“Київський фаховий коледж зв’язку”

Циклова комісія комп’ютерної та програмної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №1**

з дисципліни: «Операційні системи»

**Тема: «Ознайомлення з робочим середовищем віртуальних машин та операційних систем різних сімейств»**

Виконавли студенти

групи КСМ-23б

Команда VelikieBomji:

Бережний Т.І., Лобода Д.Р.,

Чухманенко М.В

Перевірила викладач

Сушанова В.С.

Київ 2024

**Мета роботи:**

1. Отримання практичних навиків роботи з середовищами віртуальних машин та операційними системами різних типів та сімейств – 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 10,11).

3. Віртуальна машина – Virtual Box (Oracle).

4. Операційна система GNU/Linux – CentOS.

**Завдання для попередньої підготовки**

1. Прочитавши матеріал з коротких теоретичних відомостей дайте відповіді на наступні питання:
   1. **Охарактеризуйте поняття «гіпервізор». Які бувають їх типи?**

Hypervisor or Monitor of virtual machines — a computer program or processor hardware that ensures the simultaneous and parallel execution of several virtual machines, each of which runs its own operating system, on one physical computer

Type 1 hypervisor - An autonomous hypervisor is loaded by a bootloader or firmware, and runs configured operating systems in individual virtual machines. Some standalone hypervisors have their own device drivers and scheduler.

Type 2 hypervisor - This is a component that works in the same ring with the core of the main OS ("ring 0", according to the terminology of the x86 architecture). Guest code may execute directly on the physical processor, but access to the computer's I/O devices from the guest OS is through a second component, a normal host OS process—the user-level monitor.

**2.2 Перерахуйте основні компоненти та можливості гіпервізорів відповідно до свого варіанту (порядковий номер по журналу), табл.1.**

Лобода Д.Р. № 10

Hyper-V — це віртуалізаційна технологія від Microsoft, яка дозволяє створювати та управляти віртуальними машинами. Основні компоненти та можливості Hyper-V включають:

### Основні компоненти:

1. Hyper-V Manager: Інтерфейс для створення, управління та налаштування віртуальних машин. Зазвичай використовується на комп’ютерах з Windows Server або Windows 10/11 Pro, Enterprise або Education.
2. Virtual Machine Connection: Інструмент для підключення до консолі віртуальної машини та управління нею.
3. Virtual Switch: Компонент, який забезпечує мережеве підключення віртуальних машин. Може бути External (з’єднаний з фізичним мережевим адаптером), Internal (для зв’язку між віртуальними машинами та хост-системою) або Private (для зв’язку лише між віртуальними машинами).
4. Virtual Hard Disk (VHD/VHDX): Формати для віртуальних жорстких дисків, які використовуються для зберігання даних віртуальних машин.
5. Hyper-V Integration Services: Набір драйверів та служб, що покращують інтеграцію між віртуальною машиною та хостовою операційною системою, включаючи поліпшене управління ресурсами, синхронізацію годинника та інші функції.

### Основні можливості:

1. Секретар віртуалізації: Можливість створювати і запускати кілька віртуальних машин на одному фізичному сервері.
2. Динамічне виділення пам'яті: Автоматичне регулювання обсягу оперативної пам'яті, яка виділяється віртуальним машинам на основі їхніх потреб.
3. Live Migration: Переміщення працюючих віртуальних машин між фізичними серверами без переривання їхньої роботи.
4. Snapshot (Checkpoint): Створення знімків стану віртуальної машини для можливості відновлення до попереднього стану у випадку проблем.
5. Replica: Можливість віддаленого резервного копіювання та відновлення віртуальних машин для забезпечення високої доступності.
6. Віртуалізація вкладок: Підтримка віртуалізації для кожного з віртуальних процесорів (якщо потрібно).
7. Сумісність з різними ОС: Підтримка різних операційних систем у віртуальних машинах, таких як Windows, Linux та інші.
8. Віртуальні мережі та мережеве управління: Налаштування віртуальних мереж, VLAN та інші функції для управління мережевими ресурсами віртуальних машин.

