01.1. Поняття файлової системи

Файлова система – це частина ОС, призначена для

1) зручної організації та представлення [різнорідних] даних, що зберігаються у пристроях зовнішньої пам’яті (ПЗП);

2) забезпечення зручного програмного доступу до даних (іменування, пошук, створення, знищення, запис, читання);

3) забезпечення спільної роботи з даними декількох користувачів та обчислювальних процесів (координація доступу до даних, забезпечення цілісності та несуперечливості даних).

01.2. Поняття файлової системи

Поняття «файлова система» в залежності від контексту може також означати:

- сукупність всіх файлів на жорсткому диску чи іншому пристрої зовнішньої пам’яті;
- набір структур даних, які використовуються для управління файлами (каталоги файлів, дискриптори файлів, таблиці розподілу вільного та зайнятого місця на дисках і т.п.);
- набір системного програмного забезпечення для управління файлами.

01.3. Поняття файлової системи

Приклади файлових систем:

FAT → File Allocation Table (FAT16, FAT32) (Microsoft);

NTFS → NT File System (Microsoft);

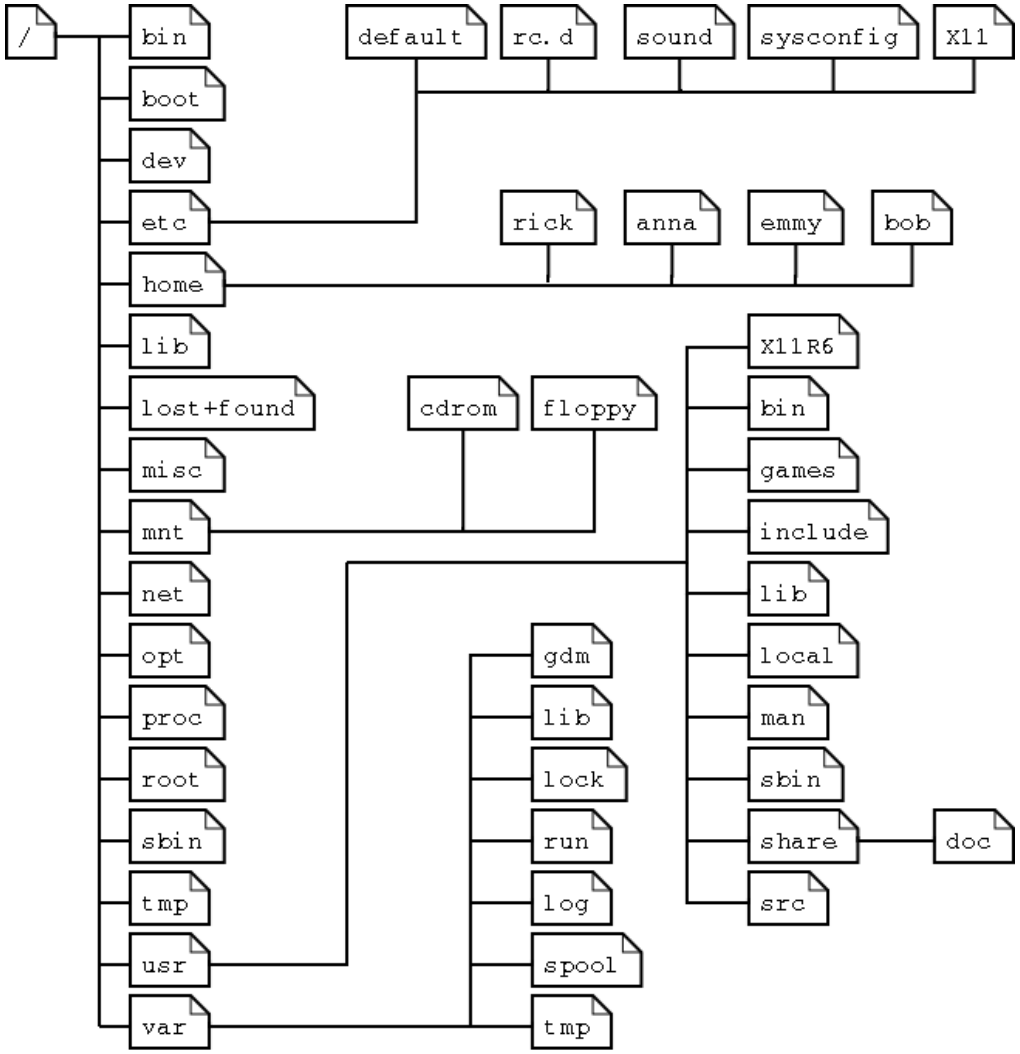
UFS → Unix File System (FreeBSD)

ext2 → ext3 → ext4 (Linux)

JFS → Journaled File System (IBM → AIX, Linux)

XFS → (Silicon Graphics → IRIX, Linux)

01.4. Поняття файлової системи



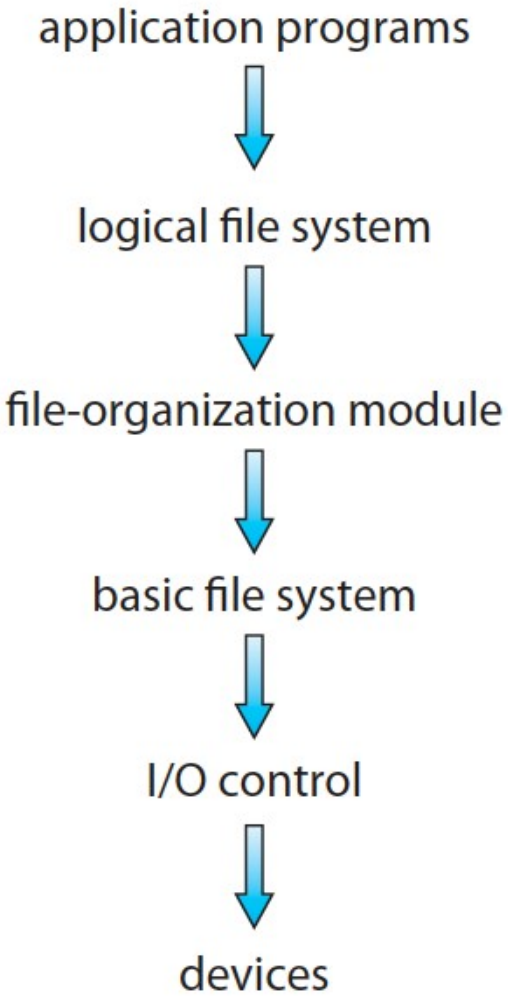
- Змістовна структура файлової системи.
- Приклад: Unix directory structure.
- Стандарт: Filesystem Hierarchy Standard (FHS)

  (v.3.0, 3 June 2015)

01.5. Поняття файлової системи

Директорія	Опис вмістимого
/bin	Основні програмні утиліти (системні та прикладні)
/boot	Файли необхідні для завантаження системи (завантажувальник, ядро та ін.)
/dev	файли пристроїв [вводу/виводу]
/etc	Основні конфігураційні файли, загальні налаштування системи
/home	Домашні директорії звичайних користувачів системи
/lib	Основні програмні бібліотеки
/mnt	Директорія, в яку тимчасово монтуються файлові системи змінних пристроїв зовнішньої пам’яті
/proc	Відображення запущених обчислювальних процесів у вигляді файлів
/root	Домашня директорія супер-користувача (адміністратора системи)
/tmp	Тимчасові файли
/usr	Вторинна ієрархія (містить /usr/bin, /usr/etc, /usr/lib та ін.), в якій зберігається більшість встановлених у системі прикладних програм та необхідні для їхнього виконання незмінні дані
/var	Змінні дані програм (в тому числі службових), які встановлені та виконуються в системі

01.6. Поняття файлової системи



Багаторівнева схема організації роботи файлової системи:

1. Базова файлова система (basic file system) виконує основні операції по запису/читанню фізичних блоків файлів.

2. Логічна файлова система (logical file system) забезпечує модуль управління файлами (file-organization module) необхідною інформацією про файли, виходячи з їх символьних назв.

