

Системне програмне
забезпечення

(Операційні системи)

Література

1. Таненбаум “Современные операционные системы”, 4-е издание.
2. Understanding the Linux Kernel. Daniel P. Bovet. Marco Cesati.
3. Solaris Internals. Richard McDougall, Jim Mauro.
4. HP-UX 11i Internals. Chris Cooper, Chris Moore.
5. Mac OS X Internals: A Systems Approach. Amit Singh.

Лекція 1

Визначення ядра ОС

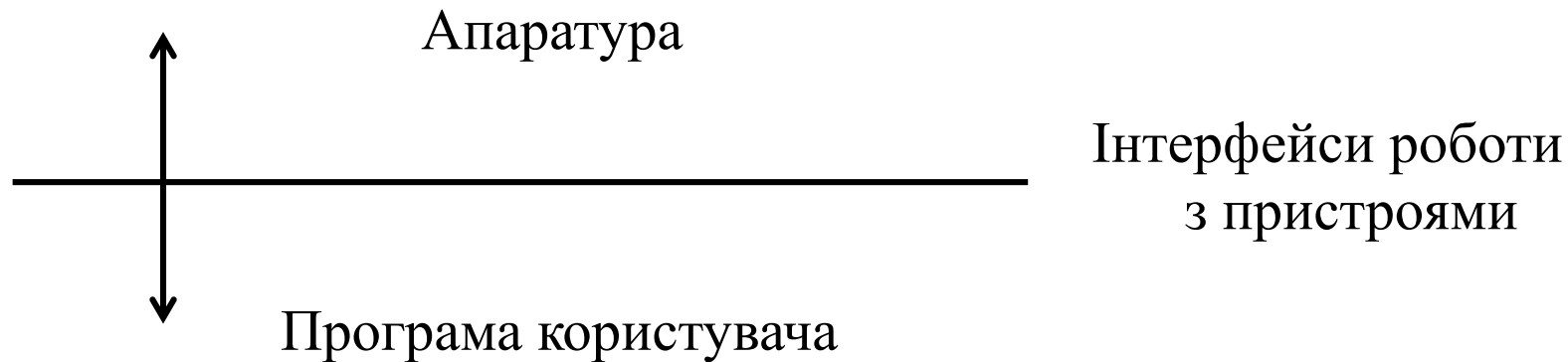
Класифікація ОС

Компоненти системного програмного забезпечення

1. Ядро (kernel).
2. Системні програми.
3. Бібліотеки.

Використання обчислювальної системи

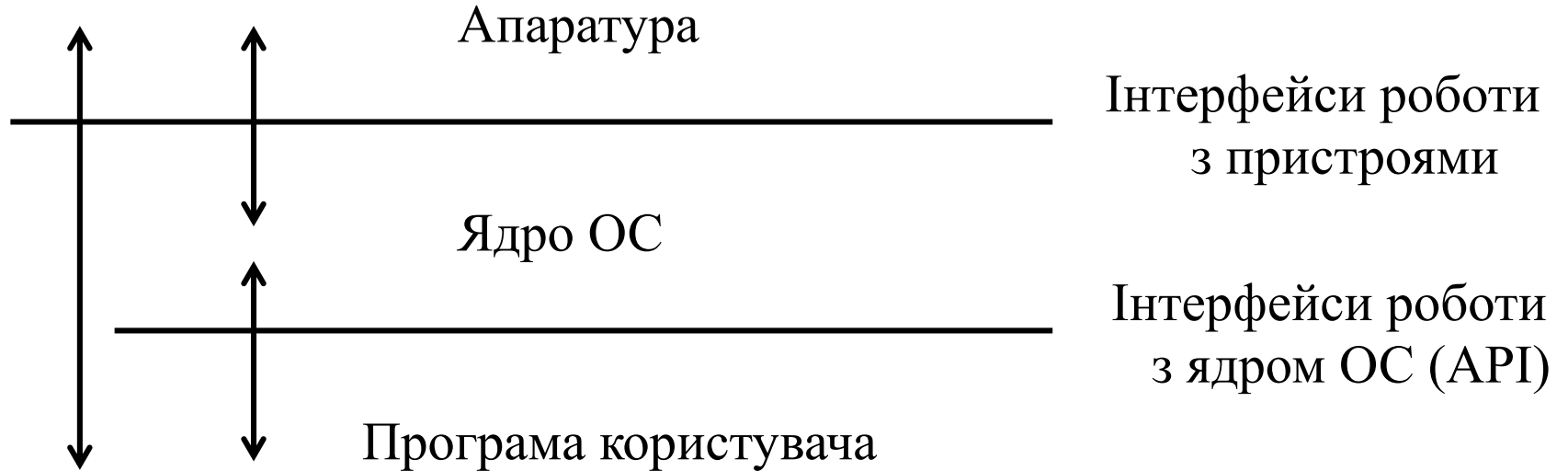
1. Без операційної системи.
2. З операційною системою.



