“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

ЗВІТ ПО ВИКОНАННЮ ЛАБОРАТОРНОЇ РОБОТИ №1

з дисципліни: «Операційні системи»

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

Виконали студенти групи Ксм-13а Команда Бюджетники;

Топчій А.С.; Топехін Б.А.; Чичкань Д.С.;

Перевірив викладач Сушанова В.С.

Київ 2023

# 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 7).
3. Віртуальна машина – Virtual Box (Oracle).
4. Операційна система GNU/Linux – CentOS.

# Завдання для попередньої підготовки

## Готував матеріал студент Чичкань Д.С.

* 1. Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеличкий словник базових англійських термінів з питань класифікації ОС.

|  |  |
| --- | --- |
| Термін англійською | Термін українською |
| Operating System | Операційна система |
| shared hosting | Спільний хостинг |
| virtual machine | віртуальна машина |
| separate computers | окремі комп'ютери |
| interchangeably | взаємозамінно |
| binary translation | двійковий переклад |
| proprietary code | власний код |
| underlying | лежить в основі |
| memorizing | запам'ятовування |

* 1. Прочитавши матеріал з коротких теоретичних відомостей дайте відповіді на наступні питання:

## Готував матеріал студент Чичкань Д.С.

* + 1. Охарактеризуйте поняття «гіпервізор». Які бувають їх типи?

Гіпервізор (VMM або Virtual Machine Monitor) - це програмна або апаратна система, яка дозволяє віртуалізувати апаратне обладнання і створювати віртуальні машини. Існують

два типи гіпервізорів:

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

Основні компоненти та можливості гіпервізора Hyper-V:

Компоненти:

Hypervisor: Основний компонент, що дозволяє створювати та керувати віртуальними машинами.

Virtual Machines (VMs): Віртуальні машини, ізольовані віртуальні середовища.

Hyper-V Manager: Інструмент для управління віртуальними машинами.

Integration Services: Покращує взаємодію між гостьовими операційними системами та гіпервізором.

Virtual Switch: Налаштовує віртуальні мережі.

Можливості:

Спільне використання ресурсів: Ефективне використання процесорів та пам'яті між віртуальними машинами.

Міграція віртуальних машин: Різні види міграції, включаючи міграцію в реальному часі.

Масштабованість: Можливість розгортати та керувати великою кількістю віртуальних машин.

Безпека: Засоби захисту віртуальних машин та ресурсів.

Підтримка гостьових операційних систем: Підтримка різних ОС.

Автоматизація та управління: Засоби для автоматизації та управління великими фермами віртуальних машин.

# Хід роботи

## Готував матеріал студент Чичкань Д.С.

1. Робота в графічному режимі в ОС сімейства Linux:.
   1. Запустіть віртуальну машину VirtualBox, ознайомтесь з її основними можливостями, прочитайте довідку по роботі з нею.

# Відповіді на контрольні запитання

## Готував матеріал студент Топехін Б.А.

1. Розкрийте поняття «GNU GPL», яка його основна концепція??

GNU GPL розшифровується як "GNU General Public License". Основна суть GNU GPL полягає в тому, що це ліцензія на вільне програмне забезпечення, яка визначає правила використання, модифікації і поширення програм, що захищають права користувачів і забезпечують вільну доступність програмного коду.

* + 2.1.Для розгортання операційної системи на базі віртуальної машини VirtualBox слід виконати такі основні етапи:

Встановлення VirtualBox:

Почніть з встановлення VirtualBox на вашому фізичному комп'ютері (господаря). Ви можете завантажити VirtualBox з офіційного веб-сайту Oracle і встановити його, виконавши інструкції для вашої операційної системи.

Завантаження образу операційної системи:

Завантажте образ операційної системи, яку ви хочете встановити на віртуальну машину. Це може бути ISO-файл для інсталяції операційної системи або інший віртуальний диск (VDI, VMDK тощо).

Створення нової віртуальної машини:

Відкрийте VirtualBox і натисніть кнопку "Нова" або "Створити віртуальну машину".

Дайте машині ім'я і виберіть тип та версію операційної системи, яку ви будете встановлювати.

