Лекція з навчальної дисципліни: «Крос-платформне програмування»

Тема 1. Віртуальні обчислювальні середовища
Заняття 1.1. Концепція та принципи організації віртуальних середовищ

КОВАЛЕНКО Олексій Єпифанович,

к.т.н., доцент СК №5

Навчальні питання:

1. Основні апаратні і програмні платформи.
2. Концепція та принципи організації середовища віртуальних машин.
3. Процесові і системні віртуальні машини.

Література:

1. Гультяев А. К. Виртуальные машины. Несколько компьютеров в одном. — СПб.: Питер, 2006. — 224 с.
2. Михеев М. О. Администрирование VMware vSphere 4.1. – М.: ДМК Пресс, 2011. – 448 с.
3. Михеев М.О. VMware Workstation 8 Network – Режим доступу: http://www.vm4.ru/2011/10/vmware-workstation-8-network.html
4. Таненбаум Э., ван Стеен М. Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. – 877 с.

1. Основні апаратні і програмні платформи

Платформна незалежність (крос-платформеність) - властивість програмного забезпечення або методів обчислень і принципів, що визначає можливість його реалізації та переносимість між різними комп'ютерними платформами.

Комп'ютерна платформа - сукупність програмного і апаратного забезпечення, що використовується для виконання програмних застосунків.

Підходи до реалізації платформної незалежності програмного забезпечення

- На рівні апаратного забезпечення;
- На рівні операційних систем;
- На рівні додатків;
- На рівні вихідних кодів і компіляторів;
- На рівні бібліотек.

1.1. Апаратні платформи

Апаратна платформа комп'ютера (архітектура комп'ютера) - рівень, утворений

- - Мікроархітектурою;
- - Мікропрограмою керування;
- - Ядром мікропроцесора;
- - Архітектурою набору команд

на апаратній базі конкретних мікросхем процесора, чіпсета, інших фізичних компонентів, які в сукупності складають апаратну модель обчислювальної системи.

Найбільш популярні платформи

Апаратна платформа	Разробник	Розрядність, біт	Типи систем
Amiga PowerPC	Eyetech Group, Genesi, bPlan GmBH, ACube Systems Srl	32/64	ПК
IA-32	Intel	32	ПК,сервер,ноутбук,кластер
x86-64	AMD	64	ПК,сервер,ноутбук,кластер
SPARCv9	Sun Microsystems	64	робоча станція,сервер
IA-64	Intel і Hewlett Packard	64	Сервер
ESA/390	IBM	32	мейнфрейм
z/Architecture	IBM	64	мейнфрейм
Xbox 360	Microsoft в співпраці з IBM, ATI і SiS	64	ігрова приставка
PlayStation 3	Sony в співпраці з Toshiba і IBM	64	ігрова приставка