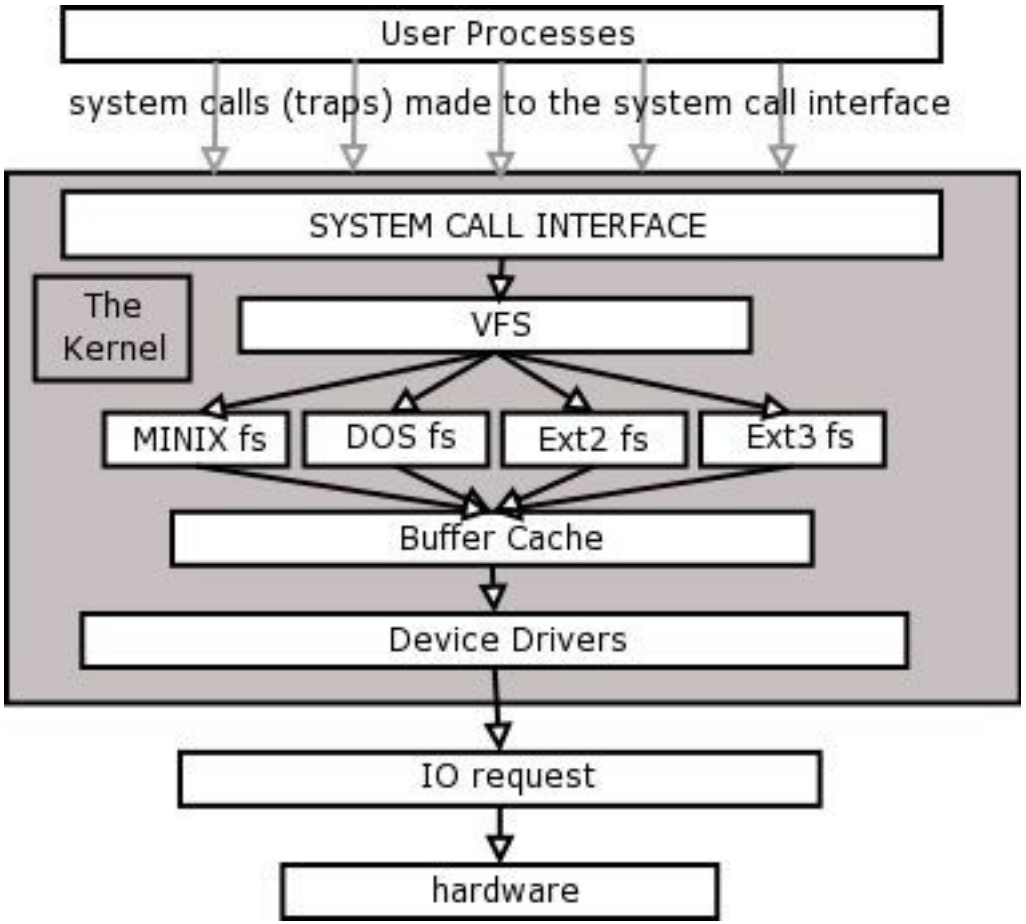
01.7. Поняття файлової системи

3. Файлова структура в логічній файловій системі визначається вмістом блоків управління файлом (file-control block, FCB).

4. FCB містить всю службову інформаію про відповідний файл (власник файлу, права доступу, місцерозташування вмістимого файлу на ПЗП та ін.).

5. В файлових системах UNIX-подібних ОС FCB називається inode.

01.8. Поняття файлової системи



Проблема уніфікації програмного інтерфейсу доступу до різних файлових систем.

Приклад:
Virtual File System (VFS) (UNIX/Linux)

02.1. Способи іменування файлів

- Базовий принцип інформаційних систем: для того, щоб отримати доступ до деякого ресурсу (в даному випадку файлу), треба дізнатися його ім’я.
- Більшість сучасних інформаційних систем відносяться до класу відкритих інформаційних систем, в яких знання імені ресурсу означає можливість беспосереднього доступу до цього ресурсу.
- З огляду на проблему захисту інофрмації у відкритих інформаційних системах додатково використовують системи контролю та управління доступом до ресурсів.

02.2. Способи іменування файлів

Узагальнена проблема іменування (naming) → потрібно забезпечити:

1. Унікальність імені (або відносна унікальність імені).

2. Незмінність імені у часі (або на деякому заданому проміжку часу).

3. Змістовність імені (ім’я має нести якусь інформацію про об’єкт/файл, наприклад, до якого типу об’єктів/файлів він відноситься).

02.3. Способи іменування файлів

Ієрархічна система імен:

- Вводяться спеціальні вузли (=каталоги =директорії =folders), на основі яких утворюєтья граф, листками (листовими вузлами) якого є файли даних.

- Відповідно ім’я файлу в такій системі імен – це шлях від деякого кореневого вузла графу до листового вузла відповідного файлу.

02.4. Способи іменування файлів

Варіанти організації ієрархії імен файлів:

1) одноріvнева (Single-Level Directory);

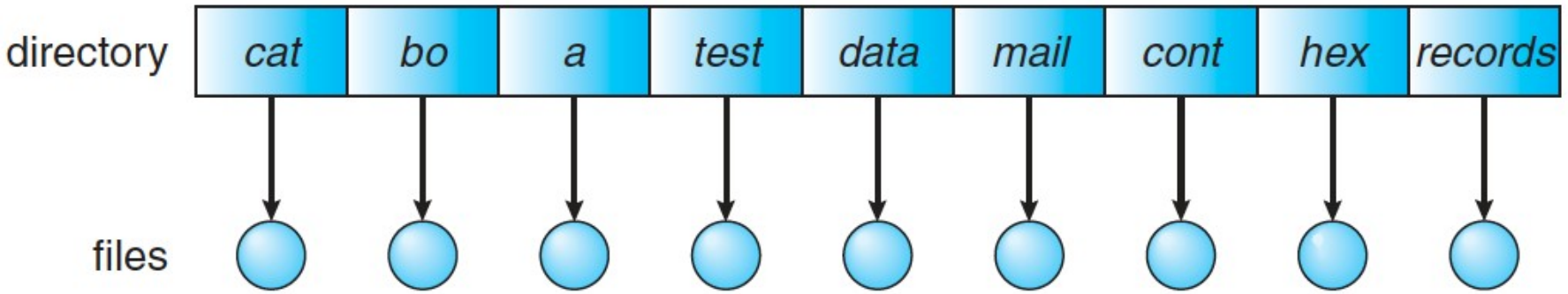
2) дворівнева (Two-Level Directory);

3) граф-дерево (Tree-Structured Directories);

4) ациклічний граф (Acyclic-Graph Directories);

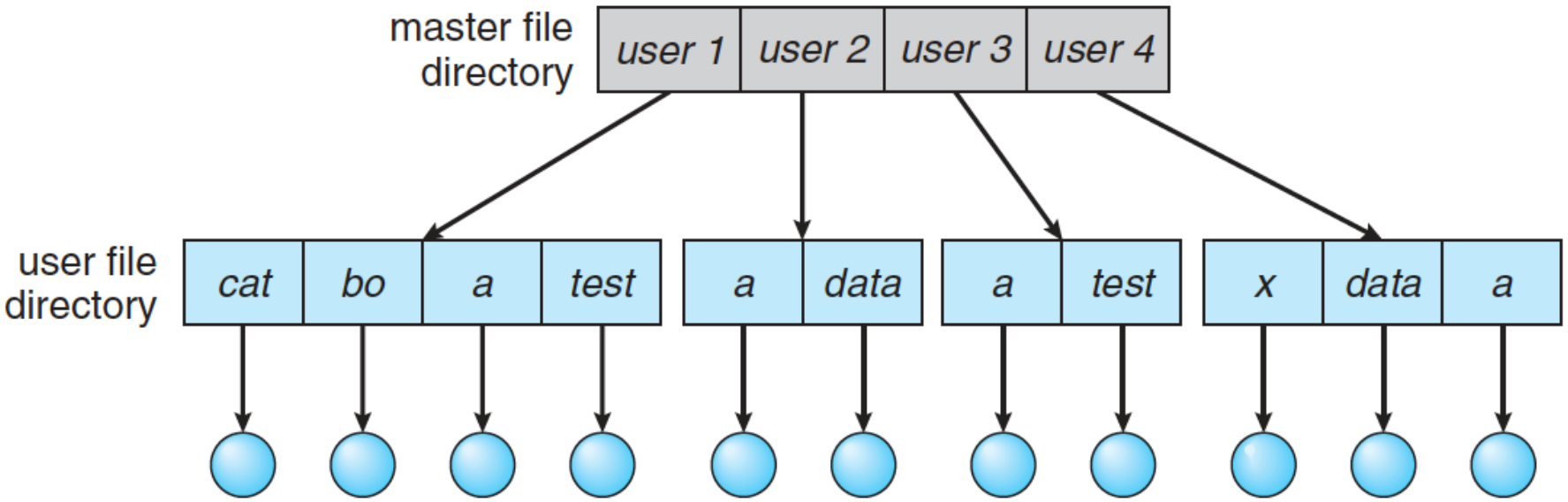
5) довільний граф (General Graph Directory).

