На цій лабораторній роботі ми теоретично вичили засади функціонування віртуальних машин та гіпервізорів, з'ясовано ключові відмінності між гіпервізорами типу 1 (наприклад, VMware ESXi), які працюють безпосередньо на апаратному забезпеченні, та гіпервізорами типу 2 (наприклад, Oracle VirtualBox), що функціонують як додатки всередині хостової ОС. Це дало розуміння їхніх сфер застосування; закріпили навички розгортання операційної системи на базі віртуальної машини, включаючи налаштування апаратних ресурсів, процес встановлення та використання Guest. Крім того, було проаналізовано апаратні обмеження для 32- та 64-бітних операційних систем, що є важливим при виборі архітектури; особлива увага була приділена ОС Linux. Було розглянуто етапи встановлення Linux у текстовому режимі та вивчено процес встановлення та налаштування різних графічних оболонок (GUI), таких як GNOME та KDE, а також їх порівняння з легковаговими альтернативами на кшталт JWM. Також було проаналізовано сутність програмного забезпечення з відкритим кодом, поняття дистрибутивів та ключові задачі системного адміністрування на базі Linux.