Операційні системи

Лекція 7

Керування оперативною пам'яттю

План лекції

- Завдання керування пам'яттю
- Типи адрес
- Пласка і сегментна моделі пам'яті
- Методи розподілу пам'яті
- Розподіл пам'яті без застосування дискового простору
- Оверлеї
- Свопінг
- Віртуальна пам'ять

Завдання керування пам'яттю

- Відстеження вільної та зайнятої пам'яті
- Виділення пам'яті процесам і звільнення пам'яті після завершення процесу
- Витіснення процесів з оперативної пам'яті на диск і повернення їх в оперативну пам'ять (віртуальна пам’ять)
- Перетворення адрес

Типи адрес

- *Символьні адреси* (ідентифікатори змінних, мітки переходів у програмах на алгоритмічних мовах)
  - Транслятор
- *Віртуальні адреси* (умовні адреси)
  1. Переміщувальний завантажувач (статичне перетворення)
  2. Динамічне перетворення апаратними засобами
- *Фізичні адреси* (номери комірок фізичної пам'яті)

- Сукупність віртуальних адрес процесу називається *віртуальним адресним простором* (у загальному випадку не дорівнює обсягу фізичної пам'яті)

Моделі пам'яті

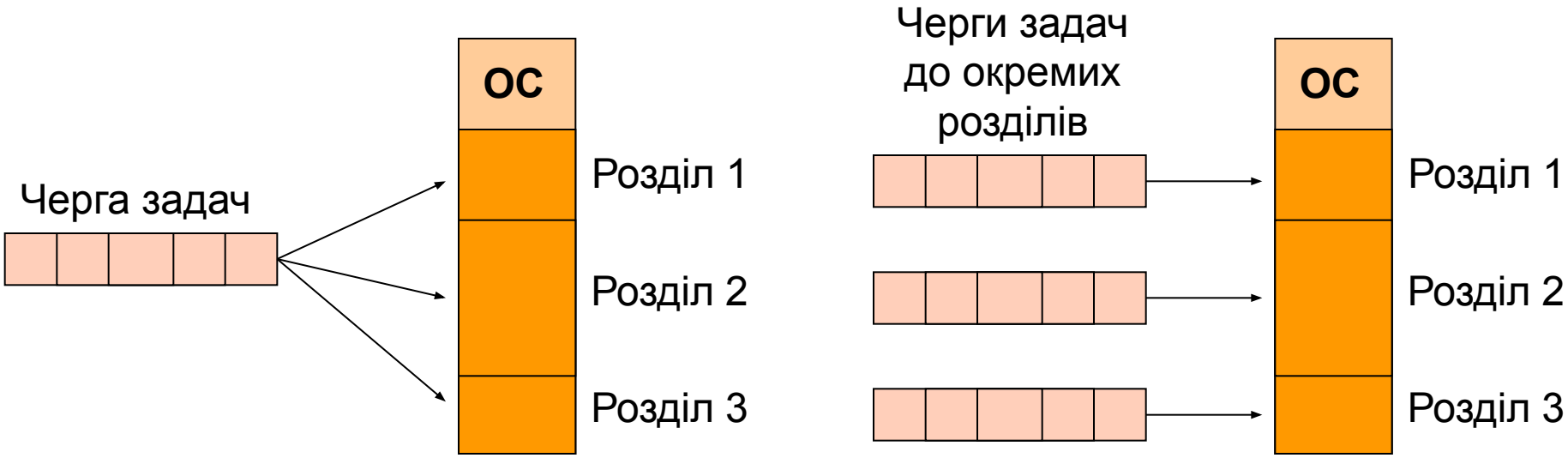
- Для забезпечення коректної адресації незалежно від розташування програми в оперативній пам’яті комп’ютера в якості віртуальних адрес використовуються *відносні адреси*, тобто зміщення від деякої *базової адреси*
- *Пласка* (*flat*) *модель пам’яті*
  - Кожному процесу виділяється єдина неперервна послідовність віртуальних адрес
  - Зміщення дозволяє однозначно вказати на положення даних або команди в адресному просторі процесу
- *Сегментна модель пам’яті*
  - Адресний простір процесу поділяється на окремі частини, які називаються *сегментами* (зустрічаються також інші назви: *секції*, *області*)
  - Віртуальна адреса задається парою чисел (*n*, *m*), де *n* визначає сегмент, а *m* – зміщення в даному сегменті
  - Сегментна модель є більш складною, але й більш гнучкою

Методи розподілу пам'яті

- Без застосування дискового простору
  - *Фіксовані розділи*
  - *Динамічні розділи* (розділи змінної величини)
  - *Переміщувані розділи*
- Із застосуванням дискового простору (віртуальна пам’ять)
  - *Сегментний розподіл*
  - *Сторінковий розподіл*
  - *Сегментно-сторінковий розподіл*

