

ТЕМА 3.1

ОРГАНІЗАЦІЯ ПАМ'ЯТІ КОМП'ЮТЕРА. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

ПЛАН

1 ФІЗИЧНА ОРГАНІЗАЦІЯ ПАМ'ЯТІ КОМП'ЮТЕРА.
ЛОКАЛЬНІСТЬ

2 ЛОГІЧНА ПАМ'ЯТЬ. ЗВ'ЯЗУВАННЯ АДРЕС

3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ
СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

Головне завдання комп'ютерної системи - виконувати програми. Програми разом з даними, до яких вони мають доступ, в процесі виконання повинні (принаймні частково) знаходитися в оперативній пам'яті.

Операційній системі доводиться вирішувати задачу розподілу пам'яті між призначеними для користувача процесами і компонентами ОС. Ця діяльність називається **управлінням пам'яттю**. Таким чином, пам'ять є найважливішим ресурсом, що вимагає ретельного управління.

Частина ОС, яка відповідає за управління пам'яттю, називається **менеджером пам'яті**.

1 ФІЗИЧНА ОРГАНІЗАЦІЯ ПАМ'ЯТІ КОМП'ЮТЕРА. ЛОКАЛЬНІСТЬ

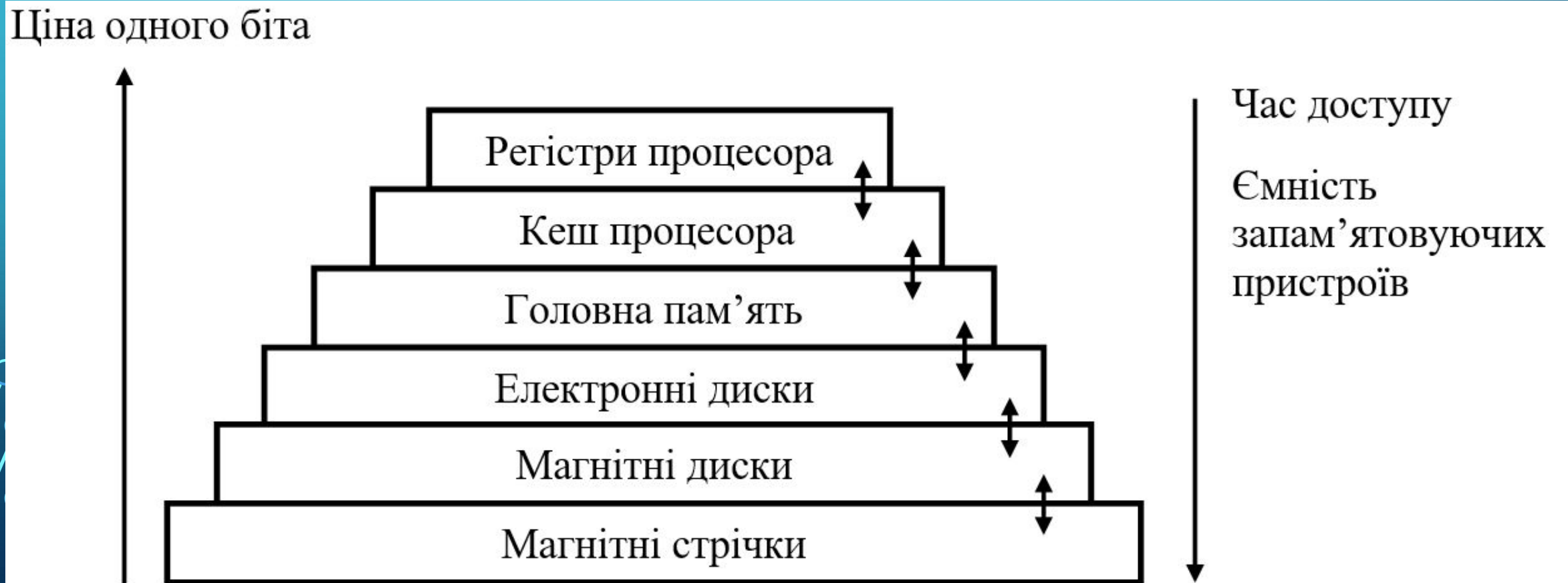
Запам'ятовуючі пристрої комп'ютера, розділяють, як мінімум, на два рівні: **основну** (головну, оперативну, фізичну) і **вторинну** (зовнішню) пам'ять.

Основна пам'ять є впорядкованим масивом однобайтових комірок, кожна з яких має свою унікальну адресу (номер). Процесор витягає команду з основної пам'яті, декодує і виконує її. Для виконання команди може бути потрібно звернення ще до декількох елементів основної пам'яті. Зазвичай основна пам'ять виготовляється із застосуванням напівпровідникових технологій і втрачає свій вміст при відключенні живлення.

