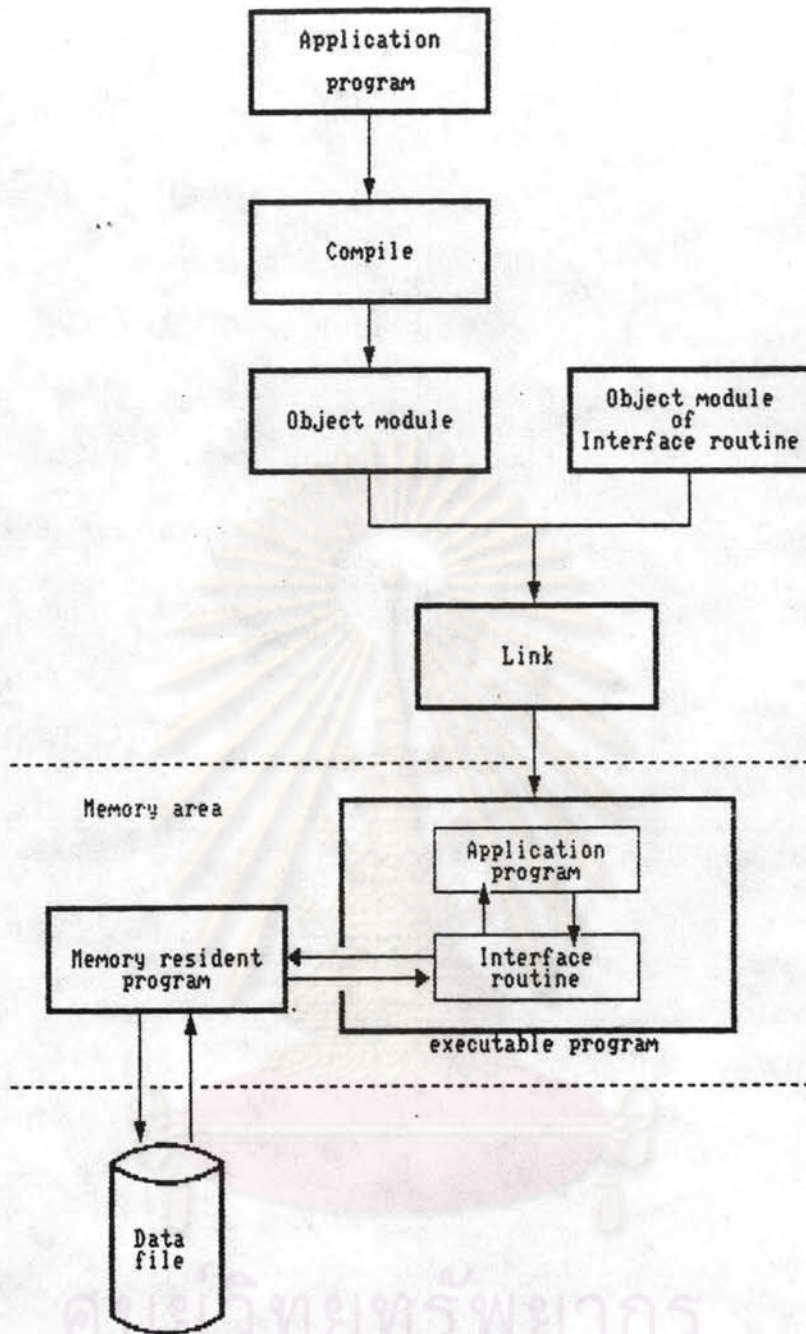




การพัฒนาโปรแกรมสำหรับฟังก์ชันจัดการแฟ้มข้อมูลดัชนี

แนวทางของการพัฒนาโปรแกรมในงานวิจัย มาจากการใช้ไลบรารีฟังก์ชัน (Library function) ซึ่งประกอบด้วยฟังก์ชันการทำงานต่าง ๆ ที่จำเป็นสำหรับโปรแกรมประยุกต์ (Application program) และถูกรวบรวมอยู่ในแฟ้มไลบรารี (Library file) การสร้างโปรแกรมประยุกต์ให้เป็นโปรแกรมที่ใช้ปฏิบัติงานได้ (Executable Program) จะต้องผ่านการเชื่อมโยง (link) กับไลบรารีฟังก์ชันจากแฟ้มไลบรารีทุกครั้ง เมื่อใดที่โปรแกรมประยุกต์จำเป็นต้องเปลี่ยนแปลงการใช้ภาษาคอมพิวเตอร์จากเดิมเป็นอีกภาษาหนึ่ง ไลบรารีฟังก์ชันที่พัฒนาขึ้นเพื่อใช้กับภาษาเดิม ก็จำเป็นต้องแก้ไขให้ใช้ได้กับภาษาใหม่เช่นกัน ดังนั้นเพื่อให้ไลบรารีฟังก์ชันที่พัฒนาขึ้น มีความสามารถในการเรียกใช้จากโปรแกรมภาษาต่าง ๆ ได้ทั่วไป จึงได้ออกแบบพัฒนาไลบรารีฟังก์ชันให้เป็นลักษณะของรันไทม์ไลบรารี (Run-time Library) ซึ่งเป็นไลบรารีฟังก์ชันที่ไม่ต้องผ่านการเชื่อมโยงกับโปรแกรมที่เรียกใช้งาน เพราะเป็นการรวบรวมฟังก์ชันทั้งหมดให้บรรจุในโปรแกรมชนิดหนึ่ง ที่สามารถบรรจุประจำในหน่วยความจำ (Memory Resident Program) ฟังก์ชันที่พัฒนาในรูปแบบนี้จึงไม่ผูกพันกับภาษาใดภาษาหนึ่ง แต่ต้องสร้างโปรแกรมติดต่อ ซึ่งเป็นโปรแกรมย่อยที่พัฒนาตามรูปแบบของภาษาที่ใช้กับโปรแกรมประยุกต์ เป็นโปรแกรมที่ใช้ติดต่อระหว่างโปรแกรมประยุกต์ และไลบรารีฟังก์ชันเหล่านั้น หน้าที่ของโปรแกรมติดต่อคือ รับพารามิเตอร์จากโปรแกรมประยุกต์ และส่งไปพร้อมกับการเรียกใช้ฟังก์ชันเหล่านั้น สำหรับไลบรารีฟังก์ชันที่พัฒนาขึ้นนี้เป็นฟังก์ชันการทำงานที่เกี่ยวกับการจัดการแฟ้มข้อมูล และมีการจัดทำดัชนี ซึ่งมีโครงสร้างการจัดเรียงของดัชนีในลักษณะ โครงสร้างบี-ทรี ทำหน้าที่ช่วยในการค้นหาระเบียบข้อมูลด้วยการระบุค่าคีย์ที่ต้องการค้นหา การพัฒนาโปรแกรมในงานวิจัยได้แบ่งออกเป็น

1. โปรแกรมบรรจุประจำในหน่วยความจำ
2. โปรแกรมติดต่อ
3. โปรแกรมอรรถประโยชน์สำหรับจัดการแฟ้มข้อมูล



ภาพที่ 5.1 ลักษณะการทำงานของโปรแกรมบรรจุประจำในหน่วยความจำ

การพัฒนาโปรแกรมบรรจุประจำในหน่วยความจำ (BTREE Program)



ภายในโปรแกรมประกอบด้วยการทำงานหลักที่สำคัญดังนี้

1. การติดตั้งโปรแกรมให้ประจำในหน่วยความจำ ประกอบด้วยการทำงาน
 - 1.1 การรับและกำหนดค่าเริ่มต้นจากพารามิเตอร์ที่ส่งมาพร้อมกับการประมวลผลโปรแกรมที่บรรทัดคำสั่ง (command line) มีรูปแบบการกำหนดคือ