02.5. Способи іменування файлів



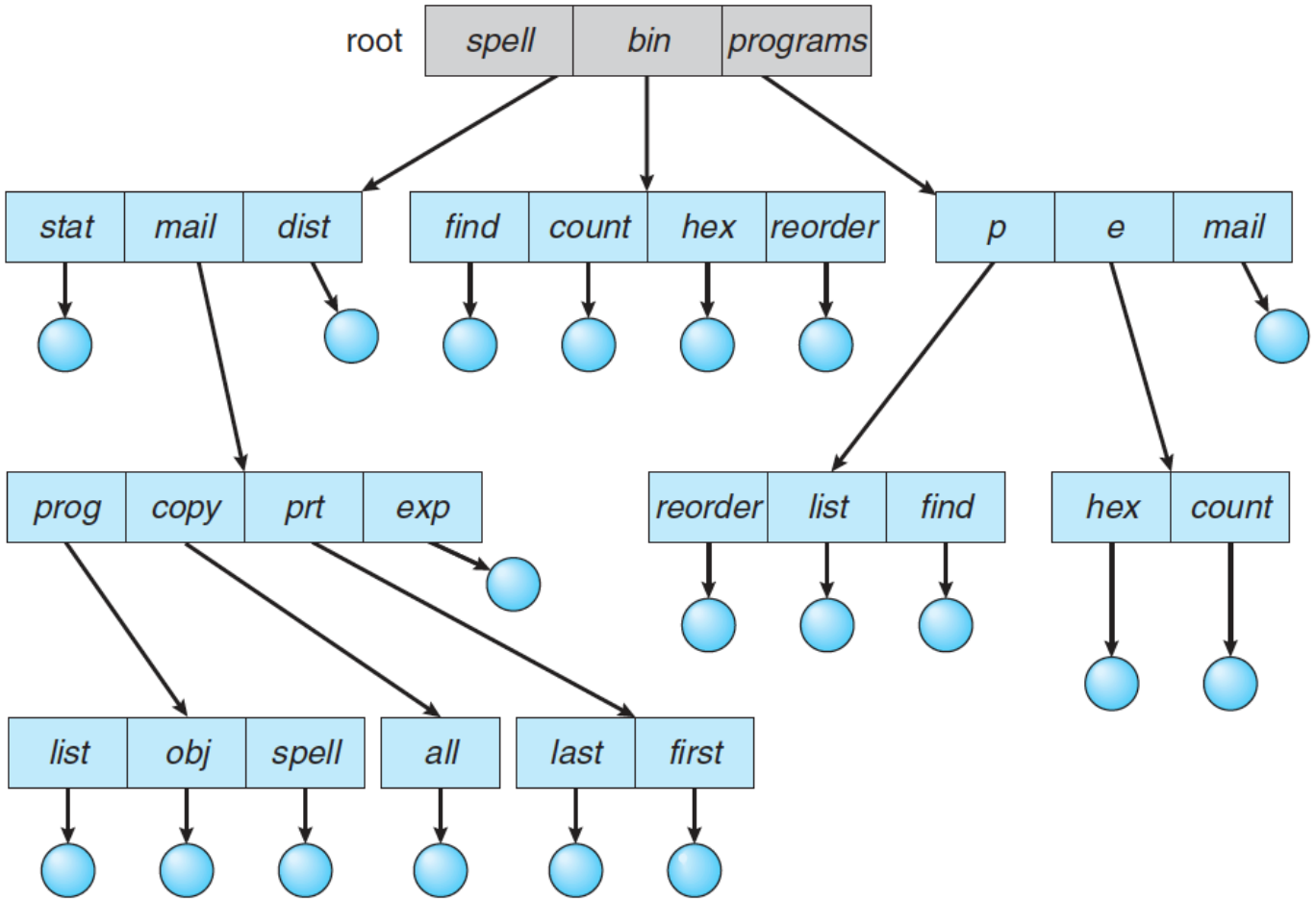
1) одноріvнева (Single-Level Directory)

02.6. Способи іменування файлів



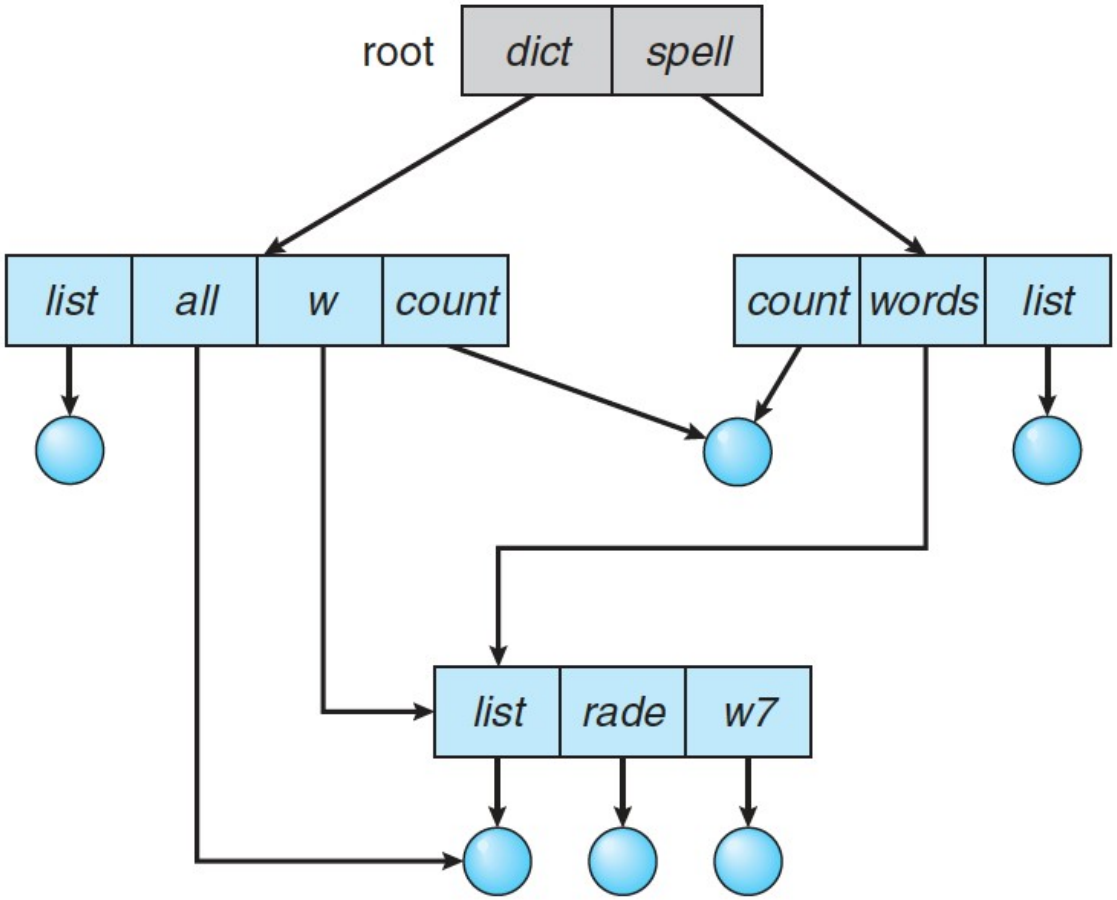
2) дворівнева (Two-Level Directory)

02.7. Способи іменування файлів



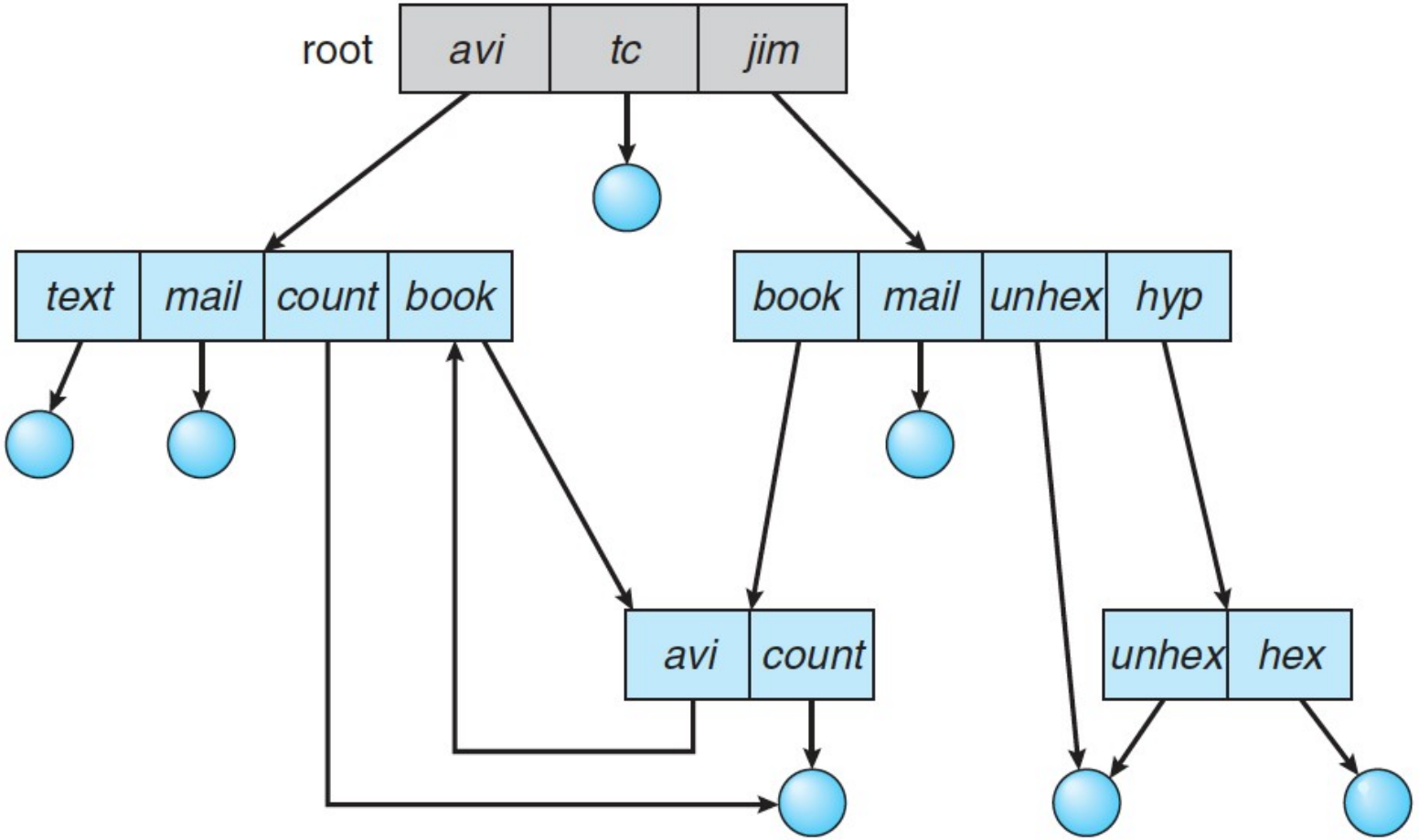
3) граф-дерево (Tree-Structured Directories)

