หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ ๕

การทำงานของคำสั่งการจัดเก็บแฟ้มข้อมูลแบบดัชนี

ในภาษาเบสิก ผู้ใช้สามารถใช้งานแฟ้มข้อมูลได้ ๒ แบบ คือแฟ้มข้อมูลแบบเรียงลำดับ และแฟ้มข้อมูลแบบสุ่ม โดยที่แฟ้มข้อมูลแต่ละแบบก็มีข้อดีและข้อเสียของตัวเอง การพัฒนาคำสั่งการจัดเก็บแฟ้มข้อมูลแบบดัชนีขึ้น เพื่อรวมข้อดีและลดข้อเสียของการจัดเก็บแฟ้มข้อมูลที่มีอยู่เดิม และเพื่อให้การทำงานของคำสั่งมีขนาดเล็ก จะมีการเรียกใช้การทำงานของคำสั่งของการจัดเก็บแฟ้มข้อมูลแบบสุ่มที่มีอยู่ในตัวอินเตอร์เพรตเตอร์

๕.๑ การจัดเก็บแฟ้มข้อมูลในภาษาเบสิก

ในภาษาเบสิก ผู้ใช้สามารถจัดเก็บแฟ้มข้อมูลได้ ๒ แบบ คือแฟ้มข้อมูลแบบเรียงลำดับ และแฟ้มข้อมูลแบบสุ่ม แฟ้มข้อมูลแต่ละแบบมีลักษณะดังนี้คือ

๕.๑.๑ แฟ้มข้อมูลแบบเรียงลำดับ เป็นแฟ้มข้อมูลที่มีลักษณะการจัดเก็บระเบียนติดต่อกันไป โดยที่ความยาวของระเบียนไม่จำเป็นต้องคงที่ ดังนั้นจะมีเครื่องหมายตัวหนึ่งคือ "ODOA" เป็นตัวแยกระเบียน และเนื่องจากความยาวและจำนวนเขตข้อมูลในระเบียนหนึ่ง ๆ ก็ไม่จำเป็นต้องคงที่ ดังนั้นก็จะมีเครื่องหมายอีกตัวหนึ่งคือ "2C" เป็นตัวแยกเขตข้อมูล นอกจากนี้จะมีเครื่องหมาย "1A" ปิดท้ายแฟ้มข้อมูลเพื่อบอกการหมดของแฟ้มข้อมูล

คำสั่งต่าง ๆ ที่เกี่ยวข้องกับการใช้แฟ้มข้อมูลแบบเรียงลำดับมีดังนี้คือ

๕.๑.๑.๑ OPEN สำหรับเปิดแฟ้มข้อมูลแบบเรียงลำดับ และมีรูปแบบดังนี้

OPEN <โมด>, [#]<หมายเลขแฟ้มข้อมูล>, <ชื่อแฟ้มข้อมูล>

โมด (Mode) เท่ากับ "O" เพื่อระบุว่าต้องการเปิดแฟ้มข้อมูลแบบเอาท์พุท

เท่ากับ "I" เพื่อระบุว่าต้องการเปิดแฟ้มข้อมูลแบบอินพุท

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขระหว่าง ๑ ถึง จำนวนแฟ้มข้อมูลที่ต้องการเปิดพร้อมกัน ซึ่งระบุตอนเริ่มต้นอินเตอร์เพรตเตอร์

ชื่อแฟ้มข้อมูล เท่ากับ ชื่อแฟ้มข้อมูลที่ต้องการใช้งาน

๔.๑.๑.๒ PRINT # สำหรับบันทึกระเบียนเข้าไปในแฟ้มข้อมูล และมีรูปแบบดังนี้

PRINT #<หมายเลขแฟ้มข้อมูล>, <รายชื่อเขตข้อมูล>

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขของหมายเลขแฟ้มข้อมูล ตอนเปิดแฟ้มข้อมูล

รายชื่อเขตข้อมูล เท่ากับ รายชื่อของตัวแปรที่เก็บค่าของเขตข้อมูล

๔.๑.๑.๓ INPUT # สำหรับอ่านระเบียนของแฟ้มข้อมูล และมีรูปแบบดังนี้

INPUT #<หมายเลขแฟ้มข้อมูล>,<รายชื่อเขตข้อมูล>

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขของหมายเลขแฟ้มข้อมูล ตอนเปิดแฟ้มข้อมูล

รายชื่อเขตข้อมูล เท่ากับ รายชื่อของตัวแปรที่ต้องการนำค่าเขตข้อมูลมาใส่

๔.๑.๑.๔ EOF สำหรับตรวจสอบการหมดของแฟ้มข้อมูล และมีรูปแบบดังนี้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

EOF (<หมายเลขแฟ้มข้อมูล>)

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขของหมายเลขแฟ้มข้อมูล ตอนเปิดแฟ้มข้อมูล

๕.๑.๑.๔ CLOSE สำหรับปิดแฟ้มข้อมูล และมีรูปแบบดังนี้

CLOSE [#]<หมายเลขแฟ้มข้อมูล>

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขของหมายเลขแฟ้มข้อมูล ตอนเปิดแฟ้มข้อมูล

การใช้แฟ้มข้อมูลแบบเรียงลำดับ ทำให้ประหยัดเนื้อที่ในการจัดเก็บบนหน่วยความจำสำรอง และง่ายต่อการเขียนโปรแกรม แต่ข้อจำกัดของแฟ้มข้อมูลแบบเรียงลำดับก็คือ การอ่านหรือบันทึกข้อมูลจะกระทำเรียงลำดับจากต้นของแฟ้มข้อมูลเป็นต้นไป ทำให้ขาดความรวดเร็วในการใช้ข้อมูล

๕.๑.๒ แฟ้มข้อมูลแบบสุ่ม เป็นแฟ้มข้อมูลที่มีลักษณะการจัดเก็บระเบียนที่มีความยาวคงที่ ดังนั้นจึงไม่จำเป็นต้องมีเครื่องหมายใด ๆ สำหรับเป็นตัวแยกระเบียน การอ่านหรือบันทึกระเบียน กระทำโดยใช้ลำดับที่ของระเบียน หรือที่เรียกว่าหมายเลขระเบียนภายในแฟ้มข้อมูลแบบสุ่ม ระเบียนต่าง ๆ อาจจะไม่ถูกบันทึกติดต่อกันไป ซึ่งในกรณีนี้ จะมีการสำรองเนื้อที่หน่วยความจำสำรอง สำหรับระเบียนที่มีหมายเลขระเบียนที่ยังไม่ได้ถูกบันทึก

คำสั่งต่าง ๆ ที่เกี่ยวข้องกับการใช้แฟ้มข้อมูลแบบสุ่ม มีดังนี้คือ

๕.๑.๒.๑ OPEN สำหรับเปิดแฟ้มข้อมูลแบบสุ่ม และมีรูปแบบดังนี้

OPEN <โมด>, [#]<หมายเลขแฟ้มข้อมูล>, <ชื่อแฟ้มข้อมูล>,
<ความยาวระเบียน>

โมด เท่ากับ "R" เพื่อระบุว่าต้องการเปิดแฟ้มข้อมูลแบบสุ่ม
หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขระหว่าง ๑ ถึง จำนวนแฟ้มข้อมูลที่ต้องการเปิดพร้อมกัน ซึ่งระบุตอนเริ่มต้นอินเตอร์เพรตเตอร์
ชื่อแฟ้มข้อมูล เท่ากับ ชื่อแฟ้มข้อมูลที่ต้องการใช้งาน
ความยาวระเบียน เท่ากับ ความยาวของระเบียน ซึ่งต้องไม่เกินความยาวสูงสุดของระเบียน ที่ระบุตอนเริ่มต้นอินเตอร์เพรตเตอร์

๔.๑.๒.๒ FIELD สำหรับกำหนดเขตข้อมูลของระเบียน และมีรูปแบบดังนี้

FIELD [#]<หมายเลขแฟ้มข้อมูล>,
<ความยาวของเขตข้อมูล> AS <ชื่อตัวแปรสตริง> ...