[drive:]BTREE [/Msize] [/Psize] [/Idrive] [/D] [/S]

drive หมายถึง หน่วยขับจานบันทึก

BTREE หมายถึง ชื่อโปรแกรมที่บรรจุประจำในหน่วยความจำ

สำหรับความหมายของพารามิเตอร์ที่กำหนด ได้แก่

/M เป็นการกำหนดเนื้อที่หน่วยความจำให้กับบัพเฟอร์สำหรับช่องใส่แผ่น โดยที่ size หมายถึง ตัวเลขที่มีหน่วยเป็นกิโลไบต์ สามารถกำหนดขนาดบัพเฟอร์ได้ตั้งแต่ 20 - 64 กิโลไบต์ ในกรณีที่ไม่กำหนดพารามิเตอร์นี้ โปรแกรมจะกำหนดขนาดบัพเฟอร์ให้เท่ากับ 20 กิโลไบต์

/P เป็นการกำหนดขนาดช่องใส่แผ่นแต่ละช่องในบัพเฟอร์ โดยที่ size หมายถึงตัวเลขที่มีหน่วยเป็นไบต์ สามารถกำหนดขนาดได้ตั้งแต่ 512 - 4096 ไบต์ และขนาดช่องใส่แผ่นที่กำหนดต้องเป็นผลคูณของ 512 เช่น /P512 /P1024 หรือ /P2048 ขนาดช่องใส่แผ่นที่ใช้ต้องมากกว่า หรือ เท่ากับขนาดแผ่นของแฟ้มข้อมูลดัชนีทุกแฟ้มที่จะนำมาใช้งานในขณะนั้น ในกรณีที่ไม่กำหนดพารามิเตอร์นี้ โปรแกรมจะกำหนดขนาดช่องใส่แผ่นให้เท่ากับ 512 ไบต์

/I เป็นการระบุหน่วยขับจานบันทึก ที่ใช้บันทึกแฟ้มสำรองแผ่นของแฟ้มข้อมูลดัชนีทุกแฟ้มที่เปิดใช้ในขณะนั้น โดย drive หมายถึง ตัวอักษรของหน่วยขับจานบันทึกที่กำหนด เช่น /IA /IB /IC ... ในกรณีที่ไม่กำหนดพารามิเตอร์นี้ โปรแกรมจะกำหนดจากหน่วยขับจานบันทึกที่ใช้ในขณะนั้น

/D เป็นพารามิเตอร์ที่กำหนดให้ฟังก์ชันต่าง ๆ ในโปรแกรม BTREE แสดงข้อความบนจอภาพ เมื่อถูกเรียกใช้จากโปรแกรมประยุกต์ เพื่อใช้ในการติดตามผลการดำเนินงานของฟังก์ชันจัดการแฟ้มข้อมูลดัชนี รวมทั้งรหัสความผิดพลาดและข้อความที่อธิบายความผิดพลาดที่เกิดขึ้น

/S เป็นพารามิเตอร์ที่ใช้ยกเลิกการทำงานของฟังก์ชันในโปรแกรม BTREE ซึ่งติดตั้งประจำในหน่วยความจำขณะนั้น และคืนเนื้อที่หน่วยความจำที่เป็นของโปรแกรมและบัฟเฟอร์สำหรับช่องใส่แผ่นทั้งหมดให้กับระบบเอมเอสดอส

1.2 การติดตั้งบัฟเฟอร์และกำหนดระบบการใช้ช่องใส่แผ่น ประกอบด้วย

1.2.1 จองเนื้อที่หน่วยความจำให้กับบัฟเฟอร์สำหรับช่องใส่แผ่น

โดยมีขนาดบัฟเฟอร์ตามพารามิเตอร์ /M

1.2.2 แบ่งเนื้อที่บัฟเฟอร์ออกเป็นช่องใส่แผ่นที่มีขนาดเท่า ๆ กัน

โดยมีขนาดช่องใส่แผ่นตามพารามิเตอร์ /P

1.2.3 สร้างเนื้อที่แถวคอยสำหรับควบคุมการใช้ช่องใส่แผ่นในบัฟเฟอร์

1.3 กำหนดตำแหน่งของฟังก์ชันในโปรแกรม BTREE ที่เป็นจุดเริ่มต้นการทำงานหลังจากถูกเรียกใช้จากโปรแกรมประยุกต์ บรรทัดตำแหน่งดังกล่าวในอินเทอร์รัปต์เวคเตอร์หมายเลข 7Ch โดยใช้ฟังก์ชันระบบหมายเลข 25h เพื่อกำหนดเวคเตอร์ใหม่ในตารางอินเทอร์รัปต์เวคเตอร์ และจบการทำงานของโปรแกรม แต่คงเนื้อที่หน่วยความจำที่ใช้สำหรับโปรแกรมและบัฟเฟอร์ไว้ โดยใช้ฟังก์ชันระบบหมายเลข 31h เพื่อจบการทำงาน

2. ฟังก์ชันเริ่มต้นการทำงานหลังจากถูกเรียกใช้จากโปรแกรมประยุกต์

เป็นฟังก์ชันที่รับการควบคุมการทำงานมาจากโปรแกรมประยุกต์ ประกอบด้วยการทำงาน

2.1 เปลี่ยนแปลงการใช้สแตคเซกเมนต์ (stack segment) ของโปรแกรมประยุกต์มาใช้สแตคเซกเมนต์ของโปรแกรม BTREE เพื่อหลีกเลี่ยงการใช้สแตคของโปรแกรมประยุกต์

2.2 เก็บบันทึกค่าสถานะเดิมของทุกรีจิสเตอร์ที่มาจากโปรแกรมประยุกต์ ได้แก่ รีจิสเตอร์ AX, BX, CX, DX, SI, DI, BP, ES และ DS ไว้ในสแตค

2.3 ตรวจสอบพารามิเตอร์ที่ส่งมาจากโปรแกรมประยุกต์ เช่น การระบุหมายเลขฟังก์ชันที่ถูกเรียกใช้จากโปรแกรมประยุกต์ ข้อมูลภายในบล็อกสารสนเทศแฟ้มที่ส่งผ่านระหว่างฟังก์ชันและโปรแกรมประยุกต์ และการระบุหมายเลขชุดคีย์ เป็นต้น

2.4 ส่งการทำงานไปให้กับฟังก์ชันที่ต้องการ และหลังจากจบการทำงาน
ของฟังก์ชัน จะบันทึกค่าสถานะเดิมของรีจิสเตอร์พร้อมกับเปลี่ยนแปลงแอสเซมบลี และส่ง
การทำงานกลับคืนให้กับโปรแกรมประยุกต์

3. ฟังก์ชันสำหรับจัดการแฟ้มข้อมูลดัชนี เป็นฟังก์ชันที่ได้พัฒนาสำหรับการเรียกใช้
(call) จากโปรแกรมประยุกต์ โดยเรียกผ่านโปรแกรมติดต่อ และต้องมีพารามิเตอร์ที่ใช้
ร่วมกันระหว่างโปรแกรมประยุกต์กับฟังก์ชันเหล่านั้น ทุกฟังก์ชันจะรับพารามิเตอร์ที่ส่งมา
เหมือนกัน แต่ความหมายและการใช้งานของแต่ละฟังก์ชันจะอาจต่างกัน พารามิเตอร์ที่ใช้
จะประกอบด้วย

1. operation number สำหรับกำหนดหมายเลขฟังก์ชันที่ต้องการ
2. file information block สำหรับบล็อกสารสนเทศแฟ้ม
3. data buffer สำหรับบัฟเฟอร์ที่ใช้ในการส่งและรับระเบียบข้อมูล
4. key buffer สำหรับบัฟเฟอร์ที่ใช้ในการส่งและรับคีย์
5. key number สำหรับระบุหมายเลขชุดคีย์

สำหรับฟังก์ชันจัดการแฟ้มข้อมูลดัชนี ประกอบด้วย

3.1 ฟังก์ชันสร้างแฟ้มข้อมูลดัชนี (Create B-Tree file)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 1
2. data buffer สำหรับบัฟเฟอร์ที่กำหนดคุณสมบัติของแฟ้มและคีย์
3. key buffer สำหรับกำหนดชื่อแฟ้มที่ต้องการสร้าง
การทำงาน