Фіксовані розділи

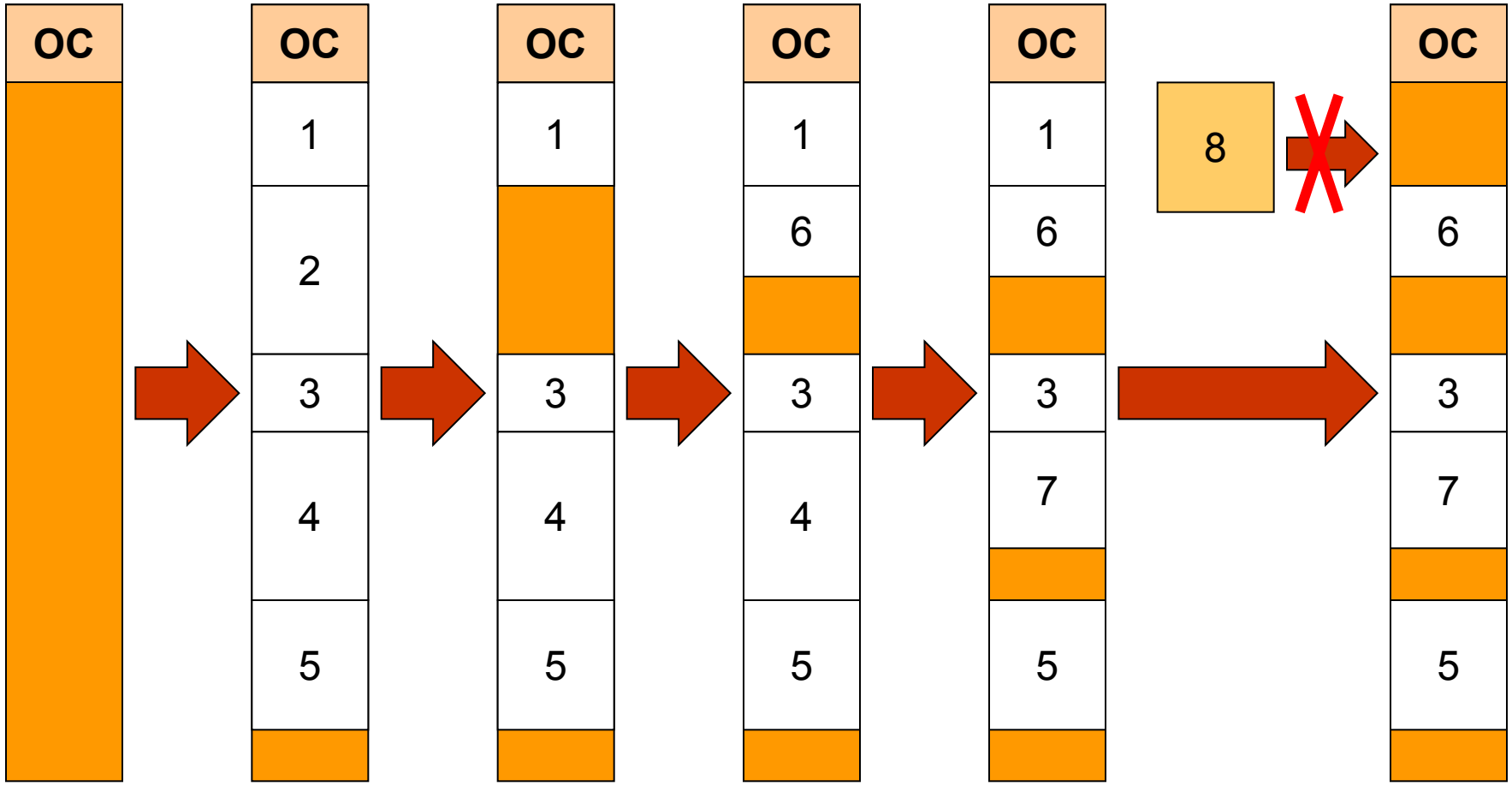
- Вибір розділу, що підходить за розміром
- Завантаження програми і налаштування адрес



Динамічні розділи

- Завдання ОС:
  - Ведення таблиць вільних і зайнятих областей (стартові адреси і розміри ділянок пам'яті)
  - Під час надходження нової задачі – аналіз запиту, перегляд таблиці вільних областей і вибір розділу за одним з алгоритмів:
    - Перший знайдений розділ достатнього розміру
    - Найменший розділ достатнього розміру
    - Найбільший розділ (достатнього розміру)
  - Завантаження задачі у виділений розділ і коригування таблиць вільних і зайнятих областей
  - По завершенні задачі – коригування таблиць вільних і зайнятих областей
- \+ Перевага
  - У процесі виконання програмний код не переміщується – можна налаштовувати адреси одноразово
- – Недолік
  - *Фрагментація*!!!