02.8. Способи іменування файлів



4) ациклічний граф (Acyclic-Graph Directories)

02.9. Способи іменування файлів



5) довільний граф (General Graph Directory)

02.10. Способи іменування файлів

Типи імен файлів:

1) [просте] символьне ім’я;

2) відносне складене ім’я файлу;

3) повне складене ім’я файлу;

4) унікальний числовий ідентифікатор

02.11. Способи іменування файлів

Атрибути файлу:

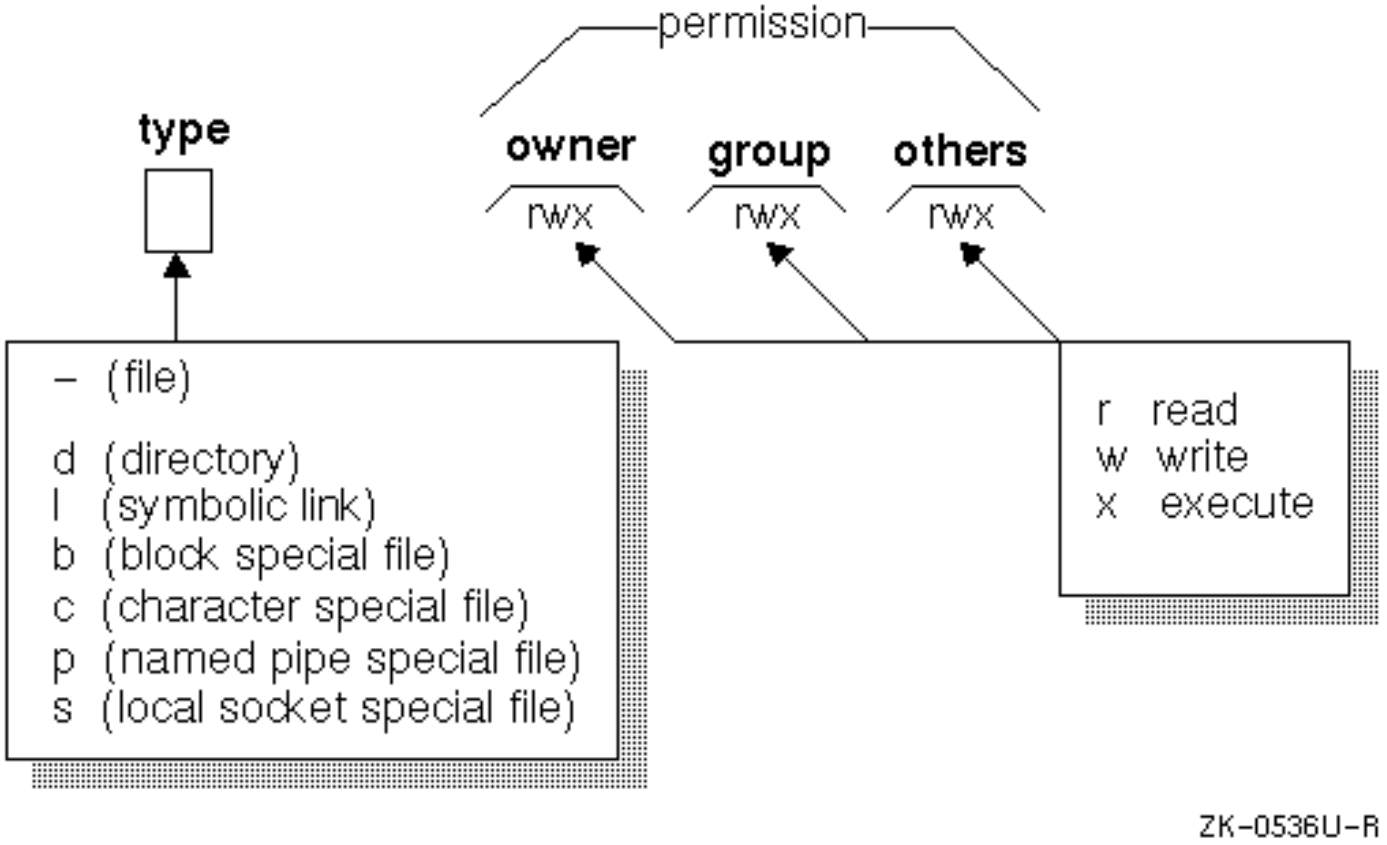
1) Символьне ім’я у зручній для людини формі.
2) Унікальний ідентифікатор файлу як об’єкту операційної системи.
3) Тип файлу, який визначає для системи допустимі операції з файлом.
4) Місцерозташування файлу на зовнішньому пристрої пам’яті.
5) Розмір файлу.
6) Час та дата створення і/або останньої модифікації файлу.
7) Ідентифікатор користувача – власника файлу.
8) Права доступу до файлу (на читання, запис та виконання).

02.12. Способи іменування файлів

```
drwxr-xr-x 2 abk abk 4096 Apr 16  2019  4.1
-rw-r--r-- 1 abk abk 9000 Dec 24  2018 'MPAu-L-results (copy).dat'
-rw-r--r-- 1 abk abk 9000 Dec 24  2018  MPAuv-L-results.dat
-rw-r--r-- 1 abk abk 9000 Dec 24  2018  MSA-1-results.dat
-rw-r--r-- 1 abk abk 9000 Dec 24  2018  PA-results.dat
-rw-r--r-- 1 abk abk 9000 Dec 24  2018  RA-results.dat
drwxr-xr-x 2 abk abk 4096 Dec 24  2018  images
-rw-r--r-- 1 abk abk 2337 Dec 24  2018  msa.gnu
drwxr-xr-x 2 abk abk 4096 Dec 24  2018  res-01
drwxr-xr-x 2 abk abk 4096 Apr  1  2019  res-02
drwxr-xr-x 3 abk abk 4096 Apr  4  2019  res-03
drwxr-xr-x 2 abk abk 4096 May  5  2019  res-04-KQ
drwxr-xr-x 2 abk abk 4096 Nov 29 23:32  res-tmp
drwxr-xr-x 2 abk abk 4096 Dec 24  2018  templates
-rw-r--r-- 1 abk abk 9000 Dec 24  2018  test.dat
```

Приклад визначння атрибутів файла в ОС UNIX командою ls

02.13. Способи іменування файлів



Значення атрибутів файла в ОС UNIX

03.1. Об’єкти файлової системи

До основних об’єктів файлової системи відносятья:

1) файли програм та даних;

2) спеціальні файли;

3) директорії (каталоги);

4) посилання («лінки»);

5) списки управління доступом.

03.2. Об’єкти файлової системи

1. Файли програм та даних:

- «Звичайні» файли → будь який окремий об’єкт даних чи коду;
- Основні типи: виконавчі файли (executables), об’єктні файли (як результат компіляції в машинний код), файли програмних бібліотек, тексти програм (на мовах високого рівня), командні файли (системні скрипти, наприклад, *.bat), текстові файли, документи, файли мультимедіа.

03.3. Об’єкти файлової системи

file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, perl, asm	source code in various languages
batch	bat, sh	commands to the command interpreter
markup	xml, html, tex	textual data, documents
word processor	xml, rtf, docx	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	gif, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	rar, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, mp3, mp4, avi	binary file containing audio or A/V information

Приклади файлів програм та даних

03.4. Об’єкти файлової системи

2. Спеціальні файли:

- Файли пристроїв (UNIX/Linux) → файл асоційований з пристроєм вводу/виводу.
- Мета використання: приховати низькорівневі деталі роботи пристрою задля збільшення зручності роботи.
- Тип файла пристрою: символьний/блочний.

03.5. Об’єкти файлової системи

3. Директорії (каталоги):

- Директорія (каталог) – це вузол ієрархічної системи імен, в якому об’єднано групу файлів та інших директорій.
- Для кожної директорії створюється запис, який містить «довідкову» інформацію про файли і каталоги, які вона містить.
- Цей запис представляє собою таблицю назв файлів/каталогів та їх атрибутів (права доступу, час останньої модифікації, розмір та ін.).
- Запис може зберігатися у вигляді окремого спеціального файлу (MS-DOS), або у вигляді посилань на відповідні рядки у службових/системних таблицях (UNIX/Linux).

03.6. Об’єкти файлової системи

4. Посилання («лінки»):

- «Псевдоніми» (aliases) файлів – це додаткові імена файлу, які вводяться для зручності користувача (наприклад, для зменшення розміру повного імені файлу в межах граф-дерева імен).

- Розрізняють два основних типи посилань:

1) жорстке посилання (hard link) – ще одне повноцінне ім’я файлу;
2) символьне посилання (symbolic/soft link) – спеціальний «текстовий» запис, який містить повний шлях до відповідного файлу.

