Архітектура операційних систем.

Виконав

Студент гр.Єс-41

Матвіїв Петро

План.

1. Поняття архітектури операційних систем.
2. Взаємодія операційної системи з апаратним забезпеченням.
3. Взаємодія операційної системи з програмним забезпеченням.
4. Підходи до реалізації архітектури операційних систем.
5. Архітектура системи UNIX.
6. Архітектура системи Linux.

1. Поняття архітектури операційних систем.

<u>Архітектура операційної системи</u> – це сукупність компонентів системи та порядок їхньої взаємодії між собою та із зовнішнім середовищем.

Основні компоненти (складові) операційної системи:

Режими виконання процесором програмного коду

Режим ядра (kernel mode).

Команди, що виконуються у цьому режимі:

- мають прямий доступ до апаратного забезпечення;
- мають доступ до усієї пам'яті комп'ютера;
- не можуть бути витіснені у файл підкачки на диск;
- виконуються з найвищим пріоритетом.

Режим користувача (user mode).

Програма, що виконується у цьому режимі:

- не має прямого доступу до апаратури;
- має обмежений адресний простір;
- може бути витіснена у віртуальну пам'ять;
- виконується з меншим пріоритетом, ніж ядро.

Команди, які є критичними для роботи системи (перемикання задач, звертання до пам'яті з заданими межами, доступ до пристроїв введення-виведення) в цьому режимі недопустимі.

Обидва режими реалізовані на апаратному рівні.

Підтримка захищеного режиму реалізована починаючи з 32- розрядних процесорів.

2. Взаємодія операційної системи з апаратним забезпеченням.

Засоби апаратної підтримки операційних систем:

- система переривань:
  - апаратні переривання
  - програмні переривання
- привілейований режим процесора
- засоби керування пам'яттю:
  - механізми трансляції адрес
  - захист пам'яті
- системний таймер
- захист пристроїв введення-виведення
- базова система введення-виведення. (BIOS).

3. Взаємодія операційної системи з програмним забезпеченням.

Засобами взаємодії операційної системи та програм є:

- **системні виклики**

Системний виклик – це засіб доступу до певної функції ядра операційної системи із прикладних програм.

- **інтерфейс програмування застосувань (API)**

Сюди входять додаткові функції, котрі доповнюють та розширюють можливості системних викликів.

Мета формування інтерфейсу програмування застосувань: надати програмісту бібліотеку функцій, котрі виконують системні виклики та реалізовані у режимі користувача.

Програмна сумісність – можливість виконувати в середовищі однієї ОС програми, розроблені для іншої ОС.

Досягається за рахунок:

- наявності стандарту на мову програмування та компілятор,
- наявності стандарту на інтерфейс програмування застосувань.

Схема взаємодії між ядром та прикладними програмами.