1. รูปแบบของชื่อแฟ้ม สามารถกำหนดได้ในลักษณะของ
drive:\path\file name.extension ซึ่งปิดท้ายชื่อภายใน key buffer ด้วยการ
เว้นช่องว่างอย่างน้อย 1 ช่อง หรือ ใช้รหัสแอสกี 0 แทน

2. รูปแบบการกำหนดคุณสมบัติของแฟ้มและคีย์ใน data buffer
ประกอบด้วยกลุ่มข้อมูลที่เรียงต่อกันของ

- กลุ่มข้อมูลคุณสมบัติเพิ่มเติม ได้แก่

1. ความยาวระเบียบข้อมูล
2. ขนาดแผ่น
3. จำนวนชุดคีย์ภายในแผ่น
4. แพลกที่ระบุการสร้างแผ่นแผนที่ข้อมูลภายในแผ่น

โดยค่าที่ใช้ระบุคือ 0 หมายถึง ไม่มีการสร้างแผ่นแผนที่ข้อมูล และ 1 หมายถึง มีการสร้างแผ่นชนิดนี้ในแผ่น

- กลุ่มข้อมูลคุณสมบัติคีย์ ประกอบด้วยกลุ่มข้อมูลของชุดคีย์ต่าง ๆ ที่กำหนดในลักษณะแถวลำดับเชกเมนต์คีย์เรียงต่อกัน เนื้อหาของแต่ละเชกเมนต์คีย์เหมือนกับบล็อกสารสนเทศสำหรับเชกเมนต์ ซึ่งแยกแต่ละชุดคีย์ด้วยการใช้แพลกที่ระบุการเป็นเชกเมนต์คีย์เนื้อหาประกอบด้วย

1. ตำแหน่งเริ่มต้นในระเบียบข้อมูล
2. ความยาวของคีย์
3. แพลกที่ระบุคุณสมบัติของคีย์

คุณสมบัติของคีย์	เลขฐานสิบหก	เลขฐานสอง	เลขฐานสิบ
กำหนดให้คีย์มีค่าซ้ำกันได้	0001	00000001	1
กำหนดให้คีย์เปลี่ยนแปลงค่าได้	0002	00000010	2
คีย์เป็นเลขจำนวนเต็ม	0004	00000100	4
คีย์เป็นเชกเมนต์คีย์	0008	00001000	8
มีการใช้ชุดการสลับลำดับตัวอักษร	0010	00010000	16
คีย์เป็นตัวอักษรภาษาไทย	0020	00100000	32

ตารางที่ 5.1 แพลกสำหรับการกำหนดคุณสมบัติของคีย์

3. การกำหนดคุณสมบัติของคีย์ที่ให้คีย์ซ้ำกันได้ และ/หรือสามารถเปลี่ยนแปลงค่าได้ จะต้องกำหนดให้มีคุณสมบัติเช่นเดียวกันทุกเซกเมนต์คีย์สำหรับชุดคีย์นั้น

4. เมื่อกำหนดให้คีย์มีการใช้ชุดการสลับลำดับตัวอักษร จะต้องกำหนดชื่อแฟ้มชุดการสลับลำดับตัวอักษรต่อท้ายกลุ่มข้อมูลคุณสมบัติคีย์ โดยต้องคั่นด้วยรหัสควบคุมที่เป็นตัวเลขจำนวนเต็มขนาด 2 ไบต์ มีค่าเท่ากับ 00ACh และต้องอยู่ในเนื้อที่เดียวกันของ data buffer รูปแบบชื่อแฟ้มกำหนดได้ลักษณะเดียวกับชื่อแฟ้มที่ต้องการสร้าง และต้องปิดท้ายด้วยการเว้นช่องว่างอย่างน้อย 1 ช่องหรือใช้รหัสแอสกี 0 แทน

5. เมื่อเกิดการผิดพลาดในการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่อแสดงสาเหตุความผิดพลาดที่เกิด

3.2 ฟังก์ชันการเปิดแฟ้มข้อมูลดัชนี (Open B-Tree file)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 2
2. file information block สำหรับเก็บบันทึกบล็อก

สารสนเทศแฟ้ม

3. key buffer สำหรับกำหนดชื่อแฟ้มที่ต้องการเปิด
4. key number สำหรับระบุลักษณะการเปิดแฟ้ม

การทำงาน

1. รูปแบบชื่อแฟ้ม กำหนดได้ในลักษณะ

drive:\path\file name.extension ปิดท้ายด้วยการเว้นช่องว่าง หรือ ใช้รหัสแอสกี 0

2. ลักษณะการเปิดแฟ้ม (Open mode) กำหนดด้วยเลขจำนวนเต็ม

จุฬาลงกรณ์มหาวิทยาลัย

key number	ลักษณะการเปิดแฟ้ม
1 2	กำหนดให้เปิดแฟ้มข้อมูลดัชนีอย่างเดียว ไม่มีการสร้างแฟ้มสำรองแผ่น กำหนดให้เปิดแฟ้มข้อมูลดัชนีเพื่อการอ่านข้อมูลจากแฟ้มเท่านั้น สำหรับการ อ่านระเบียบข้อมูลจากแฟ้มข้อมูลที่ตรวจพบว่า อาจเกิดความ เสียหายของข้อมูลในแฟ้ม
๑ หรือค่าอื่น ๆ	กำหนดให้เปิดแฟ้มข้อมูลดัชนี และสร้างแฟ้มสำรองแผ่นเพื่อใช้สำรอง แผ่นเดิมที่ถูกแก้ไขจากการเพิ่ม การลบ การแก้ไขระเบียบข้อมูล

ตารางที่ 5.2 การกำหนดลักษณะการเปิดแฟ้ม

3. เมื่อทำการเปิดแฟ้ม ฟังก์ชันจะตรวจสอบความเสียหายของ
ข้อมูลจากการหยุดทำงานของคอมพิวเตอร์ในระหว่างทำการบันทึกข้อมูลลงจานบันทึก ถ้าตรวจ
พบความเสียหาย ฟังก์ชันจะอ่านข้อมูลเดิมจากแผ่นต่าง ๆ ในแฟ้มสำรองแผ่นเพื่อบรรจุลงที่เดิม
ในแฟ้มข้อมูลดัชนี และเปิดแฟ้มใช้งานได้ต่อไป

4. เมื่อเกิดความผิดพลาดในการทำงาน ฟังก์ชันจะส่งรหัสความ
ผิดพลาดที่ไม่เท่ากับ ๑ กลับมาเพื่อแสดงสาเหตุความผิดพลาดที่เกิด

3.3 ฟังก์ชันการปิดแฟ้ม (Close B-Tree file)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 3
2. file information block สำหรับบล็อกสารสนเทศ

แฟ้มของแฟ้มข้อมูลดัชนีที่ต้องการปิด
การทำงาน

1. การปิดแฟ้มจะนำข้อมูลในช่องใส่แผ่นของบัฟเฟอร์ซึ่ง เกี่ยว
ข้องกับแฟ้มข้อมูลดัชนีนั้นออกจากทุกช่อง และลบแฟ้มสำรองแผ่นที่สร้างขึ้นออกจากจานบันทึก
และปิดแฟ้มข้อมูลดัชนีเป็นอันดับสุดท้าย

2. เมื่อเกิดความผิดพลาดในการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่อแสดงสาเหตุความผิดพลาดที่เกิดขึ้น

3.4 ฟังก์ชันการเพิ่มระเบียบข้อมูล (Insert record)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 4
2. file information block
3. data buffer สำหรับบัฟเฟอร์ของระเบียบข้อมูลที่ต้องการ

เพิ่มเติมในแฟ้ม

4. key buffer
5. key number สำหรับระบุชุดคีย์ที่ต้องการอ้างอิงถึงภายหลังจาก

การเพิ่มเติมระเบียบข้อมูล

การทำงาน

1. เนื้อหาระเบียบข้อมูล ประกอบด้วยเขตข้อมูลต่าง ๆ ที่กำหนดโดยโปรแกรมประยุกต์และไม่ควรมีรหัสควบคุมใด ๆ แทรกระหว่างเขตข้อมูลเหล่านั้น เพื่อไม่ให้เกิดความคลาดเคลื่อนของตำแหน่งคีย์ในระเบียบข้อมูลนั้น