03.7. Об’єкти файлової системи

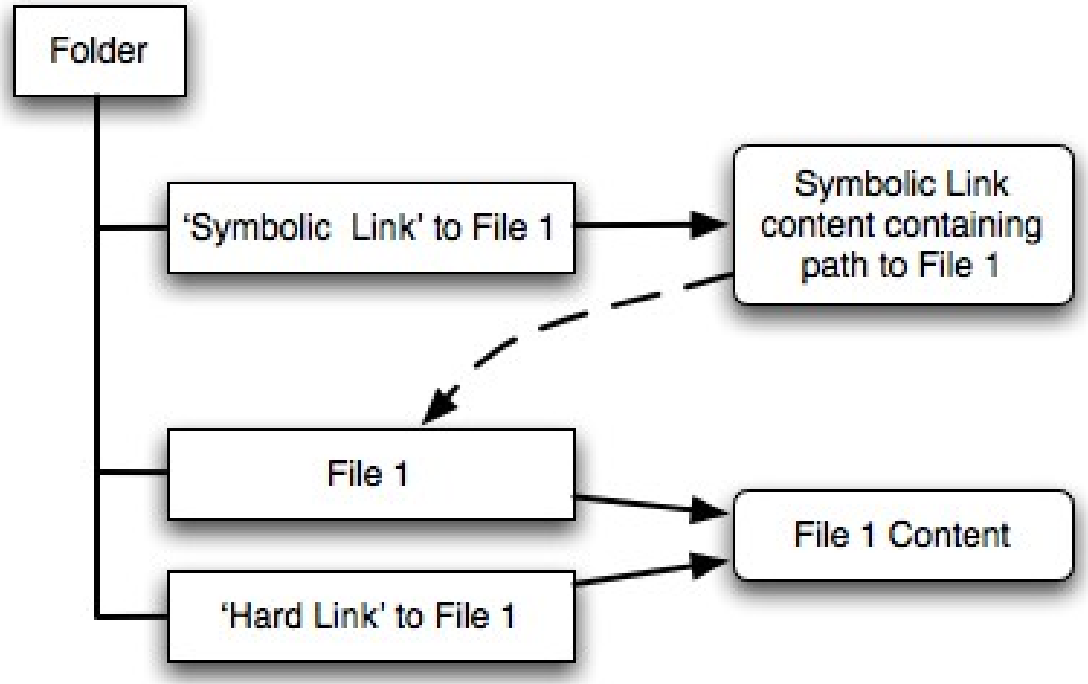


Схема hard link та symbolic/soft link

03.8. Об’єкти файлової системи

5. Списки управління доступом (access control list, ACL):

- Запис для кожного об’єкту файлової системи, який містить інформацію про права доступу до цього об’єкту (на читання, запис, виконання) для кожного користувача системи (список по користувачам та по видам доступу).
- Як правило, для кожного файлу/директорії визначається ідентифікатор відповідного ACL-запису.

04.1. Логічна організація файлу

Логічна організація файлу: представлення файлу для програміста або для користувацького процесу, який доступається до файлу (логічна структура файлу).

Способи організації програмного доступу до вмістимого файлу:

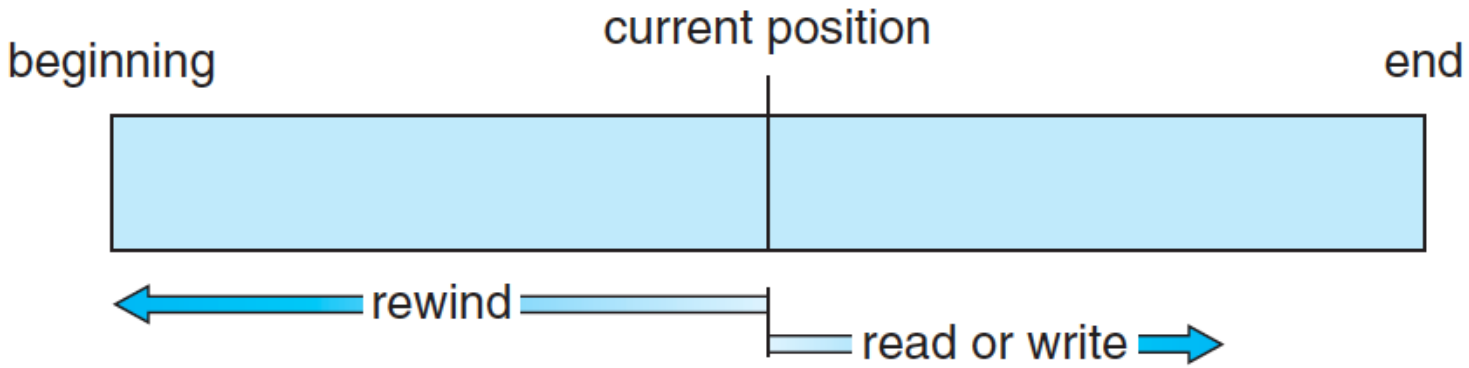
1. Послідовний доступ (Sequential Access).
2. Прямий доступ (Direct Access).
3. Доступ на основі індексів (Indexed Access).

04.2. Логічна організація файлу

1. Послідовний доступ (Sequential Access):

- Схема: початок файлу → записи → кінець файлу.
- Використовується покажчик на біжучу позицію у файлі.
- Логічна структура файлу: лінійна послідовність [однобайтних] записів (символів).
- Базова модель ПЗП: магнітна стрічка.

04.3. Логічна організація файлу



- Операції: read_next(), write_next() (автоматично зміщують покажчик на біжучу позицію у файлі після виконання операції) + skip forward or backward n records.
- Варіанти ускладнень: збільшення розміру окремого запису (слово, рядок і т.п.), використання тегів/ключів для визначення різних типів записів (HTML, XML і т.п.).

04.4. Логічна організація файлу

2. Прямий доступ (Direct Access or Relative Access):

- Файл розглядається як набір окремих логічних записів фіксованої довжини, до яких можна доступитися безпосередньо за їх ідентифікатором без послідовного перебору попередніх записів.

- Базова модель ПЗП: жорсткий диск (HDD).

04.5. Логічна організація файлу

- Приклад: в якості логічних записів можуть виступати блоки даних (запис=block), в яких файл зберігається на жорсткому диску (нумерація записів – відносна від першого блоку).

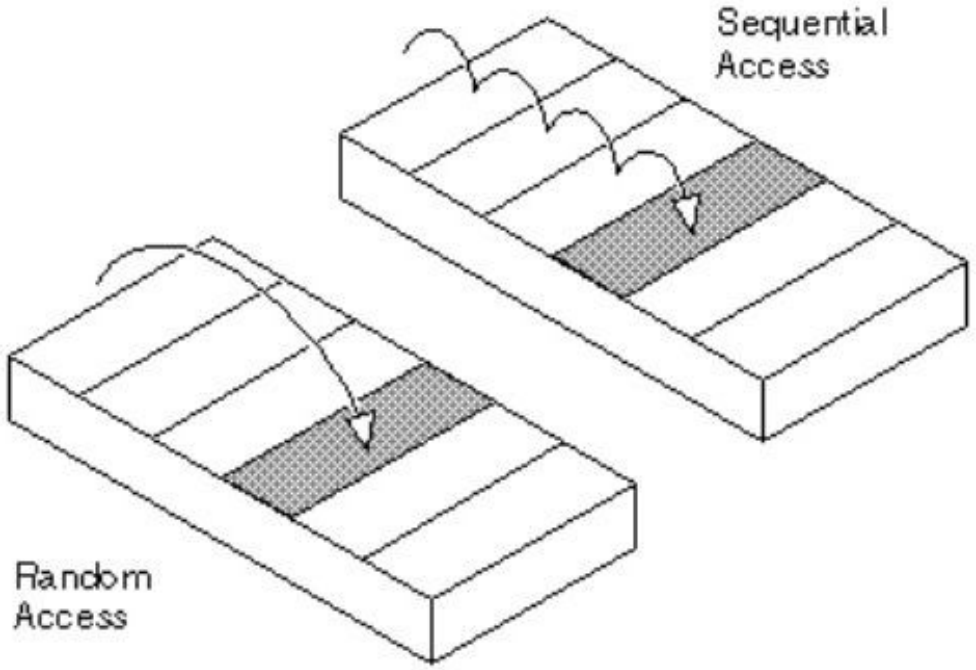
- Операції:
  1) read(n), write(n) або
  2) position_file(n) + read_next(), write_next().

04.6. Логічна організація файлу

sequential access	implementation for direct access
reset	cp = 0;
read_next	read cp ; cp = cp + 1;
write_next	write cp; cp = cp + 1;

Реалізація послідовного доступу операціями прямого доступу

04.7. Логічна організація файлу



Порівняння послідовного та прямого доступу

04.8. Логічна організація файлу

3. Доступ на основі індексів (Indexed Access):

- Для кожного файлу створюється додаткова структура даних (іноді у вигляді окремого файлу), яка містить індекси логічних записів (таблиця індексів).
- Цей спосіб доступу вигідно використовувати для великих файлів з однорідними даними (наприклад, системні лог-файли), оскільки пошук потрібного логічного запису в таблиці індексів займає значно менше часу, ніж пошук в самому файлі.
- Приклад: Indexed Sequential Access Method (ISAM) (IBM)