Динамічні розділи

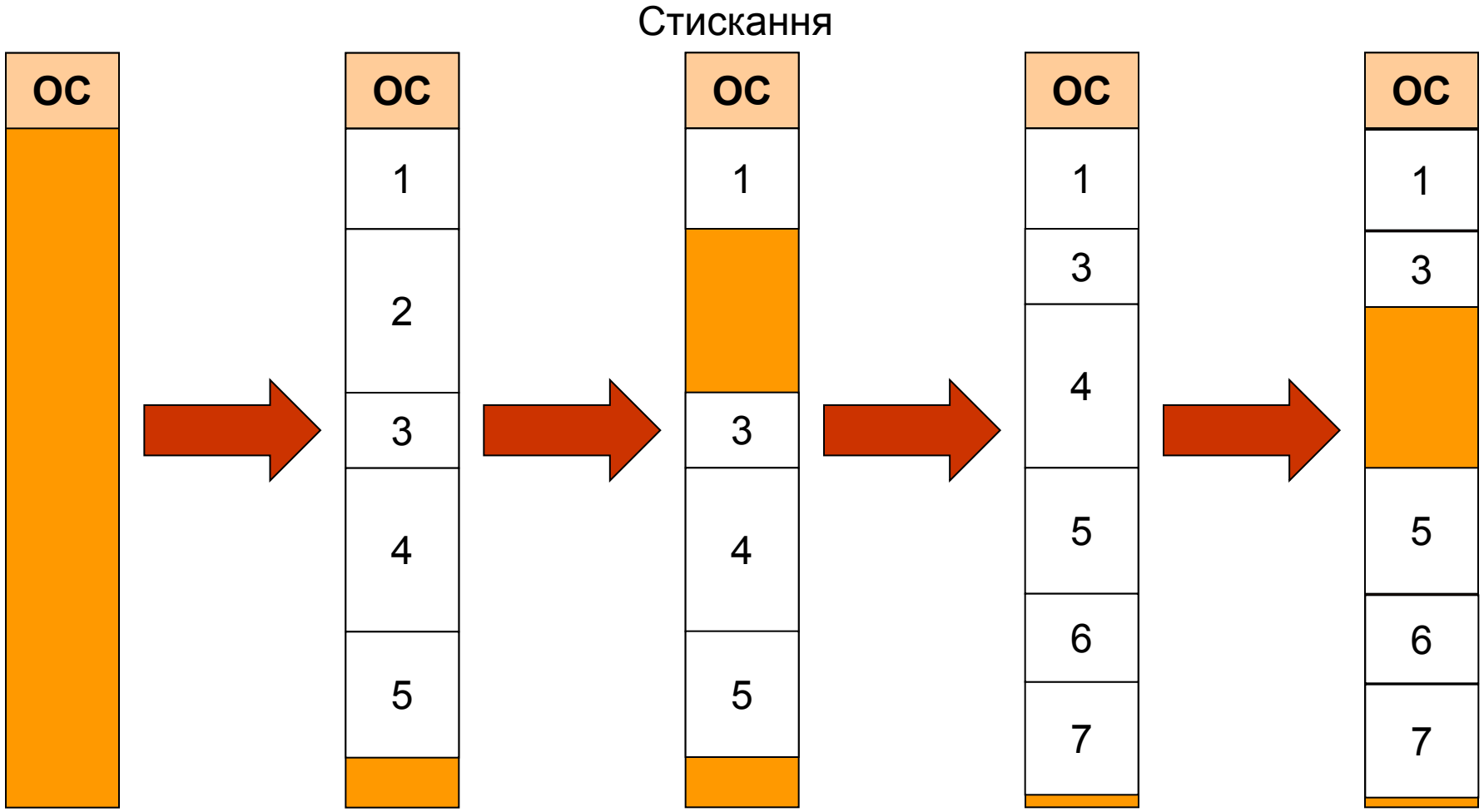


Переміщувані розділи

Те ж саме, що й динамічні розділи, плюс:

- Система періодично усуває фрагментацію пам'яті шляхом переміщення усіх розділів у бік більших (або менших) адрес
  - Процедура називається стискання пам'яті
  - Може виконуватись:
    - Або завжди, коли завершується задача
    - Або лише тоді, коли для нового розділу не вистачає пам'яті
- Недолік:
  - Необхідно динамічне перетворення адрес

Переміщувані розділи



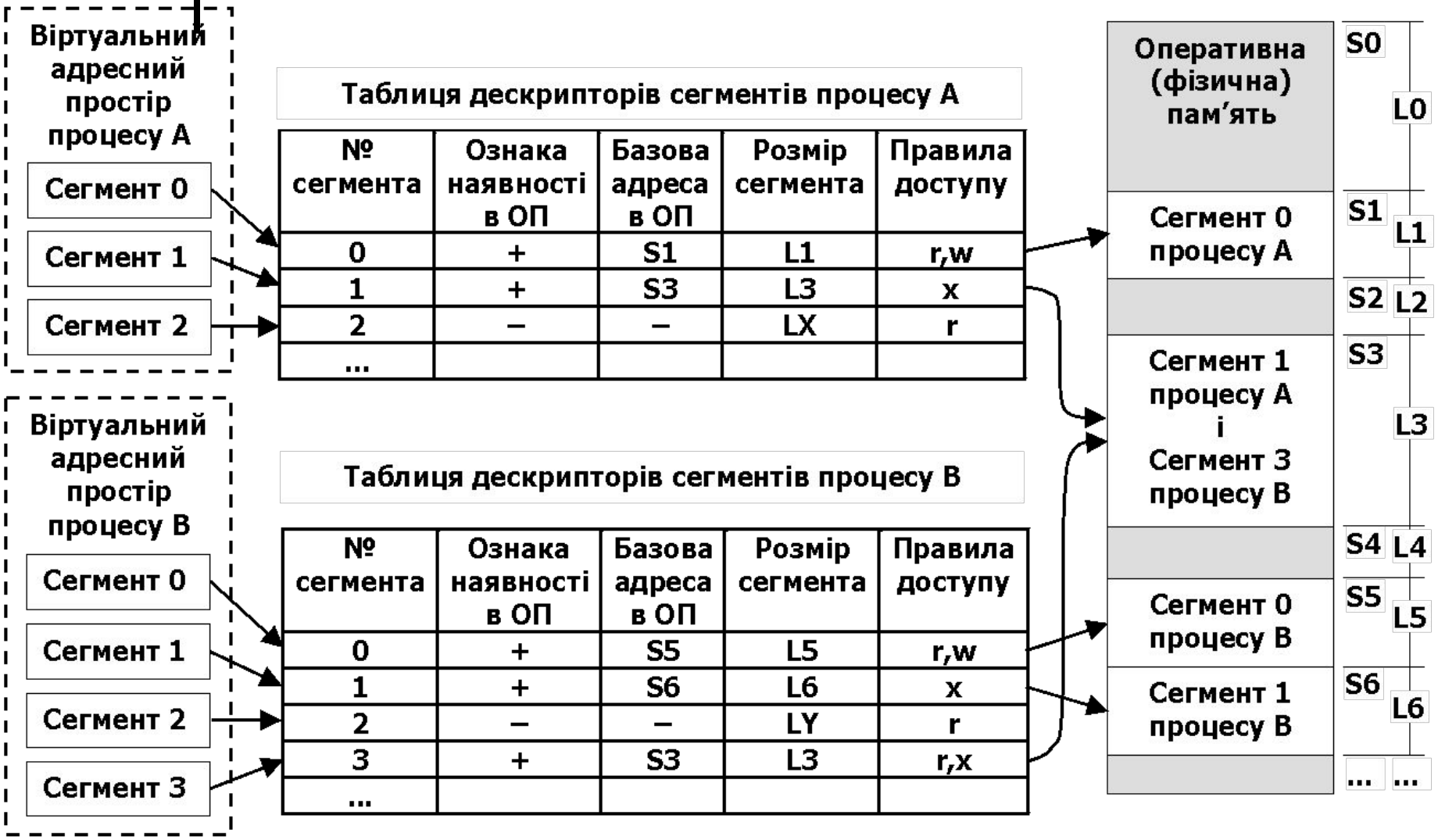
Розподіл пам'яті з використанням дискового простору

