Задання для дистанційного навчання група №29 предмет:
“Операційні системи та їх обслуговування”
тема: “Програми-емулятори ПК”
Викладач: Миронова О.Ю.

Завдання.

1. Ознайомтеся з теоретичними відомостями, що наведені нижче.
2. Законспектуйте основні поняття.
3. Виконайте практичну роботу:
   - ✓ Перегляньте навчальне відео з технологічною послідовністю встановлення програми);
   - ✓ Завантажте програму з офіційного сайту VirtualBox **(**посилання для завантаження https://www.virtualbox.org/wiki/Downloads або https://virtualbox.ru.uptodown.com/windows);
   - ✓ Встановіть програму - емулятор на ваш комп’ютер;
   - ✓ Завантажте образ операційної системи Linux Mint з офіційного сайту (https://linuxmint.com/edition.php?id=281);
   - ✓ Встановіть операційну систему Linux Mint на віртуальну машину.
4. Надішліть звіт, у вигляді презентації або відеофайлу на електронну адресу olena_mironova@yahoo.com, до 13.10.20р.

Призначення та використання програм-емуляторів ПК.

Віртуальна машина - це конкретний екземпляр деякого обчислюваного середовища ("віртуального комп'ютера"), створеного за допомогою спеціального програмного інструменту. Зазвичай такі інструменти дозволяють створювати і запускати довільну кількість віртуальних машин, яка обмежується лише фізичними ресурсами реального комп'ютера.

Інструмент для створення віртуальної машини - це звичайна програма, яка встановлюється на реальну операційну систему. Ця реальна операційна система називається "головною", або *хостовою*, ОС.

Всі задачі по управлінню віртуальними машинами виконує спеціальний модуль в складі програми віртуальної машини (ВМ) - *монітор віртуальних машин* (МВМ). Монітор грає проміжну роль між віртуальною машиною і базовим обладнанням, підтримуючи виконання всіх створених ВМ на єдиній апаратній платформі та забезпечуючи їх надійну ізоляцію.

Користувач не має повноцінного доступу до монітора віртуальних машин. В більшості програмних продуктів йому надається лише графічний інтерфейс для створення і налаштувань віртуальних машин. Цей інтерфейс зазвичай називають *консолью віртуальних машин.*

"Всередині" віртуальної машини користувач встановлює, як і на реальному комп'ютері операційну систему. Така ОС, яка належить конкретній ВМ, називається *гостьовою* (guest OS). Перелік підтримуваних ОС являється однією з найбільш головних характеристик віртуальної машини. Найбільш потужні сучасну віртуальні машини підтримують близько десятка популярних версій операційних систем із сімейства Windows, Linux, та MacOS.

Коли віртуальна машина створена і запущена, у користувача може виникнути ілюзія того, що він працює на справжньому комп'ютері, який має власний процесор, оперативну память, відеосистему та ін.

Насправді віртуальна машина не має доступу до фізичних ресурсів реального комп'ютера. Робота з ними покладена на МВМ, а також на *драйвер віртуальних машин*.

В спрощеному вигляді архітектура системи, в якій використовуються віртуальні машини виглядає наступним чином:

- хостова ОС і монітор віртуальних машин розділяють між собою права на управління апаратними компонентами комп'ютера; при цьому хостова ОС займається розподіленням ресурсів між власними програмами.

- монітор ВМ контролює розподілення ресурсів між запущеними віртуальними машинами створюючи для них ілюзію повного доступу до апаратного рівня (*віртуалізація).*

- гостьові ОС в межах виділених їм ресурсів керують роботою "своїх" програм.

Дана архітектура є цілком загальною. Однак існують системи віртуальних машин які мають значні відмінності.

Основні принципи роботи програми емулятора.

ВИДИ ВІРТУАЛЬНИХ МАШИН

Віртуальні машини з емуляцією API гостьової ОС

Зазвичай програми працюють в ізольованому адресному просторі і взаємодіють з обладнанням за допомогою інтерфейсу API (Application Programming Interface - інтерфейс прикладного програмування), що надається операційною системою. Якщо дві операційні системи сумісні за своїми інтерфейсів API (наприклад, різні версії Windows), то програми, розроблені для однієї з них, працюватимуть і на іншій. Якщо дві операційні системи несумісні за своїми інтерфейсами API (наприклад, Windows і Linux), то необхідно забезпечити перехоплення звернень додатків до API гостьової ОС і зімітувати її поведінку засобами хостової ОС. При такому підході можна встановити одну операційну систему і працювати одночасно як з її додатками, так і з додатками іншої операційної системи.

Оскільки весь код програми виповнюється без емуляції, а емулюються лише виклики API, така схема віртуалізації призводить до незначної втрати в продуктивності віртуальної машини. Однак через те, що багато програм використовують недокументовані функції API або звертаються до операційної системи в обхід API, навіть дуже хороші емулятори API мають проблеми сумісності і дозволяють запускати не більше 70% від загального числа додатків. Крім того, підтримувати емуляцію API бурхливо розвиваючоїся операційної системи (наприклад, такиої як Windows) дуже нелегко, і більшість емуляторів API так і залишаються емуляторами якоїсь конкретної версії операційної системи. Так, в Windows NT/2000 досі вбудований емулятор для додатків OS / 2 версії 1.x. Але найбільший недолік ВМ з емуляцією API гостьової ОС - це її орієнтація на конкретну операційну систему.