Вторинну пам'ять (це головним чином диски) також можна розглядати як одновимірний лінійний адресний простір, що складається з послідовності байтів. На відміну від оперативної пам'яті, вона є незалежною, має істотно велику місткість і використовується як розширення основної пам'яті.

1 ФІЗИЧНА ОРГАНІЗАЦІЯ ПАМ'ЯТІ КОМП'ЮТЕРА. ЛОКАЛЬНІСТЬ

Цю схему можна доповнити ще декількома проміжними рівнями, як показано на рисунку 1. Різновиди пам'яті можуть бути об'єднані в ієрархію по убутанню часу доступу, зростанню ціни і збільшенню місткості.



1 ФІЗИЧНА ОРГАНІЗАЦІЯ ПАМ'ЯТІ КОМП'ЮТЕРА. ЛОКАЛЬНІСТЬ

Багаторівневу схему використовують таким чином. Інформація, яка знаходиться в пам'яті верхнього рівня, зазвичай зберігається також на рівнях з великими номерами. Якщо процесор не виявляє потрібну інформацію на i -му рівні, він починає шукати її на наступних рівнях. Коли потрібна інформація знайдена, вона переноситься в швидші рівні.

Виявляється, при такому способі організації у міру зниження швидкості доступу до рівня пам'яті знижується також і частота звернень до нього.

Ключову роль тут грає **властивість реальних програм, протягом обмеженого відрізка часу здатних працювати з невеликим набором адрес пам'яті. Це емпірично спостережувана властивість відома як принцип локальності або локалізації звернень.**

1 ФІЗИЧНА ОРГАНІЗАЦІЯ ПАМ'ЯТІ КОМП'ЮТЕРА. ЛОКАЛЬНІСТЬ

Властивість локальності пояснювана, якщо врахувати, як пишуться програми і як зберігаються дані, тобто зазвичай протягом якогось відрізка часу обмежений фрагмент коду працює з обмеженим набором даних. Цю частину коду і даних вдається розмістити в пам'яті з швидким доступом.

В результаті реальний час доступу до пам'яті визначається часом доступу до верхніх рівнів, що і обумовлює ефективність використання ієрархічної схеми.

Визначення

Адреси в основній пам'яті, що характеризують реальне розташування даних у фізичній пам'яті, називаються **фізичними адресами**. Набір фізичних адрес, з яким працює програма, називають **фізичним адресним простором**.

2 ЛОГІЧНА ПАМ'ЯТЬ. ЗВ'ЯЗУВАННЯ АДРЕС

Більшість програм є набором модулів, створених незалежно один від одного. Іноді усі модулі, що входять до складу процесу, розташовуються в пам'яті один за іншим, утворюючи **лінійний простір адрес**. Проте частіше модулі поміщаються в різні області пам'яті і використовуються по-різному.

Схема управління пам'яттю, що підтримує цей погляд на те, як зберігаються програми і дані, називається **сегментацією**. **Сегмент - область пам'яті певного призначення, усередині якої підтримується лінійна адресація**. Сегменти містять процедури, масиви, стек або скалярні величини, але зазвичай не містять інформацію змішаного типу.

Сегменти пам'яті з'явилися у зв'язку з необхідністю усупільнення процесами фрагментів програмного коду (текстовий редактор, тригонометричні бібліотеки і т. д.), без чого кожен процес повинен був зберігати у своєму адресному просторі дублюючу інформацію.

Пам'ять, таким чином, перестала бути лінійною і перетворилася на двовимірну. Адреса складається з двох компонентів: номер сегменту, зміщення усередині сегменту.