- *Оверлей* (*Overlay*)
  - У процесі виконання програми окремі програмні модулі завантажуються з диску
  - Реалізується засобами прикладних програм
- *Свопінг* (*Swapping*)
  - Процеси у стані очікування повністю вивантажуються на диск
- *Віртуальна пам’ять* (*Virtual Memory*)
  - Сукупність програмно-апаратних засобів, що дозволяє процесам використовувати більший обсяг пам’яті, ніж є наявної оперативної пам'яті
    - Розміщення коду і даних у пристроях пам'яті різного типу
    - Переміщення коду і даних між пристроями пам'яті різного типу
    - Перетворення віртуальних адрес у фізичні

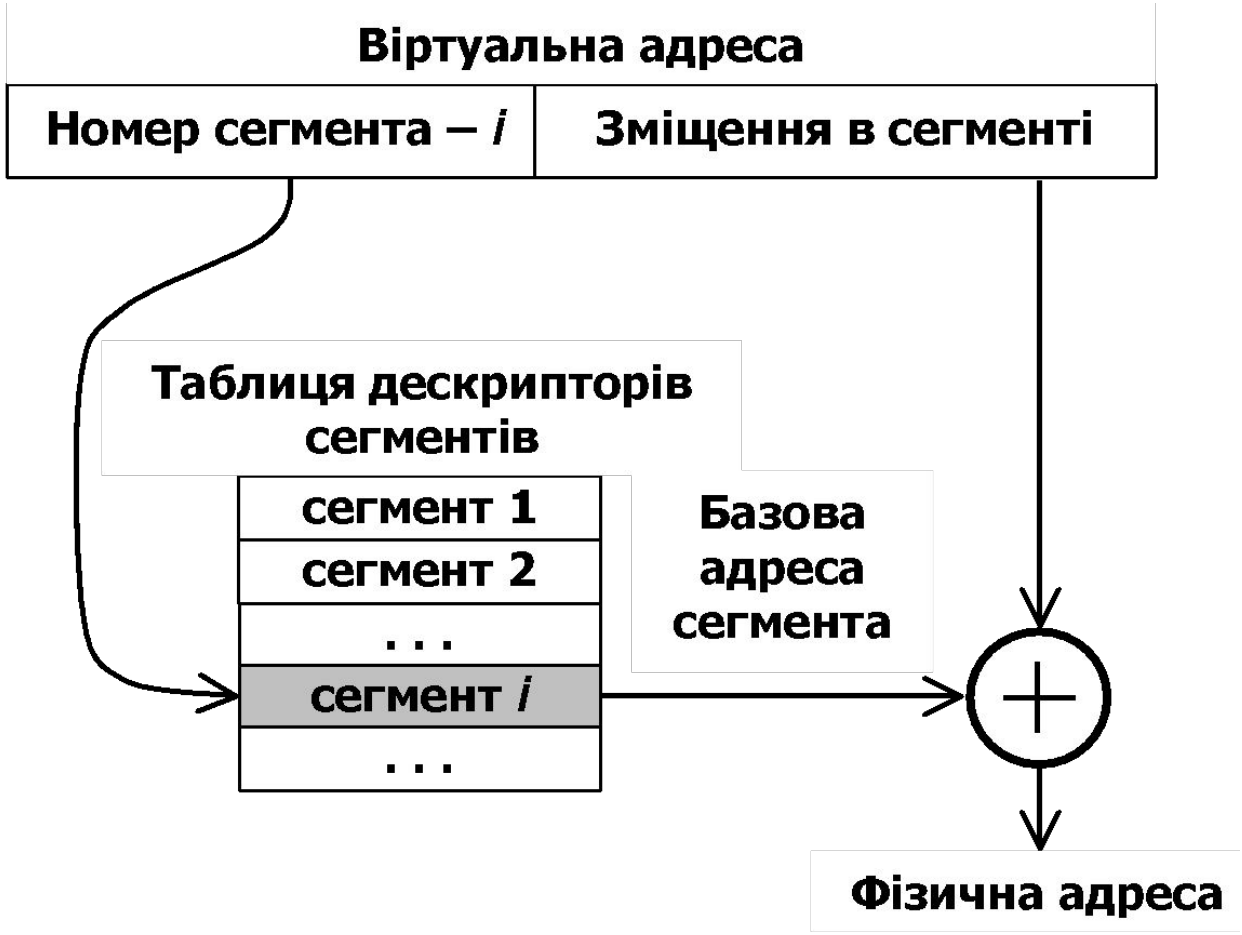
Сегментний розподіл пам'яті

- *Сегмент* – це неперервна область віртуального адресного простору довільного розміру, виділена з урахуванням типу даних, які в ній знаходяться
- Віртуальний адресний простір процесу складається з окремих сегментів, розмір кожного з яких обмежується розрядністю адресації
  - При 16-розрядній адресації – до 64 кБ
  - При 32-розрядній адресації – до 4 ГБ
- Відомості про сегменти оформлюються у вигляді таблиці, кожний рядок якої містить інформацію про окремий сегмент (*дескриптор сегмента*):
  - Базова фізична адреса процесу в оперативній пам’яті
  - Розмір сегмента
  - Тип сегмента
  - Правила доступу до сегмента
  - Ознака наявності сегмента в оперативній пам’яті
  - Ознака модифікації сегмента
  - Інші відомості
- Сегментний розподіл передбачає, що деякі сегменти можуть бути повністю витіснені на диск