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขของหมายเลขแฟ้มข้อมูล ตอนเปิดแฟ้มข้อมูล
ความยาวของเขตข้อมูล เท่ากับ ความยาวของเขตข้อมูล ซึ่งต้องไม่เกินความยาวของระเบียน
ชื่อตัวแปรสตริง เท่ากับ ชื่อของตัวแปรที่เก็บค่าของเขตข้อมูล

๔.๑.๒.๓ LSET/RSET สำหรับย้ายค่าของตัวแปรไปยังเนื้อที่แฟ้มข้อมูลทำงาน ก่อนที่จะทำการบันทึกข้อมูลโดยที่คำสั่ง LSET จะย้ายแบบชิดซ้าย ขณะที่คำสั่ง RSET จะย้ายแบบชิดขวา คำสั่งทั้งสอง มีรูปแบบดังนี้

LSET/RSET <ชื่อตัวแปรสตริง> = <นิพจน์สตริง(String Expression)>

ชื่อตัวแปรสตริง เท่ากับ ชื่อของตัวแปรสตริงที่กำหนดตอนใช้คำสั่ง FIELD

นิพจน์สตริง เท่ากับ ค่าสตริงที่จะย้ายไปยังเนื้อที่แฟ้มข้อมูลทำงาน ณ. ตำแหน่งของเขตข้อมูลนั้น ๆ

๔.๑.๒.๔ PUT สำหรับบันทึกระเบียนเข้าไปในแฟ้มข้อมูล มีรูปแบบดังนี้

PUT [#]<หมายเลขแฟ้มข้อมูล> [,<หมายเลขระเบียน>]

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขของหมายเลขแฟ้มข้อมูล ตอนเปิดแฟ้มข้อมูล

หมายเลขระเบียน เท่ากับ ตัวเลขระหว่าง ๑ ถึง ๓๒,๗๖๗

๔.๑.๒.๕ GET สำหรับอ่านระเบียนของแฟ้มข้อมูล และมีรูปแบบดังนี้

GET [#]<หมายเลขแฟ้มข้อมูล> [,<หมายเลขระเบียน>]

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขของหมายเลขแฟ้มข้อมูล ตอนเปิดแฟ้มข้อมูล

หมายเลขระเบียน เท่ากับ ตัวเลขระหว่าง ๑ ถึง ๓๒,๗๖๗

๔.๑.๒.๖ MKI$/MKS$/MKD$ สำหรับแปลงค่าตัวเลขเป็นค่าสตริงของตัวแปรจำนวนเต็ม ตัวแปรทศนิยม ๔ ตำแหน่ง และตัวแปรทศนิยม ๑๖ ตำแหน่ง ตามลำดับคำสั่งทั้ง ๓ มีรูปแบบดังนี้

MKI$ (<นิพจน์จำนวนเต็ม>)

MKS$ (<นิพจน์ทศนิยม ๔ ตำแหน่ง>)

MKD$ (<นิพจน์ทศนิยม ๑๖ ตำแหน่ง>)

๔.๑.๒.๗ CVI/CVS/CVD สำหรับแปลงค่าสตริงที่ได้จากการแปลงของคำสั่ง MKI$/MKS$/MKD$ กลับมาเป็นค่าตัวเลขจำนวนเต็ม ค่าตัวเลขทศนิยม ๔ ตำแหน่ง และค่าตัวเลขทศนิยม ๑๖ ตำแหน่งตามลำดับ คำสั่งทั้ง ๓ มีรูปแบบดังนี้

CVI(ตัวแปรสตริง ๒ ไบต์)

CVS(ตัวแปรสตริง ๔ ไบต์)

CVD(ตัวแปรสตริง ๘ ไบต์)

๔.๑.๒.๘ CLOSE สำหรับปิดแฟ้มข้อมูล และมีรูปแบบดังนี้

CLOSE [#]<หมายเลขแฟ้มข้อมูล>

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขของหมายเลขแฟ้มข้อมูล ตอนเปิดแฟ้มข้อมูล

การใช้แฟ้มข้อมูลแบบสุ่ม ทำให้การอ่านหรือบันทึกข้อมูลทำได้อย่างรวดเร็ว เนื่องจากใช้หมายเลขระเบียนมาคำนวณหาตำแหน่งของระเบียน การอ่านหรือบันทึกข้อมูลจึงกระทำได้โดยตรง แต่ข้อจำกัดของแฟ้มข้อมูลแบบสุ่มก็คือ เนื้อที่หน่วยความจำสำรองอาจจะไม่ถูกใช้อย่างเต็มที่ และการค้นหาระเบียนที่ต้องการถ้าเขตข้อมูลที่ต้องการค้นหาไม่มีความสัมพันธ์กับหมายเลขระเบียนเลย ผู้ใช้ต้องจดจำเองว่าระเบียนที่ต้องการมีหมายเลขระเบียนอะไร ซึ่งทำให้ยุ่งยากในการค้นหาระเบียน

๔.๒ แฟ้มข้อมูลแบบดัชนี

การจัดเก็บแฟ้มข้อมูลแบบดัชนี คือการจัดเก็บข้อมูลที่รวมเอาข้อดีของการจัดเก็บแฟ้มข้อมูลแบบเรียงลำดับและแบบสุ่มไว้ด้วยกัน โดยระเบียนต่าง ๆ ของแฟ้มข้อมูลจะถูกบันทึกติดต่อกันไป ทำให้เนื้อที่หน่วยความจำสำรองถูกใช้อย่างเต็มที่ แต่ขณะเดียวกันการค้นหาระเบียนก็สามารถหาได้รวดเร็วด้วย เนื่องจากการจัดเก็บแฟ้มข้อมูลแบบดัชนี ผู้ใช้ต้องกำหนดเขตข้อมูลหนึ่งของระเบียนเป็นคีย์ของระเบียน ซึ่งเขตข้อมูลที่เป็นคีย์นี้จะถูกใช้ในการอ้างอิงระเบียนเมื่อต้องการอ่านหรือบันทึกข้อมูล ลักษณะการจัดเก็บข้อมูลภายในแฟ้มข้อมูลแบบดัชนีคล้ายกับแฟ้มข้อมูลแบบเรียงลำดับ และเพื่อให้การอ่านหรือบันทึกข้อมูลกระทำได้อย่างรวดเร็ว จะมีการสร้างแฟ้มข้อมูลอีกแฟ้มหนึ่งเพื่อเก็บดัชนีคือเก็บข้อมูลที่ถูกกำหนดให้เป็นคีย์คู่กับตัวชี้ไปยังตำแหน่งของระเบียนที่มีคีย์ค่านั้น ซึ่งแฟ้มข้อมูลนี้จะถูกใช้ในการค้นหาข้อมูล เมื่อผู้ใช้ระบุคีย์ของระเบียน ผู้ใช้ไม่ต้องสร้างแฟ้มข้อมูลที่เก็บดัชนีเอง แต่โปรแกรมระบบซึ่งทำหน้าที่เกี่ยวกับการจัดเก็บข้อมูลแบบดัชนีเป็นตัวสร้างให้ โดยที่โครงสร้างข้อมูล (Data Structure) ของแฟ้มข้อมูลที่เก็บดัชนีจะต้องเป็นโครงสร้างที่ทำให้ค้นหาคีย์ได้รวดเร็ว และเสียเวลาในการปรับคีย์ไม่มาก เมื่อมีการเพิ่มหรือลดคีย์

๔.๓ โครงสร้างข้อมูลแบบทรี (Trees Data Structure)