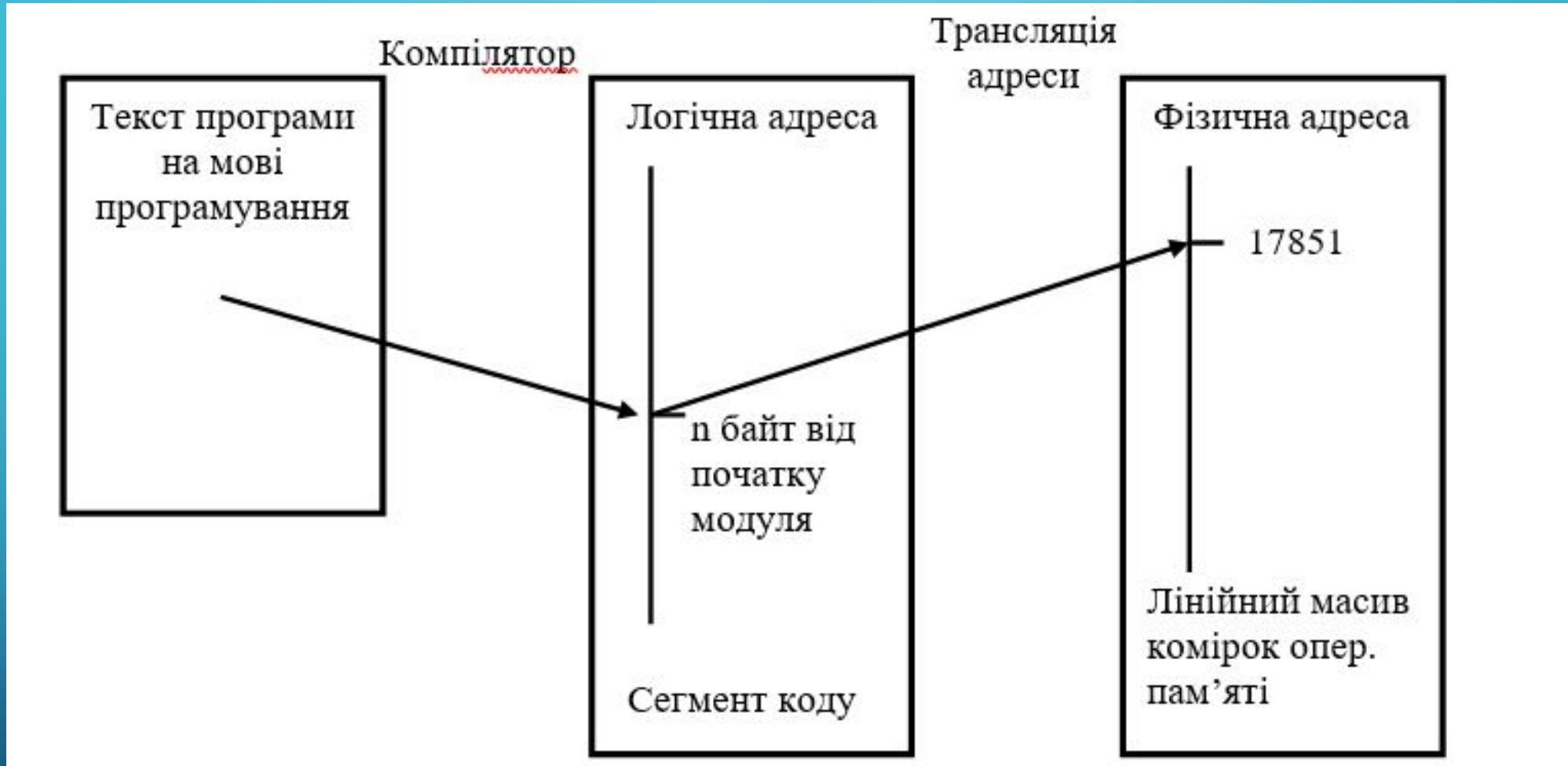
2 ЛОГІЧНА ПАМ'ЯТЬ. ЗВ'ЯЗУВАННЯ АДРЕС

Адреси, до яких звертається процес, таким чином, відрізняються від адрес, реально існуючих в оперативній пам'яті. Подібна адреса, згенерована програмою, зазвичай називається **логічною (у системах з віртуальною пам'яттю вона часто називається віртуальною) адресою**. Сукупність усіх логічних адрес називається **логічним (віртуальним) адресним простором**.

Отже, логічні і фізичні адресні простори ні по організації, ні за розміром не відповідають один одному. Максимальний розмір логічного адресного простору зазвичай визначається розрядністю процесора (наприклад, 2^{32} чи 2^{64}) і в сучасних системах значно перевищує розмір фізичного адресного простору.

Отже, процесор і ОС мають бути здатні відобразити посилання в кодї програми в реальні фізичні адреси, відповідні поточному розташуванню програми в основній пам'яті. Таке відображення адрес називають **трансляцією (прив'язкою) адреси або зв'язуванням адрес** (рисунок 2).

2 ЛОГІЧНА ПАМ'ЯТЬ. ЗВ'ЯЗУВАННЯ АДРЕС



Зв'язування логічної адреси, породженої оператором програми, з фізичною повинно бути здійснено до початку виконання оператора або у момент його виконання.

3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

Щоб забезпечити ефективний контроль використання пам'яті, ОС повинна виконувати наступні функції:

- ✓ відображення адресного простору процесу на конкретні області фізичної пам'яті;
- ✓ розподіл пам'яті між конкуруючими процесами;
- ✓ контроль доступу до адресних просторів процесів;
- ✓ вивантаження процесів (цілком або частково) в зовнішню пам'ять, коли в оперативній пам'яті недостатньо місця;
- ✓ облік вільної і зайнятої пам'яті.

3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

Розглянемо ряд конкретних схем управління пам'яттю. Кожна схема включає певну ідеологію управління, а також алгоритми і структури даних і залежить від архітектурних особливостей використовуваної системи.