Програма користувача крім рішення завдання повинна взаємодіяти з апаратурою.

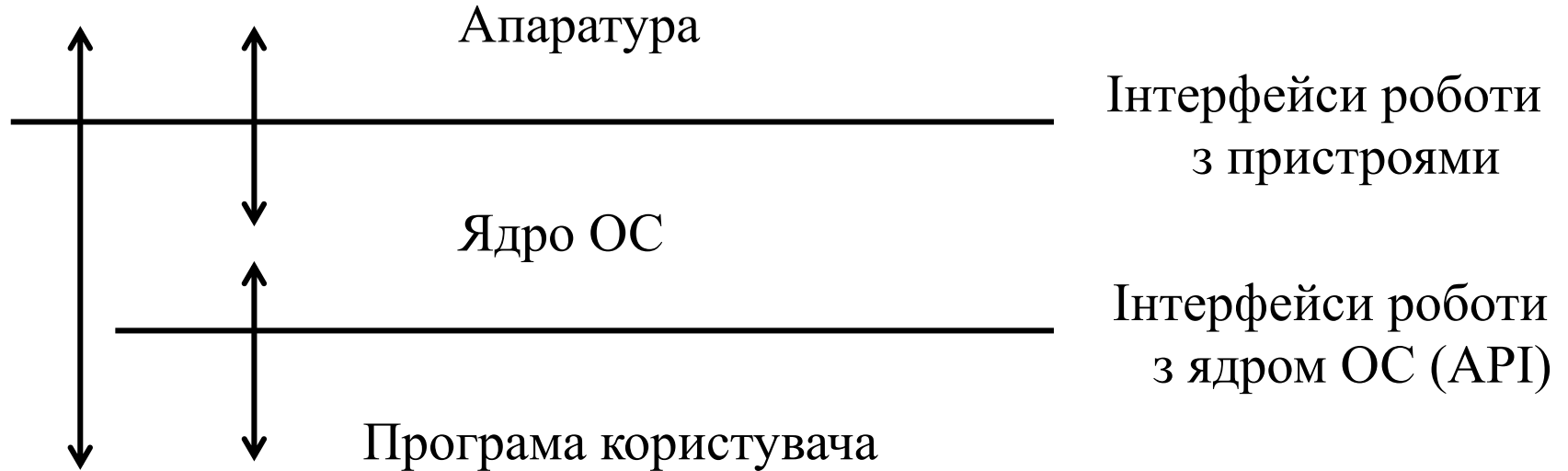
Код програми має частини, що залежать від апаратури обчислювальної системи.

Ідея використання ОС



Ядро ОС надає абстрактні інтерфейси доступу до апаратури. Таке абстрагування спрощує взаємодію програми користувача з апаратурою та системними сервісами.

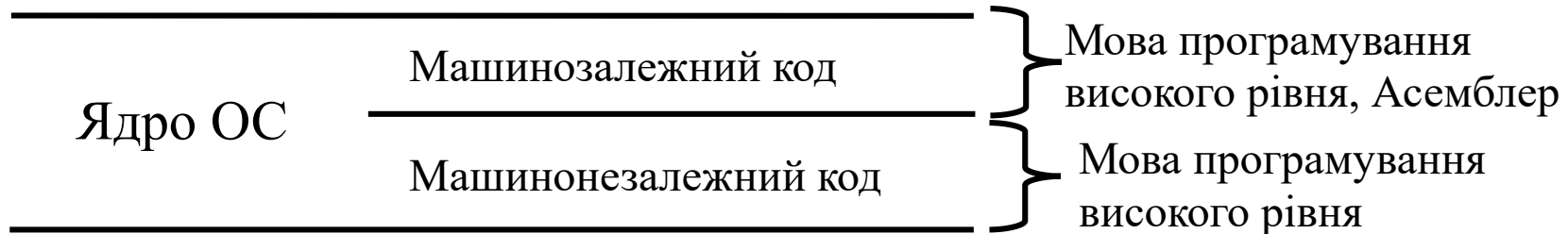
Ідея використання ОС



Програма користувача стає **системно залежною**. Наприклад програма для Windows не може бути скомпільована для Unix (різні API).