โครงสร้างข้อมูลแบบหนึ่งที่นำมาใช้จัดเก็บดัชนีของแฟ้มข้อมูลแบบดัชนี คือโครงสร้างข้อมูลแบบทรี เนื่องจากเป็นโครงสร้างข้อมูลที่มีลักษณะที่ทำให้ค้นหาคีย์ได้รวดเร็ว และการปรับคีย์เมื่อมีการเพิ่มหรือลดคีย์ใช้เวลาไม่มาก

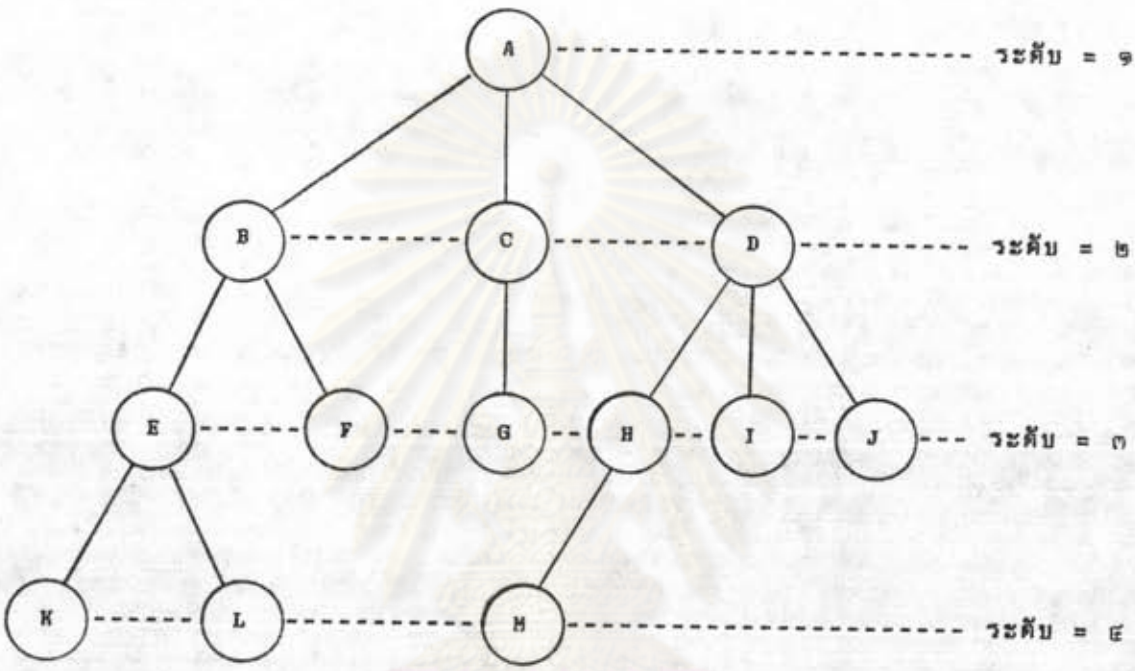
๔.๓.๑ คำจำกัดความของโครงสร้างข้อมูลแบบทรี

โครงสร้างข้อมูลแบบทรี คือเซตของสมาชิกซึ่งถูกเรียกว่าโนด (Node) จำนวนจำกัดและมีคุณสมบัติดังนี้

๔.๓.๑.๑ มีโนดพิเศษอยู่หนึ่งโนด เรียกว่า รูทโนด (Root Node)

๔.๓.๑.๒ โนดอื่น ๆ นอกเหนือไปจากรูทโนด จะแบ่งออกเป็น m

ดิสจอยท์เซ็ท (Disjoint Set) T1, T2,...Tm ซึ่งแต่ละ Ti, i = 1,2,...m ก็เป็นโครงสร้างข้อมูลแบบทรี และเรียก T1, T2,...Tm ว่าเป็นสับทรี (Subtree) ของรูทโนด



รูปที่ ๔.๑ ตัวอย่างของโครงสร้างข้อมูลแบบทรี

ทรีในรูปที่ ๔.๑ มีโนด ๑๓ โนด โดยมี A เป็นรูทโนด สับทรีของ A คือ {B,E,F,K,L} {C,G} {D,H,I,J,M} ซึ่งก็เป็นทรีด้วย

ดีกรี (Degree) ดีกรีของโนดใด ๆ คือ จำนวนสับทรีของโนดนั้น

ดังนั้นจากรูปที่ ๔.๑

A มีดีกรี = ๓

B มีดีกรี = ๒

C มีดีกรี = ๑

D มีดีกรี = ๓

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ลีฟ (Leaf) คือโนดซึ่งมีดีกรี = ๐

ดังนั้นจากรูปที่ ๔.๑ โนดที่เป็นลีฟคือ H,L,F,G,M,I,J

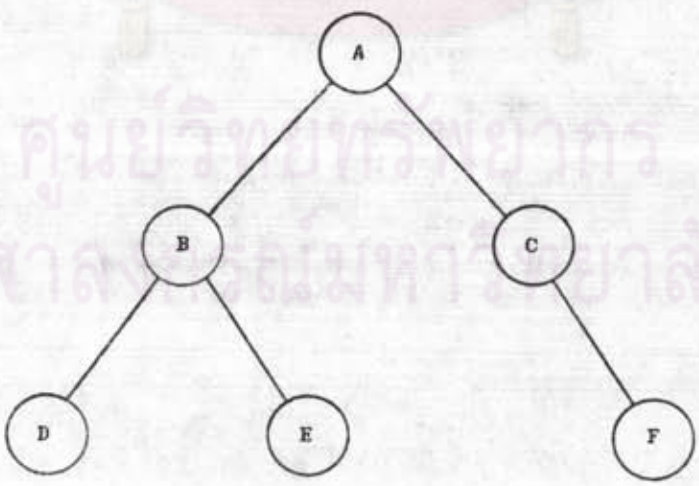
ลูก (Children) ลูกของโนดใด ๆ คือ โนดที่เป็นรูทโนดของสับทรีของโนดนั้น

ดังนั้นจากรูปที่ ๔.๑ B,C,D เป็นลูกของโนด A เพราะว่า B,C,D เป็นรูทโนดของสับทรีของ A

ระดับ (Level) ระดับของทรี กำหนดโดยให้รูทโนดมีระดับ = ๑ เมื่อเริ่มแรก และถ้าโนดใด ๆ ก็ตามซึ่งมีระดับ = L แล้ว ลูกของโนดนั้นจะมีระดับ = L+1

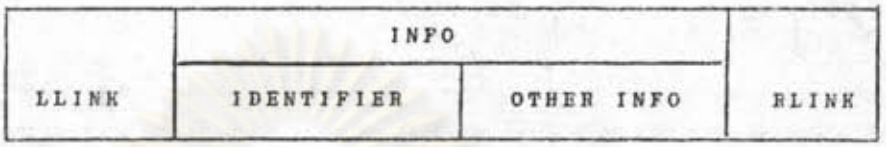
ความสูงหรือความลึก (Height หรือ Depth) ความสูงของทรี คือ ระดับสูงสุดของทรีนั้น

๔.๓.๒ ไบนารีทรี (Binary Tree) คือทรีซึ่งแต่ละโนดจะมีดีกรีสูงสุดไม่เกิน ๒ ดีกรี ดังนั้นแต่ละโนดของไบนารีทรีจะมีสับทรีไม่เกิน ๒ สับทรี คือ สับทรีทางซ้าย (Left Subtree) และสับทรีทางขวา (Right Subtree)