Схема з фіксованими розділами. Найпростішим способом управління оперативною пам'яттю є її попереднє (зазвичай на етапі генерації або у момент завантаження системи) розбиття на декілька розділів фіксованої величини. Процеси, що поступають, поміщаються в той або інший розділ. При цьому відбувається умовне розбиття фізичного адресного простору. Зв'язування логічних і фізичних адрес процесу відбувається на етапі його завантаження в конкретний розділ, іноді - на етапі компіляції.

Кожен розділ може мати свою чергу процесів, а може існувати і глобальна черга для усіх розділів (див. рисунок 3.1.3).

Ця схема була реалізована в IBM OS/360 (MFT), DEC RSX - 11 і ряду інших систем.

Підсистема управління пам'яттю оцінює розмір процесу, що поступив, вибирає відповідний для нього розділ, здійснює завантаження процесу в цей розділ і налаштування адрес.

3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

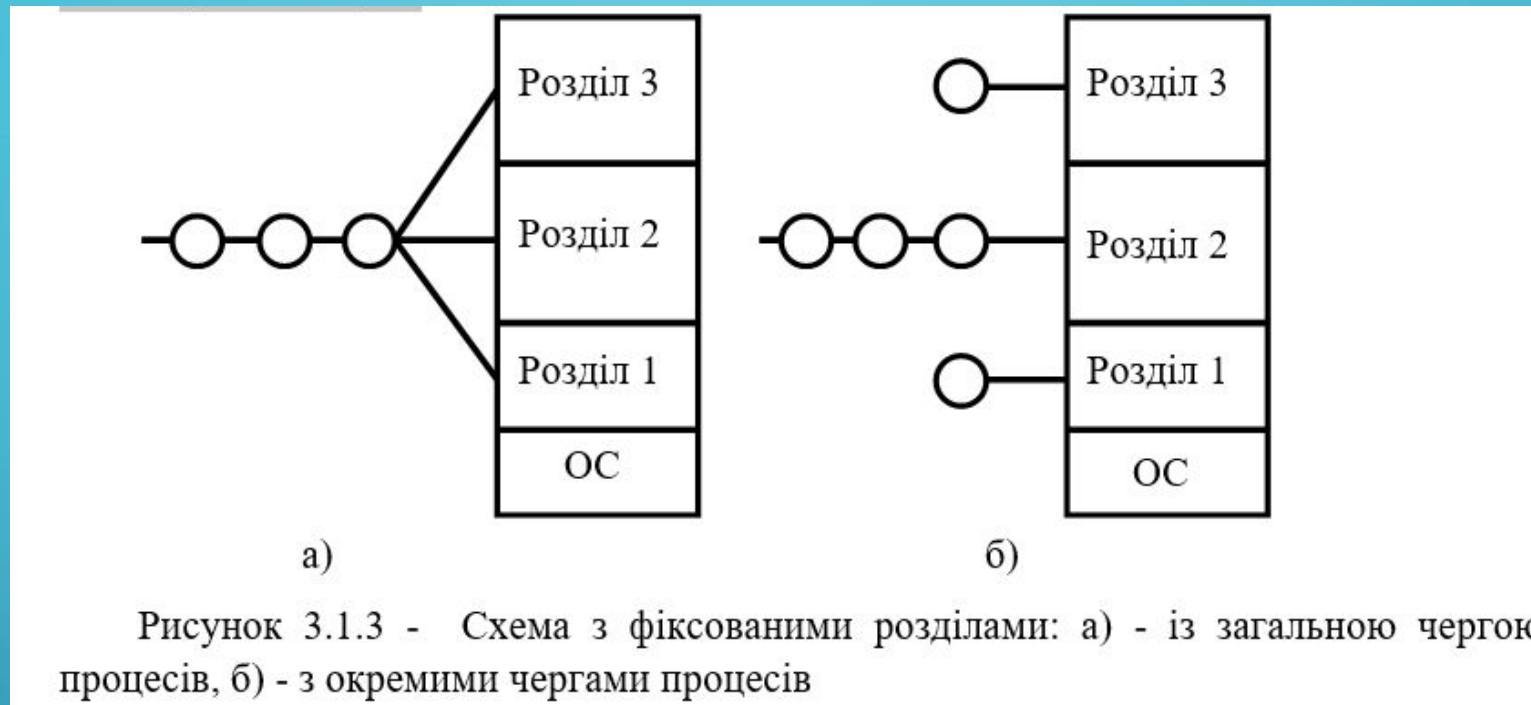


Рисунок 3.1.3 - Схема з фіксованими розділами: а) - із загальною чергою процесів, б) - з окремими чергами процесів

Недоліки схеми:

- число одночасно виконуваних процесів обмежене числом розділів.
- запропонована схема сильно страждає від внутрішньої фрагментації - втрати частини пам'яті, виділеної процесу, але не використовуваною їм. Фрагментація виникає тому, що процес не повністю займає виділений йому розділ або тому, що деякі розділи занадто малі для виконуваних призначених для користувача програм.