04.9. Логічна організація файлу

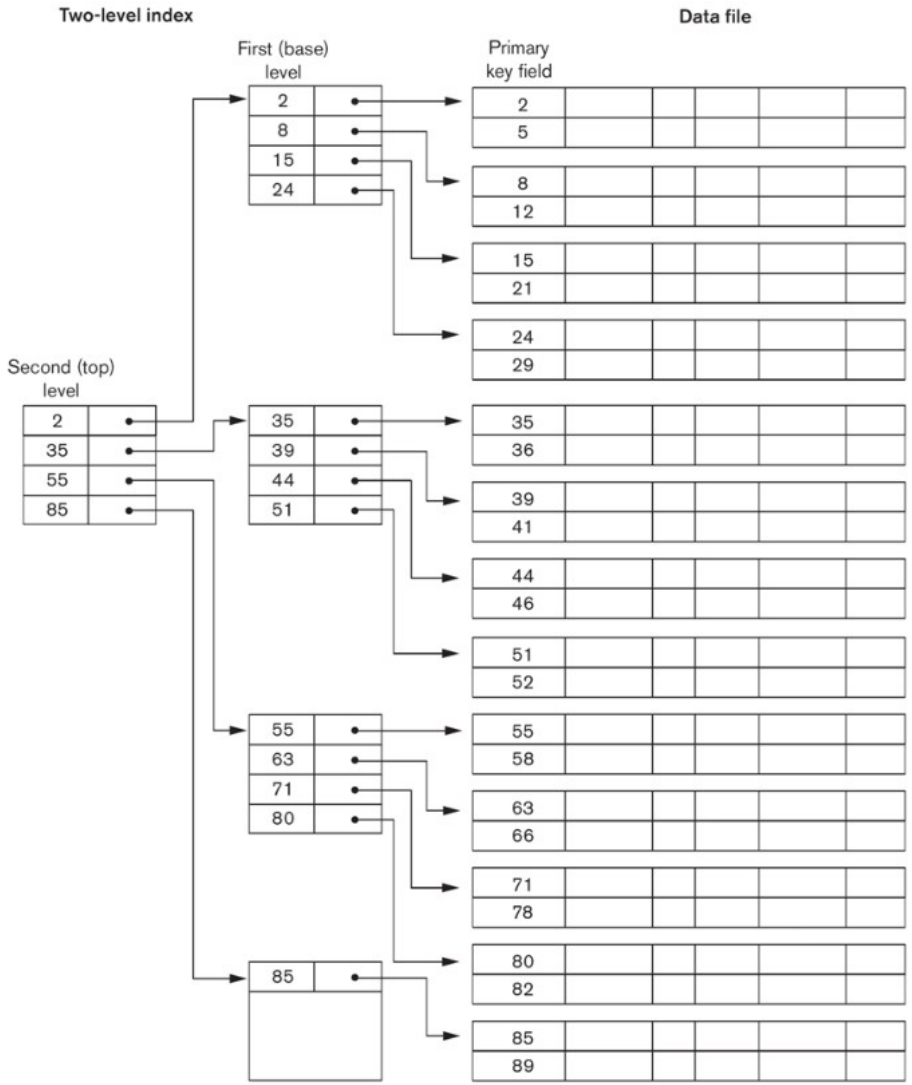
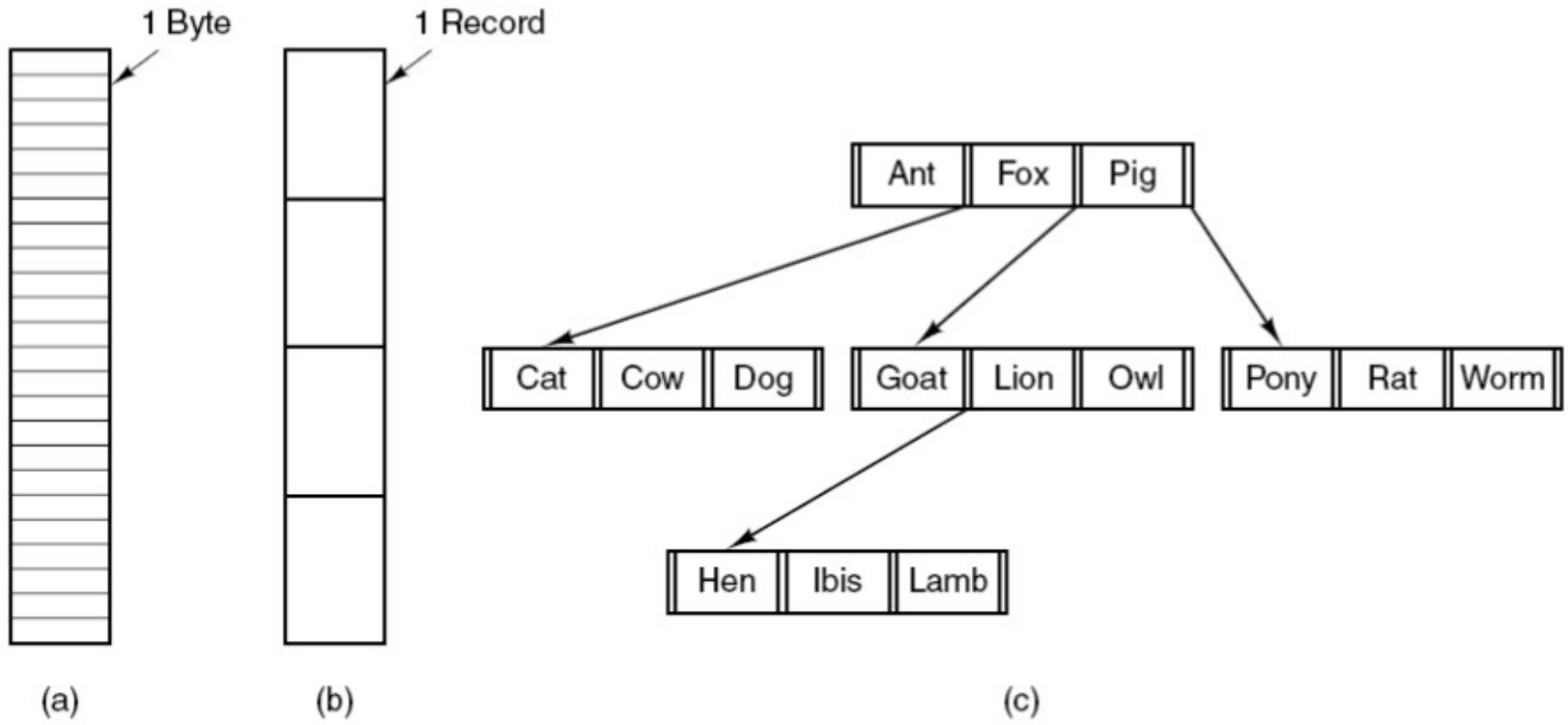


Схема роботи Indexed Sequential Access Method (ISAM) (IBM)

04.10. Логічна організація файлу



Порівняння логічних структур файла для: a) послідовного доступу, b) прямого доступу, c) доступу на основі індексів

05.1. Фізична організація файлу

- Фізична організація файлу визначає спосіб, в який файл розміщено у пристрої зовнішньої пам’яті.
- Цей спосіб залежить від специфіки цього пристрою.
- У випадку дискових накопичувачів файл розбивається на блоки однакової величини, які є одиницею обміну та зберігання даних.

05.2. Фізична організація файлу

Способи організації директорій:

1. Лінійний список (linear list) → список імен файлів і покажчиків на відповідні блоки файлу → простий в реалізації, повільний в роботі (пошук файлу повним перебором у списку) → використання кешу для службової інформації по доступу до директорії

2. Хеш-таблиця (hash table) → використовується додаткова хеш-таблиця імен файлів, яка дозволяє збільшити швидкість пошуку потрібного входження у лінійному списку (directory search time)

05.3. Фізична організація файлу

Варіанти організації фізичного розташування файлу на жорсткому диску:

1. Неперервне розташування блоків (Contiguous allocation)

2. Розташування блоків зв’язним списком (Linked allocation)

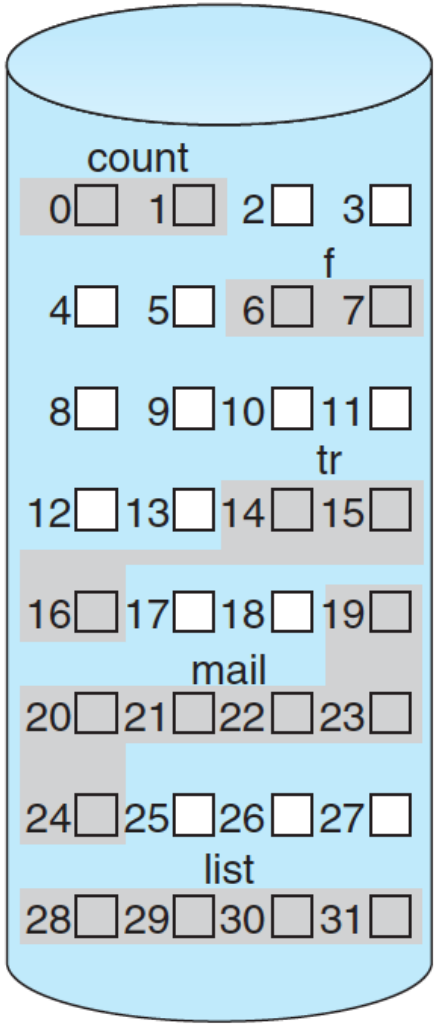
3. Використання індексного блоку (Indexed Allocation)

05.4. Фізична організація файлу

1. Неперервне розташування блоків (Contiguous allocation):

- Блоки одного файлу розміщуються єдиним кластером і йдуть впорядковано один за одним.
- Перевага: мінімізація операцій пошуку блоків на диску.
- Проблема: важко або неможливо знайти вільне місце для нового файлу в умовах фрагментації дискового простору.

05.5. Фізична організація файлу



directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

Схема роботи Contiguous allocation

05.6. Фізична організація файлу

2. Розташування блоків зв’язним списком (Linked allocation):

- Кожний блок файлу містить посилання на наступний блок.
- Недолік: складність організації доступу до довільного місця файлу.
- Перевага: вирішує проблему зовнішньої фрагментації та знаходження вільного місця.