а також **апаратні платформи мобільних терміналів зв'язку**

1.2. Рівень операційної системи

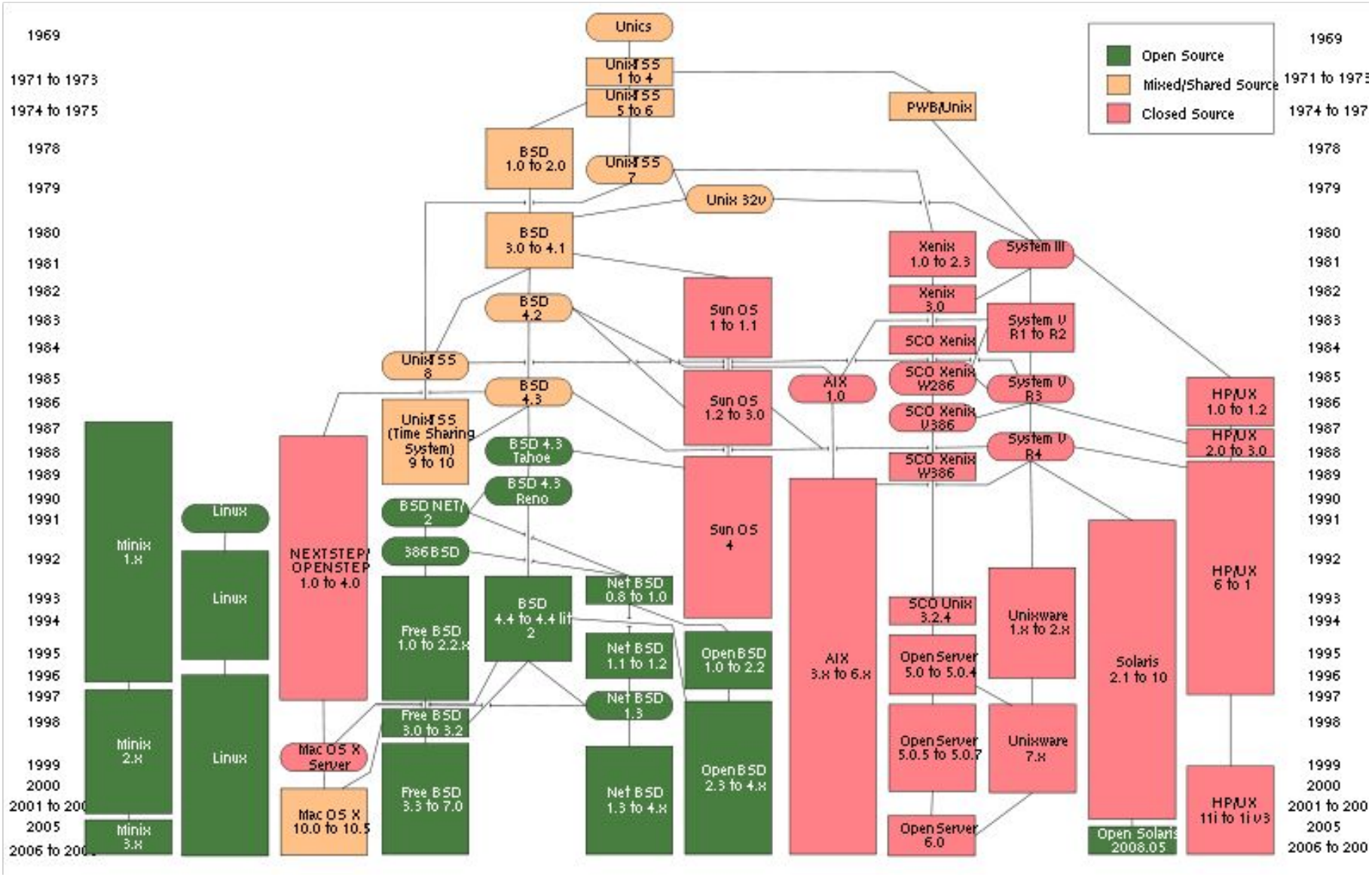
POSIX (**P**ortable **O**perating **S**ystem **I**nterface for Uni**x** - Переносимий інтерфейс операційних систем Unix) – **ISO/IEC 9945 -** набір стандартів, що описують інтерфейси між операційною системою і прикладною програмою

UNIX - сімейство

- **Linux (Arch Linux, CentOS, Debian, Fedora, Gentoo, Mandriva, Mint, openSUSE, Red Hat, Slackware, Ubuntu)**
- **ОС NetBSD, FreeBSD, OpenBSD**

Microsoft Windows: Windows CE і Windows Embedded

Еволюція UNIX-подібних ОС



Інші операційні системи:

- AmigaOS, AmigaOS 4
- Mac OSX
- IBM / Microsoft Operating System 2
- Solaris
- IBM VM/370, VM / BSEP, VM / SEP, VM / XA, VM / ESA, z / VM
- Google Chrome OS

Мобільні операційні системи:

- Android
- Bada
- Blackberry OS
- iOS
- Embedded Linux
- Palm OS
- Symbian platform
- Windows Mobile

1.3. На рівні прикладних програм, компіляторів та бібліотек

Запуск на різних архітектурах одного і того ж прикладного програмного забезпечення без необхідності забезпечувати сумісність на рівні ОС реалізується шляхом стандартизації

- - мов,
- - компіляторів,
- - бібліотек
- - середовища виконання (наприклад, POSIX),
- а також шляхом переходу на виконання ПЗ в **віртуальній машині** і **стандартному оточенні**, які реалізуються для кожної платформи та гарантують однакове виконання ПЗ незалежно від платформи (наприклад, CLI (*Common Language Infrastructure*).NET і JVM).