2. ในระหว่างทำการเพิ่มเติมระเบียบข้อมูล ฟังก์ชันจะทำการเพิ่มเติมคีย์ทุกชุดที่มาจากระเบียบข้อมูลนั้น และปรับโครงสร้างการจัดเรียงของดัชนีทุกชุดคีย์ นอกจากนี้จะตรวจสอบคุณสมบัติการมีคีย์ที่ซ้ำกันได้จากการเพิ่มเติมทุกชุดคีย์

3. เมื่อเพิ่มเติมระเบียบข้อมูลได้สำเร็จ ฟังก์ชันจะส่งคีย์ที่เกิดจากการเพิ่มเติมตามชุดคีย์ที่ระบุใน key number กลับมาใน key buffer

4. เมื่อเกิดความผิดพลาดในการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่อแสดงสาเหตุความผิดพลาดที่เกิด

3.5 ฟังก์ชันการลบระเบียบข้อมูล (Delete record)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 5
2. file information block

3. key number สำหรับระบุชุดคีย์ที่ต้องการอ้างอิงถึงภายหลังจากการลบระเบียนข้อมูล

การทำงาน

1. ก่อนการลบระเบียนข้อมูลใด ๆ ต้องทำการค้นหาเพื่อระบุตำแหน่งระเบียนข้อมูลที่ต้องการนั้น โดยการใช้ฟังก์ชันค้นหาระเบียนข้อมูล
2. การลบระเบียนข้อมูล จะส่งผลให้คีย์ทุกชุดคีย์ของระเบียนข้อมูลนั้นถูกลบออกจากแฟ้มข้อมูลดัชนีด้วย
3. เมื่อเกิดความผิดพลาดในการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่อแสดงสาเหตุความผิดพลาดที่เกิด

3.6 ฟังก์ชันการเปลี่ยนแปลงระเบียนข้อมูล (Update record)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 6
2. file information block
3. data buffer สำหรับบัฟเฟอร์ของระเบียนข้อมูลที่จะเปลี่ยนแปลงข้อมูล
4. key buffer
5. key number สำหรับระบุชุดคีย์ที่ต้องการอ้างอิง หลังจากเปลี่ยนแปลงข้อมูล

การทำงาน

1. เช่นเดียวกับการลบระเบียนข้อมูล ต้องทำการค้นหาเพื่อระบุตำแหน่งระเบียนข้อมูลที่ต้องการนั้น โดยการใช้ฟังก์ชันค้นหาระเบียนข้อมูล
2. เมื่อการเปลี่ยนแปลงข้อมูล ส่งผลให้ค่าคีย์ชุดใดเปลี่ยนแปลงไป จะต้องผ่านการตรวจสอบคุณสมบัติของคีย์ที่ให้มีการเปลี่ยนแปลงค่าได้หรือไม่ แล้วจึงทำการลบคีย์เดิมออก และเพิ่มคีย์ที่เปลี่ยนไปลงในชุดคีย์นั้นแต่ต้องตรวจสอบคุณสมบัติที่คีย์มีค่าที่ซ้ำกันได้เช่นกัน
3. เมื่อเปลี่ยนแปลงระเบียนข้อมูลได้สำเร็จ ฟังก์ชันจะส่งคีย์จากชุดคีย์ที่ระบุใน key number ซึ่งเป็นคีย์ของระเบียนข้อมูลที่ถูกเปลี่ยนแปลง กลับมาใน key buffer



4. เมื่อเกิดความผิดพลาดในการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่อแสดงสาเหตุความผิดพลาดที่เกิดขึ้น

3.7 ฟังก์ชันการค้นหาและอ่านระเบียบข้อมูลจากแฟ้มโดยการค้นหาจากคีย์พารามิเตอร์ที่ใช้

1. operation number
2. file information block
3. data buffer สำหรับรับระเบียบข้อมูลที่อ่านได้
4. key buffer สำหรับกำหนดคีย์ที่ต้องการค้นหา
5. key number สำหรับระบุชุดคีย์ที่ต้องการค้นหา

แบ่งฟังก์ชันค้นหาี้ตามลักษณะการเปรียบเทียบคีย์ที่ต้องการค้นหา

ได้ดังนี้

3.7.1 ค้นหาคีย์ที่มีค่าเท่ากับคีย์ที่ระบุ (Get the equal key) การทำงาน

1. กำหนดหมายเลขฟังก์ชันใน operation number

เท่ากับ 7

2. ทำการค้นหาคีย์ที่มีค่ากับคีย์ที่ระบุใน key buffer

เมื่อค้นหาพบจะอ่านระเบียบข้อมูลจากตำแหน่งที่พบในดัชนี และส่งระเบียบข้อมูลนั้นกลับมาใน data buffer

3. ระเบียบข้อมูลที่อยู่ในตำแหน่งข้างเคียง ได้แก่

ระเบียบข้อมูลที่ถัดไปจากระเบียบข้อมูลที่ค้นพบ จะเป็นระเบียบที่มีค่าคีย์มากกว่าคีย์ที่พบ หรือเป็นระเบียบข้อมูลลำดับต่อมาซึ่งมีค่าคีย์ซ้ำกับคีย์ที่พบ และระเบียบข้อมูลก่อนหน้าระเบียบที่พบจะเป็นระเบียบที่มีค่าคีย์ที่น้อยกว่าคีย์ที่พบ

4. เมื่อไม่พบคีย์ที่ต้องการหรือเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายสาเหตุความผิดพลาดที่เกิดขึ้น

3.7.2 ค้นหาคีย์ที่มีค่าน้อยกว่าหรือเท่ากับคีย์ที่ระบุ (Get the less than or equal key)

การทำงาน

1. กำหนดหมายเลขฟังก์ชันใน operation number เท่ากับ 8
2. ทำการค้นหาคีย์ที่มีค่าเท่ากับคีย์ที่ระบุใน key buffer ถ้าไม่พบคีย์ที่กำหนดใน key buffer จะทำการค้นหาคีย์ที่มีค่าน้อยกว่าต่อไป และอ่านคีย์ที่พบ กับระเบียบข้อมูลของคีย์นั้น ส่งกลับมาใน key buffer และ data buffer ตามลำดับ
3. ระเบียบข้อมูลที่อยู่ในตำแหน่งข้างเคียง ได้แก่ ระเบียบข้อมูลตำแหน่งถัดไป จะเป็นระเบียบที่มีค่าคีย์มากกว่าคีย์ที่พบ หรือเป็นระเบียบข้อมูลลำดับต่อมาซึ่งมีค่าคีย์ซ้ำกับคีย์ที่พบ และระเบียบข้อมูลตำแหน่งก่อนหน้า จะเป็นระเบียบข้อมูลที่มีค่าคีย์ที่น้อยกว่าคีย์ที่พบ หรือเป็นระเบียบข้อมูลลำดับสุดท้ายที่มีค่าคีย์ซ้ำกัน
4. เมื่อไม่พบคีย์ที่ต้องการ หรือเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายสาเหตุความผิดพลาดที่เกิด

3.7.3 ค้นหาคีย์ที่มีค่าน้อยกว่าคีย์ที่ระบุ (Get the less than key)

การทำงาน

1. กำหนดหมายเลขฟังก์ชันใน operation number เท่ากับ 9
2. ทำการค้นหาคีย์ที่มีค่าน้อยกว่าคีย์ที่ระบุใน key buffer และอ่านคีย์ที่พบ กับระเบียบข้อมูลของคีย์นั้น ส่งกลับมาใน key buffer และ data buffer ตามลำดับ
3. สำหรับระเบียบข้อมูลในตำแหน่งข้างเคียง จะเป็นเช่นเดียวกับการค้นหาคีย์ที่มีค่าน้อยกว่าหรือเท่ากับคีย์ที่ระบุ
4. เมื่อไม่พบคีย์ที่ต้องการ หรือเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายสาเหตุความผิดพลาดที่เกิด