3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

Схема із змінними розділами. Ефективніше представляється схема динамічного розподілу або схема із змінними розділами. В цьому випадку спочатку уся пам'ять вільна і не розділена заздалегідь на розділи. Завданню, що знову поступає, виділяється строго необхідна кількість пам'яті, не більше. Після вивантаження процесу пам'ять тимчасово звільняється. Після закінчення деякого часу пам'ять є змінним числом розділів різного розміру (рисунок 3.1.5). Суміжні вільні ділянки

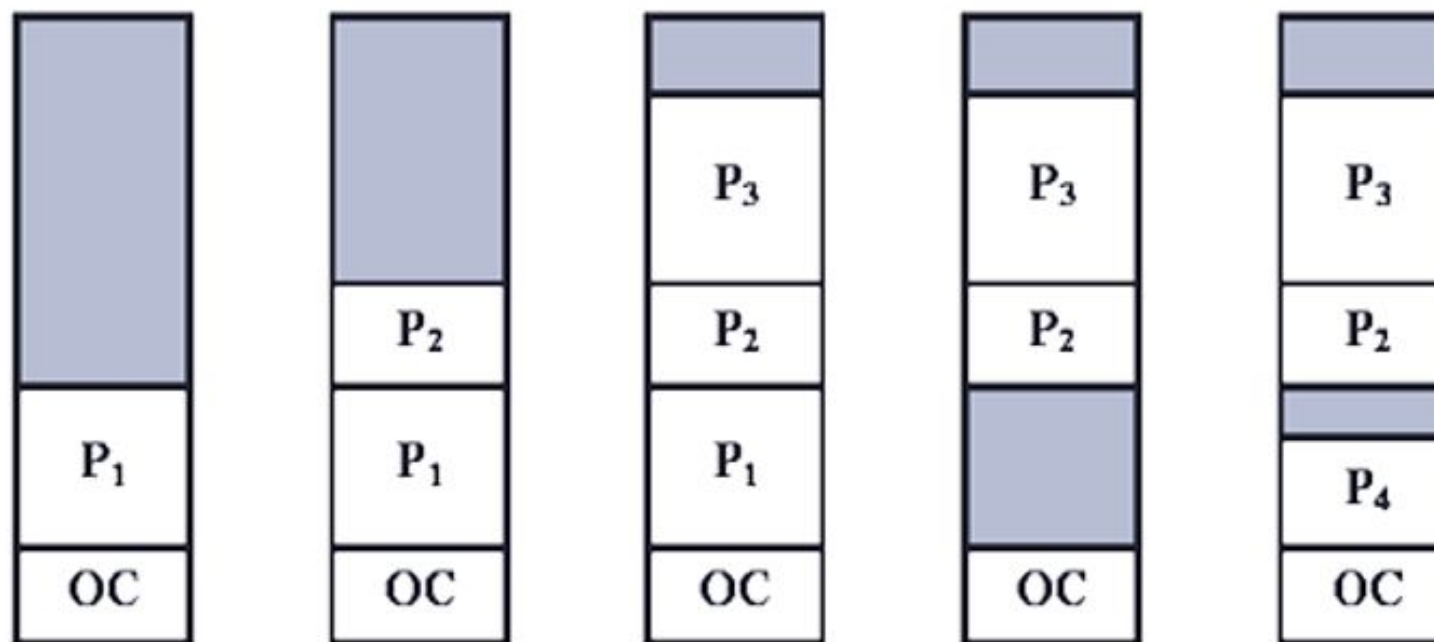


Рисунок 3.1.5 - Динаміка розподілу пам'яті між процесами (сірим кольором показана невживана пам'ять)

3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

Постає питання: «У який вільний розділ поміщати процес, що завантажується?» Найбільш поширено три стратегії:

- ✓ **стратегія першого відповідного (First fit).** Процес поміщається в перший відповідний за розміром розділ;
- ✓ **стратегія найбільш відповідного (Best fit).** Процес поміщається в той розділ, де після його завантаження залишиться менше всього вільного місця;
- ✓ **стратегія найменш відповідного (Worst fit).** При приміщенні в найбільший розділ в ньому залишається досить місця для можливого розміщення ще одного процесу.

Недоліки схеми: зовнішня фрагментація - наявність великого числа ділянок невживаної пам'яті, не виділеної жодному процесу.

3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

Одне з вирішень проблеми зовнішньої фрагментації - організувати стискування, тобто переміщення усіх зайнятих (вільних) ділянок у бік зростання (убування) адрес, так, щоб уся вільна пам'ять утворила безперервну область. Цей метод іноді називають **схемою з переміщуваними розділами**. У ідеалі фрагментація після стискування має бути відсутньою. Проте, алгоритм вибору оптимальної стратегії стискування дуже важкий і, як правило, стискування здійснюється в комбінації з вивантаженням і завантаженням по інших адресах.