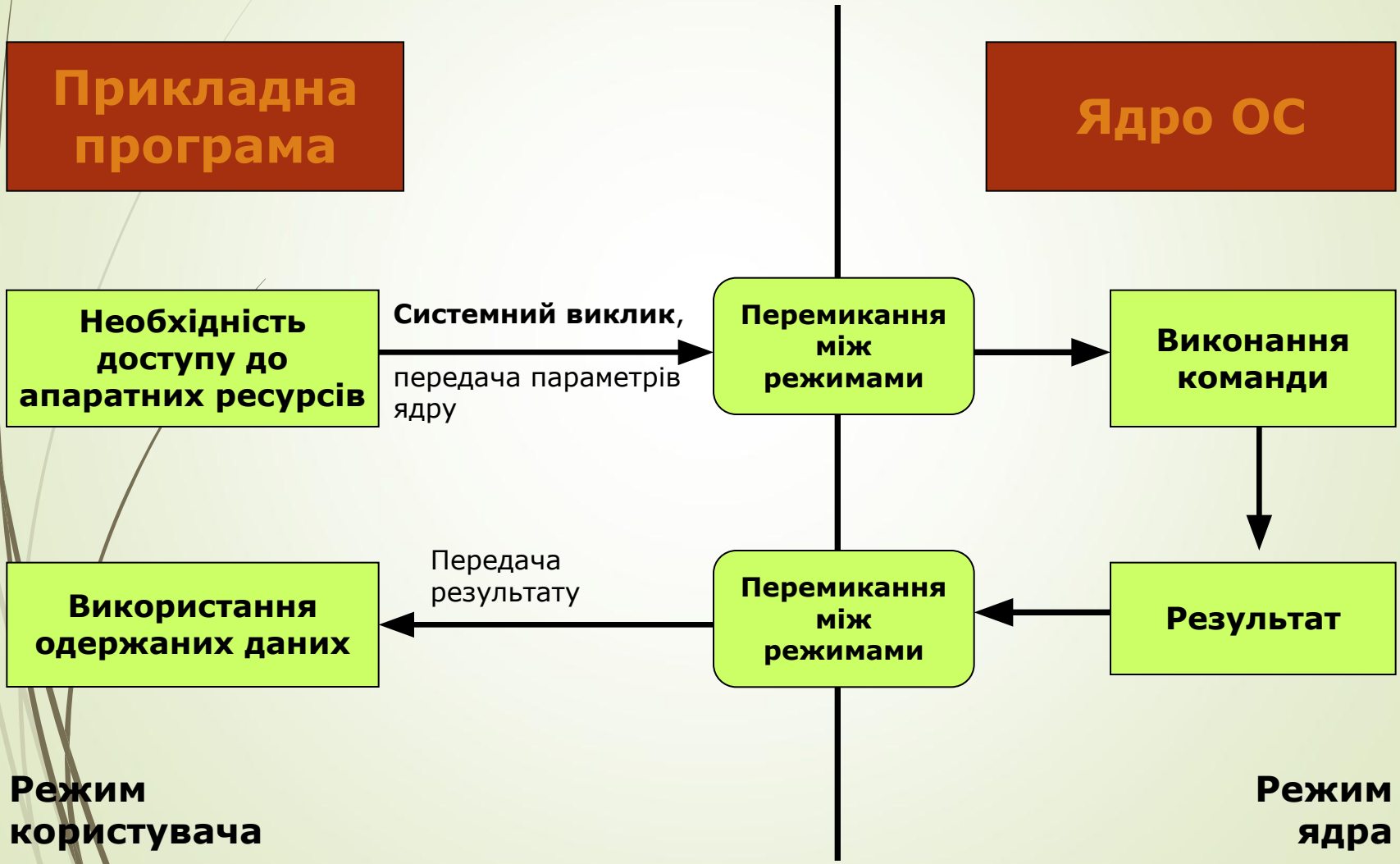


Схема взаємодії між ядром та прикладними програмами (приклад).

Механізм операційної системи та політика управління компонентами.

Механізм операційної системи – набір фундаментальних можливостей, які надають її компоненти.

(ЩО реалізовано компонентом)

Політика управління компонентами – рішення щодо використання визначених можливостей.

(ЯК можна використати компоненти)

4. Підходи до реалізації архітектури операційних систем.

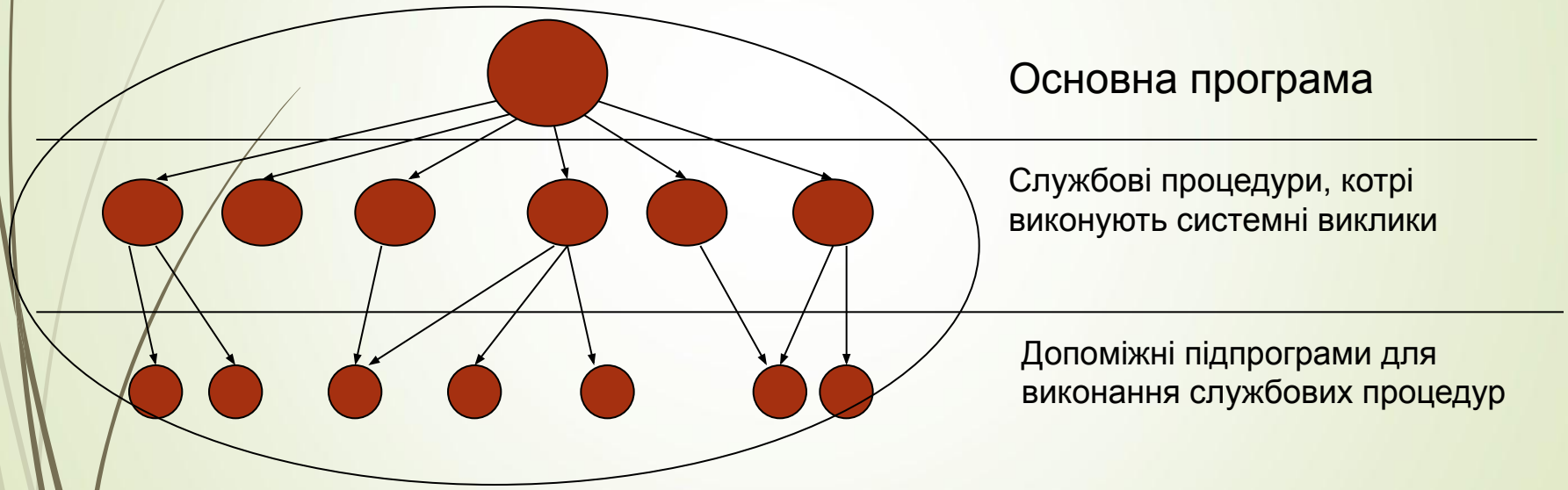
Виділяють наступні основні реалізації архітектури операційних систем:

