# บทที่ 2
# วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

ทฤษฎีบทนี้เป็นหลักการที่ถูกนำไปใช้ในการพัฒนาโปรแกรมป้องกันการละเมิดลิขสิทธิ์ โดยกล่าวถึง การค้นหาข้อมูลแบบเชิงลำดับ(Sequential Search), FileSystemWatcher, MPEG, ID3

## 1. การค้นหาแบบเรียงลำดับ (Sequential Search)

การค้นหาข้อมูลแบบเชิงลำดับ(Sequential Search) เป็นวิธีการค้นหาข้อมูลที่ง่ายและตรงไปตรงมาที่สุด การค้นหาทำได้โดยนำค่าหลักไปเปรียบเทียบกับข้อมูลทั้งหมดทีละตัวตั้งแต่ตัวแรกเรียงตามลำดับจนกว่าจะพบข้อมูลที่ต้องการ หรือเปรียบเทียบไปจนถึงตัวสุดท้ายและพบว่าไม่มีข้อมูลนี้อยู่ ตัวอย่าง เช่น มีข้อมูลอยู่ 10 จำนวน ดังนี้

18 3 39 70 27 8 1 31 2 50

เมื่อต้องการค้นหาเลขจำนวนเต็ม 2 ให้นำไปเปรียบเทียบกับข้อมูลทีละตัว เริ่มที่ตัวแรกคือ 18 ถ้าไม่ใช่ข้อมูลที่ต้องการให้ไปเปรียบเทียบกับข้อมูลตัวถัดไป จนกว่าจะพบข้อมูลที่ต้องการก็หยุดค้น หรือค้นหาจนถึงข้อมูลตัวสุดท้ายแล้วยังไม่พบข้อมูลที่ต้องการก็สรุปได้ว่าไม่มีข้อมูลที่ต้องการ ในตัวอย่างนี้การค้นหา 2 ต้องมีการเปรียบเทียบกับค่าต่าง ๆ ทั้งหมด 9 ครั้ง จึงจะพบข้อมูลที่ต้องการ

ดังนั้นจะเห็นว่าการค้นหาข้อมูลแต่ละครั้ง ถ้ามีข้อมูลทั้งหมด n จำนวน จำนวนครั้งของการเปรียบเทียบเป็นดังนี้

กรณีที่ดีที่สุด คือกรณีที่ข้อมูลที่ต้องการค้นอยู่ที่ตำแหน่งแรก จำนวนครั้งของการเปรียบเทียบเท่ากับ 1 ครั้ง

กรณีที่แย่ที่สุด คือกรณีที่ข้อมูลที่ต้องการค้นอยู่ที่ตำแหน่งสุดท้าย หรือไม่มีข้อมูลนั้นอยู่เลย จำนวนครั้งของการเปรียบเทียบเท่ากับ n ครั้ง

กรณีเฉลี่ย การค้นหาข้อมูลใด ๆ ในแต่ละครั้งมีจำนวนครั้งของการเปรียบเทียบเฉลี่ยเป็นดังนี้จำนวนครั้งของการเปรียบเทียบเฉลี่ย $= (n+1)/2$ ครั้ง

กรณีที่ข้อมูลที่มีการเรียงลำดับอยู่แล้ว การค้นหาข้อมูลจะรวดเร็วขึ้น เนื่องจากสามารถใช้ความสัมพันธ์ของการเรียงลำดับที่มีอยู่มาช่วยในการค้นหาได้ โดยเฉพาะการค้นหา

ข้อมูลที่ไม่มีอยู่ในโครงสร้าง การค้นหาไม่จำเป็นต้องค้นไปจนถึงข้อมูลตัวสุดท้าย เพราะเมื่อเจอข้อมูลที่มีค่ามากกว่า (ข้อมูลเรียงจากน้อยไปมาก) สามารถหยุดค้นได้ทันที ตัวอย่างการค้นหาในข้อมูลที่เรียงลำดับจากน้อยไปมากดังต่อไปนี้