Приклади продуктів, виконаних по технології емуляції API гостьової ОС:

■ проект з відкритим кодом Wine (Wine Is Not an Emulator, «Wine - це не емулятор»), що дозволяє запускати DOS-, Win16-і Win32-додатки під управлінням операційної системи Linux і Unix;

■ продукт Win4Lin компанії Netraverse, що дозволяє запускати операційні системи сімейства Windows під управлінням операційної системи Linux;

■ проект з відкритим кодом DOSEMU, що дозволяє запускати DOS-додатки під управлінням операційної системи Linux;

■ проект з відкритим кодом User Mode Linux (UML), що дозволяє запускати декілька копій операційної системи Linux на одному комп'ютері (в даний час вбудований і ядро Linux версії 2.6);

■ технологія Virtuozzo, розроблена російською компанією SWsoft і дозволяє запускати декілька копій операційної системи Linux на одному комп'ютері.

Віртуальні машини з повною емуляцією гостьової ОС

Проекти, що підтримують технологію повної емуляції, працюють за принципом інтерпретації інструкцій з системи команд гостьової ОС. Оскільки при цьому повністю емулюється поведінка як процесора, так і всіх зовнішніх пристроїв, то існує можливість емулювати комп'ютер з архітектурою Intel x86 на комп'ютерах з абсолютно іншою архітектурою, наприклад на робочих станціях Mac або на серверах Sun з RISC-процесорами. Головний недолік повної емуляції полягає в істотній втраті продуктивності гостьової операційної системи (швидкість роботи «гостьових» додатків може впасти в 100-1000 разів). Тому до недавнього часу ВМ з повною емуляцією найчастіше використовувалися як низькорівневі відладники для дослідження і трасування операційних систем. Однак завдяки значному зростанню обчислювальних потужностей навіть «настільних» комп'ютерів ВМ з повною емуляцією стали сьогодні цілком конкурентоспроможними. Найбільш яскравий представник цього виду ВМ - продукт Virtual PC фірми Connectix (нині купленої Microsoft).

В якості прикладів проектів, виконаних за технологією повної емуляції, можна назвати наступні:

■ проект з відкритим кодом Bochs, що дозволяє запускати різні операційні системи Intel x86 під Linux, Windows, BeOS і Mac OS;

■ продукт Simics компанії Virtutech, що дозволяє запускати і налагоджувати різні операційні системи Intel x86 під управлінням Windows та інших операційних систем;

■ проект Qemu - емулятор різних архітектур на PC,

Віртуальні машини з квазіемуляцією гостьової ОС

Технологія квазіемуляції гостьової ОС заснована на тому, що далеко не всі інструкції гостьової ОС потребують емуляції засобами хостової операційної системи. Багато з інструкцій, необхідних для коректної роботи «гостьових» додатків, можуть бути безпосередньо адресовані хостової ОС. Виняток становлять інструкції для керування такими пристроями, як відеокарта, IDE-контролер, таймер, і деякими іншими.

Таким чином, в процесі роботи RM з квазіемуляцією відбувається вибіркова емуляція інструкцій гостьової ОС. Очевидно, що продуктивність такої ВМ повинна бути вище, ніж у ВМ з повною емуляцією. Тим не менш, як було сказано, при досягнутих рівнях продуктивності персональних комп'ютерів різниця виявляється не настільки відчутною. Приклади проектів, виконаних за технологією квазіемуляції:

■ технологія Virtual Platform, на базі якої компанія VMware пропонує чотири продукти: VMware Workstation для Windows NT/2000/XP, VMware Workstation для Linux, VMware GSX Server (group server) і VMware LSX Server (enterprise server);

■ віртуальна машина Serenity Virtual Station (SVISTA) (колишня twoOStwo), розроблена російською компанією Паралелі (Parallels) на замовлення німецької компанії NetSys GmbH ;

■ проект з відкритим кодом Plex86, що дозволяє запускати різні операційні системи Intel x86 під управлінням Linux.

■ проект з відкритим кодом L4Ka, що використовує мікроядро;

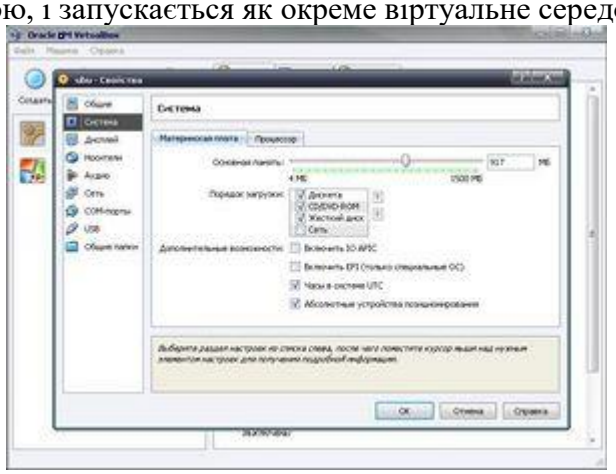
■ проект з відкритим кодом Xen, що дозволяє запускати модифіковані ОС Linux, FreeBSD, NetBSD і Windows XP під управлінням Linux, FreeBSD, NetBSD і при дотриманні деяких умов забезпечує навіть приріст продуктивності.

Інтерфейс програми.

ПРИКЛАДИ ВІРТУАЛЬНИХ МАШИН

VirtualBox

Яскравим представником віртуальних машин є VirtualBox. Програма віртуалізації для операційних систем, розроблена німецькою фірмою innotek, зараз вона належить Oracle Corporation. Вона встановлюється на існуючу операційну систему, яка називається хостовою, у середину цієї програми встановлюється друга операційна система, яку називають гостьовою операційною системою, і запускається як окреме віртуальне середовище.



VirtualBox. Основна пам'ять

Програма була створена компанією Innotek з використанням вихідного коду Qemu. Перша публічно доступна версія VirtualBox з'явилась 15 січня 2007 року. В лютому 2008 року Innotek був викуплений компанією Sun Microsystems, модель поширення VirtualBox при цьому не змінилася. В січні 2010 року Sun Microsystems була поглинена Oracle Corporation, модель поширення залишилась попередньою.

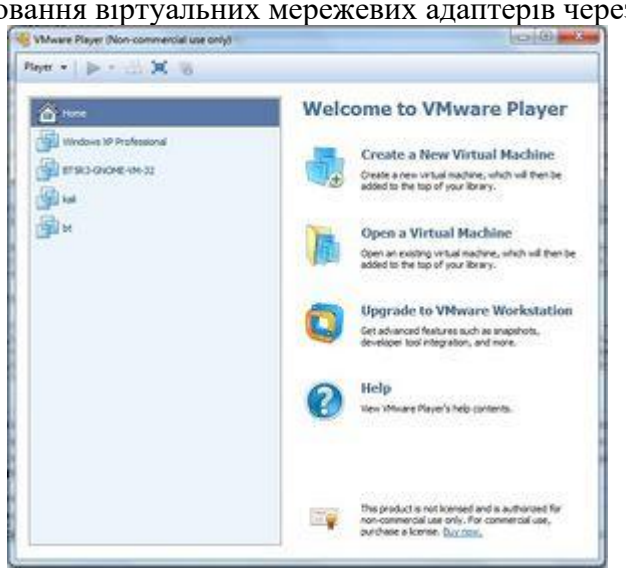
Існують різні варіанти VirtualBox, для різних хостових операційних систем. На даний момент існують варіанти для Windows, Linux, OS X та Solaris, як для x86 архітектури, так і 64-бітних. Завантажити програму можливо на офіційному сайті VirtualBox.

В налаштуваннях є багато цінних функцій: встановлення розміру оперативної пам'яті, керування накопичувачами інформації(підтримка IDE, SATA, SCSI, SAS, Floppy), можливість встановлення декількох мережевих адаптерів, трансляція роботи COM-портів, підтримка "Спільної теки" для обміну даними між хостовою та гостьовою ОС-ми. Використовуючи такі широкі можливості налаштувань дозволяють експерементувати з гостьовою ОС, без шкоди для основної операційної системи.

VMware Player

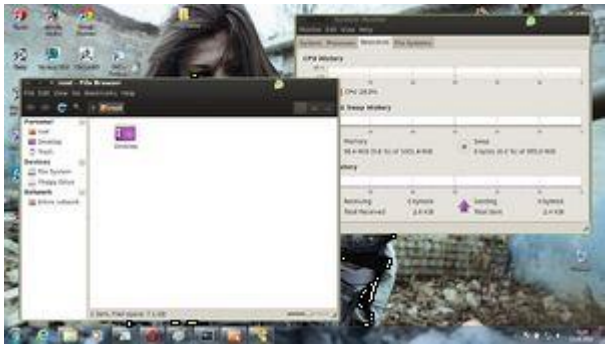
VMware Player - безкоштовний для некомерційного використання програмний продукт, на основі віртуальної машини VMware Workstation, але з обмеженою функціональністю, призначений для запуску образів віртуальних машин, створених в інших продуктах VMware, а також в Microsoft VirtualPC і Symantec LiveState Recovery. Починаючи з версії 3.0 VMware Player дозволяє також

створювати образи віртуальних машин. Обмеження функціональності тепер стосується в основному функцій, призначених для IT-фахівців і розробників ПЗ. Наприклад, відсутня можливість тонкого настроювання віртуальних мережевих адаптерів через Virtual Network Editor.



VMware Player

Налаштування має значну кількість функцій. Цікавою функцією є режим Unity. Режим Unity дає можливість відображати додатки віртуальної машини безпосередньо на робочому столі хост-системи. Наприклад, на малюнку можна побачити запущений System monitor та File Browzer.



Режим Unity