Дві частини ядра

Апаратура



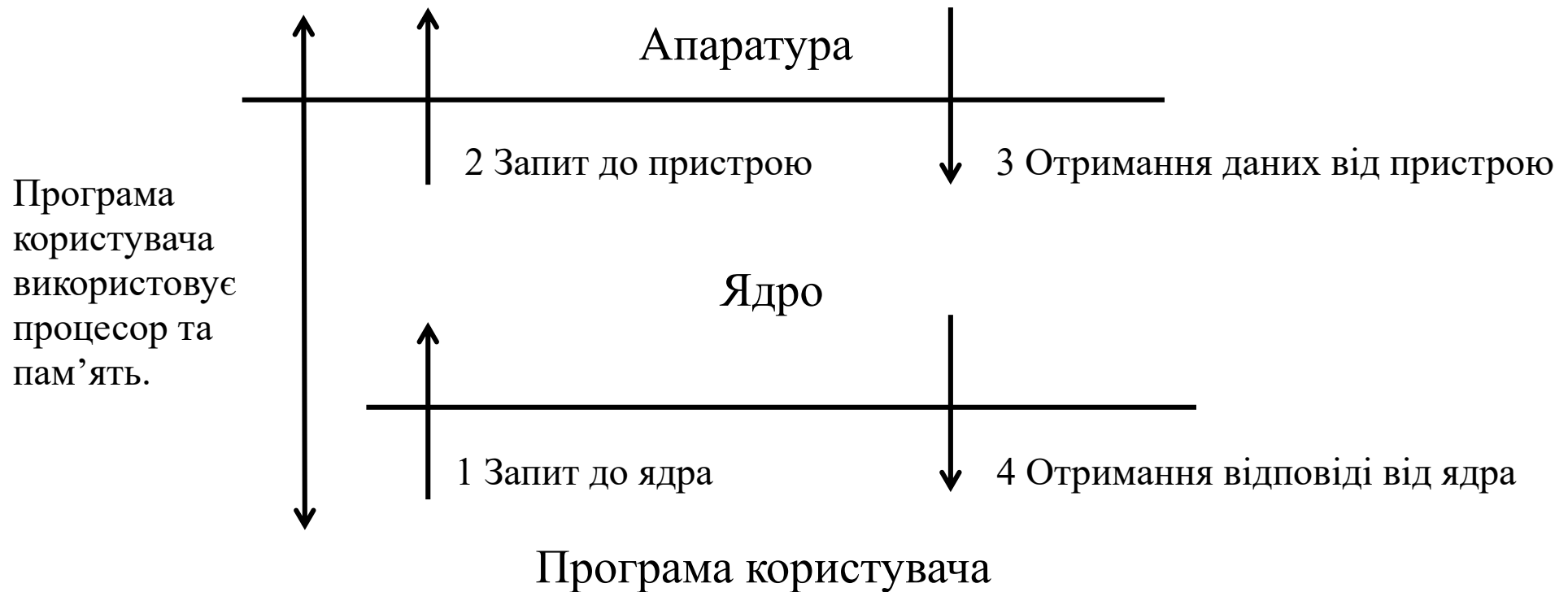
Програма користувача

Наприклад:

`mi_switch()` та `md_switch()`

Machine independent (MI), machine dependent (MD)

Взаємодія програми користувача і ядра ОС



Запуск програми користувача на іншій ОС

1. Якщо програма використовує API іншої ОС, то треба перекомпілювати програму (необхідний вихідний код програми).
2. Якщо програма використовує інше API, то треба переробити весь вихідний код, що працює з ядром (необхідний вихідний код програми).
3. Якщо програма доступна тільки в скомпільованому вигляді, то можливо використати режим бінарної сумісності (має сенс тільки для подібних ОС).

Ще більше абстракції

Програма користувача може бути розроблена на мові програмування, що інтерпретується, або виконується в віртуальній машині. В такому випадку програма користувача напряду не виконує команди процесора, та не звертається до пам'яті обчислювальної системи.