Сегментний розподіл пам'яті



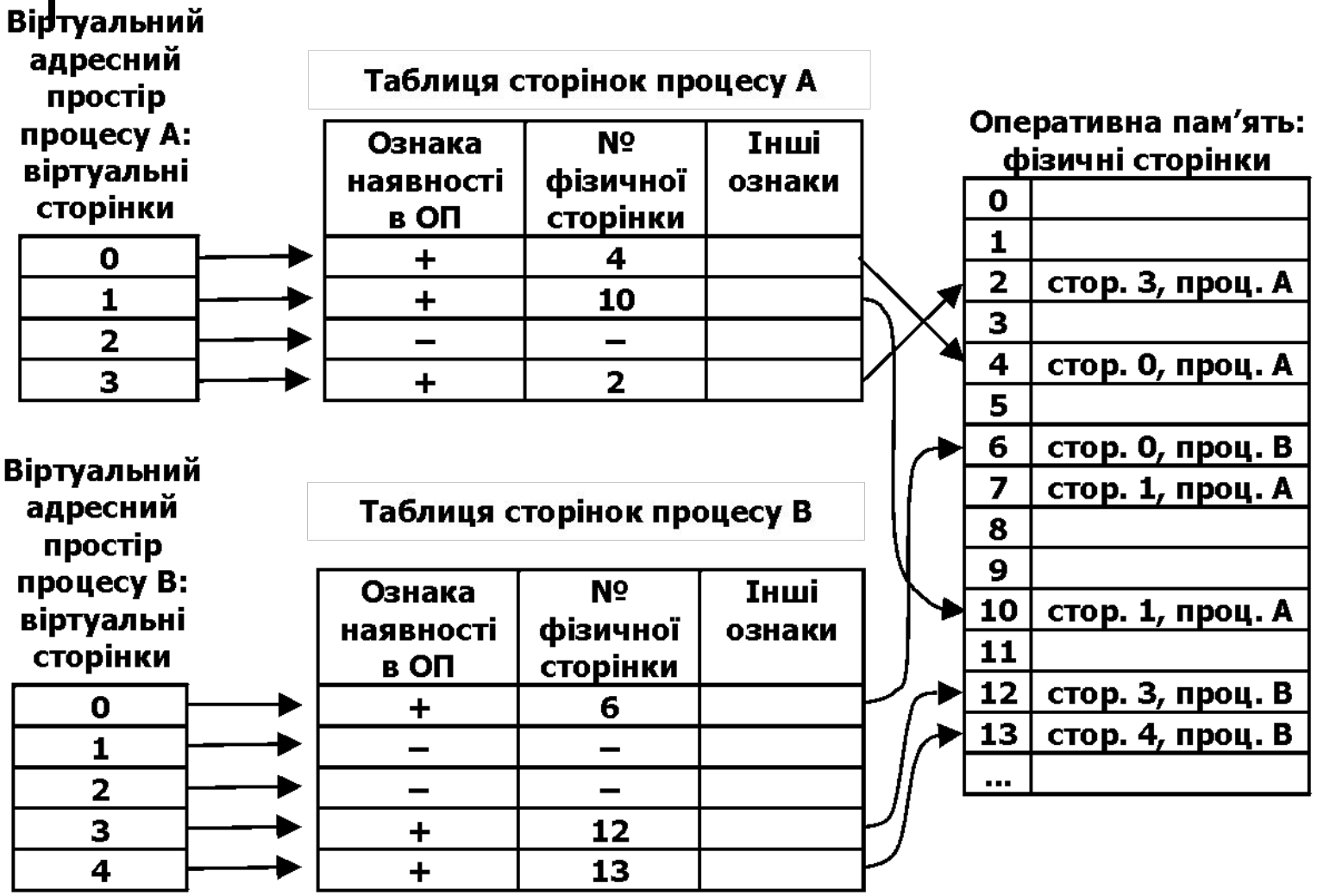
Трансляція віртуальної адреси при сегментній організації пам’яті