**Хід роботи**

**2.1 Перерахуйте етапи для розгортання операційної системи на базі віртуальної машини VirtualBox.**

**2.2 Чи є якісь апаратні обмеження при встановленні 32- та 64-бітних ОС? (Лобода Д.Р.)**

**Підтримка процесором**

**32-бітна ОС**: Підтримується як 32-бітними, так і 64-бітними процесорами.

**64-бітна ОС**: Вимагає 64-бітного процесора. Якщо процесор лише 32-бітний, встановити 64-бітну ОС неможливо.

**Оперативна пам'ять (RAM)**

**32-бітна ОС**: Має обмеження щодо адресації пам'яті – вона може використовувати максимум близько **4 ГБ оперативної пам'яті**. Навіть якщо встановлено більше пам'яті, система використовуватиме лише 4 ГБ.

**64-бітна ОС**: Підтримує значно більший обсяг пам'яті, який може досягати десятків терабайтів (залежить від ОС та її версії). Це дозволяє використовувати повністю всю встановлену оперативну пам'ять.

**Сумісність з програмним забезпеченням**

**32-бітна ОС**: Підтримує лише 32-бітні програми. Вони також можуть працювати на 64-бітних системах (якщо ОС має відповідну підтримку), але не використовуватимуть всі можливості 64-бітної архітектури.

**64-бітна ОС**: Підтримує як 64-бітні, так і більшість 32-бітних програм (через емуляцію або сумісні шари).

**Драйвери та обладнання**

**32-бітна ОС**: Використовує 32-бітні драйвери. Якщо пристрої не мають 32-бітних драйверів, їхнє використання буде неможливим.

**64-бітна ОС**: Вимагає 64-бітні драйвери для всіх пристроїв. Тому для старого обладнання, для якого немає 64-бітних драйверів, можуть виникнути проблеми з сумісністю.**Ефективність**

**32-бітна ОС**: Менш ефективна при обробці великих обсягів даних або ресурсомістких завдань, таких як редагування відео, робота з великими базами даних тощо.

**64-бітна ОС**: Забезпечує вищу продуктивність при виконанні таких завдань, оскільки може обробляти більше даних за один такт і працювати з більшою оперативною пам'яттю.

**2.3 Які основні етапи при встановленні OS Linux в текстовому режимі?**

**2.4\*Яким чином можна до установити графічні оболонки Gnome та KDE в Linux, якщо вона вже встановлена в текстовому режимі (вкажіть необхідні команди та пакети)?**

**2.5\*\*Дайте коротку характеристику графічних інтерфейсів, що використовуються в різних дистрибутивах Linux відповідно до свого варіанту (порядковий номер по журналу), табл.2..**

Лобода Д.Р. (10)

### KDE (K Desktop Environment)

KDE — це потужний та функціональний графічний інтерфейс для дистрибутивів Linux. Основні характеристики:

* Панель інструментів: Має схожу на Windows панель з меню, кнопками швидкого запуску та віджетами.
* Віджети: Можливість додавання різноманітних віджетів на робочий стіл для розширення функціональності.
* Темізація: Можливість налаштування зовнішнього вигляду через теми, значки та кольори.
* Налаштування: Розширені налаштування для всіх аспектів інтерфейсу, включаючи управління вікнами та ефектами.
* Продуктивність: Відносно ресурсомісткий через велику кількість функцій та можливостей.

### Fluxbox

Fluxbox — це легкий і простий графічний інтерфейс для Linux. Основні характеристики:

* Мінімалізм: Прості та ефективні елементи без зайвих функцій.
* Кастомізація: Можливість значної налаштування через конфігураційні файли, але без графічного інтерфейсу для налаштувань.
* Швидкість: Низьке споживання ресурсів, що робить його підходящим для старих або малопотужних систем.
* Панель інструментів: Може бути налаштована з базовими функціями, такими як запуск програм і перемикання вікон.

**Відповіді на контрольні запитання**

1. **Порівняйте гіпервізори типу 1 та типу 2, яка між ними відмінність та сфера їх застосування? (Лобода Д.Р.)**

Гіпервізори (або віртуалізатори) поділяються на два основних типи — гіпервізор типу 1 та гіпервізор типу 2.