05.7. Фізична організація файлу

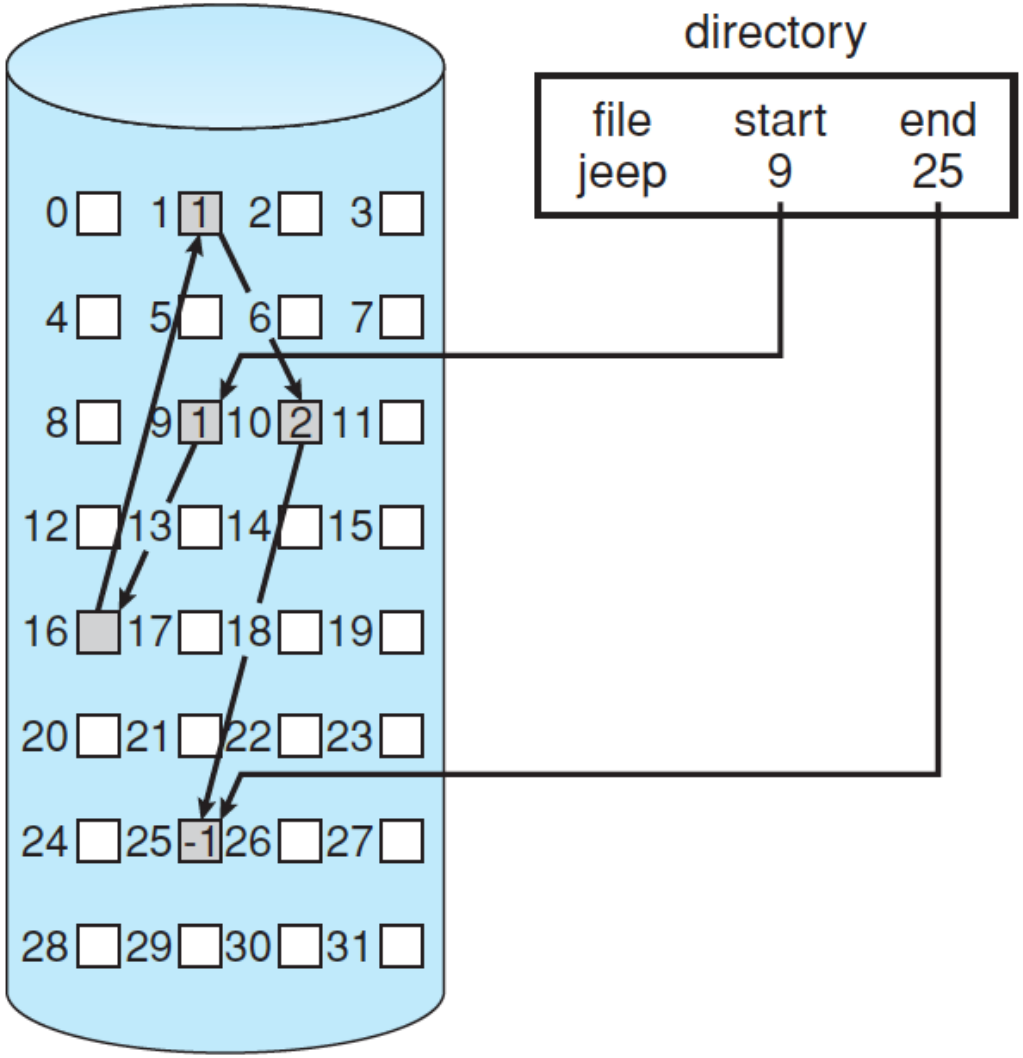
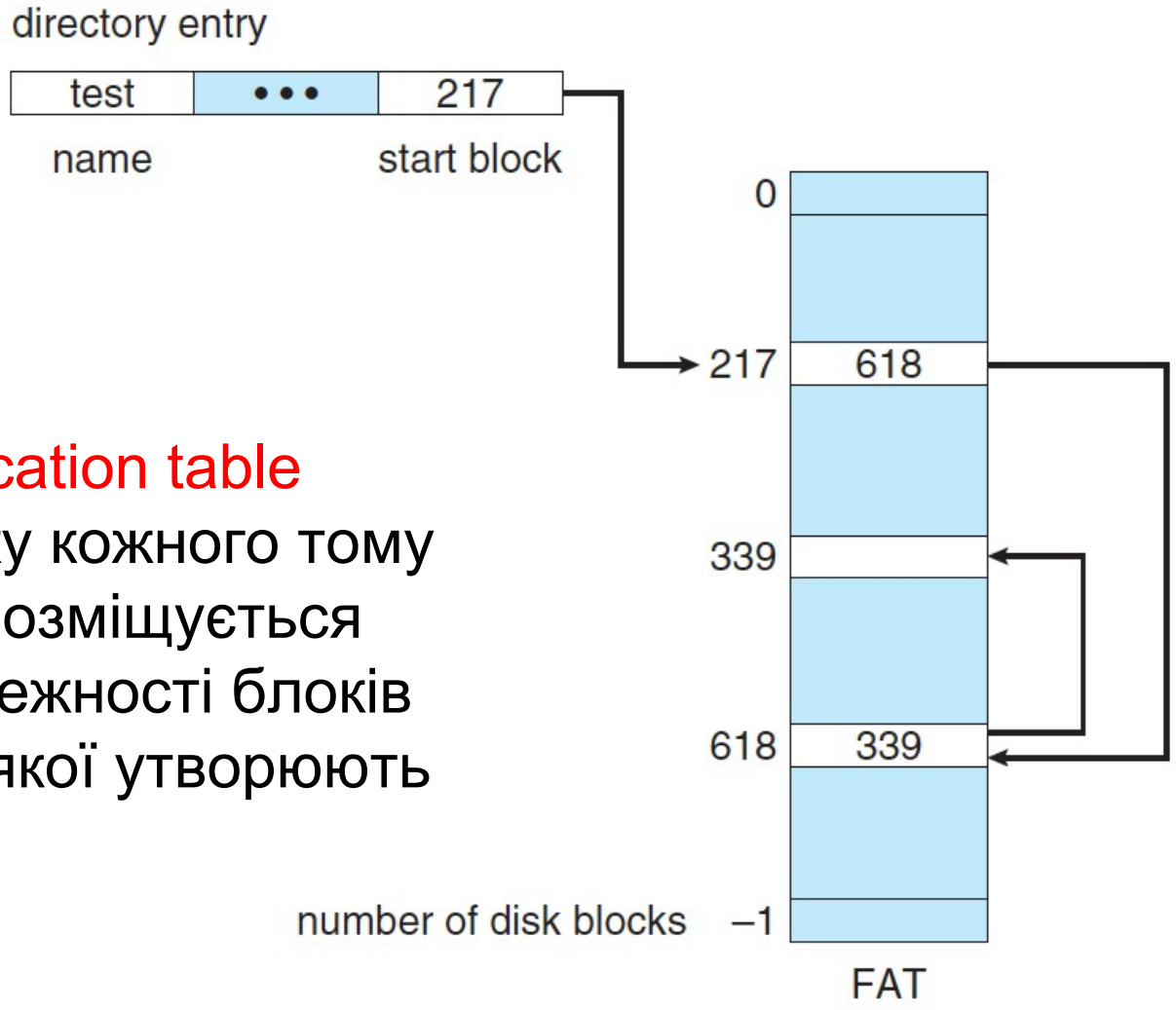


Схема роботи
Linked allocation

05.8. Фізична організація файлу



Варіант: file-allocation table (FAT): на початку кожного тому (volume) диску розміщується таблиця приналежності блоків файлам, рядки якої утворюють зв’язні списки.

05.9. Фізична організація файлу

3. Використання індексного блоку (Indexed Allocation):

- Для кожного файлу виділяється спеціальний блок, який містить покажчики на усі блоки файлу.
- Покажчик → пара: порядковий номер блоку, покажчик на блок
- Перевага: такий спосіб значно зменшує час пошуку довільного блоку файлу.