รูปที่ ๔.๒ ตัวอย่างของไบนารีทรี

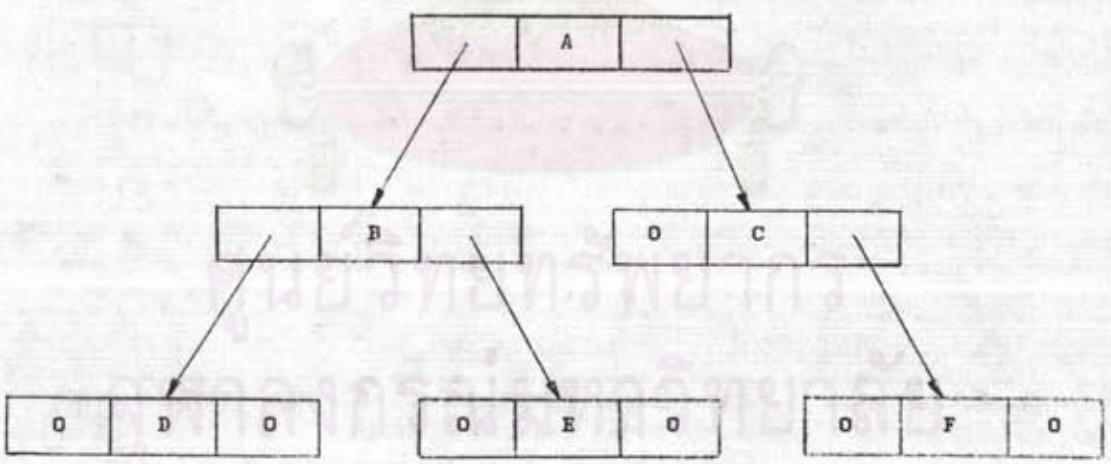
๔.๓.๓ ความสัมพันธ์ระหว่างลิงค์ลิสท์ (Linked List) และทรี โนดแต่ละโนดของทรี สามารถนำโครงสร้างข้อมูลแบบลิงค์ลิสท์คู่ (Double Linked List) มาใช้ดังนี้



โดยที่ INFO คือ ค่าของข้อมูล ซึ่งอาจจะประกอบด้วยหลายเขตข้อมูล แต่จะมีเขตข้อมูลหนึ่งเป็นตัวแทนโนด (Identifier)

LLINK คือ ตัวชี้ไปยังสับทรีทางซ้าย

RLINK คือ ตัวชี้ไปยังสับทรีทางขวา



รูปที่ ๔.๓ การนำโครงสร้างแบบลิงค์ลิสท์คู่มาแทนโนดของทรีในรูป ๔.๒

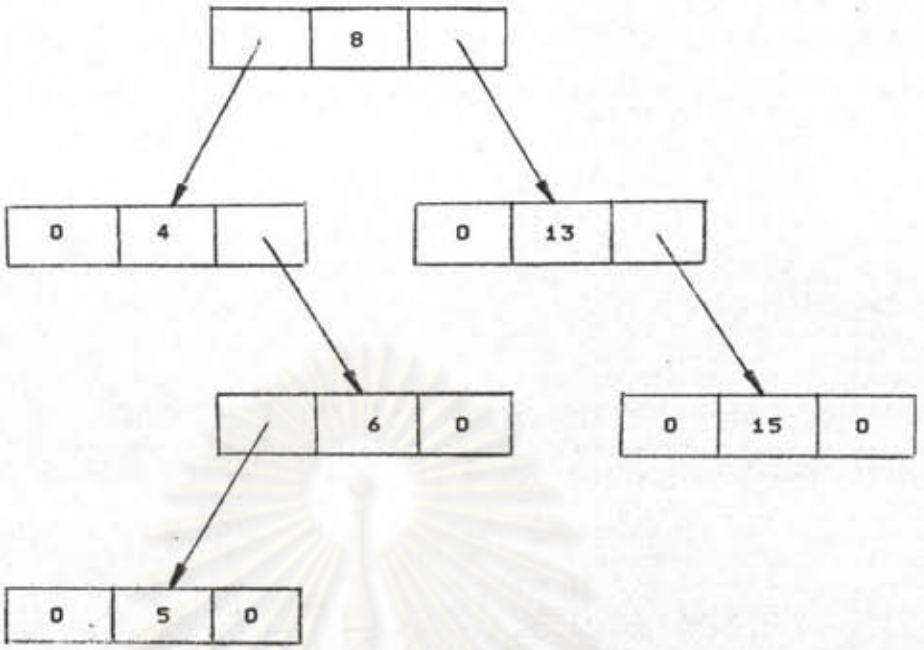
๔.๓.๔ ไบนารีเสิชทรี (Binary Search Tree) คือไบนารีทรีซึ่งมีคุณสมบัติดังนี้

๔.๓.๔.๑ ค่าของตัวแทนโนดทุกตัวในสับทรีทางซ้าย มีค่าน้อยกว่าค่าของตัวแทนโนดของรูทโนด

๔.๓.๔.๒ ค่าของตัวแทนโนดทุกตัวในสับทรีทางขวา มีค่ามากกว่าค่าของตัวแทนโนดของรูทโนด

๔.๓.๔.๓ สับทรีทางซ้ายและสับทรีทางขวาของโนดใด ๆ ต้องเป็นไบนารีเสิชทรีด้วย

โครงสร้างข้อมูลแบบไบนารีเสิชทรี เป็นโครงสร้างข้อมูลที่สามารถนำมาใช้จัดเก็บดัชนีของแฟ้มข้อมูลแบบดัชนี โดยกำหนดให้ส่วน INFO ของแต่ละโนดเก็บข้อมูล ๒ เขตข้อมูล เขตข้อมูลหนึ่งที่เป็นตัวแทนโนดให้เก็บค่าของคีย์ อีกเขตข้อมูลหนึ่งให้เก็บตำแหน่งของระเบียนที่มีคีย์ค่านั้น ดังนั้นการค้นหาคีย์สามารถทำได้โดยใช้วิธีเปรียบเทียบคีย์ที่ต้องการค้นหากับคีย์ในไบนารีเสิชทรีตั้งแต่รูทโนด ถ้าเปรียบเทียบแล้วน้อยกว่า ก็เปรียบเทียบต่อไปเฉพาะกับคีย์ที่อยู่ในสับทรีทางซ้ายของรูทโนด แต่ถ้าเปรียบเทียบแล้วมากกว่า ก็เปรียบเทียบต่อไปเฉพาะกับคีย์ที่อยู่ในสับทรีทางขวาของรูทโนด ทำเช่นนี้ไปเรื่อย ๆ จนพบคีย์ที่ต้องการค้นหาก็จะได้ตำแหน่งของระเบียนที่มีคีย์ค่านั้น ซึ่งจะถูกนำไปใช้อ่านระเบียนนั้นอีกทีหนึ่ง จะเห็นว่าวิธีการค้นหาเช่นนี้ ถ้าจำนวนโนดของสับทรีทางซ้ายมีค่าใกล้เคียงกับจำนวนโนดของสับทรีทางขวา ทุกครั้งที่ทำการเปรียบเทียบถ้ายังไม่เท่ากันก็จะเหลือจำนวนโนดที่ยังไม่เปรียบเทียบประมาณครึ่งหนึ่งของที่มีอยู่ ทำให้สามารถค้นหาคีย์ที่ต้องการได้ภายในเวลาที่รวดเร็ว นอกจากนี้เมื่อมีการเพิ่มหรือลดคีย์ โครงสร้างข้อมูลแบบไบนารีเสิชทรีก็ไม่ทำให้เสียเวลาเท่าใดสำหรับการปรับคีย์ เพราะเพียงแต่ใช้วิธีปรับตัวชี้ใหม่ให้เหมาะสมเท่านั้น