1 2 3 8 18 27 31 39 50 70

เมื่อต้องการค้นหา 2 ต้องเปรียบเทียบทั้งหมด 2 ครั้ง และถ้าต้องการค้นหาค่าหลักที่ไม่มีอยู่ในข้อมูลชุดนี้ เช่น 7 จะทำการเปรียบเทียบตั้งแต่ข้อมูลตัวแรกจนกระทั่งพบตัวที่มีค่ามากกว่า 7 คือ 8 แสดงว่าไม่มี 7 อยู่ในข้อมูลชุดนี้อย่างแน่นอน จำนวนครั้งของการเปรียบเทียบทั้งหมดเท่ากับ 4 ครั้ง ซึ่งถ้าเป็นข้อมูลที่ไม่ได้เรียงลำดับจะต้องค้นหาไปจนถึงตัวสุดท้ายจึงจะสรุปได้ว่าไม่พบข้อมูลที่ต้องการ

การค้นหาข้อมูลแบบเรียงลำดับเป็นวิธีการที่ไม่ซับซ้อน เป็นการค้นหาแบบตรงไปตรงมา ใช้กับข้อมูลที่เรียงลำดับแล้วหรือยังไม่เรียงลำดับก็ได้ทั้งสิ้น แต่ถ้าเป็นข้อมูลเรียงลำดับแล้วจำนวนครั้งของการเปรียบเทียบเฉลี่ยจะน้อยกว่าข้อมูลที่ยังไม่ได้เรียงลำดับ

**2. FileSystemWatcher**

FileSystemWatcher การสร้างตัวอย่างส่วนประกอบ FileSystemWatcher ซึ่งจะแสดงตัวอย่างของส่วนประกอบ FileSystemWatcherเมื่อต้องการสร้างระบบแฟ้มที่ติดตามเข้าไปในโปรแกรมของคุณ ส่วนประกอบ FileSystemWatcher ที่จะเชื่อมต่อกับไดเรคทอรี่ และนาฬิกาสำหรับการเปลี่ยนแปลงอย่างจำเพาะเจาะจงภายในมัน เช่นเดียวกันกับการสร้างของแฟ้มใหม่การเพิ่มของไดเรคทอรี่ย่อย และการเปลี่ยนชื่อของแฟ้มหรือไดเรคทอรี่ย่อยคุณสามารถเพิ่มตัวอย่างของส่วนประกอบFileSystemWatcher ให้ฟอร์มวินโดวส์ , ฟอร์มเว็ป , และคลาสส่วนประกอบ

ส่วนประกอบ FileSystemWatcher ไม่มีส่วนติดต่อผู้ใช้จำลอง ถ้าทำการเพิ่มตัวอย่างของส่วนประกอบ FileSystemWatcher ให้ผู้ออกแบบจำลอง เช่นเดียวกันกับส่วนประกอบผู้ออกแบบฟอร์มวินโดวส์ปรากฏในพื้นที่เล็กข้างล่างเส้นขอบข้างใต้ของฟอร์ม พื้นที่นี้รู้ว่าเป็นถาดส่วนประกอบ และกฎหมายเป็นสถานที่ที่จะแสดงรายการมองไม่เห็นทั้งหมดที่เกี่ยวข้อง กับฟอร์ม

การใช้งานเพื่อตรวจจับความเปลี่ยนแปลงของไฟล์ต่างๆในระบบสามารถทำได้โดย

1) การสร้าง FileSystemWatch object ขึ้นมาก่อนแล้วกำหนดตำแหน่งของ directory ที่ต้องการตรวจจับความเปลี่ยนแปลงด้วยการใช้propertyPath

2) เพิ่มและลบeventลงในobjectด้วยการใช้delegate

3) เปิดการทำงานให้ event ด้วยการกำหนดค่า property EnableRaisingEvents ให้เป็น true

ตัวอย่าง

```
FileSystemWatchermy Watcher = new System.IO.FileSystemWatcher();
```