Середовища розробки програмного забезпечення

Для комп'ютерних систем:

- Adobe AIR
- Java - JDK and JRE
- Mono
- Mozilla Prism XUL and XULRunner
- .NET Framework
- Oracle Database
- Steam
- uniPaaS
- Vexi

Для мобільних пристроїв

- APOXI
- BREW
- Java ME
- JavaFX Mobile
- Qt

1.4. Пов'язані поняття

Портовність (переносимість, *portability*) зазвичай розріняють у двох змістах:

1. Портованість - як можливість один раз відкомпілювати код (зазвичай в певний проміжний код, який потім компілюється під час виконання, «на льоту» - JIT (Just-In-Time)), потім запускати його на інших платформах без яких-небудь змін.

2. Портованість - як властивість програмного забезпечення, яке описує, наскільки легко це ПЗ може бути портовано. Додаткові переваги в плані портованості можуть мати програми, що задовольняють спеціальним стандартам і правилам написання (наприклад, **Smart Package Manager**).

порт і **форк** - в першому випадку все користувальницькі властивості пакета намагаються зберегти, а в другому це базована на спільній основі самостійна розробка з новими корисними властивостями.

Крос-компілятор (*cross compiler*) - компілятор, що виробляє виконуваний код для платформи, відмінної від тієї, на якій виповнюється сам крос-компілятор.

Крос-браузерність - властивість сайту відображатися і працювати у всіх популярних браузерах ідентично.

Емуляція (*emulation*) - відтворення програмними або апаратними засобами або їх комбінацією роботи інших програм або пристроїв.

Віртуалізація

Віртуалізація в обчисленнях - процес представлення набору обчислювальних ресурсів, або їх логічного об'єднання, який дає певні переваги перед оригінальною конфігурацією.

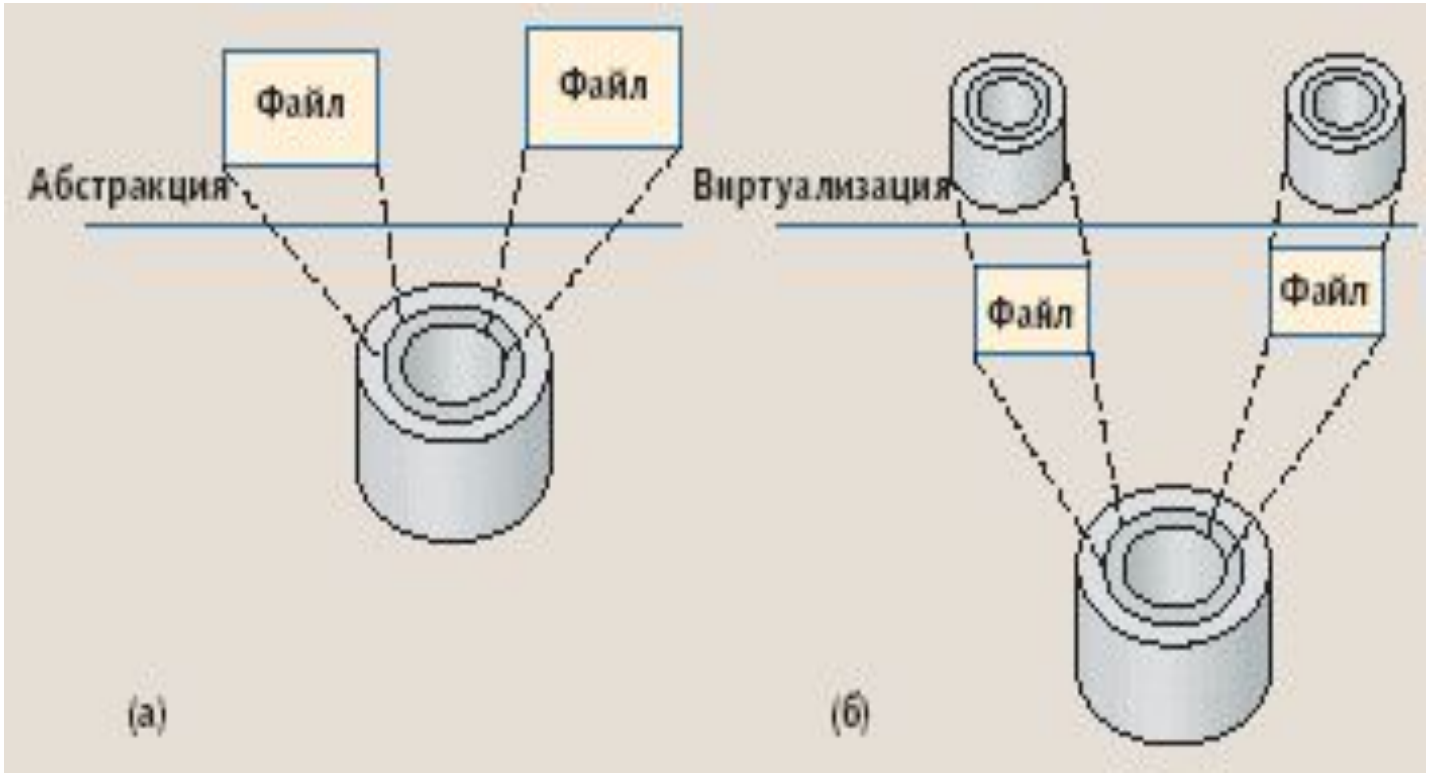
2. Концепція та принципи організації середовища віртуальних машин