3.7.4 ค้นหาคีย์ที่มีค่ามากกว่าหรือเท่ากับคีย์ที่ระบุ

(Get the greater than or equal key)

การทำงาน

1. กำหนดหมายเลขฟังก์ชันใน operation number เท่ากับ 10
2. ทำการค้นหาคีย์ที่มีค่าเท่ากับคีย์ที่ระบุใน key buffer ถ้าไม่พบคีย์ที่กำหนดใน key buffer จะทำการค้นหาคีย์ที่มีค่ามากกว่าต่อไป และอ่านคีย์ที่พบกับระเบียบข้อมูลของคีย์นั้น ส่งกลับมาใน key buffer และ data buffer ตามลำดับ
3. สำหรับระเบียบข้อมูลที่ตำแหน่งข้างเคียง จะเป็น เช่นเดียวกับการค้นหาคีย์ที่มีค่าน้อยกว่าหรือเท่ากับคีย์ที่ระบุ
4. เมื่อไม่พบคีย์ที่ต้องการ หรือเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายสาเหตุความผิดพลาดที่เกิด

3.7.5 ค้นหาคีย์ที่มีค่ามากกว่าคีย์ที่ระบุ (Get the greater than key)

การทำงาน

1. กำหนดหมายเลขฟังก์ชันใน operation number เท่ากับ 11
2. ทำการค้นหาคีย์ที่มีค่ามากกว่าคีย์ที่ระบุใน key buffer และอ่านคีย์ที่พบกับระเบียบข้อมูลของคีย์นั้น ส่งกลับมาใน key buffer และ data buffer ตามลำดับ
3. สำหรับระเบียบข้อมูลในตำแหน่งข้างเคียง จะเป็น เช่นเดียวกับการค้นหาคีย์ที่มีค่าน้อยกว่าหรือเท่ากับคีย์ที่ระบุ
4. เมื่อไม่พบคีย์ที่ต้องการ หรือเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายสาเหตุความผิดพลาดที่เกิด

3.8 ฟังก์ชันการค้นหาและอ่านระเบียบข้อมูลจากคีย์ลำดับก่อนหน้า

(Get the previous key)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 12
2. file information block
3. data buffer สำหรับรับระเบียบข้อมูลที่สามารถอ่านได้
4. key buffer
5. key number สำหรับระบุชุดคีย์ที่ต้องการค้นหา

การทำงาน

1. ทำการค้นหาคีย์ที่อยู่ในลำดับก่อนหน้าของคีย์จากระเบียบข้อมูลตำแหน่งปัจจุบัน ซึ่งต้องระบุลงไว้ใน key buffer ระเบียบข้อมูลที่สามารถอ่านจากคีย์ที่พบ อาจเป็นระเบียบซึ่งมีคีย์น้อยกว่าคีย์ใน key buffer หรือเป็นระเบียบข้อมูลในลำดับก่อนหน้าซึ่งระเบียบข้อมูลทั้งสองมีคีย์ที่ซ้ำกัน ระเบียบข้อมูล และคีย์ที่ค้นพบจะส่งกลับมาใน data buffer และ key buffer ตามลำดับ
2. ระเบียบข้อมูลในตำแหน่งข้างเคียงหลังจากการเลื่อนตำแหน่งได้แก่ ระเบียบข้อมูลตำแหน่งถัดไป จะเป็นระเบียบข้อมูลเดิมก่อนที่จะเรียกใช้ฟังก์ชันนี้ และระเบียบข้อมูลตำแหน่งก่อนหน้า จะเป็นระเบียบข้อมูลที่อยู่ลำดับก่อนหน้าระเบียบที่พบ ถ้าระเบียบทั้งสองมีคีย์ที่ซ้ำกัน หรืออาจเป็นระเบียบข้อมูลที่มีคีย์น้อยกว่าคีย์ใน key buffer
3. เมื่อไม่พบคีย์ที่ต้องการ หรือเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายสาเหตุความผิดพลาดที่เกิดขึ้น

3.9 ฟังก์ชันการค้นหาและอ่านระเบียบข้อมูลจากคีย์ลำดับถัดไป

(Get the next key)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 13
2. file information block
3. data buffer สำหรับรับระเบียบข้อมูลที่สามารถอ่านได้
4. key buffer
5. key number สำหรับระบุชุดคีย์ที่ต้องการค้นหา

การทำงาน

1. ทำการค้นหาคีย์ที่อยู่ในลำดับถัดไปจากคีย์ของระเบียบข้อมูลตำแหน่งปัจจุบัน ซึ่งต้องระบุงไว้ ใน key buffer ระเบียบข้อมูลที่อ่านจากคีย์ที่พบ อาจเป็นระเบียบข้อมูลซึ่งมีค่าคีย์มากกว่าคีย์ใน key buffer หรือเป็นระเบียบข้อมูลในลำดับถัดไป ซึ่งระเบียบข้อมูลทั้งสองมีค่าคีย์ที่ซ้ำกัน ระเบียบข้อมูลและคีย์ที่ค้นพบจะส่งกลับมาใน data buffer และ key buffer ตามลำดับ
2. ระเบียบข้อมูลในตำแหน่งข้างเคียงหลังจากการเลื่อนตำแหน่งได้แก่ ระเบียบข้อมูลตำแหน่งถัดไป จะเป็นระเบียบข้อมูลที่อยู่ลำดับถัดไปจากระเบียบที่พบ ถ้าระเบียบทั้งสองมีค่าคีย์ที่ซ้ำกัน หรือ อาจเป็นระเบียบข้อมูลที่มีค่าคีย์มากกว่าคีย์ใน key buffer และระเบียบข้อมูลลำดับก่อนหน้า จะเป็นระเบียบข้อมูลเดิมก่อนที่จะเรียกใช้ฟังก์ชัน
3. เมื่อไม่พบคีย์ที่ต้องการ หรือเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายสาเหตุความผิดพลาดที่เกิด

3.10 ฟังก์ชันการค้นหาและอ่านระเบียบข้อมูลจากคีย์ที่มีค่าน้อยที่สุด

(Get the lowest key)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 14
2. file information block
3. data buffer สำหรับรับระเบียบข้อมูลที่อ่านได้
4. key buffer สำหรับรับคีย์ที่ค้นพบ
5. key number สำหรับระบุชุดคีย์ที่ต้องการค้นหา

การทำงาน

1. ทำการค้นหาคีย์ที่มีค่าน้อยที่สุดจากชุดคีย์ที่ระบุใน key number และอ่านระเบียบข้อมูลกับคีย์ที่พบส่งกลับมาใน data buffer และ key buffer ตามลำดับ
2. เมื่อเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายสาเหตุความผิดพลาดที่เกิด



3.11 ฟังก์ชันการค้นหาและอ่านระเบียบข้อมูลจากคีย์ที่มีค่ามากที่สุด

(Get the highest key)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 15
2. file information block
3. data buffer สำหรับรับระเบียบข้อมูลที่สามารถอ่านได้
4. key buffer สำหรับรับคีย์ที่ค้นพบ
5. key number สำหรับระบุชุดคีย์ที่ต้องการค้นหา

การทำงาน

1. ทำการค้นหาคีย์ที่มีค่ามากที่สุดจากชุดคีย์ที่ระบุใน key number และอ่านระเบียบข้อมูลกับคีย์ที่พบส่งกลับมาใน data buffer และ key buffer ตามลำดับ
2. เมื่อเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายความผิดพลาดที่เกิด

3.12 ฟังก์ชันการอ่านตำแหน่งระเบียบข้อมูล (Get position)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 16
2. file information block
3. data buffer สำหรับรับตำแหน่งระเบียบข้อมูลปัจจุบันในแฟ้ม

การทำงาน

1. ทำการส่งค่าที่ระบุตำแหน่งระเบียบข้อมูลกลับมาใน data buffer ซึ่งมีขนาดตำแหน่งระเบียบข้อมูลเท่ากับ 4 ไบต์
2. เมื่อเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายสาเหตุความผิดพลาดที่เกิด

3.13 ฟังก์ชันการค้นหาและอ่านระเบียบข้อมูลจากตำแหน่งระเบียบข้อมูล

(Get direct)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 17

2. file information block
3. data buffer
4. key buffer สำหรับรับคีย์ของระเบียบข้อมูลที่พบ
5. key number สำหรับระบุชุดคีย์ที่ต้องการ

การทำงาน

1. ทำการอ่านระเบียบข้อมูลจากตำแหน่งระเบียบที่ได้จากฟังก์ชันการอ่านตำแหน่งระเบียบข้อมูล ซึ่งกำหนดใน data buffer สำหรับระเบียบข้อมูลที่จะอ่านได้จะส่งกลับมาใน data buffer เดียวกัน ดังนั้นเนื้อที่สำหรับ data buffer จึงต้องมีขนาดที่เพียงพอสำหรับรับระเบียบข้อมูลที่ส่งกลับมา
2. ทำการค้นหาคีย์ของระเบียบข้อมูลที่ได้ตามชุดคีย์ที่ระบุใน key number คีย์ที่ค้นพบจะส่งกลับมาใน key buffer
3. เมื่อเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายความผิดพลาดที่เกิด

3.14 ฟังก์ชันการเลื่อนตำแหน่งตามลำดับของระเบียบข้อมูล (Step direct)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 18
2. file information block
3. data buffer สำหรับรับระเบียบข้อมูล

การทำงาน

1. ทำการเลื่อนตำแหน่งระเบียบข้อมูลปัจจุบันไปยังตำแหน่งลำดับถัดไปตามการจัดเรียงของระเบียบข้อมูลที่ได้จากการเพิ่มเติม และไม่เกี่ยวข้องกับการใช้คีย์ระเบียบข้อมูลที่จะอ่านจากตำแหน่งนั้นจะส่งกลับมาใน data buffer จุดประสงค์ในการใช้ฟังก์ชันนี้ เป็นการอ่านระเบียบข้อมูลจากแฟ้มที่เกิดความเสียหายที่ละระเบียบ โดยอ่านจากแผ่นข้อมูลในแฟ้มข้อมูลดัชนีโดยตรง
2. ฟังก์ชันจะทำการอ่านระเบียบข้อมูลตามลำดับจากตำแหน่งของระเบียบปัจจุบัน ถ้าใช้ฟังก์ชันนี้หลังจากการเปิดแฟ้ม จะเริ่มต้นอ่านระเบียบข้อมูลจากระเบียบแรกของแผ่นข้อมูลแรกในแฟ้มข้อมูลดัชนี

3. เมื่อเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายสาเหตุความผิดพลาดที่เกิด

3.15 ฟังก์ชันการค้นหาและอ่านระเบียบข้อมูลจากหมายเลขระเบียบ

(Get record number)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 19
2. file information block
3. data buffer

การทำงาน

1. ทำการค้นหาตำแหน่งระเบียบข้อมูลจากหมายเลขระเบียบข้อมูลที่กำหนดใน data buffer ซึ่งเป็นข้อมูลเลขจำนวนเต็มขนาด 4 ไบต์ โดยต้องอาศัยแผ่นแผนที่ข้อมูลซึ่งกำหนดขึ้นในระหว่างการสร้างแฟ้มข้อมูลดัชนี และอ่านระเบียบข้อมูลส่งกลับมาใน data buffer ขนาดของบัพเฟอร์จะต้องมีขนาดที่เพียงพอสำหรับรับระเบียบข้อมูลที่ส่งกลับมา

2. เมื่อเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายสาเหตุความผิดพลาดที่เกิด

3.16 ฟังก์ชันการอ่านสถานะภาพของแฟ้มข้อมูลดัชนี

(Report B-Tree file status)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 20
2. file information block
3. data buffer สำหรับรับข้อมูลคุณสมบัติของแฟ้มและคีย์
4. key buffer สำหรับรับชื่อชุดการสลับลำดับตัวอักษร

การทำงาน

1. ฟังก์ชันจะอ่านสถานะภาพภายในแฟ้มข้อมูลดัชนีจากแฟ้มที่ระบุด้วย file information block

2. ข้อมูลที่เป็นคุณสมบัติของแฟ้มและคีย์จะส่งกลับมาใน data buffer ซึ่งมีรูปแบบเช่นเดียวกับการกำหนดคุณสมบัติในขณะสร้างแฟ้มข้อมูลดัชนี โดยประกอบด้วยกลุ่มข้อมูลที่เรียงติดกัน ไม่มีรหัสใด ๆ คั่นระหว่างข้อมูล ได้แก่

- กลุ่มข้อมูลคุณสมบัติแฟ้ม มีเนื้อหาที่ประกอบด้วย
 1. ความยาวระเบียบข้อมูล
 2. ขนาดแผ่น
 3. จำนวนชุดคีย์ในแฟ้ม
 4. จำนวนระเบียบข้อมูลและที่ว่างของระเบียบที่
 5. จำนวนที่ว่างของระเบียบข้อมูล
 6. จำนวนที่ว่างของแผ่น
 7. แพลกที่ระบุการสร้างแผ่นแผนที่ข้อมูล
- กลุ่มข้อมูลคุณสมบัติคีย์ มีเนื้อที่ประกอบด้วย
 1. ตำแหน่งเริ่มต้นในระเบียบข้อมูล
 2. ความยาวของคีย์
 3. แพลกที่ระบุคุณสมบัติของคีย์
 4. จำนวนคีย์ทั้งหมดในชุดคีย์

ถูกลบทิ้ง

3. เมื่อเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายสาเหตุความผิดพลาดที่เกิด

3.17 ฟังก์ชันกำหนดให้มีข้อความแสดงผลการทำงานทางจอภาพ

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 21
2. key number สำหรับระบุค่าแพลกที่กำหนดการทำงาน

การทำงาน

1. ทุกฟังก์ชันที่พัฒนาในโปรแกรมที่บรรจุประจำในหน่วยความจำ จะมีขั้นตอนที่แสดงผลการทำงานของแต่ละฟังก์ชันทางจอภาพเป็นระยะ ๆ การกำหนดให้ข้อความต่าง ๆ แสดงบนจอภาพได้ จะต้องกำหนดให้ key number มีค่าเท่ากับ 1 เท่านั้น

2. สำหรับฟังก์ชันนี้ไม่มีการส่งรหัสความผิดพลาดใด ๆ ที่เกิดจากการทำงานให้กับโปรแกรมประยุกต์ นอกจากโปรแกรมติดต่อจะตรวจพบว่าไม่มีการติดตั้งฟังก์ชันจัดการเพิ่มข้อมูลดัชนีไว้ในหน่วยความจำ จะส่งรหัสความผิดพลาดเท่ากับ 99 กลับมา

3.18 ฟังก์ชันการอ่านไดเรกทอรีปัจจุบันของแผ่นจานบันทึก

(Get the current directory)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 22
2. key buffer สำหรับรับไดเรกทอรีปัจจุบันที่อ่านได้
3. key number สำหรับระบุหน่วยขับจานบันทึก

การทำงาน

1. ทำการอ่านไดเรกทอรีปัจจุบันจากจานบันทึกที่ระบุไว้ใน key number โดยใช้หมายเลขในการระบุหน่วยขับจานบันทึก เช่น 1 แทน หน่วยขับจานบันทึก A 2 แทน หน่วยขับจานบันทึก B ตามลำดับ และ 0 แทน หน่วยขับจานบันทึกที่ใช้งานในขณะนั้น
2. ไดเรกทอรีที่อ่านได้ จะส่งกลับมาใน key buffer โดยมีขนาดเนื้อที่ไม่เกิน 64 ไบต์ และปิดท้ายข้อมูลใน key buffer ด้วยรหัสแอสกี 0
3. เมื่อเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับมาเพื่ออธิบายสาเหตุความผิดพลาดที่เกิด