รูปที่ ๔.๔ ตัวอย่างของไบนารีเสิชทรี

๔.๔ แนวความคิดของโครงสร้างแฟ้มข้อมูลแบบดัชนี

การพัฒนาโปรแกรมระบบขึ้นมาเพื่อจัดเก็บแฟ้มข้อมูลแบบดัชนี ถ้าออกแบบลักษณะการจัดเก็บใหม่หมด โปรแกรมระบบที่เขียนขึ้นจะต้องเป็นตัวจัดการข้อมูลเองทั้งหมด ดังนั้นโปรแกรมระบบจะมีขนาดใหญ่ ซึ่งจะมีผลให้ตัวอินเตอร์เพรตเตอร์มีขนาดใหญ่ขึ้นตามไปด้วย ทำให้เหลือเนื้อที่ว่างสำหรับใช้งานน้อยลง วิธีการหนึ่งที่จะลดขนาดของโปรแกรมระบบได้คือนำลักษณะการจัดเก็บข้อมูลที่มีอยู่แล้วมาใช้ ซึ่งจะทำให้โปรแกรมระบบสามารถเรียกใช้การทำงานของคำสั่งต่าง ๆ ที่มีอยู่

ถ้าพิจารณาการจัดเก็บแฟ้มข้อมูลที่มีอยู่ จะเห็นว่าการจัดเก็บแฟ้มข้อมูลแบบเรียงลำดับไม่เหมาะสมที่จะนำมาใช้จัดเก็บข้อมูลของแฟ้มข้อมูลแบบดัชนี เนื่องจากไม่มีลักษณะที่เข้าถึงข้อมูลได้โดยตรง ทำให้ไม่มีทางที่จะค้นหาข้อมูลได้อย่างรวดเร็ว ส่วนการจัดเก็บแฟ้มข้อมูลแบบสุ่ม มีลักษณะที่เหมาะสำหรับนำมาใช้จัดเก็บข้อมูลของแฟ้มข้อมูลแบบดัชนี

เนื่องจากสามารถเข้าถึงข้อมูลได้โดยตรง แต่เพื่อให้ระเบียนต่าง ๆ ถูกบันทึกติดต่อกันไป และการค้นหาระเบียนสามารถใช้คีย์แทนหมายเลขระเบียนเป็นตัวอ้างอิง โปรแกรมระบบต้องเป็นตัวจัดการเรื่องเหล่านี้แทนผู้ใช้ ซึ่งทำได้โดยเก็บหมายเลขระเบียนต่อไปที่ใช้งานได้เพื่อนำมาใช้บันทึกระเบียนต่อไป และสร้างแฟ้มข้อมูลดัชนีขึ้นสำหรับเก็บคีย์และหมายเลขระเบียนที่มีคีย์ค่านั้น เพื่อให้สามารถค้นหาระเบียนได้โดยอ้างอิงคีย์แทนหมายเลขระเบียน โดยให้แฟ้มข้อมูลดัชนีก็เป็นแฟ้มข้อมูลแบบสุ่ม และมีโครงสร้างข้อมูลแบบไบนารีเสิชทรี

จากแนวความคิดที่กล่าวข้างต้น แฟ้มข้อมูลแบบดัชนีจะประกอบด้วยแฟ้มข้อมูลแบบสุ่ม ๒ แฟ้ม ดังนี้

๕.๔.๑ แฟ้มข้อมูลใช้งาน คือแฟ้มข้อมูลแบบสุ่มที่เก็บข้อมูลของระเบียน ซึ่งจะมีลักษณะเช่นเดียวกับแฟ้มข้อมูลแบบสุ่มทั่ว ๆ ไป ดังนั้นแฟ้มข้อมูลใช้งานของแฟ้มข้อมูลแบบดัชนีสามารถนำไปใช้เป็นแฟ้มข้อมูลแบบสุ่มได้ด้วย

๕.๔.๒ แฟ้มข้อมูลดัชนี คือแฟ้มข้อมูลแบบสุ่มที่เก็บดัชนีของแฟ้มข้อมูลแบบดัชนี นอกจากเก็บดัชนีแล้ว แฟ้มข้อมูลดัชนียังเก็บรายละเอียดบางอย่างของแฟ้มข้อมูลไว้ด้วย ได้แก่ ความยาวของระเบียน ตำแหน่งเริ่มต้นของคีย์ภายในระเบียน ความยาวของคีย์ หมายเลขระเบียนต่อไปที่ใช้งานได้ในแฟ้มข้อมูลใช้งาน หมายเลขระเบียนต่อไปที่ใช้งานได้ในแฟ้มข้อมูลดัชนี จำนวนระเบียนที่กำลังใช้งาน ฯลฯ รายละเอียดต่าง ๆ นี้จะใช้สำหรับการบันทึกและอ่านระเบียน และถูกปรับค่าเองโดยโปรแกรมระบบ แฟ้มข้อมูลดัชนีเป็นแฟ้มข้อมูลแบบสุ่มที่มีความยาว ๑๒๘ ไบต์เสมอ และมีโครงสร้างข้อมูลแบบไบนารีเสิชทรี

เนื่องจากแฟ้มข้อมูลใช้งานและแฟ้มข้อมูลดัชนี ต่างก็เป็นแฟ้มข้อมูลแบบสุ่ม ดังนั้นการเปิดแฟ้มข้อมูล การบันทึกระเบียน การอ่านระเบียน และการปิดแฟ้มข้อมูลของแฟ้มข้อมูลทั้งสอง โปรแกรมระบบสามารถเรียกใช้การทำงานของคำสั่งที่มีอยู่ได้แก่ คำสั่ง OPEN, PUT, GET และ CLOSE ตามลำดับ มาประกอบกับการทำงานของคำสั่งใหม่โดยไม่จำเป็นต้องเขียนการทำงานของคำสั่งใหม่หมด

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

๔.๔ รูปแบบและการทำงานของคำสั่งการจัดเก็บแฟ้มข้อมูลแบบดัชนี

คำสั่งการจัดเก็บแฟ้มข้อมูลแบบดัชนี เป็นชุดของคำสั่ง ๗ คำสั่ง แต่ละคำสั่งมีรูปแบบและการทำงานของของคำสั่งดังนี้

๔.๔.๑ CREATE สำหรับสร้างโครงสร้างของแฟ้มข้อมูลแบบดัชนี มีรูปแบบดังนี้

CREATE [#]<หมายเลขแฟ้มข้อมูล>,<ชื่อแฟ้มข้อมูล>,<ความยาวระเบียน>,
<ตำแหน่งเริ่มต้นของคีย์>,<ความยาวของคีย์>

หมายเลขแฟ้มข้อมูล	เท่ากับ	ตัวเลขระหว่าง ๑ ถึงจำนวนแฟ้มข้อมูลที่ต้องการเปิดพร้อมกัน ซึ่งระบุตอนเริ่มต้นอินเตอร์เพรตเตอร์
ชื่อแฟ้มข้อมูล	เท่ากับ	ชื่อของแฟ้มข้อมูลใช้งาน
ความยาวระเบียน	เท่ากับ	ความยาวของระเบียน ต้องไม่เกิน ๓๒,๗๖๗ ไบต์
ตำแหน่งเริ่มต้นของคีย์	เท่ากับ	ตำแหน่งเริ่มต้นของคีย์ในระเบียน ต้องไม่เกินความยาวของระเบียน
ความยาวของคีย์	เท่ากับ	ความยาวของคีย์ นับจากตำแหน่งเริ่มต้นของคีย์ต้องไม่เกินความยาวของระเบียน

ตัวอย่าง CREATE #1,"STOCK.DAT",256,1,10

การทำงานของคำสั่ง CREATE จะสร้างแฟ้มข้อมูลแบบสุ่มขึ้น ๒ แฟ้มข้อมูล แฟ้มข้อมูลแรกคือแฟ้มข้อมูลใช้งานมีชื่อเดียวกับชื่อแฟ้มข้อมูลที่ระบุในคำสั่ง จากตัวอย่างของคำสั่ง CREATE จะได้แฟ้มข้อมูลใช้งานชื่อ "STOCK.DAT" แฟ้มข้อมูลที่สองคือแฟ้มข้อมูล