Розглянемо їх ключові відмінності та сферу застосування.

Гіпервізор типу 1 (Bare-metal): Гіпервізори типу 1 працюють на фізичному обладнанні (сервері), тобто не потребують операційної системи для своєї роботи. Вони взаємодіють напряму з апаратними ресурсами і керують віртуальними машинами на рівнях "заліза".

Приклад: Microsoft Hyper-V, VMware ESXi, Xen.

Переваги: Висока продуктивність: Хоча працює з апаратним забезпеченням, гіпервізор типу 1 має менші затримки та кращу продуктивність.

Безпека: Менша кількість компонентів знижує ймовірність уразливостей, тому такі гіпервізори вважаються більш безпечними. - Висока ефективність масштабування: Використовується в корпоративних середовищах і центрах даних для хмарних і серверних рішень.

Сфера застосування: традиційне використання у великих інфраструктурах, де потрібно віртуалізувати багато серверів та забезпечити високу доступність ресурсів (хмарні платформи, великі компанії, дата-центри).

Гіпервізор типу 2 (Hosted): Гіпервізори типу 2 працюють над операційною системою (гостової ОС), яка вже встановлена ​​на фізичному комп'ютері. Вони забезпечують віртуалізацію всередині цієї ОС, і віртуальні машини запускаються в межах середовища операційної системи.

Приклад: VMware Workstation, Oracle VirtualBox, Parallels Desktop.

Переваги: - Простота використання: Легше встановлювати і на аналізувати, потім не потребує попередньої установки на чисте обладнання. - Гнучкість: Може використовуватися для запуску віртуальних машин на робочих станціях і звичайних ПК, що робить його зручним для розробників і тестувальників. - Сумісність: Підтримка багатьох операційних систем і широких можливостей для інтеграції з існуючою ОС.

Сфера застосування: Використовується переважно в домашніх або офісних умовах для тестування програмного забезпечення, розробки або навчання. Застосовується там, де високопродуктивна віртуалізація не є критичною.

Основні відмінності:

Рівень роботи: - Тип 1 працює на фізичному апаратному забезпеченні. - Тип 2 працює поверх хостової операційної системи.

Продуктивність: - Тип 1 має вищу продуктивність за рахунок прямого доступу до апаратних ресурсів. - Тип 2 менш продуктивний, залишається працювати через хостову ОС.

1. **Розкрийте поняття «GNU GPL», яка його основна концепція?**
2. **В чому суть програмного забезпечення з відкритим кодом?**
3. **Що таке дистрибутив? (Лобода Д.Р.)**

Дистрибутив — це готовий комплект операційної системи, що включає ядро (часто Linux), набір системних утиліт, додатків і драйверів. Він дозволяє користувачам легко встановлювати і налаштовувати операційну систему.

1. **Які задачі системного адміністрування можна реалізувати на базі ОС Linux?**
2. **Як пов'язані між собою ОС Android та Linux?**
3. **Основні можливості та сфера використання Embedded Linux? (Лобода Д.Р.)**

Embedded Linux — це спеціалізована версія Linux, оптимізована для вбудованих систем, таких як побутові прилади, автомобільна електроніка, промислові контролери тощо.

### Основні можливості:

1. Linux підтримує багато архітектур, таких як ARM, MIPS, x86.
2. Малий розмір і висока продуктивність — адаптоване ядро та мінімальний набір програм для роботи на пристроях з обмеженими ресурсами (пам'ять, процесор).
3. Гнучкість і масштабованість — можливість налаштувати систему під конкретні потреби пристрою.
4. Підтримка різноманітного апаратного забезпечення — Відкритий код — дозволяє розробникам змінювати та налаштовувати систему відповідно до вимог.
5. Реальний час — можливість інтеграції патчів для забезпечення виконання завдань у реальному часі (наприклад, в промислових системах).

### Сфера використання:

1. Промислові пристрої — контролери для автоматизації, сенсори, IoT-пристрої.
2. Автомобільна електроніка — системи управління автомобілем, інформаційно-розважальні комплекси.
3. Побутова техніка — телевізори, пральні машини, холодильники зі смарт-функціями.
4. Медичні прилади — монітори, діагностичне обладнання.
5. Телекомунікаційні пристрої — маршрутизатори, комутатори, базові станції.

Embedded Linux широко застосовується завдяки його відкритості, надійності та можливості тонкого налаштування під конкретне апаратне забезпечення.

1. **Яким чином можна змінити типу завантаження Linux: в текстовому режимі (3 рівень) або графічному (рівень 5)? Чим відрізняються режими CLI та GUI? (Лобода Д.Р.)**

Щоб змінити тип завантаження Linux (текстовий режим — рівень 3, або графічний — рівень 5), можна скористатися різними способами. Основні дії залежатимуть від вашого дистрибутиву та його конфігурації.

Як змінити режим завантаження:

1. Тимчасова зміна (при завантаженні):

- На етапі завантаження системи відкрийте меню GRUB.

- Виберіть необхідний пункт завантаження і натисніть `e` для редагування.

- У параметрі ядра (рядок, який починається з `linux` або `linux16`) додайте `3` для текстового режиму або `5` для графічного перед `quiet` або в кінці рядка.

- Натисніть `Ctrl + X` для завантаження з новими параметрами.

2. Постійна зміна режиму через systemd (для сучасних дистрибутивів):

- Відкрийте файл конфігурації `/etc/default/grub` у текстовому редакторі (наприклад, `nano`):

```bash

sudo nano /etc/default/grub

```

- Знайдіть рядок, який починається з `GRUB\_CMDLINE\_LINUX`, і додайте `3` для текстового режиму або `5` для графічного:

```bash

GRUB\_CMDLINE\_LINUX="... 3" # для текстового режиму

```

- Оновіть налаштування GRUB:

```bash

sudo update-grub # для Debian/Ubuntu

sudo grub2-mkconfig -o /boot/grub2/grub.cfg # для RHEL/CentOS

```

- Перезавантажте систему.

3. Зміна через systemd target:

- Для переходу в текстовий режим використовуйте:

```bash

sudo systemctl set-default multi-user.target

```

- Для переходу в графічний режим:

```bash

sudo systemctl set-default graphical.target

```

- Ця команда змінює режим завантаження на постійній основі.

Відмінність між CLI (Command-Line Interface) та GUI (Graphical User Interface):

1. CLI (Command-Line Interface):

* Інтерфейс: Взаємодія відбувається через текстову консоль, де користувач вводить команди для виконання завдань.
* Ресурси: Використовує мінімум системних ресурсів, оскільки не потребує графічного інтерфейсу.
* Гнучкість: Більше можливостей для виконання складних операцій через командний рядок.
* Сфера застосування: Адміністрування системи, сервери, автоматизація, скрипти, робота в середовищах з обмеженими ресурсами.

2. GUI (Graphical User Interface):

* Інтерфейс: Графічне середовище, де взаємодія з системою відбувається через вікна, кнопки, меню та інші елементи.
* Ресурси: Потребує більше ресурсів, оскільки запускає графічні середовища (наприклад, GNOME, KDE).
* Зручність: Інтуїтивний та легкий у використанні для новачків і непрофесіоналів.
* Сфера застосування: Робочі станції, персональні комп'ютери, системи, де потрібна взаємодія з користувачем.

Таким чином, CLI забезпечує більшу ефективність і контроль, але потребує знань команд, тоді як GUI робить систему більш дружньою для користувачів, але вимагає більше ресурсів.

**Висновок**

Ознайомлення з робочим середовищем віртуальних машин та операційних систем різних сімейств підкреслює важливість розуміння різних платформ і інструментів, які використовуються для віртуалізації та управління операційними системами. Віртуальні машини (VMs) надають потужний інструмент для ізоляції та безпечного тестування різних операційних систем, що дозволяє користувачам і адміністраторам зменшити витрати на фізичне обладнання та підвищити ефективність інфраструктури. Операційні системи різних сімейств (Windows, Linux, macOS) забезпечують різні рівні функціональності та інтерфейсів, тому розуміння їх особливостей і управління віртуальним середовищем є ключовими аспектами для сучасного ІТ-фахівця.