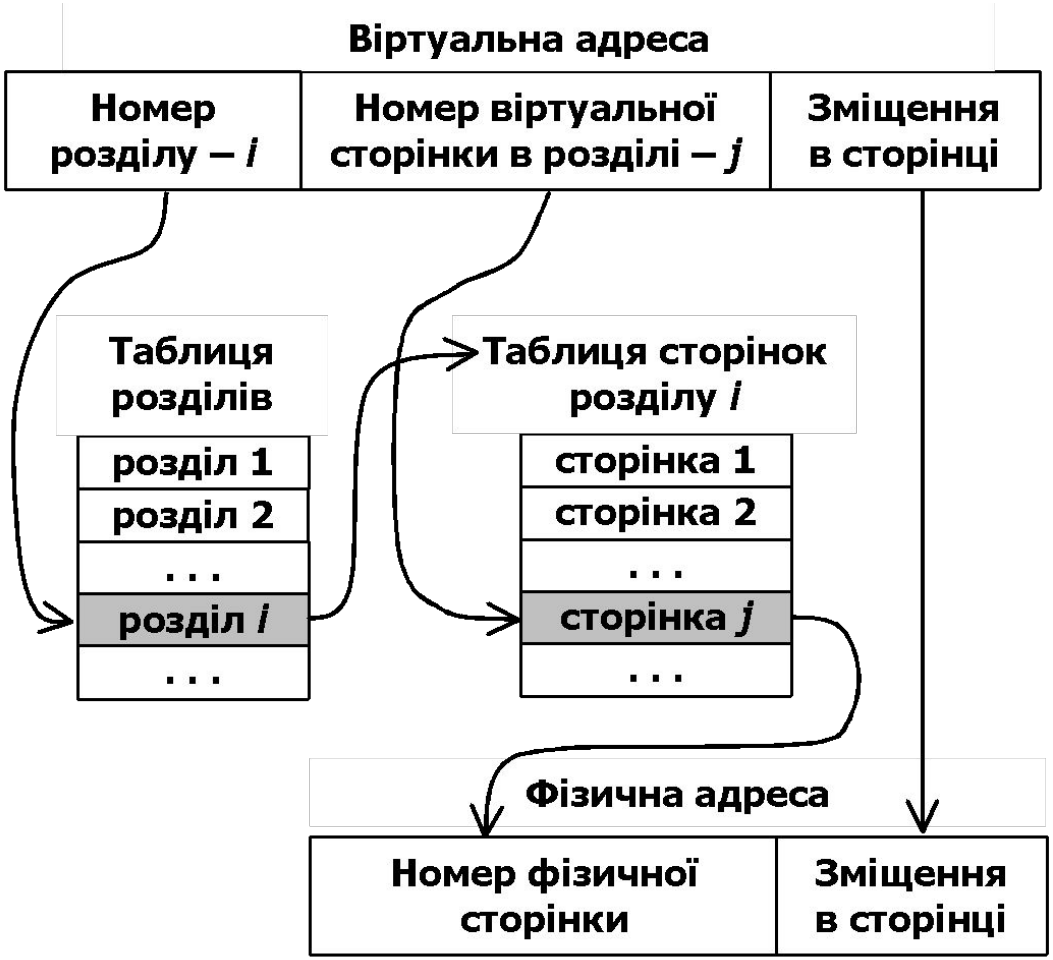
Сторінковий розподіл пам'яті

- *Сторінка* – це неперервна область віртуального адресного простору порівняно невеликого фіксованого розміру, виділена без урахування типу даних, які в ній знаходяться
- Для сторінок, як і для сегментів, застосовуються дескриптори, але структура їх значно простіша:
  - Номер фізичної сторінки в оперативній пам’яті, в яку завантажена ця віртуальна сторінка
  - Ознака наявності в оперативній пам’яті
  - Ознаку модифікації сторінки (якщо модифікації не було, в разі необхідності звільнити пам’ять сторінку можна просто “затерти”)
  - Ознаку звернення до сторінки (використовується для вибору сторінок-кандидатів для витіснення на диск)
- Типовий розмір сторінки – 4 кБ (величезна таблиця сторінок)
  - Віртуальний адресний простір поділяють на розділи однакового розміру, який підбирають таким чином, щоби таблиця сторінок одного розділу займала рівно одну сторінку
  - Для кожного розділу формують свою таблицю сторінок
  - Таблиці витискаються на диск разом із відповідними розділами
  - Дескриптори таблиць сторінок аналогічні дескрипторам звичайних сторінок, вони формують окрему таблицю, яку називають *таблицею розділів* або *каталогом сторінок*.