Налаштування параметрів віртуальної машини:

Встановіть обсяг оперативної пам'яті (RAM), яку виділите віртуальній машині. Створіть віртуальний жорсткий диск та виберіть його розмір.

Виберіть тип віртуального жорсткого диску (VDI, VMDK тощо) та налаштуйте його параметри.

Налаштування параметрів системи:

У віртуальний диспетчер VirtualBox перейдіть до налаштувань віртуальної машини.

В розділі "Загальні" встановіть параметри загальної конфігурації, такі як пристрій завантаження і порядок завантаження.

Завантаження віртуальної машини з образу операційної системи:

У розділі "Збережені відображення" виберіть образ операційної системи, який ви завантажите на віртуальну машину.

Запустіть віртуальну машину, натиснувши кнопку "Пуск". Інсталяція операційної системи:

Віртуальна машина запуститься з обраного образу операційної системи, і ви можете розпочати процес інсталяції, слідуючи вказівкам на екрані.

Встановіть операційну систему так, як ви це робили б на звичайному фізичному комп'ютері.

Настройка додаткових параметрів (за бажанням):

Після інсталяції операційної системи можна додатково налаштувати параметри віртуальної машини, такі як мережеві налаштування, додаткові пристрої, снапшоти та інші.

Завершення і використання віртуальної машини:

Після налаштування інстальованої операційної системи ви можете користуватися віртуальною машиною так, як будь-яким фізичним комп'ютером. Ці етапи дозволяють вам створити і налаштувати віртуальну машину з операційною системою в середовищі VirtualBox для виконання різних завдань, таких як розробка, тестування або віддалена робота.

* + 2.2. Так, є апаратні обмеження при встановленні 32- і 64-бітних операційних систем:

Процесор (CPU): Для встановлення 64-бітної операційної системи потрібен 64-бітний процесор. 32-бітні операційні системи можна встановлювати як на 32-бітних, так і на 64-бітних процесорах.

Оперативна пам'ять (RAM): 64-бітні операційні системи можуть краще використовувати більший обсяг оперативної пам'яті (більше 4 гігабайт). Для оптимального використання 64-бітної ОС може знадобитися більше оперативної пам'яті.

Підтримка 64-бітних інструкцій: Ваш процесор повинен підтримувати 64-бітні інструкції для встановлення та використання 64-бітної операційної системи.

Більшість сучасних процесорів це роблять, але старіші моделі можуть бути обмежені в цьому відношенні.

Біос (UEFI/BIOS): Деякі біоси (у сучасних комп'ютерах використовуються UEFI) можуть мати обмеження на встановлення або завантаження 64-бітних операційних систем. У таких випадках вам може знадобитися оновити біос або вибрати 32-бітну ОС.

* 1. Основні етапи встановлення CentOS в текстовому режимі

1. Завантаження зі зберіженого носія або мережі.
2. Вибір мови і локалізації.
3. Вибір режиму встановлення (текстовий режим).
4. Налаштування мережі (за бажанням).
5. Налаштування сховища та розділів.
6. Вибір груп пакунків для встановлення.
7. Налаштування користувача та пароля.
8. Налаштування графічного сервера (за бажанням).
9. Перевірка конфігурації.
10. Завершення встановлення та перезавантаження.
    1. Встановлення графічного інтерфейсу GNOME; 1.GNOME

sudo yum groupinstall "Server with GUI" -y

Після завершення встановлення перезавантажте систему: sudo reboot

Після перезавантаження GNOME буде доступний як графічний інтерфейс. 2.KDE:

Встановлення графічного інтерфейсу KDE:

sudo yum groupinstall "KDE Plasma Workspaces" -y

Після завершення встановлення перезавантажте систему: sudo reboot

Після перезавантаження KDE буде доступний як графічний інтерфейс.

Оберіть одну з цих команд, відповідно до вашого вибору (GNOME або KDE), та виконайте її в командному рядку. Після перезавантаження ви зможете вибрати бажаний графічний інтерфейс при вході в систему.

* + 1. Xfce:

Легкий та швидкий графічний інтерфейс для Linux та Unix. Призначений для роботи на комп'ютерах з обмеженими ресурсами. Модульний і налаштовується, з можливістю вибору компонентів.

Дружній до користувача інтерфейс, схожий на класичний робочий стіл.

