

## 16.3 ФІЗИЧНА ОРГАНІЗАЦІЯ ФАЙЛУ

Файл, що має образ цілісного набору байт, насправді дуже часто розкиданий «ділянками» по всьому диску, причому це розбиття ніяк не пов'язане з логічною структурою файлу. Наприклад, його окремий логічний запис може бути розташований в несуміжних секторах диска. Логічно об'єднані файли з одного каталогу зовсім не зобов'язані бути сусідами на диску. Принципи розміщення файлів, каталогів і системної інформації на реальному пристрої описуються фізичною організацією файлової системи.

### 16.3.1 Диски, розділи, сектори, кластери

Основним типом пристрою, який використовується в сучасних обчислювальних системах для зберігання файлів, є дискові накопичувачі. Ці пристрої призначені для зчитування і запису даних на жорсткі диски.

Ще раз нагадаємо топологію жорсткого диска, який складається з однієї або декількох скляних або металевих пластин, кожна з яких покрита з однієї або двох сторін магнітним матеріалом. Таким чином, диск у загальному випадку складається з пакету пластин (рис. 16.4).

На кожній стороні кожної пластини розмічені тонкі концентричні кільця – *доріжки* (traks), на яких зберігаються дані. Кількість доріжок залежить від типу диска. Нумерація доріжок починається з 0 від зовнішнього краю до центру диска. Коли диск обертається, елемент, що називається головкою, прочитує двійкові дані з магнітної доріжки або записує їх на магнітну доріжку.

Головка може позиціонуватися над заданою доріжкою. Головки переміщуються над поверхнею диска дискретними кроками, кожен крок відповідає зрушенню на одну доріжку. Запис на диск здійснюється завдяки здатності головки змінювати магнітні властивості доріжки. У деяких дисках уздовж кожної поверхні переміщається одна головка, а в інших – є по головці на кожну доріжку.

У першому випадку для пошуку інформації головка повинна переміщатися по радіусу диска. Зазвичай усі головки закріплені на єдиному рухомому механізмі (коромислі) і рухаються синхронно (сервопривід). Тому, коли головка фіксується на заданій доріжці однієї поверхні, усі інші головки зупиняються над доріжками з

такими ж номерами. У тих же випадках, коли на кожній доріжці є окрема головка, ніякого переміщення головок з однієї доріжки на іншу не потрібно, за рахунок цього економиться час, що витрачається на пошук даних.