Віртуальна машина (ВМ, virtual machine) -

- програмна та/або апаратна система, що **емулює апаратне забезпечення деякої платформи** та виконуюча програми для цієї платформи (target - цільова або гостьова платформа) на іншій платформі (host - хост-платформа, платформа-господар)
- або **віртуалізує деяку платформу і створює на ній середовища**, ізолюючі один від одного програми і навіть операційні системи (напоклад, пісочниця);
- або **специфікація** деякого обчислювального середовища (наприклад: «віртуальна машина мови програмування Сі»).

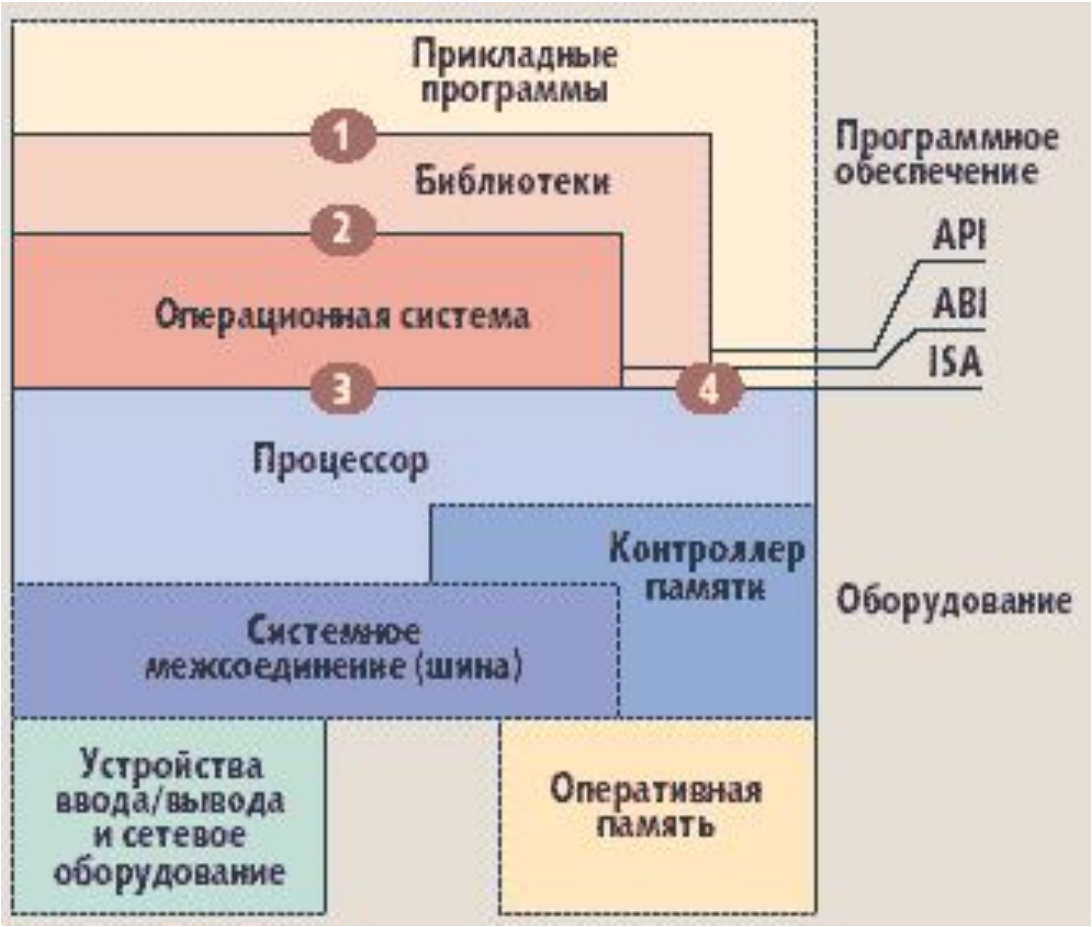
Віртуальні машини можуть використовуватися для