ดัชนีมีชื่อเดียวกับชื่อแฟ้มข้อมูลที่ระบุในคำสั่ง แต่มีประเภทของแฟ้มเป็น "NDX" จากตัวอย่างเดียวกัน จะได้แฟ้มข้อมูลดัชนีชื่อ "STOCK.NDX" แฟ้มข้อมูลใช้งานจะมีความยาวระเบียนเท่ากับความยาวระเบียนที่ระบุในคำสั่งและเป็นแฟ้มข้อมูลว่าง ๆ ส่วนแฟ้มข้อมูลดัชนีจะมีความยาวระเบียนเท่ากับ ๑๒๘ ไบต์เสมอ และมีการบันทึกระเบียนเข้าไป ๑ ระเบียน เพื่อเก็บรายละเอียดบางอย่างของแฟ้มข้อมูลแบบดัชนี

รูปแบบของระเบียนในแฟ้มข้อมูลดัชนี เป็นดังนี้

ระเบียนที่ ๑ เป็นระเบียนที่เก็บลักษณะของแฟ้มข้อมูลดัชนี และรายละเอียดการใช้งานบางอย่างดังนี้

ไบต์ที่ ๑ ถึง ๑๑ เก็บชื่อของแฟ้มข้อมูลใช้งาน

ไบต์ที่ ๑๒ ถึง ๑๓ เก็บความยาวของระเบียน

ไบต์ที่ ๑๔ ถึง ๑๕ เก็บตำแหน่งเริ่มต้นของคีย์

ไบต์ที่ ๑๖ ถึง ๑๗ เก็บความยาวของคีย์ลบออกหนึ่ง

ไบต์ที่ ๑๘ ถึง ๑๙ เก็บหมายเลขระเบียนต่อไปที่ใช้งานได้ของแฟ้มข้อมูลใช้งาน โดยกำหนดให้เท่ากับ ๑ ตอนเริ่มต้น

ไบต์ที่ ๒๐ ถึง ๒๑ เก็บหมายเลขระเบียนต่อไปที่ใช้งานได้ของแฟ้มข้อมูลดัชนี โดยกำหนดให้เท่ากับ ๒ ตอนเริ่มต้น

ไบต์ที่ ๒๒ ถึง ๒๓ เก็บตำแหน่งไบต์ต่อไปที่ใช้งานได้ของระเบียนต่อไปที่ใช้งานได้ของแฟ้มข้อมูลดัชนี โดยกำหนดให้เท่ากับ ๑ ตอนเริ่มต้น

ไบต์ที่ ๒๔ ถึง ๒๕ เก็บหมายเลขระเบียนของโนดที่เป็นรูทโนดของทรี โดยกำหนดให้เท่ากับ ๒ ตอนเริ่มต้น

ไบต์ที่ ๒๖ ถึง ๒๗ เก็บตำแหน่งไบต์ของโนดที่เป็นรูทโนดของทรี โดยกำหนดให้เท่ากับ ๑ ตอนเริ่มต้น

ไบต์ที่ ๒๘ ถึง ๒๙ เก็บจำนวนระเบียนที่กำลังใช้งาน โดยกำหนดให้เท่ากับ ๐ ตอนเริ่มต้น

ไบต์ที่ ๓๐ ถึง ๑๒๘ สำรองไว้

ระเบียนที่ ๒ เป็นต้นไป เป็นระเบียนที่เก็บดัชนีของแฟ้มข้อมูลดัชนี ดัชนีแต่ละตัวมีลักษณะดังนี้

ไบต์ที่ ๑ ถึง 1 เก็บคีย์ของระเบียน โดยที่ 1 เท่ากับความยาวของคีย์

ไบต์ที่ 1+๑ ถึง 1+๒ เก็บหมายเลขระเบียนของแฟ้มข้อมูลใช้งานที่มีคีย์ค่านั้น

ไบต์ที่ 1+๓ ถึง 1+๔ เก็บตัวชี้ไปยังสับทรีทางซ้ายของโนดปัจจุบัน โดยที่

ไบต์ที่ 1+๓ ถึง 1+๔ เก็บหมายเลขระเบียนในแฟ้มข้อมูลดัชนีของรูทโนดของสับทรีทางซ้าย

ไบต์ที่ 1+๔ ถึง 1+๕ เก็บตำแหน่งไบต์ของหมายเลขระเบียนในแฟ้มข้อมูลดัชนีของรูทโนดของสับทรีทางซ้าย

ไบต์ที่ 1+๖ ถึง 1+๘ เก็บตัวชี้ไปยังสับทรีทางขวาของโนดปัจจุบัน โดยที่

ไบต์ที่ 1+๖ ถึง 1+๗ เก็บหมายเลขระเบียนในแฟ้มข้อมูลดัชนีของรูทโนดของสับทรีทางขวา

ไบต์ที่ 1+๘ ถึง 1+๙ เก็บตำแหน่งไบต์ของหมายเลขระเบียนในแฟ้มข้อมูลดัชนีของรูทโนดของสับทรีทางขวา

การสร้างโครงสร้างแฟ้มข้อมูลแบบดัชนี โปรแกรมระบบจะเรียกใช้การทำงานของคำสั่ง OPEN เพื่อเปิดแฟ้มข้อมูลแบบสุ่ม โดยเปิดแฟ้มข้อมูลใช้งานให้มีความยาวระเบียนเท่ากับความยาวระเบียนที่ระบุในคำสั่ง และเปิดแฟ้มข้อมูลดัชนีให้มีความยาว ๑๒๙ ไบต์ และเรียกใช้การทำงานของคำสั่ง PUT เพื่อบันทึกระเบียนแรกเข้าไปในแฟ้มข้อมูลดัชนี โดยเก็บรายละเอียดที่ได้จากการวิเคราะห์คำสั่ง CREATE และค่าเริ่มต้นต่าง ๆ ที่กล่าวข้างต้น หลังจากนั้นจะเรียกใช้คำสั่ง CLOSE เพื่อปิดแฟ้มข้อมูลทั้ง ๒

๔.๔.๒ OPEN "X" สำหรับเปิดแฟ้มข้อมูลแบบดัชนี และมีรูปแบบดังนี้

OPEN <โมด>,[#]<หมายเลขแฟ้มข้อมูล>,<ชื่อแฟ้มข้อมูล>,
<ความยาวระเบียน>

โมด เท่ากับ "X" เพื่อระบุว่าต้องการเปิดแฟ้มข้อมูลแบบดัชนี

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขระหว่าง ๑ ถึง จำนวนแฟ้มข้อมูลที่ต้องการเปิดพร้อมกัน ซึ่งระบุตอนเริ่มต้นอินเตอร์เพรตเตอร์

ชื่อแฟ้มข้อมูล เท่ากับ ชื่อแฟ้มข้อมูลแบบดัชนี ซึ่งต้องสร้างโครงสร้างแฟ้มข้อมูลแบบดัชนีด้วยคำสั่ง CREATE มาแล้ว

ความยาวระเบียน เท่ากับ ความยาวระเบียนที่ระบุตอนสร้างโครงสร้างแฟ้มข้อมูลแบบดัชนี

ตัวอย่าง OPEN "X", #1, "STOCK.DAT", 256

การทำงานของคำสั่ง OPEN "X" จะเปิดแฟ้มข้อมูลขึ้น ๒ แฟ้ม คือ แฟ้มข้อมูลใช้งานและแฟ้มข้อมูลดัชนี จากตัวอย่างของคำสั่ง OPEN "X" จะเปิดแฟ้มข้อมูลใช้งานชื่อ "STOCK.DAT" และเปิดแฟ้มข้อมูลดัชนีชื่อ "STOCK.NDX" สำหรับแฟ้มข้อมูลดัชนี หลังจากเปิดแฟ้มข้อมูลเรียบร้อย จะมีการอ่านระเบียนแรกเข้ามา และเก็บรายละเอียดต่าง ๆ ของแฟ้มข้อมูลไว้ในหน่วยความจำ เพื่อใช้งานในขั้นตอนต่อไป