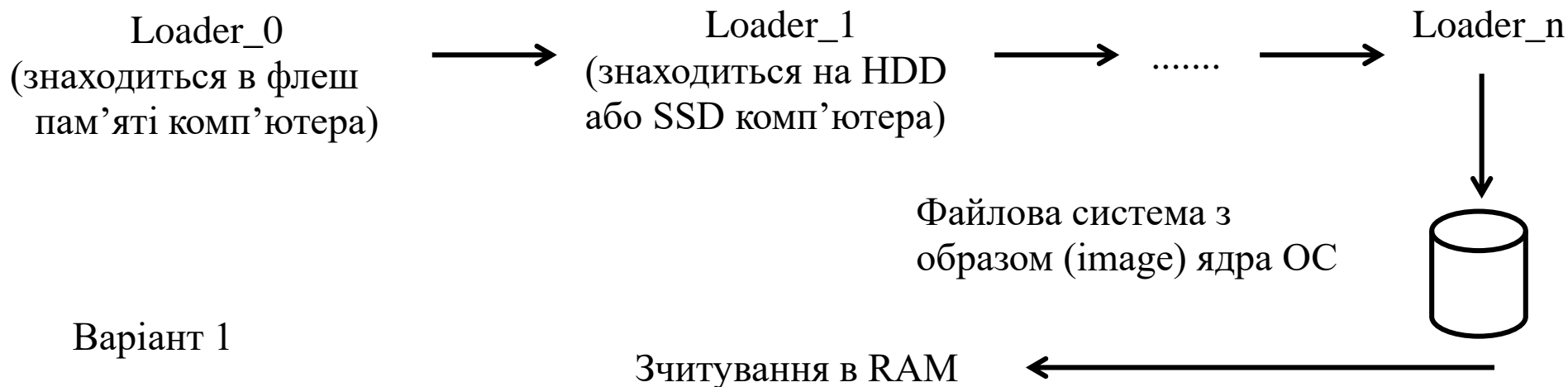
Але інтерпретатор або віртуальна машина виконується в оточенні створеним ядром ОС, тому взаємодіє з ядром ОС, виконує команди процесора та звертається до пам'яті.

Завдання ядра ОС

1. Абстрагування інтерфейсів.
2. Підтримка роботи програм користувача.
3. Робота з пам'яттю.
4. Файлові системи.
5. Підтримка мереж (стек мережових протоколів).
6. Безпека.

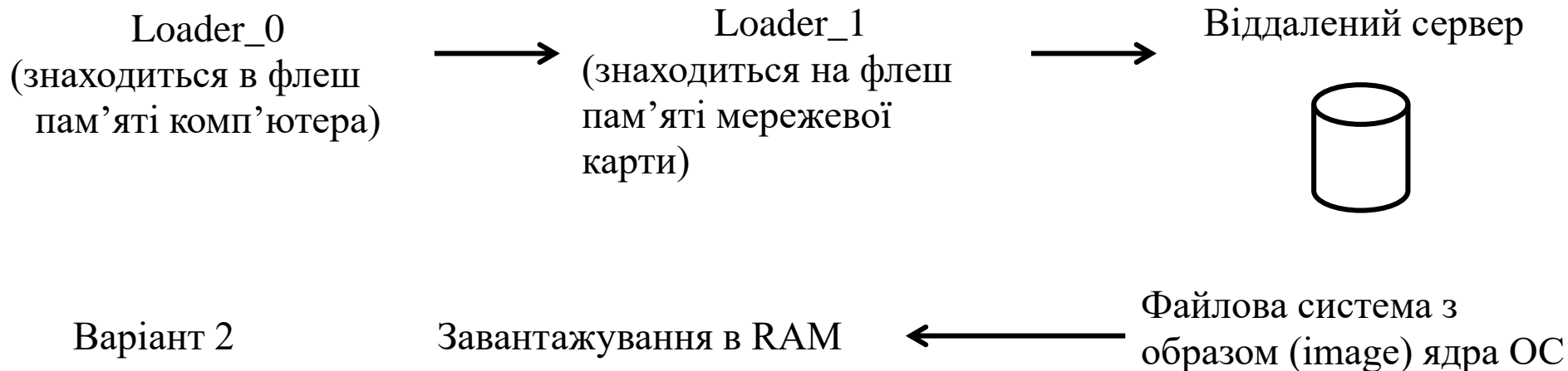
Завантаження ядра ОС в пам'ять.

Спеціальна програма **завантажувач** (loader) відповідає за завантаження ядра ОС в пам'ять комп'ютера.



Завантаження ядра ОС в пам'ять.

Спеціальна програма **завантажувач** (loader) відповідає за завантаження ядра ОС в пам'ять комп'ютера.



Класифікація ОС за призначенням

1. ОС загального призначення (general purpose OS). Не гарантують наявність обіцяних ресурсів, не обмежують в часі виконання запитів.
2. ОС реального часу (real-time OS). Гарантують виконання запиту в межах наперед встановленого часового вікна.
3. Вбудовані ОС (embedded OS). Зазвичай це системи загального призначення з додаванням підтримки необхідного устаткування.
4. Спеціалізовані ОС. Вирішують конкретні завдання в конкретній обчислювальній системі.
5. Розподілені ОС (distributed OS). Підтримують прозоре (transparent) використання ресурсів.