3.19 ฟังก์ชันการเปลี่ยนไดเรกทอรีปัจจุบันของแผ่นจานบันทึก

(Set the current directory)

พารามิเตอร์ที่ใช้

1. operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 23
2. key buffer สำหรับกำหนดไดเรกทอรีที่ต้องการ

การทำงาน

1. ทำการเปลี่ยนไดเรกทอรีปัจจุบันของจานบันทึกที่ต้องการการทำงานจะคล้ายกับคำสั่ง CHDIR ที่ใช้กับเอ็มเอสดอส ไดเรกทอรีที่ต้องการ ให้กำหนดใน key buffer และต้องปิดท้ายด้วยรหัสแอสกี 0

2. เมื่อเกิดความผิดพลาดจากการทำงาน ฟังก์ชันจะส่งรหัส
ความผิดพลาดที่ไม่เท่ากับ ๐ กลับมาเพื่ออธิบายความผิดพลาดที่เกิด

3.2๐ ฟังก์ชันยกเลิกการติดตั้ง โปรแกรมที่ประจำในหน่วยความจำ
(Stop B-Tree function)

พารามิเตอร์ที่ใช้

operation number กำหนดหมายเลขฟังก์ชันเท่ากับ 24

การทำงาน

ยกเลิกการติดตั้ง โปรแกรมที่ประจำในหน่วยความจำ และการจอง
เนื้อที่บัฟเฟอร์สำหรับช่องใส่แผ่น พร้อมกับคืนเนื้อที่ทั้งหมดให้แก่ระบบเอมเอสดอล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

การพัฒนาโปรแกรมติดต่อ (Interface Routine)

โปรแกรมติดต่อ เป็นโปรแกรมการทำงานส่วนหนึ่ง ที่ต้องมีในโปรแกรมประยุกต์ ทำหน้าที่รับและส่งพารามิเตอร์จากโปรแกรมประยุกต์ ให้กับการทำงานของฟังก์ชันจัดการแฟ้มข้อมูลดัชนีที่ติดตั้งในหน่วยความจำ แต่เนื่องจากวิธีเรียกใช้ฟังก์ชันจัดการแฟ้มข้อมูลดัชนี ได้กำหนดให้ใช้คำสั่งการอินเทอร์รัปต์ โปรแกรมติดต่อจึงต้องเตรียมบล็อกข้อมูลเพื่อที่หนึ่งสำหรับบรรจุข้อมูลพารามิเตอร์เหล่านั้นก่อนการอินเทอร์รัปต์ ฟังก์ชันจัดการแฟ้มข้อมูลดัชนีสามารถเข้าถึงข้อมูลต่าง ๆ ได้จากการอ้างอิงตำแหน่งของบล็อกข้อมูลนั้นด้วยรีจิสเตอร์ DS:DX ซึ่งระบุตำแหน่งเซกเมนต์และออฟเซตของบล็อกข้อมูล ภายในบล็อกข้อมูลจะประกอบด้วยลำดับข้อมูลดังนี้

1. ตำแหน่งอ้างอิงของบัพเฟอร์สำหรับระเบียบข้อมูล ซึ่งระบุในลักษณะฟาร์พอยต์เตอร์ (far pointer) คือ ประกอบด้วย ตำแหน่งออฟเซตของบัพเฟอร์ และ ตำแหน่งเซกเมนต์ของบัพเฟอร์
2. ตำแหน่งอ้างอิงในลักษณะฟาร์พอยต์เตอร์ ที่ระบุตำแหน่งของบล็อกข้อมูลบอกตำแหน่ง ซึ่งอยู่ภายในบัพเฟอร์สำหรับบล็อกสารสนเทศแฟ้ม โดยเริ่มจากตำแหน่งไบต์ที่ 38 มีความยาวของบล็อกเท่ากับ 90 ไบต์
3. ตำแหน่งอ้างอิงในลักษณะฟาร์พอยต์เตอร์ ที่ระบุตำแหน่งของเอฟซีบี (FCB) ซึ่งอยู่ภายในบัพเฟอร์สำหรับบล็อกสารสนเทศแฟ้ม โดยเริ่มจากตำแหน่งไบต์ที่ 0 มีความยาวของบล็อกเท่ากับ 38 ไบต์
4. หมายเลขฟังก์ชันของฟังก์ชันจัดการแฟ้มข้อมูลดัชนี
5. ตำแหน่งอ้างอิงในลักษณะฟาร์พอยต์เตอร์ ที่ระบุตำแหน่งของบัพเฟอร์สำหรับคีย์
6. หมายเลขชุดคีย์ที่ใช้ในการค้นหาคีย์
7. ตำแหน่งอ้างอิงในลักษณะฟาร์พอยต์เตอร์ ที่ระบุตำแหน่งของตัวแปรที่กำหนดภายในโปรแกรมติดต่อ เพื่อใช้รับรหัสความผิดพลาดที่ส่งมาจากการทำงานของฟังก์ชัน

หลักการการทำงานที่ต้องเตรียมในโปรแกรมติดต่อ ประกอบด้วยขั้นตอนดังนี้

1. การตรวจสอบการติดตั้งฟังก์ชันจัดการแฟ้มข้อมูลดัชนี ที่ประจำในหน่วยความจำก่อนการใช้คำสั่งอินเทอร์รัปต์ 7Ch โดยทดสอบจากรหัสควบคุมที่มีขนาด 2 ไบต์ ซึ่งอยู่ใน

ตำแหน่งก่อนจุดเริ่มต้นการทำงานของอินเทอร์รัปต์ 7Ch รหัสความผิดนี้ต้องมีค่าเท่ากับ 0ABCDh

2. บรรจุข้อมูลของพารามิเตอร์ต่าง ๆ ลงในบล็อกข้อมูลที่เตรียมไว้ในโปรแกรมติดต่อ และใช้รีจิสเตอร์ DS ระบุดำแหน่งเซกเมนต์และ DX ระบุดำแหน่งออฟเซตของบล็อกข้อมูลนั้น

3. ส่งการทำงานของโปรแกรมให้กับฟังก์ชันจัดการเพิ่มข้อมูลดัชนี โดยการให้คำสั่งอินเทอร์รัปต์ 7Ch

4. เมื่อจบการทำงานของฟังก์ชันที่ถูกเรียกใช้ ฟังก์ชันจะส่งรหัสความผิดพลาดมาที่ตัวแปรรับค่ารหัสความผิดพลาด เพื่อโปรแกรมติดต่อยจะส่งให้กับโปรแกรมประยุกต์ที่เรียกใช้ต่อไป รหัสความผิดพลาดจะเท่ากับ 0 ทุกครั้งที่การทำงานของฟังก์ชันเป็นไปอย่างถูกต้องภายในโปรแกรมประยุกต์ จึงควรตรวจสอบรหัสความผิดพลาดทุกครั้งหลังจากการเรียกใช้ฟังก์ชันจัดการเพิ่มข้อมูลดัชนี

แต่เนื่องจาก โดยทั่วไปการส่งผ่านพารามิเตอร์ให้กับโปรแกรมที่ถูกเรียกใช้ (called program) เป็นการส่งผ่านข้อมูลของพารามิเตอร์ทางสแตคของโปรแกรม ลักษณะการส่งผ่านข้อมูล และการจัดเรียงลำดับข้อมูลของพารามิเตอร์แต่ละตัวภายในสแตค จะถูกกำหนดจากตัวแปลภาษา แต่ละภาษามีลักษณะของการกำหนดที่ไม่เหมือนกัน ในที่นี้จึงได้พัฒนาโปรแกรมติดต่อที่ใช้สำหรับโปรแกรมภาษาไมโครซอฟท์ปาสคาล ภาษาไมโครซอฟท์ฟอร์แทรน และภาษาไมโครซอฟท์ซี เท่านั้น