การเปิดแฟ้มข้อมูลแบบดัชนี โปรแกรมระบบจะเรียกใช้การทำงานของคำสั่ง OPEN เพื่อเปิดแฟ้มข้อมูลแบบสุ่ม โดยเปิดแฟ้มข้อมูลใช้งานด้วยหมายเลขแฟ้มข้อมูลที่ระบุในคำสั่ง และเปิดแฟ้มข้อมูลดัชนีด้วยหมายเลขแฟ้มข้อมูลที่ระบุในคำสั่งบวกด้วย ๑ ดังนั้นจาก ตัวอย่างของคำสั่ง OPEN "X" ผู้ใช้จะเปิดแฟ้มข้อมูลอื่นด้วยหมายเลขแฟ้มข้อมูลเท่ากับ ๒ ไม่ได้ นอกจากจะปิดแฟ้มข้อมูลดัชนีที่เปิดด้วยหมายเลขแฟ้มข้อมูลเท่ากับ ๑

เสียก่อน หลังจากเปิดแฟ้มข้อมูลใช้งานและแฟ้มข้อมูลดัชนีแล้ว โปรแกรมระบบจะเรียกใช้การทำงานของคำสั่ง GET เพื่ออ่านระเบียนแรกของแฟ้มข้อมูลดัชนีเข้ามาสำหรับใช้งานต่อไป

๕.๕.๓ INSERT #n สำหรับเพิ่มระเบียนใหม่เข้าไปในแฟ้มข้อมูลแบบดัชนี มีรูปแบบดังนี้

INSERT #<หมายเลขแฟ้มข้อมูล>

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขของหมายเลขแฟ้มข้อมูลที่ระบุตอนเปิดแฟ้มข้อมูลแบบดัชนี

ตัวอย่าง INSERT #1

เนื่องจากแฟ้มข้อมูลในระบบการจัดเก็บแฟ้มข้อมูลแบบดัชนี เป็นแฟ้มข้อมูลแบบสุ่ม การกำหนดเขตข้อมูลของระเบียนจะยังคงใช้คำสั่ง FIELD เหมือนการใช้แฟ้มข้อมูลแบบสุ่มโดยไม่มีการแก้ไขการทำงานของคำสั่ง FIELD แต่อย่างใด และสำหรับแฟ้มข้อมูลแบบสุ่ม ก่อนที่จะใช้คำสั่ง PUT จะต้องย้ายค่าของตัวแปรไปยังเนื้อที่แฟ้มข้อมูลทำงานเสียก่อน การใช้คำสั่ง INSERT ก็เช่นเดียวกัน ผู้ใช้ต้องใช้คำสั่ง LSET หรือ RSET เพื่อย้ายค่าของตัวแปรไปยังเนื้อที่แฟ้มข้อมูลทำงาน และโปรแกรมระบบจะใช้ค่าที่อยู่ในเนื้อที่แฟ้มข้อมูลทำงานสำหรับการทำงานของคำสั่ง INSERT

การทำงานของคำสั่ง INSERT จะมีการบันทึกระเบียนใหม่เข้าไปในแฟ้มข้อมูลใช้งาน ระเบียนที่ถูกบันทึกจะมีหมายเลขระเบียนเท่ากับหมายเลขระเบียนต่อไปที่ใช้งานได้ของแฟ้มข้อมูลใช้งานที่เก็บอยู่ในระเบียนแรกของแฟ้มข้อมูลดัชนี และหมายเลขระเบียนต่อไปที่ใช้งานได้จะถูกบวกเพิ่มขึ้น ๑ นอกจากนี้จะมีการเพิ่มดัชนีใหม่เข้าไปในแฟ้มข้อมูลดัชนี ดัชนีใหม่ที่ถูกบันทึกจะอยู่ในระเบียนที่มีหมายเลขระเบียนเท่ากับหมายเลขระเบียนต่อไปที่ใช้งานได้ของแฟ้มข้อมูลดัชนี และมีตำแหน่งเริ่มต้นเท่ากับตำแหน่งไบต์ต่อไป

ที่ใช้งานได้ในระเบียนนั้น ซึ่งหมายเลขระเบียนต่อไปที่ใช้งานได้ของแฟ้มข้อมูลดัชนีและตำแหน่งไบต์ต่อไปที่ใช้งานได้ก็จะถูกปรับค่าใหม่ เช่นเดียวกับหมายเลขระเบียนต่อไปที่ใช้งานได้ของแฟ้มข้อมูลใช้งาน

การบันทึกระเบียนใหม่เข้าไปในแฟ้มข้อมูลใช้งาน โปรแกรมระบบจะเรียกใช้การทำงานของคำสั่ง PUT โดยใช้หมายเลขระเบียนต่อไปที่ใช้งานได้ของแฟ้มข้อมูลใช้งานสำหรับการบันทึก ส่วนการเพิ่มดัชนีใหม่เข้าไปในแฟ้มข้อมูลดัชนีจะเรียกใช้การทำงานของคำสั่ง GET และ PUT เพื่ออ่านดัชนีที่มีอยู่ และเพิ่มดัชนีใหม่เข้าไปยังตำแหน่งที่เหมาะสม

๔.๔.๔ SEARCH #n สำหรับค้นหาระเบียนในแฟ้มข้อมูลแบบดัชนี มีรูปแบบดังนี้

SEARCH #<หมายเลขแฟ้มข้อมูล>

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขของหมายเลขแฟ้มข้อมูลที่ระบุตอนเปิดแฟ้มข้อมูลแบบดัชนี

ตัวอย่าง SEARCH #1

ก่อนที่จะใช้คำสั่ง SEARCH จะต้องย้ายค่าของคีย์ที่ต้องการค้นหาเข้าไปในเนื้อที่แฟ้มข้อมูลทำงานตรงตำแหน่งเริ่มต้นของคีย์โดยใช้คำสั่ง LSET หรือ RSET การทำงานของคำสั่ง SEARCH จะมีการนำค่าของคีย์ที่อยู่ในเนื้อที่แฟ้มข้อมูลทำงานไปค้นหาในแฟ้มข้อมูลดัชนี โดยเปรียบเทียบคีย์ที่ต้องการค้นหากับคีย์ในแฟ้มข้อมูลดัชนี ถ้าเปรียบเทียบแล้วได้ค่าเท่ากับ จะนำหมายเลขระเบียนที่อยู่คู่กับคีย์ไปอ่านในแฟ้มข้อมูลใช้งาน และนำค่าของระเบียนเข้าไปไว้ในเนื้อที่แฟ้มข้อมูลทำงาน ถ้าเปรียบเทียบแล้วไม่พบค่าที่เท่ากับ จะมีการส่งรหัสข้อผิดพลาดมาให้สำหรับใช้ตรวจสอบ

การค้นหาระเบียนในแฟ้มข้อมูลแบบดัชนี โปรแกรมระบบจะเรียกใช้การทำงานของคำสั่ง GET เพื่ออ่านดัชนีในแฟ้มข้อมูลดัชนีแล้วนำค่าของคีย์ในแฟ้มข้อมูลดัชนีมาเปรียบเทียบกับค่าของคีย์ที่ต้องการค้นหา และถ้าเปรียบเทียบแล้วได้ค่าเท่ากัน ก็จะเรียกใช้การทำงานของคำสั่ง GET เพื่ออ่านระเบียนที่มีคีย์ค่านั้นมาไว้ในเนื้อที่แฟ้มข้อมูลทำงาน