- захисту інформації та обмеження можливостей програм;
- дослідження продуктивності ПЗ або нової комп'ютерної архітектури;
- емуляції різних архітектур (наприклад, емулятор ігрової приставки);
- оптимізації використання ресурсів мейнфреймів і інших потужних комп'ютерів (наприклад: IBM eServer);
- шкідливого коду для управління інфікованої системою: вірус PMBS, виявлений в 1993 році, а також руткіт SubVirt, створений в 2006 році в Microsoft Research, створювали віртуальну систему, якою обмежувався користувач і всі захисні програми (антивіруси та інші);
- моделювання інформаційних систем з клієнт-серверною архітектурою на одній ЕОМ (емуляція комп'ютерної мережі за допомогою декількох віртуальних машин).
- спрощення управління кластерами - віртуальні машини можуть просто мігрувати з однієї фізичної машини на іншу під час роботи;
- тестування і налагодження системного програмного забезпечення.



Абстракція і віртуалізація в застосуванні до дискової пам'яті:
(а) абстракція підтримує спрощений інтерфейс до базових ресурсів;
(б) на даному рівні абстракції віртуалізація забезпечує альтернативний інтерфейс до ресурсів та організацію їх поділу

Архітектура комп'ютерної системи



Вертикальні зв'язки між ключовими рівнями реалізації забезпечуються архітектурою системи команд (ISA), двійковим інтерфейсом додатків (ABI) і інтерфейсом прикладного програмування (API)