**3. MPEG**

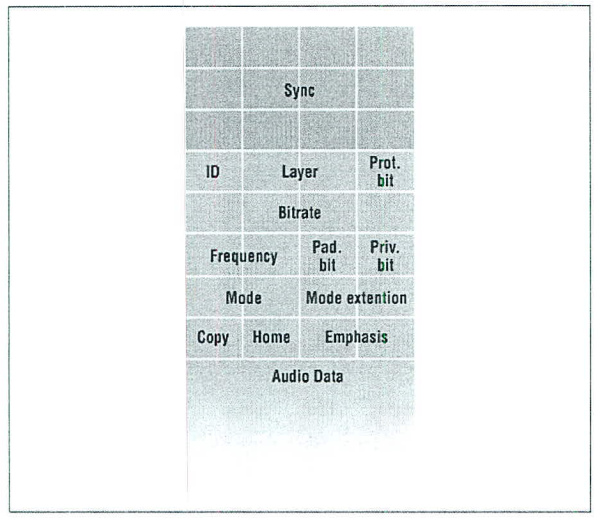
MPEG ย่อมาจาก Moving Picture Experts Group เป็นมาตรฐานการบีบอัดสัญญาณภาพและเสียงของภาพเคลื่อนไหวโดยการใช้ระบบ DCT หรือ Discrete Cosine Transform ซึ่งเป็นการแทนค่าตัวแปลของสัญญาณต่าง ๆ ด้วยสมการทางคณิตศาสตร์ ซึ่งก็คล้ายกับขั้นตอนการเข้ารหัสภาพนิ่งแบบ JPEG แต่จะลดจำนวนข้อมูลที่ซ้ำ ๆ กันของภาพต่อ ๆ ไป เพราะขั้นตอนในการเข้ารหัสเพื่อบีบอัดสัญญาณวีดีโอ จะนานกว่าขั้นตอนการถอดรหัสข้อมูลออกไปเป็นภาพและเสียง

MP3 (เอ็มพีสาม หรือ เอ็มพีทรี) เป็นวิธีการเข้ารหัสสัญญาณเสียงดิจิตัลที่เป็นที่นิยมแบบหนึ่ง ใช้วิธีการบีบอัดข้อมูลแบบมีการสูญเสียข้อมูลบางส่วนบางส่วน หรือ lossy ออกแบบมาเพื่อใช้ลดปริมาณข้อมูลเสียงให้เหลือเพียงเล็กน้อย (ส่วนมากจะได้ที่อัตรา 10 ต่อ 1) แต่ข้อมูลที่ลดลงมานี้ก็ยังให้คุณภาพเสียงที่ดีใกล้เคียงกับสัญญาณเสียงต้น ฉบับโดยทดสอบกับผู้ฟังส่วนใหญ่ ในการใช้งานส่วนใหญ่คำว่า MP3 จะเป็นกล่าวอ้างถึงแฟ้มที่ใช้เก็บเสียงหรือดนตรีในรูปแบบ MP3 บนเครื่องคอมพิวเตอร์ หรือเครื่องเล่น MP3

คำว่า MP3 ได้มาจากคำว่า "MPEG-1 Audio Layer 3" หรือในคำที่เป็นทางการว่า "ISO/IEC 11172-3 Layer 3" อย่างไรก็ตามแฟ้มนามสกุล ".mp3" บางแฟ้มก็ใช้การเข้ารหัสแบบใหม่ที่มีชื่อว่า "MPEG-2 Audio Layer 3" หรือ "ISO/IEC 13818-3 Layer 3"

MP3 เป็นรูปแบบการบีบอัดข้อมูลแบบมีการสูญเสียข้อมูลบางส่วน ใช้แทนข้อมูลเสียงที่เข้ารหัสแบบ PCM ให้มีขนาดที่เล็กโดยตัดข้อมูลบางส่วนที่พิจารณาแล้วว่าระบบการได้ยินของมนุษย์เกือบจะไม่สามารถรับฟังได้ (แนวคิดนี้คล้ายกับการบีบอัดข้อมูลภาพแบบ JPEG) วิธีการต่างๆที่ช่วยให้สามารถตัดข้อมูลบางส่วนออกไปได้ได้ถูกนำมาใช้กับ MP3 รวมทั้ง psychoacoustics ข้อมูลเสียงแบบ MP3 สามารถบีบอัดให้มีขนาดที่แตกต่าง หรือมี อัตราบิท ที่หลากหลายขึ้นกับขนาดของข้อมูลและคุณภาพเสียง

เป็นรูปแบบแฟ้มที่เป็นการบีบอัดข้อมูลแบบมีการสูญเสียข้อมูลบางส่วนออกมา เพื่อให้ปริมาณข้อมูลลดลง แต่ยังคงคุณภาพใกล้เคียงกับสัญญาณเสียงเดิม อาจมีการเปลี่ยนแปลงบ้าง แต่เป็นเพียงเล็กน้อย และในการเข้ารหัสแบบ MP3 เป็นการเข้ารหัสของเพลงเป็นส่วนใหญ่ เพื่อให้สามารถใช้เนื้อที่ได้มากขึ้น และสามารถรวบรวมแฟ้มเพลงหลายๆแฟ้มมารวมอยู่ในรูปของ MP3 ทำให้ไม่สิ้นเปลืองเนื้อที่ในการจัดเก็บข้อมูล




**ภาพที่ 1** The MP3 frame header

**ภาพที่ 2** ภาพโครงสร้าง data bits ของ MP3 frame header โดยเปรียบเทียบกับตารางที่ 1

**ตารางที่ 1** ตำแหน่งโครงสร้าง Data bits ของ MP3 frame header

| Position | Purpose | Length (in bits) |
|---|---|---|
| A | Frame sync | 11 |
| B | MPEG audio version (MPEG 1, 2, etc.) | 2 |
| C | MPEG layer (Layer I, II, III, etc.) | 2 |
| D | Protection (if on, then checksum follows header) | 1 |
| E | Bitrate index (lookup table used to specify bitrate for this MPEG version and layer) | 4 |
| F | Sampling rate frequency (44.1kHz, etc., determined by lookup table) | 2 |
| G | Padding bit (on or off, compensates for unfilled frames) | 1 |
| H | Private bit (on or off, allows for application-specific triggers) | 1 |
| I | Channel mode (stereo, joint stereo, dual channel, single channel) | 2 |
| J | Mode extension (used only with joint stereo, to conjoin channel data) | 2 |
| K | Copyright (on or off) | 1 |
| L | Original (off if copy of original, on if original) | 1 |
| M | Emphasis (respects emphasis bit in the original recording; now largely obsolete)) | 2 |
| | | 32 total header bits |

4. ID3

ID3v1 หลังจากการสร้างมาตรฐาน MP3 ปรากฏปัญหา กับข้อมูลที่เก็บเกี่ยวกับแฟ้ม ใน 1996 Eric Kemp จึงมีแนวความคิดที่จะเพิ่มกลุ่มเล็กของข้อมูลที่แฟ้มเสียงการแก้ดังเช่นปัญหา มาตรฐานถูกเรียก ID3v1 และมันคือการถอด facto อย่างรวดเร็ว มาตรฐานสำหรับ metadata ที่เก็บในMP3รูปแบบนี้ถูกใช้กับแฟ้มจำนวนมากโดย tag ID3v1 ครอบครอง 128 ไบท์tagเก็บ 30 ไบท์สำหรับหัวเรื่องศิลปินอัลบัมและ"ข้อความ"และไบท์ที่จะระบุประเภทของเพลงจาก predefined แสดงรายการของ 80 ค่าขยายต่อมารายการนี้ถึง 148 ค่า การปรับปรุงหนึ่งถึง ID3v1 ถูกทำโดย

Michael Mutschler ใน1997 โดยฟิลด์ข้อความเล็กเกินไปที่จะเขียนประโยชน์สิ่งใดๆได้ เขาได้แบ่งมันเป็นสองไบท์ และใช้เก็บตัวเลขแทร็ค

Enhanced tag Enhanced tagคือบล็อคข้อมูลพิเศษ tag ID3v1 เป็นตัวขยายหัวเรื่องศิลปินและฟิลด์อัลบัมโดย 60 ไบท์แต่ละครั้งกลายเป็น freetext ประเภทไบท์หนึ่ง (ค่า0-5) ความเร็ว และเริ่มต้น และเวลาหยุดของเพลงในแฟ้ม MP3 ถ้าไม่มีฟิลด์ถูกใช้มันจะที่ถูกละเว้น