3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

Сторінкова пам'ять. Описані вище схеми недостатньо ефективно використовують пам'ять, тому в сучасних схемах управління пам'яттю не прийнято розміщувати процес в оперативній пам'яті одним безперервним блоком.

У найпростішому і найбільш поширеному випадку сторінкової організації пам'яті (чи paging) як логічний адресний простір, так і фізичний представляються такими, що складаються з наборів блоків або сторінок однакового розміру. При цьому утворюються **логічні сторінки (page)**, а відповідні одиниці у фізичній пам'яті називають **фізичними сторінками або сторінковими кадрами (page frames)**. Сторінки (і сторінкові кадри) мають фіксовану довжину, що зазвичай є мірою числа 2, і не можуть перекриватися. Кожен кадр містить одну сторінку даних. При такій організації зовнішня фрагментація відсутня, а втрати із-за внутрішньої фрагментації, оскільки процес займає ціле число сторінок, обмежені частиною останньої сторінки процесу.

3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

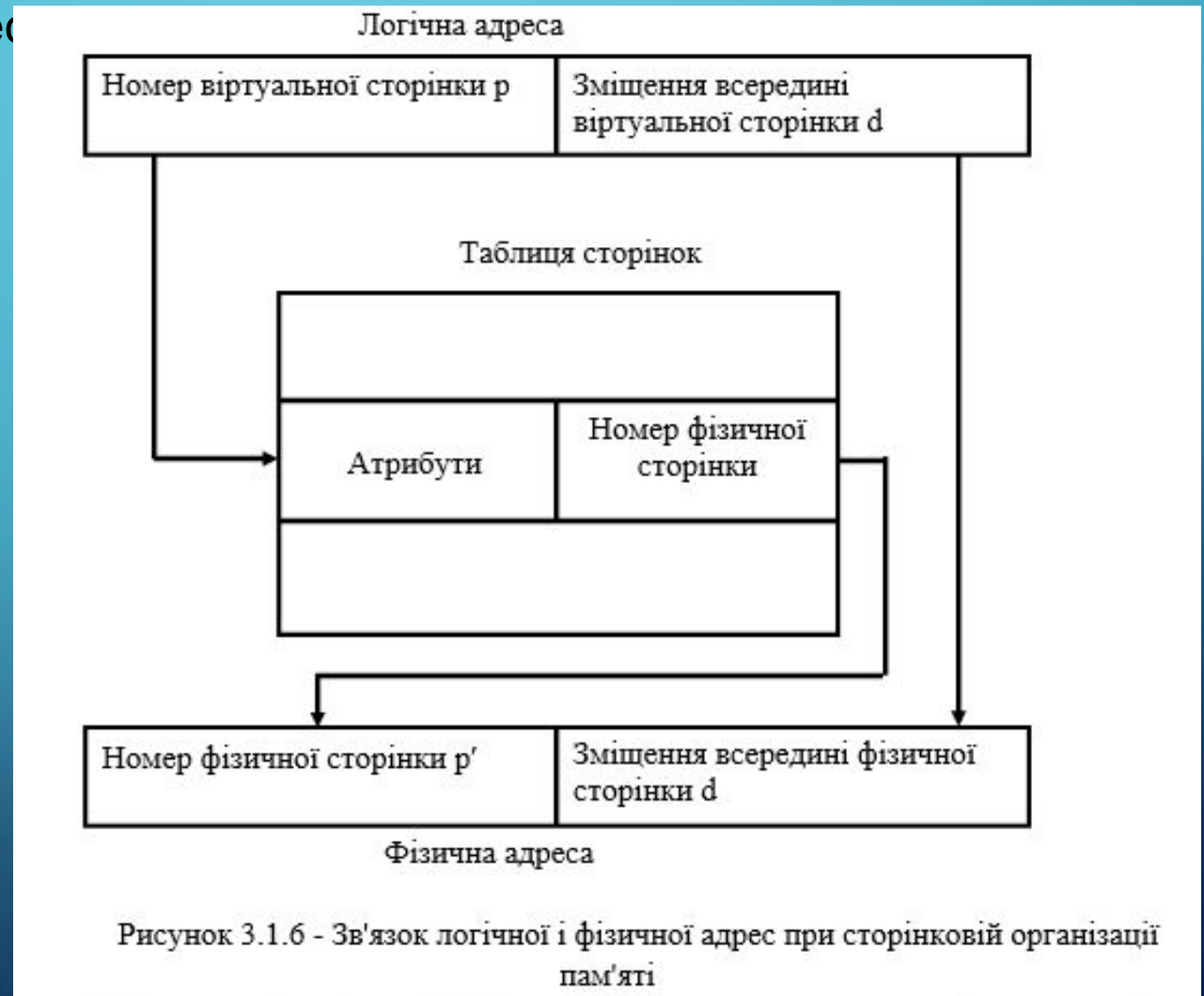
Логічна адреса в сторінковій системі – це впорядкована пара (p, d) , де p - номер сторінки у віртуальній пам'яті, а d - зміщення у рамках сторінки p , на якій розміщується елемент, що адресується.