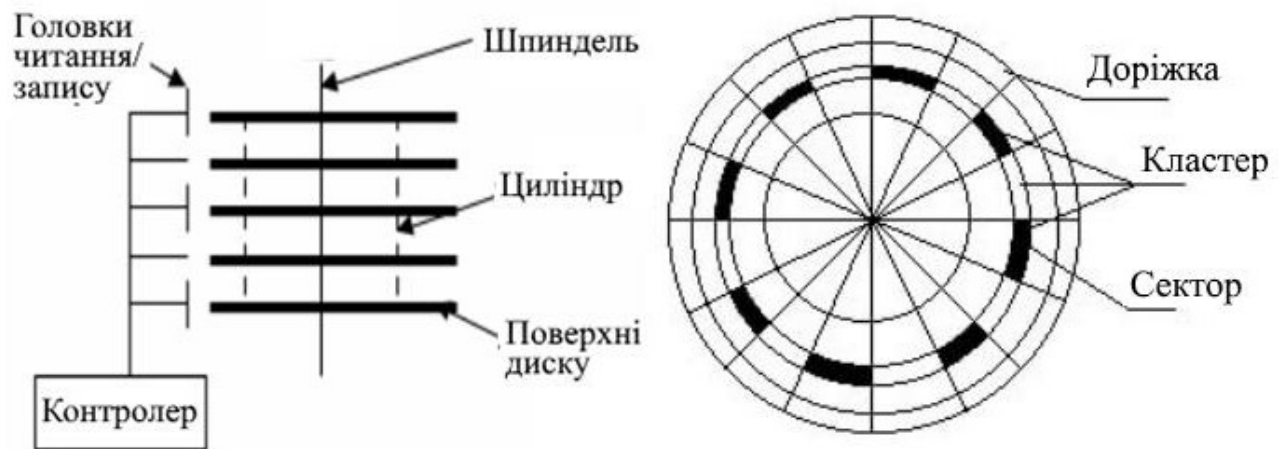


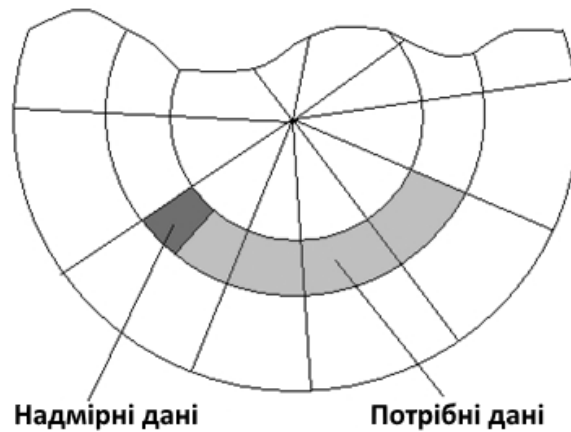
Рисунок 16.4 – Схема пристрою жорсткого диска

Сукупність доріжок одного радіусу на усіх поверхнях усіх пластин пакету називається **циліндром** (cylinder). Кожна доріжка розбивається на фрагменти, що називаються **секторами** (sectors), або **блоками** (blocks), так що усі доріжки мають рівне число секторів, в які можна максимально записати одне і те ж число байт. Іноді зовнішня доріжка має декілька додаткових секторів, які

використовуються для заміни пошкоджених секторів в режимі гарячого резервування.

Сектор має фіксований для конкретної системи розмір, що виражається степенем двійки. Найчастіше розмір сектора складає 512 байт. Враховуючи, що доріжки різного радіусу мають однакове число секторів, щільність запису стає тим вище, чим ближче доріжка до центру.

Для того щоб контролер міг знайти на диску потрібний сектор, необхідно задати йому усі складові адреси сектора: номер циліндра, номер поверхні і номер сектора. Оскільки прикладній програмі в загальному випадку потрібний не сектор, а деяка кількість байт, то типовий запит включає читання декількох секторів, що містять необхідну інформацію і надлишкові дані (рис. 16.5).



**Рисунок 16.5** – Прочитування надмірних даних при обміні з диском

Операційна система при роботі з диском використовує, як правило, власну одиницю дискового простору, що називається **кластером** (cluster). Іноді кластер називають **блоком** (наприклад, в ОС Unix), що може призвести до термінологічної плутанини. Взагалі, термінологія, яка використовується при описі форматів дисків і файлових систем, залежить від апаратної платформи (RISC, Wintel тощо) і операційної системи. Це треба враховувати і трактувати терміни залежно від контексту.

При створенні файлу йому виділяється місце на диску кластерами. Наприклад, якщо файл має розмір 2560 байт, а розмір кластера у файловій системі визначений в 1024 байти, то файлу буде виділено на диску 3 кластери.

Розмітку диска під конкретний тип файлової системи виконують процедури високорівневого, або логічного форматування. При високорівневому форматуванні визначається розмір кластера і на диск записується інформація, що необхідна для роботи файлової системи, у тому числі інформація про доступний і невживаний простір, про межі областей, відведених під файли і каталоги, інформація про пошкоджені області. Крім того, на диск записується завантажувач ОС – невелика програма, яка починає процес ініціалізації операційної системи після включення живлення або рестарту комп'ютера.

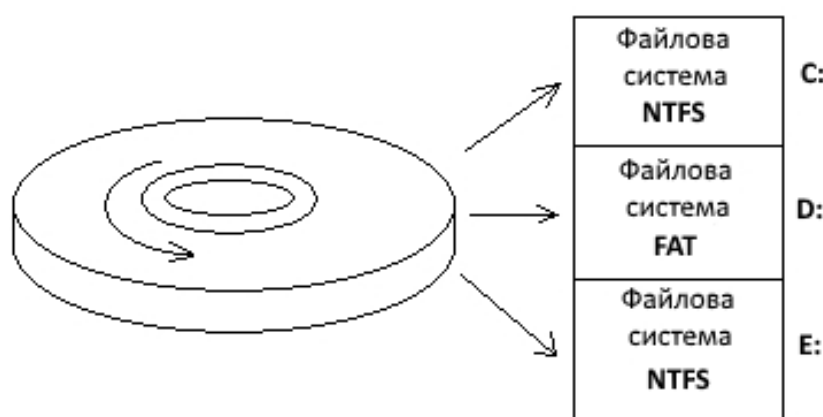
Перш ніж форматувати диск під певну файлову систему, він може бути розбитий на розділи. **Розділ** – це неперервна частина фізичного диска, яку операційна система представляє користувачеві як логічний пристрій (використовуються також назви логічний диск і логічний розділ). У багатьох операційних системах

використовується термін «*том*» (volume). У різних ОС тлумачення цього терміну має свої нюанси, але найчастіше він означає логічний пристрій, що відформатований під конкретну файлову систему.