Сторінковий розподіл пам'яті



Трансляція віртуальної адреси при дворівневій сторінковій організації пам’яті



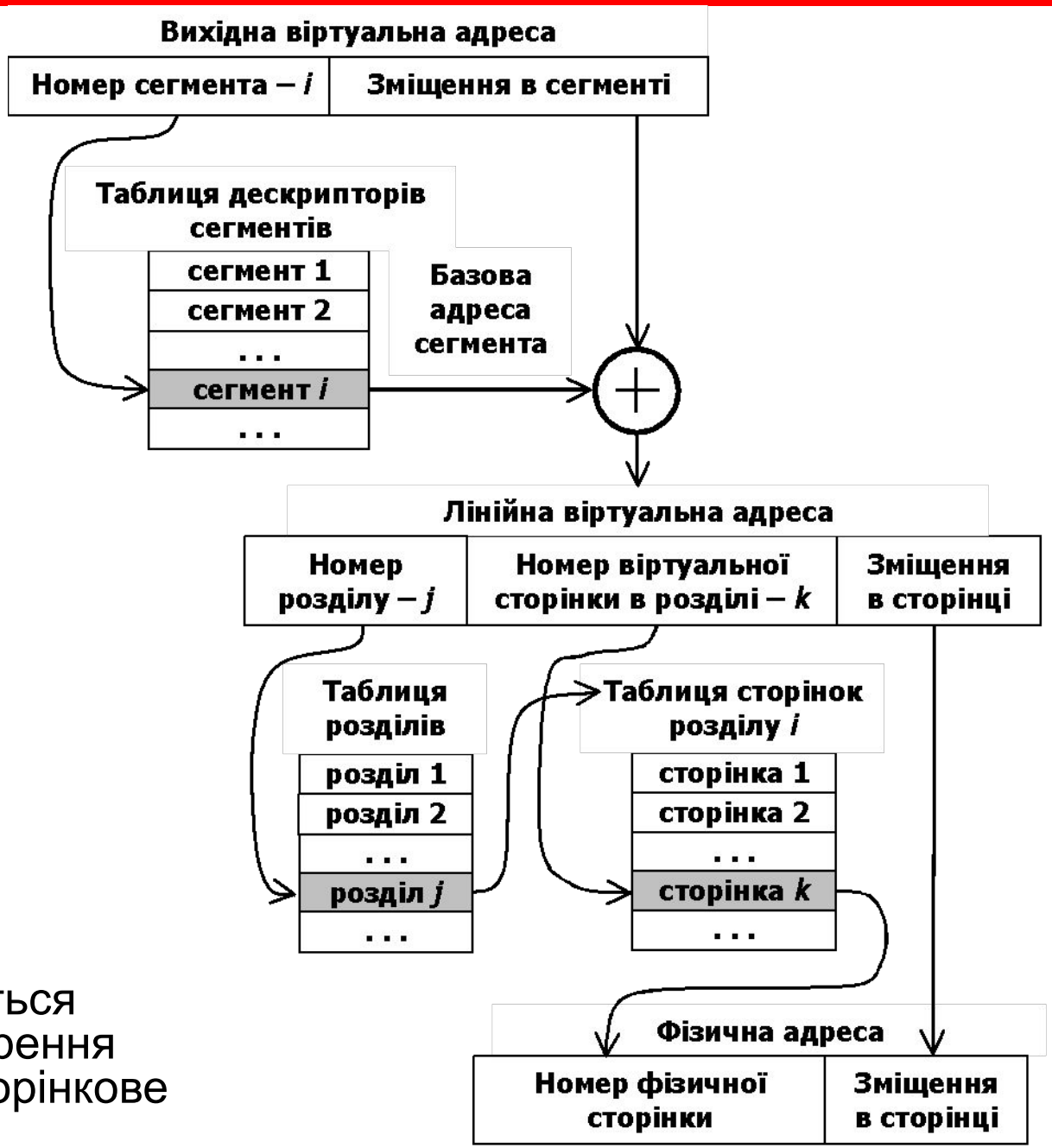
Завантаження-вивантаження сторінок

- Під час кожного звернення до пам'яті здійснюється зчитування інформації про віртуальну сторінку з таблиці сторінок
- Якщо сторінка є в пам'яті – здійснюється перетворення віртуальної адресу у фізичну
- Якщо сторінки немає – здійснюється така послідовність дій:
  - Сторінкове переривання
  - Процес переводять у стан очікування
  - Обробник сторінкового переривання знаходить сторінку на диску і намагається завантажити її у пам’ять
  - Якщо вільне місце є, сторінка завантажується у пам’ять
  - Якщо місця немає – здійснюється вибір сторінки, яку необхідно вивантажити з пам'яті
    - Перша знайдена сторінка
    - Сторінка, що довше за усіх не використовувалась
    - Сторінка, звернень до якої було менше за усіх
  - Якщо обрану сторінку модифікували – її записують на диск
  - Якщо обрану сторінку не модифікували – її просто видаляють з пам'яті

Сегментний і сторінковий розподіли пам’яті: переваги і недоліки

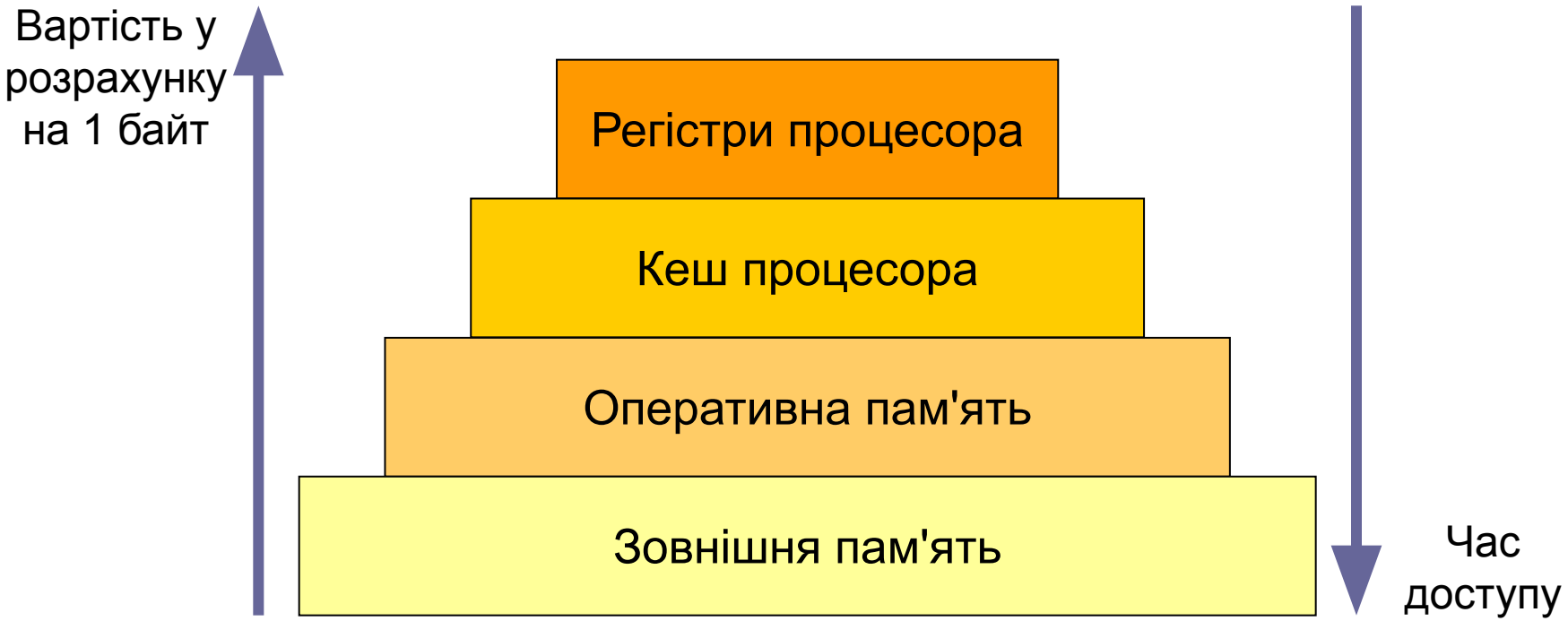
- Перевага сегментів – в їх типізації
  - Дозволяє здійснювати диференційоване керування доступом у відповідності до типу даних, що містяться у сегменті:
    - Заборона записування у сегмент, де містяться коди програми
    - Заборона виконання процесором фрагментів програмного коду, що містяться у сегменті даних
  - Сегментний розподіл є основою для реалізації захисту областей пам’яті
- Перевага сторінок – в однаковому і невеликому розмірі
  - Легше і швидше завантажити і вивантажити певну кількість сторінок однакового розміру, ніж один великий сегмент
  - Сторінковий розподіл переважно застосовується для реалізації механізму обміну інформацією між фізичною пам’яттю і диском.