๔.๔.๔ UPDATE #n สำหรับแก้ไขระเบียนเก่าในแฟ้มข้อมูลแบบดัชนี มีรูปแบบดังนี้

UPDATE #<หมายเลขแฟ้มข้อมูล>

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขของหมายเลขแฟ้มข้อมูลที่ระบุตอนเปิดแฟ้มข้อมูลแบบดัชนี

ตัวอย่าง UPDATE #1

ก่อนที่จะใช้คำสั่ง UPDATE จะต้องค้นหาระเบียนโดยใช้คำสั่ง SEARCH เสียก่อน จึงจะได้ค่าของระเบียนเข้ามาอยู่ในเนื้อที่แฟ้มข้อมูลทำงาน ซึ่งผู้ใช้สามารถแก้ไขระเบียนได้ด้วยการใช้คำสั่ง LSET หรือ RSET เขตข้อมูลทุกเขตสามารถแก้ไขได้ยกเว้นเขตข้อมูลที่เป็นคีย์ หลังจากแก้ไขเรียบร้อย จึงใช้คำสั่ง UPDATE

การทำงานของคำสั่ง UPDATE จะนำหมายเลขระเบียนของระเบียนที่กำลังแก้ไข ซึ่งหามาได้ตอนใช้คำสั่ง SEARCH มาใช้สำหรับบันทึกระเบียนข้ำลงที่เดิม

๔.๔.๖ REMOVE #n สำหรับลบระเบียนเก่าในแฟ้มข้อมูลแบบดัชนี มีรูปแบบดังนี้

REMOVE #<หมายเลขแฟ้มข้อมูล>

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขของหมายเลขแฟ้มข้อมูลที่ระบุตอนเปิดแฟ้มข้อมูลแบบดัชนี

ตัวอย่าง REMOVE #1

ก่อนที่จะใช้คำสั่ง REMOVE จะต้องย้ายค่าของคีย์ที่ต้องการลบทิ้งเข้าไปในเนื้อที่แฟ้มข้อมูลทำงานตรงตำแหน่งเริ่มต้นของคีย์โดยใช้คำสั่ง LSET หรือ RSET การทำงานของคำสั่ง REMOVE จะมีการนำค่าของคีย์ที่อยู่ในเนื้อที่แฟ้มข้อมูลทำงานไปค้นหาในแฟ้มข้อมูลดัชนี โดยเปรียบเทียบคีย์ที่ต้องการลบทิ้งกับคีย์ในแฟ้มข้อมูลดัชนี ถ้าเปรียบเทียบแล้วได้ค่าเท่ากัน จะทำการลบคีย์นั้นออกจากแฟ้มข้อมูลดัชนี และนำหมายเลขระเบียนที่มีคีย์ค่านั้นไปอ่านในแฟ้มข้อมูลใช้งาน แล้วบันทึกค่า "00" ลงไปทุกไบต์ของระเบียนนั้น

การลบระเบียนเก่าในแฟ้มข้อมูลแบบดัชนี โปรแกรมระบบจะเรียกใช้การทำงานของคำสั่ง GET เพื่ออ่านดัชนีในแฟ้มข้อมูลดัชนี แล้วนำค่าของคีย์ในแฟ้มข้อมูลดัชนีมาเปรียบเทียบกับค่าของคีย์ที่ต้องการลบทิ้ง และถ้าเปรียบเทียบแล้วได้ค่าเท่ากัน ก็จะลบคีย์นั้นออกจากแฟ้มข้อมูลดัชนี และนำหมายเลขระเบียนที่มีคีย์ค่านั้นไปอ่านในแฟ้มข้อมูลใช้งานโดยเรียกใช้การทำงานของคำสั่ง GET และบันทึกค่า "00" ลงไปทุกไบต์ของระเบียนนั้นโดยเรียกใช้การทำงานของคำสั่ง PUT

๔.๔.๗ CLOSE #n สำหรับปิดแฟ้มข้อมูลแบบดัชนี และมีรูปแบบดังนี้

CLOSE #<หมายเลขแฟ้มข้อมูล>

หมายเลขแฟ้มข้อมูล เท่ากับ ตัวเลขของหมายเลขแฟ้มข้อมูลที่ระบุตอนเปิดแฟ้มข้อมูลแบบดัชนี

ตัวอย่าง CLOSE #1

การทำงานของคำสั่ง CLOSE จะปิดแฟ้มข้อมูล ๒ แฟ้มคือแฟ้มข้อมูลใช้งานและแฟ้มข้อมูลดัชนี สำหรับแฟ้มข้อมูลดัชนีก่อนที่จะทำการปิดแฟ้มข้อมูล จะมีการบันทึกรายละเอียดต่าง ๆ ของแฟ้มข้อมูลกลับไปที่ระเบียนแรกของแฟ้มข้อมูลดัชนีด้วย

การปิดแฟ้มข้อมูลแบบดัชนี โปรแกรมระบบจะเรียกใช้การทำงานของคำสั่ง GET เพื่ออ่านระเบียนแรกของแฟ้มข้อมูลดัชนีเข้ามาในเนื้อที่แฟ้มข้อมูลทำงานหลังจากแก้ไขรายละเอียดบางอย่างของแฟ้มข้อมูลแล้ว ก็จะเรียกใช้การทำงานของคำสั่ง PUT เพื่อบันทึกระเบียนแรกชำลงที่เดิม หลังจากนั้นจะเรียกใช้การทำงานของคำสั่ง CLOSE เพื่อปิดแฟ้มข้อมูลใช้งานและแฟ้มข้อมูลดัชนี

๔.๖ <u>สรุปการใช้งานคำสั่งการจัดเก็บแฟ้มข้อมูลแบบดัชนี</u>

การใช้แฟ้มข้อมูลแบบดัชนี จะต้องเริ่มต้นด้วยการสร้างโครงสร้างแฟ้มข้อมูลก่อนเสมอ ซึ่งทำได้โดยการใช้คำสั่ง CREATE ในกรณีที่เกิดข้อผิดพลาดในการสร้างโครงสร้างแฟ้มข้อมูล ผู้ใช้ต้องทำการลบแฟ้มข้อมูลใช้งานและแฟ้มข้อมูลดัชนีก่อน แล้วจึงสร้างโครงสร้างแฟ้มข้อมูลขึ้นใหม่ ทุกครั้งที่ต้องการใช้งานแฟ้มข้อมูลแบบดัชนี จะต้องทำการเปิดแฟ้มข้อมูลด้วยการใช้ คำสั่ง OPEN "X" โดยที่แฟ้มข้อมูลและแฟ้มข้อมูลดัชนีจะต้องอยู่ในจานแม่เหล็กจานเดียวกันเสมอ หลังจากเปิดแฟ้มข้อมูลเรียบร้อย ผู้ใช้สามารถจะเพิ่ม แก้ไข ลบ หรือค้นหาระเบียนโดยใช้รูปแบบของคำสั่งและขั้นตอนการใช้งานตามที่กล่าวในหัวข้อ ๔.๔ เมื่อจบการใช้แฟ้มข้อมูลแบบดัชนี จะต้องปิดแฟ้มข้อมูลก่อนทุกครั้ง มิฉะนั้นรายละเอียดของแฟ้มข้อมูลแบบดัชนีที่เปลี่ยนแปลงไประหว่างการใช้งาน จะไม่ถูกบันทึกกลับเข้าไปในแฟ้มข้อมูลดัชนี

เนื่องจากแฟ้มข้อมูลใช้งานและแฟ้มข้อมูลดัชนีต่างก็เป็นแฟ้มข้อมูลแบบสุ่ม และแฟ้มข้อมูลดัชนีจะมีความยาวระเบียนเท่ากับ ๑๒๘ ไบต์เสมอ ดังนั้นการเริ่มต้นอินเดอร์-

เพรดเดอร์ จะต้องระบุตัวแปรต้นให้มีความยาวสูงสุดของระเบียนไม่น้อยกว่า ๑๒๘ ไบต์เสมอ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย