**Work-case 2**

Ланковський Ерік

**1. Клонування віртуальної робочої ОС (Work-case 2):**

Процес клонування залежить від використовуваного програмного забезпечення для віртуалізації (наприклад, VirtualBox, VMware). Загальний алгоритм дій наступний:

1. **Вимкніть віртуальну машину:** Переконайтесь, що віртуальна машина, яку ви хочете клонувати, повністю вимкнена.
2. **Знайдіть файл віртуальної машини:** Знайдіть папку, де зберігаються файли вашої віртуальної машини. Зазвичай це папка з розширенням .vbox (VirtualBox) або .vmx (VMware).
3. **Скопіюйте папку:** Скопіюйте всю папку з файлами віртуальної машини в нове місце. Це буде ваш клон.
4. **Зареєструйте клон (за необхідності):** Якщо ви використовуєте VirtualBox або VMware, вам може знадобитися "зареєструвати" клон віртуальної машини в програмі віртуалізації. Зазвичай це робиться через меню "Машина" -> "Додати" або "Зареєструвати".
5. **Запустіть клон:** Запустіть клоновану віртуальну машину. Можливо, вам потрібно буде змінити деякі налаштування, такі як ім'я машини та MAC-адреса мережевого адаптера, щоб уникнути конфліктів з оригінальною машиною.

**Експорт віртуальної робочої ОС:**

Для експорту віртуальної машини також використовуються інструменти віртуалізації. Зазвичай це робиться через меню "Файл" -> "Експорт" або "Export Appliance". Виберіть формат експорту (наприклад, OVA, OVF) та вкажіть місце збереження. Цей файл можна потім імпортувати в інше віртуальне середовище.

**2. Типи організації мережевих з’єднань:**

* **Трансляція мережевих адрес (NAT):** Віртуальна машина отримує доступ до Інтернету через IP-адресу хост-машини. Машина "бачить" зовнішній світ під IP-адресою хоста. Інші машини в мережі не можуть напряму звернутися до віртуальної машини. Підходить для випадків, коли віртуальній машині потрібен доступ до Інтернету, але не потрібно, щоб до неї зверталися ззовні.
* **Мережевий міст (Bridged):** Віртуальна машина отримує власну IP-адресу в тій же мережі, що й хост-машина. Вона "бачиться" в мережі як окремий пристрій. Підходить для випадків, коли віртуальна машина повинна бути доступна з інших машин в мережі.
* **Віртуальний адаптер хоста (Host-only):** Віртуальна машина створює власну віртуальну мережу, до якої підключена тільки хост-машина. Віртуальна машина не має доступу до Інтернету, якщо хост-машина не налаштована як маршрутизатор. Підходить для ізольованих мереж, де віртуальні машини повинні взаємодіяти тільки між собою та хост-машиною.
* **Внутрішня мережа (Internal Network):** Створюється ізольована мережа тільки між віртуальними машинами. Хост-машина не має доступу до цієї мережі. Підходить для випадків, коли потрібно створити окрему мережу для віртуальних машин, незалежну від хост-машини та зовнішнього світу.

**3. Розгортання мережі між віртуальними машинами:**

Для з'єднання клонованої та оригінальної віртуальних машин найкраще підійде режим "Мережевий міст" (Bridged).

1. **Налаштування мережевих параметрів ОС:**
   * **Linux:**
     + ip addr show: Показує інформацію про мережеві інтерфейси.
     + ifconfig <інтерфейс> <IP-адреса> netmask <маска\_мережі>: Призначає IP-адресу та маску мережі для інтерфейсу.
     + route add default gw <шлюз>: Встановлює шлюз за замовчуванням.
     + nameserver <DNS\_сервер>: Встановлює DNS-сервер.
   * **Windows:**
     + ipconfig: Показує інформацію про мережеві інтерфейси.
     + netsh interface ip set address name="<інтерфейс>" static <IP-адреса> <маска\_мережі> <шлюз> 1: Призначає статичну IP-адресу.
2. **Доступ до Інтернету:**

Після налаштування IP-адрес та шлюзу, обидві віртуальні машини повинні мати доступ до Інтернету. Відкрийте браузер та перевірте з'єднання.

1. **Обмін повідомленнями:**
   * **Linux/Windows:**
     + ping <IP-адреса\_іншої\_машини>: Перевіряє з'єднання з іншою машиною.
     + traceroute <IP-адреса\_іншої\_машини>: Показує маршрут до іншої машини.
     + net send <ім'я\_машини> <повідомлення> (Windows, застаріла команда): Відправляє повідомлення на іншу машину. Для сучасних систем краще використовувати інші засоби, такі як месенджери.
2. **Спільна мережева папка:**

Налаштування спільної папки залежить від операційної системи та програмного забезпечення віртуалізації. Зазвичай це робиться через налаштування "Спільні папки" в програмі віртуалізації. Після налаштування папка буде доступна з обох віртуальних машин. Скопіюйте файли за допомогою звичайних команд копіювання (cp в Linux, copy в Windows).

**4. Обмін інформацією між основною та віртуальними ОС:**

Найпростіший спосіб обміну файлами - це використання спільних папок, налаштованих в програмі віртуалізації. Ви можете створити спільну папку на основній ОС, яка буде доступна для віртуальних машин. Скопіюйте аудіофайл до цієї папки, і він буде доступний на робочому столі віртуальних машин. Аналогічно, для копіювання файлу з віртуальної машини на основну ОС, скопіюйте його до спільної папки.

## Практичне завдання “Work-case 2”

**Створення нової віртуальної машини:**

**VBoxManage createvm:** Ця команда використовується для створення нової віртуальної машини.

**--name <name>:** Задає ім'я віртуальної машини.

**--ostype <OS>:** Визначає тип операційної системи, яку буде встановлено на віртуальній машині.

**--register:** Реєструє віртуальну машину в VirtualBox.

**VBoxManage createvm --name "MyVM" --ostype "Linux" --register**

**Вибір/додавання доступного для віртуальної машини обладнання:**

**VBoxManage modifyvm:** Ця команда використовується для зміни налаштувань віртуальної машини, включаючи додавання обладнання.

**--memory <size>:** Встановлює об'єм оперативної пам'яті для віртуальної машини.

**--cpus <count>:** Встановлює кількість процесорів для віртуальної машини.

**--hda <path>:** Додає віртуальний жорсткий диск до віртуальної машини.

**--nic<number> <type>:** Додає мережевий адаптер до віртуальної машини.

**VBoxManage modifyvm "MyVM" --memory 2048 --cpus 2 --hda "path/to/disk.vdi" --nic1 bridged**

**Налаштування мережі та підключення до точок Wi-Fi:**

**VBoxManage modifyvm:** Як зазначено вище, ця команда також використовується для налаштування мережі.

**--nic<number> <type>:** Визначає тип мережевого підключення (наприклад, bridged, NAT, host-only).

**--bridgeadapter<number> <adapter>:** Для bridged-з'єднання, вказує мережевий адаптер, який буде використано для підключення до мережі.

**VBoxManage modifyvm "MyVM" --nic1 bridged --bridgeadapter1 "Wi-Fi Adapter"**

**Можливість роботи з зовнішніми носіями (flash-пам'ять):**

**--vmname <name>:** Вказує ім'я віртуальної машини.

**--storagectl <name>:** Вказує контролер зберігання.

**--port <port>:** Вказує порт підключення.

**--device <device>:** Вказує номер пристрою.

**--type <type>:** Вказує тип носія.

**--medium <path>:** Вказує шлях до файлу.

**VBoxManage storageattach "MyVM" --storagectl "IDE Controller" --port 1 --device 0 --type hdd --medium /dev/sdb**

Порівняння FluxBox та Gnome.

|  |  |  |
| --- | --- | --- |
| Feature | Fluxbox | GNOME |
| Тип | Window Manager | Desktop Environment |
| Ресурси | Дуже легкий, споживає мало ресурсів | Важкий, споживає багато ресурсів |
| Швидкість | Швидкий та чуйний | Може бути повільним на слабкому обладнанні |
| Налаштування | Висока гнучкість налаштування через текстові файли конфігурації | Менш гнучкий, налаштування через графічний інтерфейс |
| Зовнішній вигляд | Мінімалістичний, простий | Сучасний, з різноманітними візуальними ефектами |
| Функціональність | Базова функціональність, фокус на керуванні вікнами | Повноцінне середовище з багатьма вбудованими програмами |
| Простота використання | Потребує певних знань для налаштування | Інтуїтивно зрозумілий для новачків |
| Встановлення | Зазвичай встановлюється окремо | Часто встановлюється за замовчуванням в дистрибутивах Linux |
| Підтримка | Менша спільнота, менше документації | Велика спільнота, багато документації та підтримки |
| Програми | Не включає в себе багато програм, користувач сам обирає потрібні | Має великий набір вбудованих програм для різних потреб |
| Оновлення | Оновлюється рідше | Оновлюється часто, з новими функціями та покращеннями |
| Висновок | Ідеально підходить для користувачів, які цінують швидкість та гнучкість | Підходить для користувачів, які хочуть сучасне та функціональне середовище |

**Висновки:**

Виконавши цю роботу, я отримав глибше розуміння принципів роботи з командним інтерфейсом **Bash** та його можливостей. Для мене було особливо цікаво дізнатися про:

* Різноманітність команд: Існує безліч команд для виконання різноманітних завдань, від навігації файловою системою до управління процесами та налаштування системи.
* Змінні оточення: Змінні оточення відіграють ключову роль у налаштуванні поведінки програм та процесів. Особливо цікавим було дізнатися про змінну $PS1 та її вплив на вигляд рядка запрошення.
* Гнучкість налаштування: Bash надає широкі можливості для налаштування під свої потреби, включаючи створення псевдонімів, функцій та зміну значення змінних оточення.
* Інструкції керування: Інструкції керування дозволяють створювати складні конструкції для автоматизації виконання завдань та управління потоком виконання програм.
* Робота з файлами: Команди whereis та locate є потужними інструментами для пошуку файлів та інформації про команди.

Ця робота допомогла мені не тільки ознайомитися з основними командами та поняттями Bash, але й зрозуміти, як використовувати їх для вирішення практичних завдань. Я також навчився користуватися документацією та шукати необхідну інформацію в інтернеті.

В цілому, я вважаю, що Bash є незамінним інструментом для кожного, хто хоче ефективно працювати з операційною системою Linux. Розуміння принципів роботи з Bash відкриває широкі можливості для автоматизації завдань, налаштування системи та управління нею.

Я впевнений, що отримані знання та навички будуть корисними мені в подальшому навчанні та роботі.