- Монолітна система
- Багаторівнева система
- Система з мікроядром
- Віртуальна машина

Монолітні системи

Монолітна система – це система, у якій всі базові функції сконцентровані у ядрі.

Базова структура монолітної системи:



!!! Увесь код виконується у привілейованому режимі

\+ 1) підвищується продуктивність (не витрачається час на перемикання між режимами).

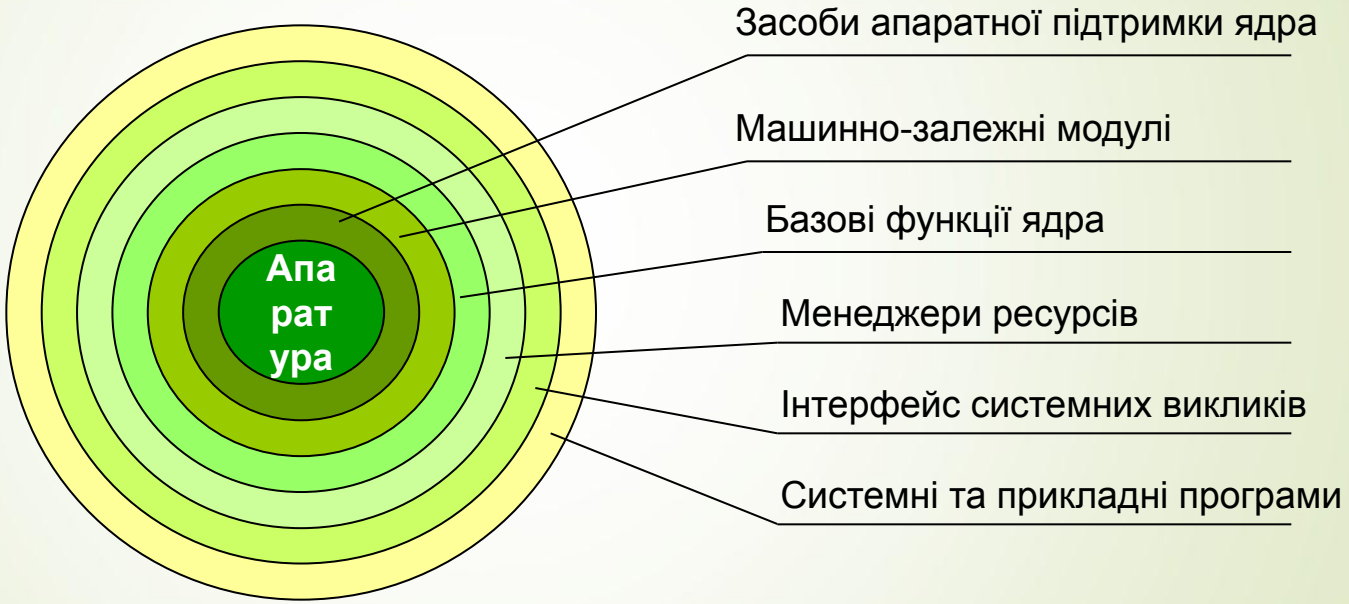
\- 1) менш надійні (оскільки увесь код виконується у привілейованому режимі, то кожна помилка може бути критичною).

Приклади: OS/360, UNIX ранніх версій.

Багаторівневі системи

Багаторівнева система – це система, компоненти котрої утворюють ієрархію рівнів, кожен з яких спирається на функції попереднього рівня.

Схема рівнів системи:



\+ 1) Висока продуктивність;

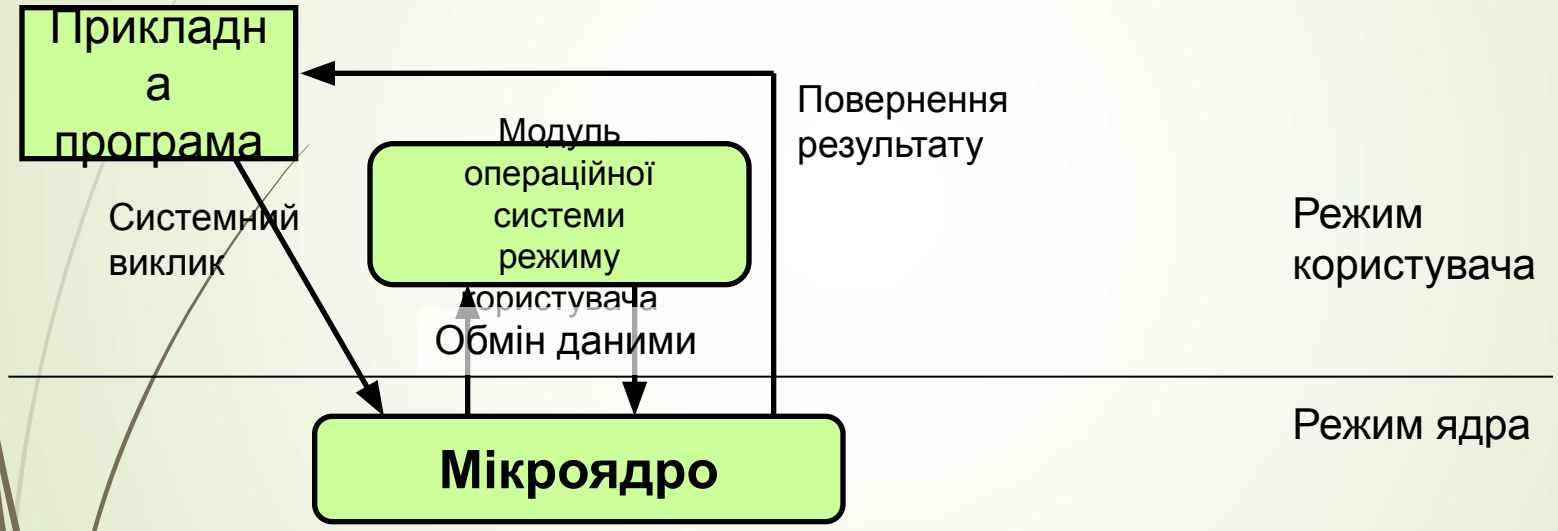
\- 1) великий обсяг коду ядра знижує надійність системи.

Приклад: THE, створена Е.Дейкстрою в Technische Hogeschool Eindhoven, 1968 р.

MULTICS, Масачусетський університет, Bell Labs, General Electrics.

Системи з мікроядром

Система з мікроядром – це система, у якій невелика частка функцій ядра реалізована у привілейованому режимі (*mikrokernel*), інші функції – виконуються процесами режиму користувача.



+

1) невеликий розмір ядра спрощує його розробку та налагодження;
2) висока надійність (тільки невелика кількість команд мікроядра має доступ до апаратного забезпечення);
3) більша гнучкість та розширюваність (нові функції можна додати шляхом додавання нового модуля);
4) можливість адаптації до умов мережі (функції ядра та режиму користувача можуть знаходитись на різних комп'ютерах).

-

1) зниження продуктивності через велику кількість переключень між режимами

Приклад: Symbian OS, Minix.

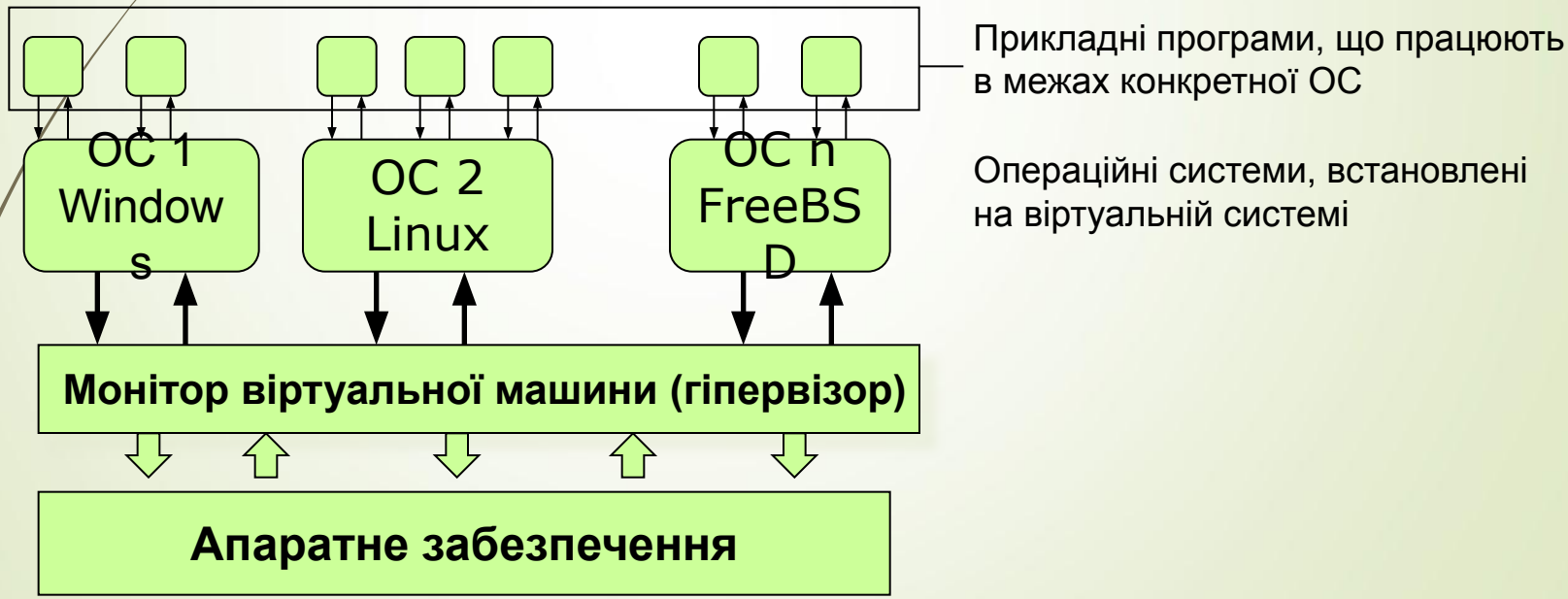
Віртуальні машини

Віртуальна машина – це створена програмно копія апаратного забезпечення, на якій функціонує одна чи декілька інших операційних систем.

Існує два типи віртуальних машин:

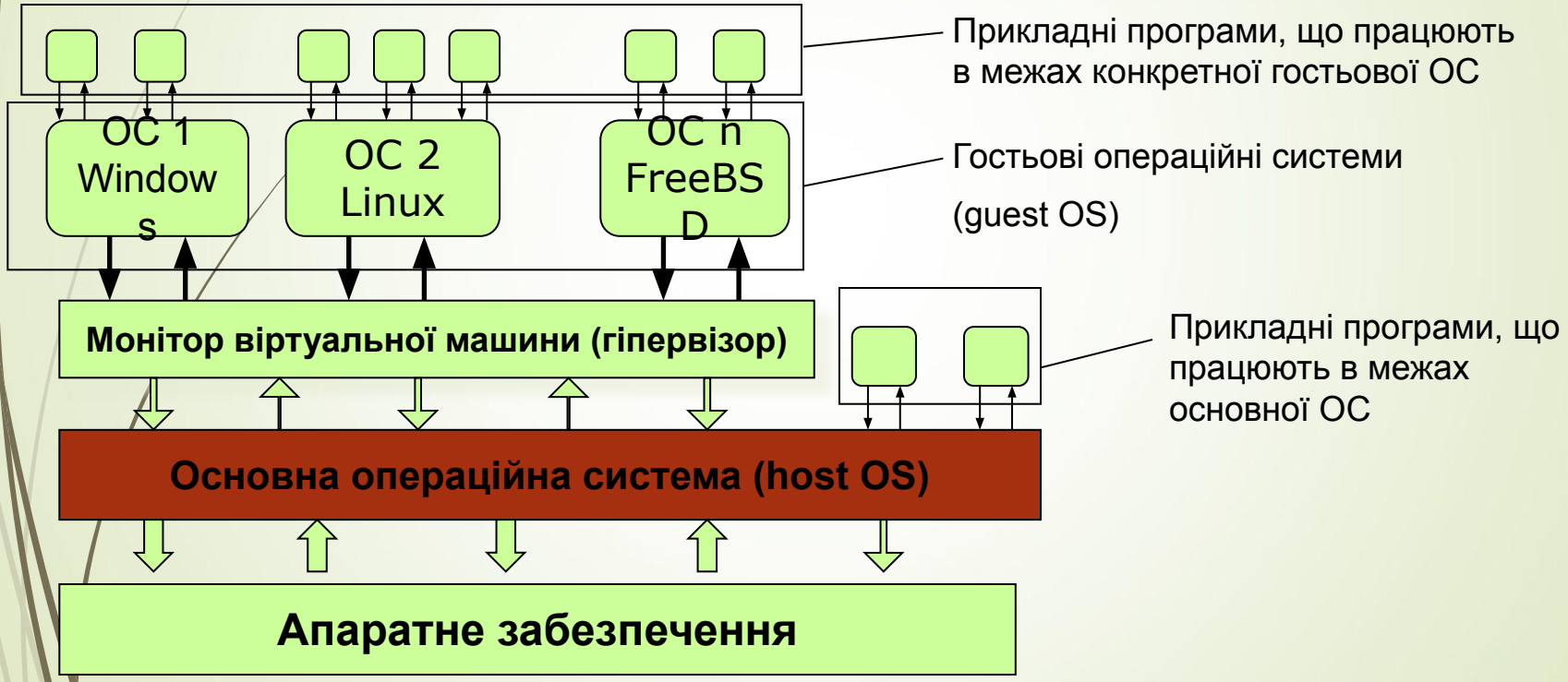
1) Віртуальна машина, що працює безпосередньо на реальному обладнанні,

2) Віртуальна машина, котра встановлена як прикладна програма над існуючою ОС.



Віртуальні машини

Схема роботи віртуальної машини, котра встановлена **як прикладна програма над існуючою ОС.**



Приклади: VMware, розробка дослідників Стенфордського університету, 1999р.
Virtual PC, розробку розпочала фірма Connectix, згодом проект викупила Microsoft.

5. Архітектура операційної системи UNIX (базова)

Архітектура операційної системи UNIX

Компоненти режиму ядра

1. Підсистема керування процесами – контролює створення та вилучення процесів, розподілення системних ресурсів між ними, міжпроцесову взаємодію, керування пам'яттю.

2. Файлова підсистема – забезпечує єдиний інтерфейс доступу до даних, розташованих на дискових накопичувачах та до периферійних пристроїв, контролює права доступу до файлів.

!!! В UNIX-подібних системах одні і ті самі системні виклики використовують як для обміну даними з диском, так і для виведення на термінал або принтер (програма працює з принтером як з файлом).

3. Підсистема введення-виведення – виконує запити файлової підсистеми.

Символьні пристрої – файли цих пристроїв потрібно зчитувати/записувати послідовно, потоками символів. (*принтер, модем*)

Блокові пристрої – допускають прямий доступ до будь-якої ділянки файлу. (*диск*)

4. Інтерфейс системних викликів – обов'язково має містити визначений стандартом POSIX мінімальний набір функцій для управління процесами, файловою системою, правами користувачів, системним часом.

Стандарт POSIX (Portable Operating System) розроблений інститутом IEEE для забезпечення сумісності усіх UNIX-подібних систем.

6. Архітектура операційної системи Linux

В ОС Linux можна виділити три основні частини:

Системні бібліотеки - стандартний набір функцій для використання у прикладних програмах

Системні утиліти - прикладні програми, які виконують спеціалізовані задачі

Архітектура операційної системи Linux

Компоненти режиму ядра:

1. Планувальник процесів - відповідає за реалізацію багатозадачності (робота з таймером, створення та завершення процесів та ін.)

2. Менеджер пам'яті - виділяє адресний простір для кожного процесу та реалізує підтримку віртуальної пам'яті.

3. Віртуальна файлова система - надає універсальний інтерфейс взаємодії з різними файловими системами та пристроями введення-виведення.

4. Драйвери пристроїв - забезпечують роботу з периферійними пристроями.

5. Мережний інтефейс - забезпечує доступ до мережних протоколів та драйверів мережних пристроїв.

6. Підсистема міжпроцесорної взаємодії - забезпечує обмін даними між процесами.

Висновки

- **Архітектура операційної системи** – це сукупність компонентів системи та порядок їхньої взаємодії між собою та із зовнішнім середовищем.
- **Режими роботи процесора**: режим ядра, режим користувача.
- Виділяють наступні **основні реалізації архітектури операційних систем**: монолітна система, багаторівнева система, система з мікроядром, віртуальна машина.
- Операційна система UNIX є прикладом реалізації багаторівневої архітектури.
- Операційна система Linux має монолітну архітектуру.