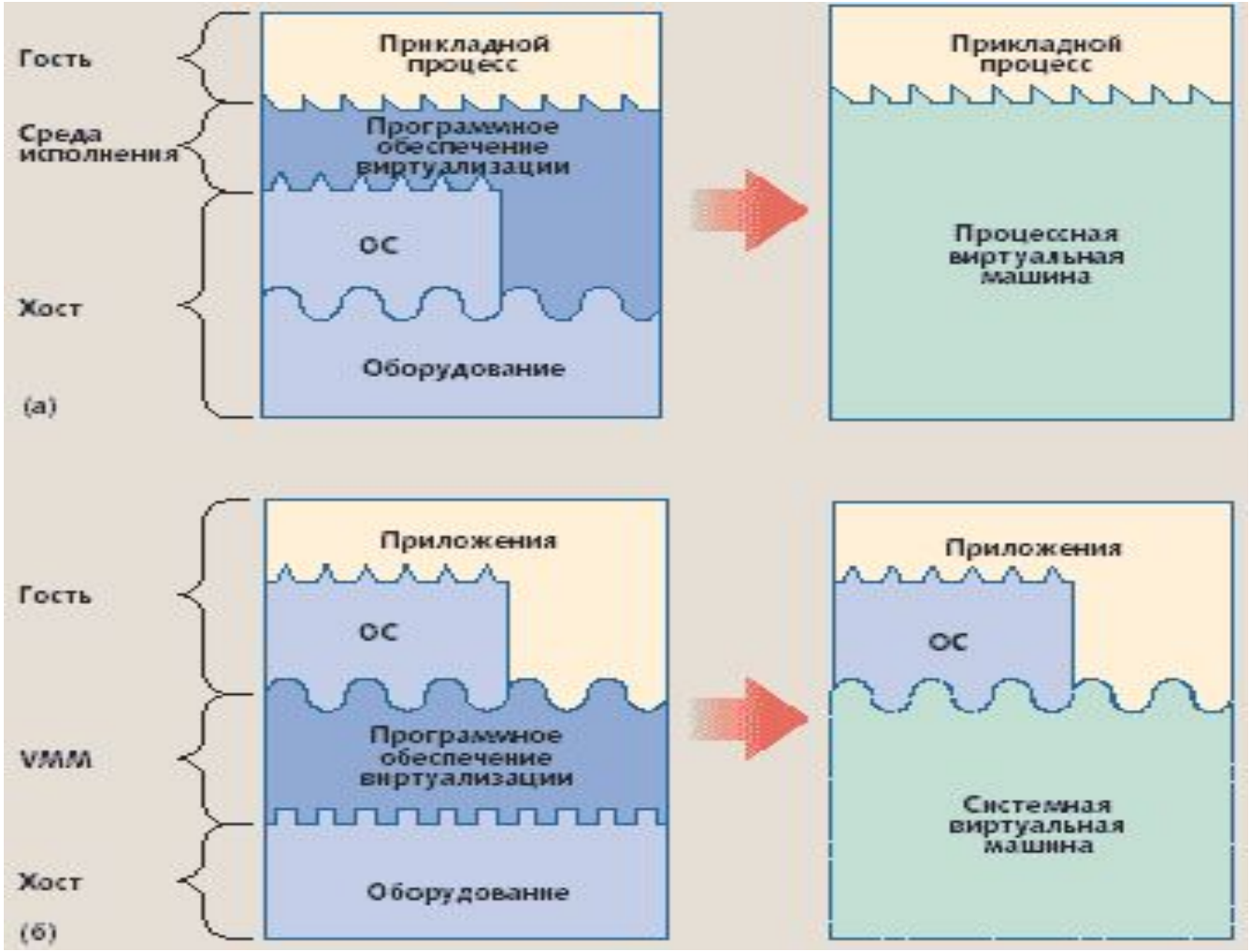
3. Процесні і системні віртуальні машини

Процес або система, виконувана на ВМ, називається **гостем**, а базова платформа, що підтримує ВМ, - **хостом**.

Програмне забезпечення віртуалізації, що реалізує процесну ВМ, часто скорочено іменують **робочим середовищем**.

Програмне забезпечення віртуалізації системної ВМ зазвичай називають **монітором віртуальних машин** (virtual machine monitor, **VMM**).

Процесні і системні віртуальні машини



- (а) програмне забезпечення віртуалізації процесної ВМ перетворює системні виклики та машинні команди користувацького рівня у відповідні виклики і команди іншої програмно-апаратної платформи;
- (б) в системній ВМ програмне забезпечення віртуалізації перетворює машинні команди однієї апаратної платформи в команди для іншої

4.1. Процесні віртуальні машини

- **Багатозадачні системи**
- **Інтерпретатори і динамічні транслятори двійкового коду**
- **Оптимізатори двійкового коду**
- **Віртуальні машини для мов високого рівня**

4.2. Системні віртуальні машини

- **Класичні системні ВМ**
- **Вкладені ВМ**
- **Інтегральні ВМ**
- **Багатопроцесорна віртуалізація**
- **Вбудовані ВМ**

Класифікація віртуальних машин