* + 1. FVWM (F Virtual Window Manager):

Надзвичайно настраюваний графічний інтерфейс для Linux та Unix.

Дозволяє користувачам налаштовувати практично всі аспекти інтерфейсу.

Легкий на ресурси і працює на старих комп'ютерах.

Має великі можливості для створення незвичайних інтерфейсів.

## Готував матеріал студент Топчій А.С.

Контрольні запитання

1.

Гіпервізори типу 1:

* Запускаються на "голому металі": Робота безпосередньо на апаратному рівні без операційної системи хоста.
* Висока продуктивність і ефективність: Зазвичай надають кращу продуктивність і ресурсоємність.
* Сфера застосування: Великі дата-центри, хмарні інфраструктури, високонавантажені сервери.

Гіпервізори типу 2:

* Запускаються в операційній системі хоста: Робота як додаток у вже існуючій операційній системі.
* Зазвичай менш продуктивні: Можуть бути менш ефективними у використанні ресурсів через додатковий шар операційної системи.
* Сфера застосування: Розробка, тестування, віртуалізація на робочих станціях, локальні тестові середовища.

2.

* + - Вільна розповсюдження: Кожен має право копіювати, поширювати та передавати програмне забезпечення, включаючи його внесок у відкритий код.
    - Право доступу до вихідного коду: Ліцензія вимагає, щоб вихідний код програмного забезпечення був доступним для всіх, хто отримує програму. Це дозволяє користувачам розуміти, аналізувати та модифікувати код.
    - Право модифікації: Ви маєте право змінювати та модифікувати програмне забезпечення, якщо ви використовуєте його.
    - Вимога збереження ліцензії: Якщо ви розповсюджуєте або публікуєте модифіковану версію програмного забезпечення, ви зобов'язані зробити це за умовами GNU GPL, тобто зберегти вільну ліцензію для цієї версії.
    - Захист від патентів: Ліцензія містить положення, що захищає користувачів від патентних позовів щодо використання програмного забезпечення.
    - Відсутність дискримінації: Ліцензія не дозволяє обмеження за релігійними, політичними або особистими переконаннями користувачів.

Головна мета GNU GPL - захист вільності програмного забезпечення та забезпечення можливості спільного розвитку та спільного використання вільних технологій та ресурсів.

3.

* + - * Вільний доступ до вихідного коду: Вихідний код програми відкритий і доступний для всіх.
      * Вільність використання, модифікації та розповсюдження: Користувачі можуть використовувати, змінювати та поширювати програму за власними потребами.
      * Спільнота розробників і користувачів: Утворюється спільнота, яка спільно працює над вдосконаленням програми та надає підтримку.
      * Безкоштовність: Більшість програм з відкритим кодом є безкоштовними або мають безкоштовну альтернативу.

4. Дистрибутив - це комплект програмного забезпечення, який включає в себе операційну систему та інші компоненти, призначені для встановлення і використання на комп'ютері або сервері. Дистрибутиви дозволяють користувачам швидко і зручно встановлювати та налаштовувати операційні системи і програми. Їх можна використовувати для створення робочих станцій, серверів, спеціалізованих систем і багато іншого. Найвідомішими прикладами дистрибутивів є Linux-дистрибутиви, такі як Ubuntu, Fedora, CentOS і Debian.

5.

1. Управління користувачами і групами: Створення, видалення та налаштування облікових записів користувачів і груп.
2. Керування правами доступу: Налаштування прав доступу до файлів і каталогів для забезпечення безпеки даних.
3. Моніторинг системи: Відстеження ресурсів, включаючи використання процесора, пам'яті і мережі, для виявлення проблем і оптимізації продуктивності.
4. Забезпечення безпеки: Встановлення заходів безпеки, оновлення системи та моніторинг на предмет потенційних загроз.
5. Керування пакунками і оновленнями: Встановлення, оновлення та видалення програмних пакетів.
6. Створення резервних копій: Регулярне створення резервних копій системи і даних для забезпечення їхньої безпеки.
7. Керування мережею: Налаштування мережевих інтерфейсів, маршрутизація, налаштування файрволу і інше.
8. Віртуалізація: Управління віртуальними машинами та контейнерами для оптимізації ресурсів і розгортання додатків.
9. Автоматизація завдань: Створення скриптів і автоматичних завдань для спрощення та автоматизації рутинних процесів.
10. Налаштування серверів: Установка, налаштування та управління серверами, такими як веб-сервери, бази даних і поштові сервери.
11. Вирішення проблем і підтримка: Виявлення і вирішення проблем у системі та надання підтримки користувачам.

6. Операційна система Android та ядро Linux мають тісну взаємозв'язок:

* + Ядро Linux: Android базується на ядрі Linux. Це означає, що основною операційною системою, яка керує апаратурою пристрою, є Linux. Ядро Linux забезпечує роботу з апаратурою, драйвери пристроїв, планування завдань і багато іншого.
  + Відкритий код: Ядро Linux і Android - це обидва вільне та відкрите програмне забезпечення. Це означає, що користувачі та розробники можуть переглядати, змінювати і розповсюджувати їхній вихідний код.
  + Спільні компоненти: Android використовує багато компонентів і бібліотек з Linux, таких як бібліотеки C, мережеві стеки і файлові системи.
  + Спільні принципи безпеки: Ядро Linux і Android використовують спільні принципи безпеки, такі як управління правами доступу і ізоляція процесів.
  + Розробка на основі Linux: Розробка додатків для Android відбувається на базі Linux-систем, що спрощує процес розробки та тестування.

7.

Основні можливості:

* + Відкритий код і спільнота розробників: Embedded Linux базується на вільному та відкритому коді, що дозволяє розробникам вільно використовувати, змінювати і розповсюджувати його.
  + Підтримка для різних архітектур: Embedded Linux підтримує різні архітектури, що робить його придатним для використання на різних вбудованих платформах, таких як ARM, x86, MIPS і інші.
  + Низькі вимоги до ресурсів: Embedded Linux може працювати на пристроях з обмеженими обсягами пам'яті і обчислювальної потужності.
  + Модульність і налаштовуваність: Вбудовані системи на базі Linux можуть бути налаштовані та збудовані з різними компонентами залежно від потреби, що дозволяє оптимізувати розмір і функціональність системи.

Сфера використання:

* + Мобільні пристрої: Embedded Linux використовується в смартфонах, планшетах, смарт-годинниках і інших мобільних пристроях.
  + Мережеві пристрої: Вбудовані системи на базі Linux застосовуються у мережевих роутерах, комутаторах, маршрутизаторах, мережевих серверах і ін.
  + Вбудовані системи в автомобілях: Embedded Linux використовується в автомобільних системах навігації, мультимедіа, діагностики і управління.
  + Промислові пристрої: Вбудовані системи застосовуються в промисловому обладнанні, такому як контролери, автоматизовані системи, промислові комп'ютери і монітори.
  + Медичні пристрої: Embedded Linux використовується в медичних пристроях, таких як ЕКГ, УЗД, медичні монітори і обладнання для діагностики.
  + Смарт-телевізори і приставки: Embedded Linux використовується в телевізорах і приставках для стрімінгу мультимедійного контенту.

1. K
   1. Зміна рівня завантаження під час завантаження:

* Під час завантаження Linux, ви можете вибрати рівень завантаження або "runlevel" на екрані завантаження, зазвичай використовуючи клавішу "е для редагування параметрів завантаження і додавання "3° для текстового режиму або 5 ° для графічного режиму.
  1. За допомогою команди командного рядка (CLI):
* В текстовому режимі (CLI) ви можете ввести команду, таку як 'systemctl set-

default multi-user. target\* для перемикання на текстовий режим.

* В графічному режимі (GUI) ви можете ввести команду, таку як 'systemetl set-default graphical.target для перемикання на графічнии режим.

Embedded Linux є популярним вибором для вбудованих систем завдяки своїм можливостям, гнучкості і великій спільноті розробників.

**Висновки**

В ході виконання лабораторної роботи мною було досліджено різні аспекти операційних систем і програмного забезпечення, більш детально теоретично досліджено питання відмінностей між режимами CLI та GUI. Отримано практичні навики роботи з командами у Linux, налаштування віртуальних машин у гіпервізорі VirtualBox. Ці навички будуть корисні у подальших дослідженнях і роботі з інформаційними технологіями.