Сегментно-сторінковий розподіл пам’яті



- Спочатку здійснюється сегментне перетворення адреси, а далі – сторінкове

Ієрархія пристроїв пам'яті



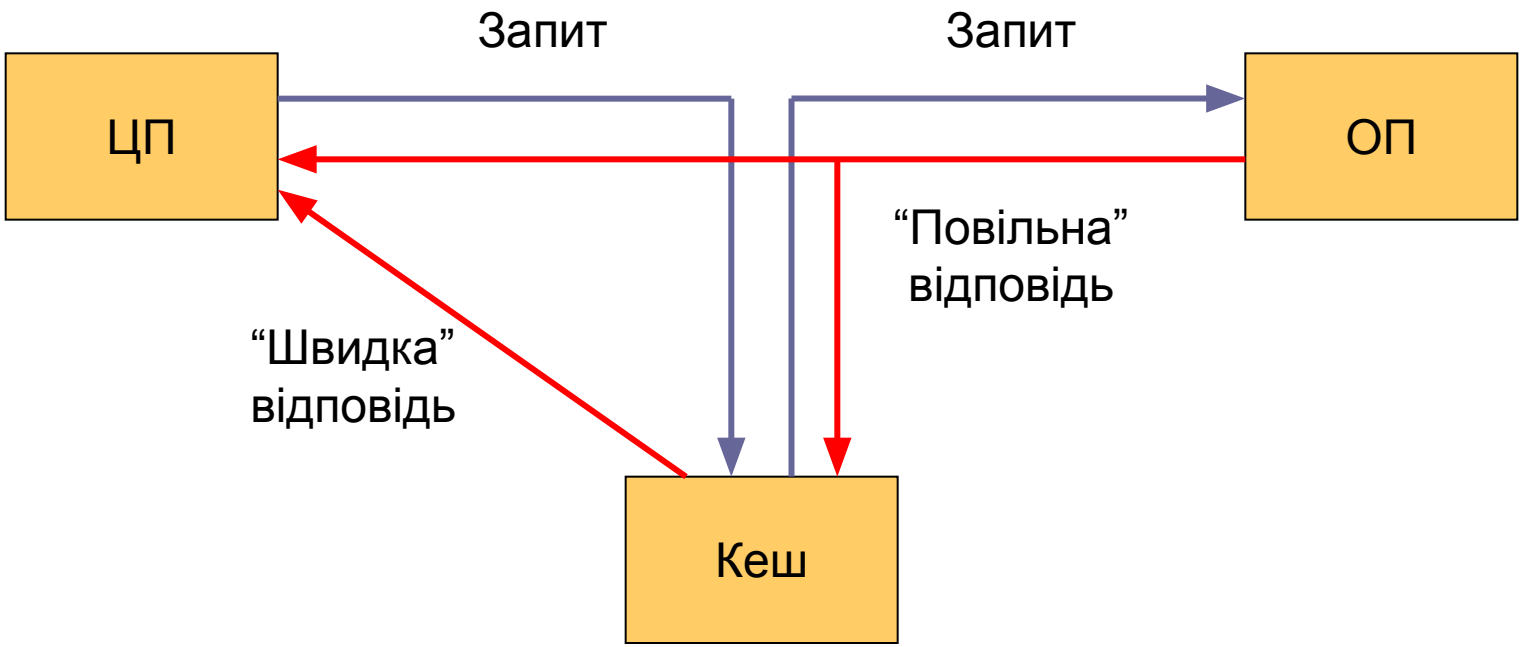
Кеш-пам'ять

- *Кеш-пам’ять* – це спосіб організації сумісного функціонування 2-х типів пристроїв пам'яті, що дозволяє знизити середній час доступу до даних за рахунок копіювання у “швидкий” пристрій частини даних з “повільного” пристрою
- Іноді кеш-пам'яттю називають не лише спосіб, але й сам швидкий пристрій
- Дані у кеш-пам'яті зберігаються прозоро (немає власної адресації, застосовуються адреси з повільного пристрою)
- Середній час доступу:

$$t = t_{\text{повільн}} (1 - p) + t_{\text{швидк}}\, p$$

p – ймовірність потрапляння в кеш (велика! ~0,9)

Схема функціонування



Локальність даних

- *Часова локальність*
  - Якщо відбулося звернення до пам'яті за певною адресою, то з великою ймовірністю найближчим часом відбудеться повторне звернення за тією ж адресою
  - Обґрунтовує ефективність копіювання даних в кеш під час зчитування
- *Просторова локальність*
  - Якщо відбулося звернення до пам'яті за певною адресою, то з великою ймовірністю наступне звернення відбудеться за сусідньою адресою
  - Обґрунтовує ефективність випереджаючого зчитування даних в кеш