**ตารางที่ 2** Layout

| Field | Length | Description |
|---|---|---|
| header | 3 | "TAG" |
| title | 30 | 30 characters of the title |
| artist | 30 | 30 characters of the artist name |
| album | 30 | 30 characters of the album name |
| year | 4 | A four-digit year |
| comment | 28 or 30 | The comment. |
| zero-byte | 1 | If a track number is stored, this byte contains a binary 0. |
| track | 1 | The number of the track on the album, or 0. Invalid, if previous byte is not a binary 0. |
| genre | 1 | Index in a list of genres, or 255 |

**ตารางที่ 3** Extended tag

| Field | Length | Description |
|---|---|---|
| header | 4 | "TAG+" |
| title | 60 | Next 60 characters of the title (90 characters total) |
| artist | 60 | Next 60 characters of the artist name |
| album | 60 | Next 60 characters of the album name |
| speed | 1 | 0=unset, 1=slow, 2= medium, 3=fast, 4=hardcore |
| genre | 30 | A free-text field for the genre |
| start-time | 6 | the start of the music as mmm:ss |
| end-time | 6 | the end of the music as mmm:ss |

ID3v2 มาตรฐานใหม่ที่เรียก ID3v2 ถูกสร้างใน1998 โดยเฟรมจำนวนมากมายแต่ละตัวจะมีmetadata 1 ตัว เป็นต้นว่าเฟรม TIT2 บรรจุหัวเรื่อง และเฟรม WOAR บรรจุ URL ของเว็บไซท์ของศิลปินในมาตรฐาน ID3v2 มี 84 ชนิดของเฟรม มีเฟรมมาตรฐานสำหรับบรรจุครอบคลุมศิลปะ BPMลิขสิทธิ์ และใบอนุญาตเป็นลักษณะของเพลงและข้อความและ URL ข้อมูลมี สามเวอร์ชันของ ID3v2 โดย ID3v2.2 คือเวอร์ชันแรกของID3v2 มันใช้ตัวค้นหาเฟรมตัวอักษรค่อนข้างดี( TT2 สำหรับหัวเรื่องแทนที่ TIT2 )แต่ v2.3 และ v2.4 ได้พัฒนารูปแบบการค้นหาเฟรมทำให้มาตรฐาน v2.2 นี้ถูกคิดว่าล้าสมัย ตอนนี้ID3v2.3 ที่ขยายตัวค้นหาเฟรมถึงสี่ตัวอักษร, และเพิ่มเฟรมจำนวนมากมาย โครงค้ำตรงรูปตัว ? หรือ Vสามารถบรรจุค่าหลายอัน เพราะถูกที่แยกออกมา กับตัวอักษร สิ่งนี้คือส่วนมากใช้กว้างขวางเวอร์ชันของ tag ID3v2 ถึงแม้ว่าเวอร์ชันของ ID3v2 จะแตกต่างกันตรงที่วิธีการอ่านข้อมูลแต่ แทบมองไม่เห็นแต่ความแตกต่างระหว่างเวอร์ชันทั้งหมด

## 5. สรุป

การตรวจสอบไฟล์เพลงต่างๆนอกจากจะตรวจสอบจากนามสกุลของไฟล์ยังจะสามารถตรวจสอบได้จาก Header File ซึ่งมีมาตรฐานการกำหนด Header โดยการตรวจสอบจาก Header File ต้องใช้วิธีการค้นหาซึ่งมีหลายแบบ ซึ่งแบบหนึ่งที่มีความซับซ้อนในการค้นหาน้อย คือ การค้นหาแบบเชิงลำดับ (Sequential Search) ส่วนต่างๆของการคัดลอกไฟล์ลงเครื่องคอมพิวเตอร์ในระบบปฏิบัติการวินโดวส์ จะมีตัว FileSystemWatcher เป็นตัวช่วยในการแจ้งเตือนเมื่อมีการคัดลอกไฟล์เกิดขึ้น