Логічний пристрій функціонує так, якби це був окремий фізичний диск. Саме з логічними пристроями працює користувач, звертаючись до них за символічними іменами, використовуючи, наприклад, позначення A, B, C, SYS тощо.

На різних логічних пристроях одного і того ж фізичного диска можуть розташовуватися файлові системи різного типу. На рис. 16.6 показаний приклад диска, розбитого на три розділи, в яких встановлені дві файлові системи NTFS (розділи C і E) і одна файлова система FAT (розділ D).

Усі розділи одного диска мають однаковий розмір блоку, визначений для цього диска в результаті низькорівневого форматування. Проте в результаті високорівневого форматування в різних розділах одного і того ж диска, представлених різними логічними пристроями, можуть бути встановлені файлові системи, в яких визначені кластери різних розмірів.



**Рисунок 16.6** – Розбиття диска на розділи

### 16.3.2 Фізична організація і адресація файлу

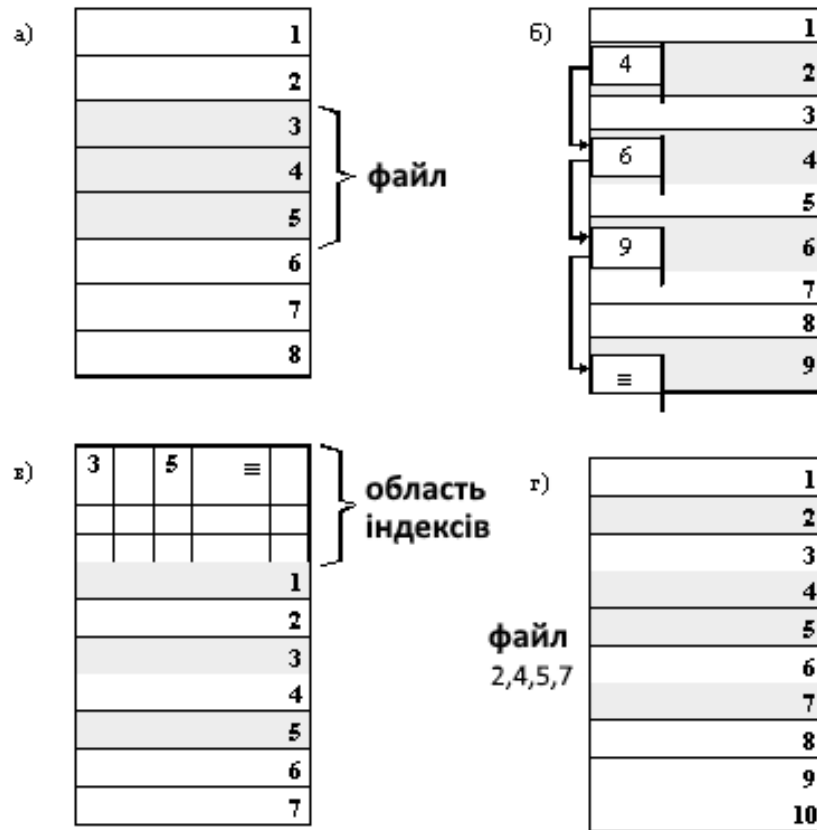
Важливим компонентом фізичної організації файлової системи є фізична організація файлу, тобто спосіб розміщення файлу на диску. Основними критеріями ефективності фізичної організації файлів є:

- швидкість доступу до даних;
- об'єм адресної інформації файлу;
- міра фрагментації дискового простору;

– максимально можливий розмір файлу.

Безперервне розміщення – простий варіант фізичної організації (рис. 16.7, *a*), при якому файлу надається послідовність кластерів диска, що утворюють безперервну ділянку дискової пам'яті. Основною перевагою цього методу є висока швидкість доступу, оскільки витрати на пошук і прочитування

кластерів файлу мінімальні. Також мінімальний об'єм адресної інформації – досить зберігати тільки номер першого кластера і об'єм файлу.



**Рисунок 16.7** – Фізична організація файлу *a*) – безперервне розміщення; *б*) – зв'язаний список блоків; *в*) – зв'язаний список індексів; *г*) – перелік номерів блоків

Ця фізична організація не обмежує максимально можливий розмір файлу. Проте цей варіант має істотні недоліки, які утрудняють його застосування на практиці, незважаючи на всю його логічну простоту. Дійсно, якого розміру має бути безперервна область, що виділяється файлу, якщо файл при кожній модифікації може збільшити свій розмір? Ще серйознішою проблемою є фрагментація.

Через деякий час після створення файлової системи в результаті виконання численних операцій створення і видалення файлів простір диска неминуче

перетворюється на «клаптеву ковдру», що включає велике число вільних областей невеликого розміру. При фрагментації, сумарний об'єм вільної пам'яті може бути дуже великим, а вибрати місце для розміщення цілого файлу неможливо.

Наступний спосіб фізичної організації – розміщення файлу у вигляді зв'язаного списку кластерів дискової пам'яті (див. рис. 16.7, б). При такому способі на початку кожного кластера міститься покажчик на наступний кластер. У цьому випадку адресна інформація мінімальна: розташування файлу може бути задане одним числом – номером першого кластера. На відміну від попереднього способу кожен кластер може бути приєднаний до ланцюжка кластерів якого-небудь файлу, отже, фрагментація на рівні кластерів відсутня.

Файл може змінювати свій розмір під час свого існування, нарощуючи число кластерів.

Недоліком є складність реалізації доступу до довільно заданого місця файлу. Щоб прочитати  $n$ -тий за порядком кластер файлу, необхідно послідовно прочитати чотири перші кластери, простежуючи ланцюжок номерів кластерів. Крім того, при цьому способі кількість даних файлу, що містяться в одному кластері, не дорівнює степені двійки (одне слово витрачене на номер наступного кластера), а багато із програм читають дані кластерами, розмір яких дорівнює степені двійки.

Популярним способом у файловій системі FAT, є використання зв'язного списку індексів (див. рис. 16.7, в). Цей спосіб є деякою модифікацією попереднього. Файлу виділяється пам'ять у вигляді зв'язного списку кластерів. Номер першого кластера запам'ятовується в записі каталогу, де зберігаються характеристики цього файлу. Інша адресна інформація відокремлена від кластерів файлу. З кожним кластером диска зв'язується деякий елемент – *індекс*. Індокси розташовуються в окремій області диска – в MS-DOS це таблиця

FAT (File Allocation Table), що займає один кластер. Коли пам'ять вільна, всі індокси мають нульове значення. Якщо деякий кластер  $N$  призначений деякому файлу, то індекс цього кластера стає рівним або номеру  $M$  наступного кластера цього файлу, або набуває спеціального значення, що є ознакою того, що цей кластер для даного файлу є останнім. Індекс же попереднього кластера файлу набуває значення  $N$ , вказуючи на новопризначений кластер.

При такій фізичній організації зберігаються усі переваги попереднього способу: мінімальність адресної інформації, відсутність фрагментації, відсутність проблем при зміні розміру. Крім того, цей спосіб має додаткові переваги. По-перше, для доступу до довільного кластера файлу не вимагається послідовно прочитувати його кластери, досить прочитати тільки сектори диска, що містять таблицю індексів, відлічити потрібну кількість кластерів файлу по ланцюжку і визначити номер потрібного кластера. По-друге, дані файлу заповнюють кластер цілком, тобто мають об'єм, рівний степені двійки.

Необхідно відмітити, що за відсутності фрагментації на рівні кластерів на диску все одно є певна кількість областей пам'яті невеликого розміру, які неможливо використати, тобто фрагментація все ж існує. Ці фрагменти є невживаними частинами останніх кластерів, призначених файлам, оскільки об'єм файлу в загальному випадку не кратний розміру кластера. На кожному файлі в середньому втрачається половина кластера. Це втрати особливо великі, коли на диску є велика кількість маленьких файлів, а кластер має великий розмір. Ще один спосіб фізичного розташування файлу полягає в простому перерахуванні номерів кластерів, які зайняті цим файлом (див. рис. 16.7, з). Цей перелік і служить адресою файлу. Недолік цього способу очевидний: довжина адреси залежить від розміру файлу і для великого файлу може скласти значну величину. Перевагою ж є висока швидкість доступу до довільного кластера файлу, оскільки тут застосовується пряма адресація, яка виключає перегляд ланцюжка покажчиків при пошуку адреси довільного кластера файлу.

Фрагментація на рівні кластерів у цьому способі також відсутня.