Описувана схема дозволяє завантажити процес, навіть якщо немає безперервної області кадрів, достатньої для розміщення процесу цілком.

Система відображення логічних адрес у фізичні зводиться до системи відображення логічних сторінок у фізичні і є **таблицею сторінок**, яка зберігається в оперативній пам'яті. Іноді говорять, що таблиця сторінок - це кусочно-лінійна функція відображення, задана в табличному виді.

3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

Інтерпретація логічної адреси у фізичну показана на рисунку 3.1.6. Якщо виконуваний процес звертається до логічної адреси $v = (p, d)$, механізм відображення шукає номер сторінки p в таблиці сторінок і визначає, що ця сторінка знаходиться в сторінковому кадрі p' , формуючи реальну адресу



3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

Таблиця сторінок (page table) адресується за допомогою спеціального регістра процесора і дозволяє визначити номер кадру за логічною адресою. Окрім цього основного завдання, за допомогою атрибутів, записаних в рядку таблиці сторінок, можна організувати контроль доступу до конкретної сторінки і її захист.

Реальне відображення приховане від користувача і контролюється ОС. Помітимо, що процесу користувача чужа пам'ять недоступна. Він не має можливості адресувати пам'ять за межами своєї таблиці сторінок, яка включає тільки його власні сторінки.

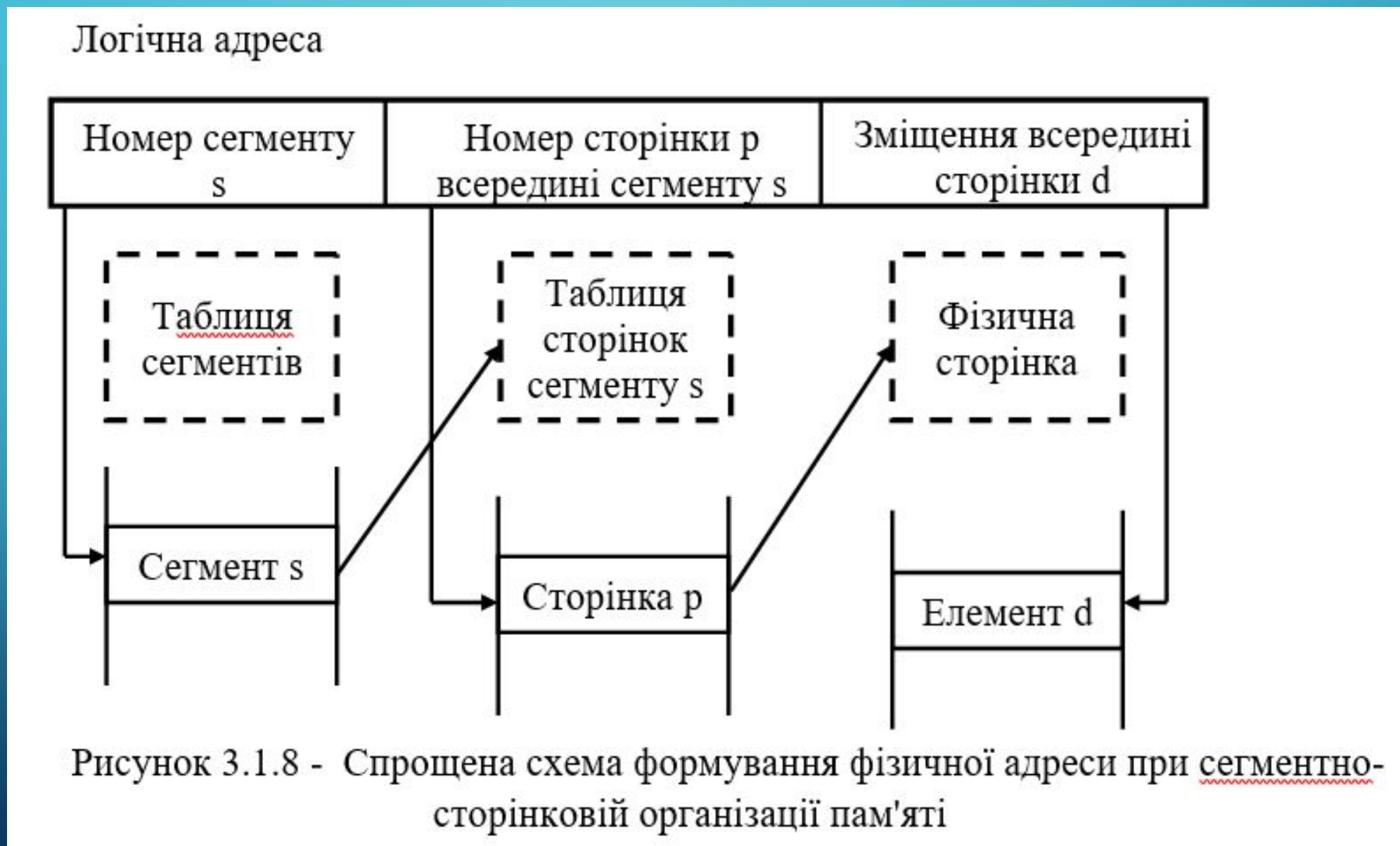
Для управління фізичною пам'яттю ОС підтримує структуру **таблиці кадрів**. Вона має один запис на кожен фізичний кадр, що показує його стан.