05.10. Фізична організація файлу

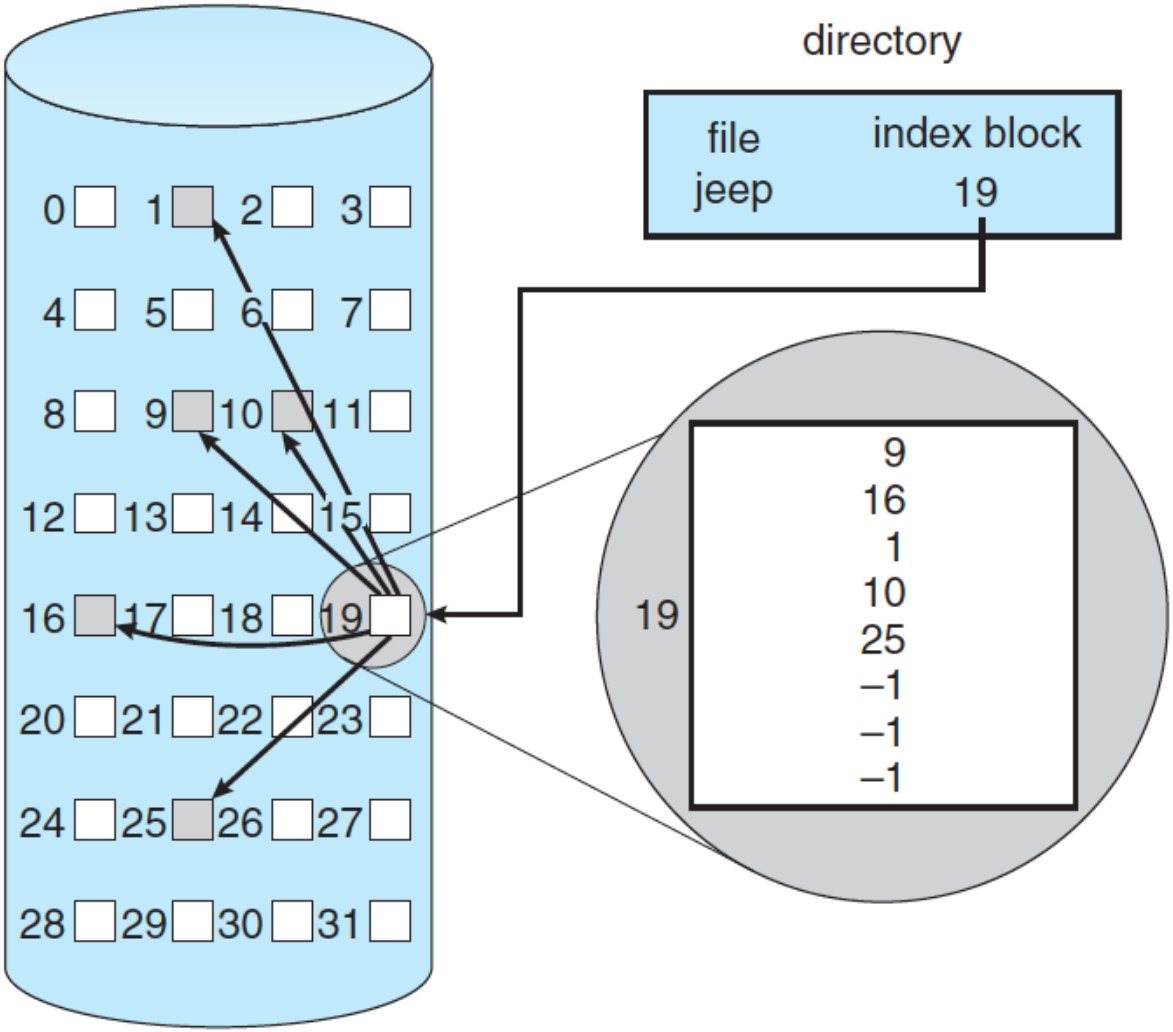


Схема роботи Indexed Allocation

05.11. Фізична організація файлу

Проблема Indexed Allocation:

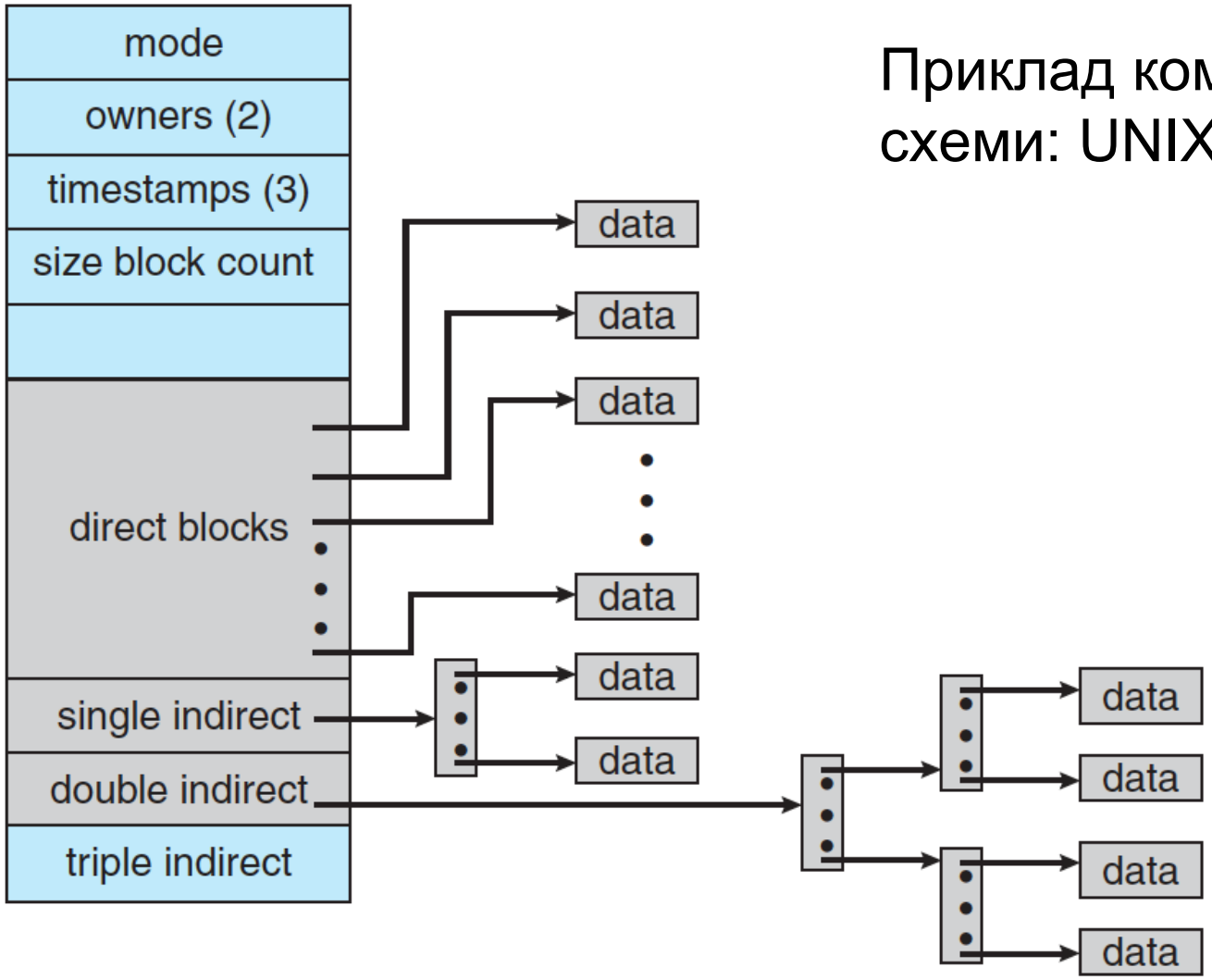
як зберігати файли з великою кількістю блоків?

1) Зв’язний список індексних блоків (Linked scheme);

2) Ієрархія індексних блоків (Multilevel index);

3) Комбінована схема (Combined scheme).

05.12. Фізична організація файлу



Приклад комбінованої схеми: UNIX inode

05.13. Фізична організація файлу

Проблеми, які виникають в процесі фізичного збереження даних:

1. Контроль вільного місця на диску (або ПЗП іншого типу).

2. Фрагментація дискового простороу → методи дефрагментації + методи розміщення нових даних, які зменшують фрагментацію.

3. Рівномірний розподіл кількості операцій читання/запису по ділянкам пам’яті (флеш-накопичувачі, SSD).

06.1. Класифікація файлових систем

Варіанти фізичного розташування файлової системи:

1) Один пристрій зовнішньої пам’яті – одна файлова система;

2) Розбиття одного пристрою зовнішньої пам’яті на декілько ділянок (partitions), кожна з яких містить свою файлову систему (використовується також термін «логічний диск»);

3) Використання декількох пристроїв зовнішньої пам’яті для одної файлової системи → RAID (Redundant Array of Independent Disks).

06.2. Класифікація файлових систем

Дві основні «архітектури» ФС (виникли історично):

1) окремі логічні диски з незалежними файловими системами → драйвери файлових систем → менеджер файлових систем (Windows)

2) загальна файлова система, до якої підключаються підпорядкованими частинами файлові системи пристроїв зовнішньої пам’яті (UNIX/Linux) → монтування файлової системи → точка монтування

06.3. Класифікація файлових систем

Інші «архітектурні» особливості ФС:

- внесення змін → ФС з веденням журналу змін: Journaling FS + versioning FS (ext4) → можливість відкату до попередньої [непошкодженої] версії ФС
- стиснення даних → відбувається на льоту в моменти запису даних у пристрій зовнішньої пам’яті

06.4. Класифікація файлових систем

- безпека → шифрування даних → Encrypting FS → real-time encryption/decryption
- віртуальні файлові системи в рамках технологій апаратної та програмної віртуалізації
- мережні/розподілені файлові системи → NFS, Google File System, Lustre file system, etc.
- хмарні системи збереження даних (Cloud storage)

06.5. Класифікація файлових систем

Альтернативні способи організації даних:

- Системи лінійних каталогів (приклад: бібліотечний каталог) → для збереження однорідних / однотипних даних → Oracle Berkeley DB
- Реляційні системи управління базами даних (relational database management system) → формалізація вимог до системи збереження даних → ініціатива Oracle: заміна операційної системи системою управління базами даних

06.6. Класифікація файлових систем

- Семантичні файлові системи (Semantic file systems), приклади: SemFS, Tagxfs
- Об’єктні «файлові системи» → Object storage → Each object typically includes the data itself, a variable amount of metadata, and a globally unique identifier → OceanStore project (UC Berkeley)
- Гіпертекстові системи. Приклади: Web, Wiki, personal wiki.