Відображення адрес має бути здійснене коректно навіть в складних випадках і зазвичай реалізується апаратно. Для посилання на таблицю процесів використовується спеціальний регістр. При перемиканні процесів необхідно знайти таблицю сторінок нового процесу, покажчик на яку входить в контекст процесу.

3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

Сегментно-сторінкова організація пам'яті. Зберігати в пам'яті сегменти великого розміру цілком так само незручно, як і зберігати процес безперервним блоком. Напрошується ідея розбиття сегментів на сторінки. При сегментно-сторінковій організації пам'яті відбувається дворівнева трансляція віртуальної адреси у фізичну. В цьому випадку логічна адреса складається з трьох полів $v=(s, p, d)$: номера сегменту логічної пам'яті, номера сторінки усередині сегменту і зміщення усередині сторінки. Відповідно, використовуються дві таблиці відображення - таблиця сегментів, що зв'язує номер сегменту з таблицею сторінок, і окрема таблиця сторінок для кожного сегменту. (рисунок 3.1.8)

3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ



3 ФУНКЦІЇ СИСТЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ. ПРОСТІ СХЕМИ УПРАВЛІННЯ ПАМ'ЯТТЮ

Сегментно-сторінкова і сторінкова організація пам'яті дозволяє легко організувати спільне використання одних і тих же даних і програмного коду різними завданнями. Для цього різні логічні блоки пам'яті різних процесів відображують в один і той же блок фізичної пам'яті, де розміщується фрагмент коду або даних, що розділяється.

УЗАГАЛЬНЕННЯ ВИВЧЕНОГО МАТЕРІАЛУ:

Усно дайте відповіді на питання.

- 1 Як називається властивість реальних програм, протягом обмеженого відрізка часу здатних працювати з невеликим набором адрес пам'яті?
- 2 Як називають адреси в основній пам'яті, що характеризують реальне розташування даних у пам'яті?
- 3 Як називається область пам'яті певного призначення, усередині якої підтримується лінійна адресація?
- 4 З яких компонентів складається адреса пам'яті при сегментній організації пам'яті?
- 5 Як називають сукупність усіх логічних адрес пам'яті?
- 6 Як називається відображення логічних адрес пам'яті у фізичні адреси?
- 7 Поясніть принцип роботи схеми з фіксованими розділами;
- 8 Поясніть принцип роботи схеми зі змінними розділами;
- 9 При реалізації організації пам'яті з фіксованими розділами застосовуються декілька стратегій вибору вільного розділу пам'яті. В чому полягає стратегія першого відповідного (First fit)?

УЗАГАЛЬНЕННЯ ВИВЧЕНОГО МАТЕРІАЛУ:

Усно дайте відповіді на питання.

- 10 При реалізації організації пам'яті з фіксованими розділами застосовуються декілька стратегій вибору вільного розділу пам'яті. В чому полягає стратегія найбільш відповідного (Best fit)?
- 11 При реалізації організації пам'яті з фіксованими розділами застосовуються декілька стратегій вибору вільного розділу пам'яті. В чому полягає стратегія найменш відповідного (Worst fit)?
- 12 Поясніть принцип роботи схеми зі змінними розділами;
- 13 Яким чином формується логічна адреса в сторінковій системі?
- 14 Який механізм відображення логічних сторінок у фізичні кадри використовується при сторінковій організації пам'яті?
- 15 Яким чином формується логічна адреса в сегментно-сторінковій системі?
- 16 Який механізм відображення логічних сторінок у фізичні кадри використовується при сегментно-сторінковій організації пам'яті?
- 17 Для чого використовується таблиця сегментів при сегментно-сторінковій організації пам'яті?

ДОМАШНЄ ЗАВДАННЯ

В Moodle пройти Тест до теми 3.1.