โปรแกรมติดต่อสำหรับโปรแกรมไมโครซอฟท์ปาสคาล (PASXBTR routine)

การพัฒนาโปรแกรมติดต่อสำหรับโปรแกรมไมโครซอฟท์ปาสคาล กำหนดให้มีรูปแบบการเรียกใช้ในลักษณะแบบฟังก์ชัน (function type) ของภาษาปาสคาล ชื่อฟังก์ชันของโปรแกรมคือ XBTRREE ซึ่งเป็นฟังก์ชันที่ส่งผลกลับให้โปรแกรมที่เรียกใช้ (calling program) เป็นข้อมูลเลขจำนวนเต็มมีขนาด 2 ไบต์ ข้อมูลที่ส่งกลับมาก็คือ รหัสความผิดพลาด (error code) เนื่องจากในขั้นตอนการใช้คำสั่งอินเทอร์รัปต์ 7Ch ของโปรแกรมติดต่อ ภาษาไมโครซอฟท์ปาสคาลไม่ได้เตรียมไลบรารีฟังก์ชันที่ใช้สำหรับการอินเทอร์รัปต์ให้กับผู้เขียนโปรแกรม การพัฒนาโปรแกรมติดต่อให้กับโปรแกรมไมโครซอฟท์ปาสคาลจึง

จำเป็นต้องใช้โปรแกรมแอสเซมบลี(assembly program) ส่วนของโปรแกรมติดต่อกัน
 ผ่านการแปลจะนำมาเชื่อมโยงกับโปรแกรมปาสคาลในภายหลัง นอกจากนี้ภายในโปรแกรม
 ปาสคาลต้องมีการกำหนดรูปแบบฟังก์ชัน XBTRREE ในลักษณะฟังก์ชันภายนอก(external
 function) โดยการกำกับ extern ไว้ตอนท้ายของฟังก์ชัน

```
function XBTRREE ( OPERATION : integer;
                  vars FIL_BLK : string;
                  vars DAT_BUF : string;
                  vars KEY_BUF : string;
                  KEY_NUM : integer ) : integer; extern;
```

OPERATION เป็นการกำหนดฟังก์ชันจัดการแฟ้มข้อมูลดัชนีที่ต้องถูกเรียกใช้
 มีทั้งหมด 24 ฟังก์ชัน กำหนดเป็นหมายเลขของฟังก์ชันที่ต้องการ ข้อมูลที่ส่งมาทางสแตก
 โดยโปรแกรมไมโครซอฟท์ปาสคาล จะเป็นข้อมูลหมายเลขฟังก์ชันที่ต้องการ ใช้เนื้อที่ภายใน
 สแตกเท่ากับ 2 ไบต์

FIL_BLK เป็นเนื้อที่บัฟเฟอร์สำหรับบล็อกสารสนเทศแฟ้มของแฟ้มข้อมูลดัชนี
 การกำหนดเนื้อที่บล็อกสารสนเทศแฟ้มภายในโปรแกรมปาสคาล จะต้องกำหนดในรูปแบบ
 string ซึ่งมีขนาดเท่ากับ 128 ไบต์

```
var FIL_BLK : string(128);
```

ข้อมูลที่ส่งมาทางสแตกโดยโปรแกรมปาสคาล จะเป็นข้อมูลตำแหน่งอ้างอิงในลักษณะ
 ฟาร์พอยต์เตอร์ที่ระบุตำแหน่งของบล็อกสารสนเทศแฟ้ม ใช้เนื้อที่ภายในสแตกเท่ากับ 4 ไบต์

DAT_BUF เป็นเนื้อที่บัฟเฟอร์สำหรับระเบียบข้อมูล ที่ใช้รับและส่งในระหว่างการทำงาน
 ของฟังก์ชันจัดการแฟ้มข้อมูลดัชนี เนื้อที่บัฟเฟอร์สำหรับระเบียบข้อมูลที่กำหนดใน
 โปรแกรมปาสคาล จะต้องกำหนดในรูปแบบ string ซึ่งมีขนาดเท่ากับระเบียบข้อมูลที่
 ข้อมูลที่ส่งมาทางสแตกโดยโปรแกรมปาสคาล จะเป็นข้อมูลตำแหน่งอ้างอิงในลักษณะฟาร์พอยต์เตอร์

ที่ระบุตำแหน่งบัพเฟอร์นั้น ใช้เนื้อที่ภายในสแตกเท่ากับ 4 ไบต์ แต่บางครั้งการทำงานจำเป็นต้องใช้บัพเฟอร์ในลักษณะโครงสร้างระเบียบ (record structure) การกำหนดบัพเฟอร์สำหรับระเบียบข้อมูล จึงได้กำหนดในรูปแบบ variant record ตามลักษณะภาษาปาสคาล โดยตัวแปรของบัพเฟอร์สำหรับพารามิเตอร์นี้ จะต้องเป็นตัวแปรของโครงสร้างระเบียบที่กำหนดในรูปแบบ string มีขนาดเท่ากับขนาดของโครงสร้างระเบียบที่ใหญ่ที่สุด

```

type DATA_FIELD = record
    case integer of
        1 : (Field1 : string(4);
            Field2 : string(30);
            Field3 : string(6));
        2 : (BUFFER : string(40));
    end;
var DATA : DATA_FIELD;

```



DATA.BUFFER จะเป็นตัวแปรที่ส่งมาสำหรับพารามิเตอร์ DAT_BUF

KEY_BUF เป็นเนื้อที่บัพเฟอร์สำหรับคีย์ ที่ใช้รับและส่งในระหว่างการทำงานของฟังก์ชันจัดการแฟ้มข้อมูลคีย์นี้ การกำหนดเนื้อที่บัพเฟอร์สำหรับคีย์ภายในโปรแกรมภาษาปาสคาลจะต้องกำหนดในรูปแบบ string ซึ่งมีขนาดเท่ากับความยาวคีย์ที่ใช้ ข้อมูลที่ส่งมาทางสแตกโดยโปรแกรมปาสคาล จะเป็นข้อมูลตำแหน่งอ้างอิงในลักษณะพาร์พอนต์เตอร์ที่ระบุตำแหน่งบัพเฟอร์นั้น ใช้เนื้อที่ภายในสแตกเท่ากับ 4 ไบต์ ในกรณีที่คีย์ประกอบด้วยหลายเชกเมนต์คีย์ การกำหนดบัพเฟอร์สำหรับคีย์ได้กำหนดในรูปแบบ variant record เช่นเดียวกับ DAT_BUF

KEY_NUM เป็นการกำหนดหมายเลขชุดคีย์ที่ใช้ในการค้นหาคีย์ หมายเลขชุดคีย์กำหนดได้ตั้งแต่ 0 ถึง 23 ขึ้นกับจำนวนชุดคีย์ที่กำหนดในแฟ้ม ข้อมูลที่ส่งมาทางสแตกภายในโปรแกรมปาสคาล จะเป็นข้อมูลหมายเลขชุดคีย์ ใช้เนื้อที่ภายในสแตกเท่ากับ 2 ไบต์

การกำหนดรูปแบบ vars สำหรับพารามิเตอร์ภายในฟังก์ชัน XBTRREE จะเป็นการระบุให้ตัวแปลภาษาไมโครซอฟท์ปาสคาล จัดเตรียมตำแหน่งอ้างอิงแบบพาร์พอยต์เตอร์สำหรับพารามิเตอร์เหล่านั้นส่งมาทางสแตกภายในโปรแกรม แต่เนื่องจากการกำหนดข้อมูลในรูปแบบ string จะทำให้ตัวแปลภาษาต้องเตรียมเนื้อที่ภายในสแตกขนาด 2 ไบต์ สำหรับบันทึกขนาดของแต่ละพารามิเตอร์

โปรแกรมติดต่อสำหรับโปรแกรมไมโครซอฟท์ฟอร์แทรน (FORXBTR routine)

การพัฒนาโปรแกรมติดต่อสำหรับโปรแกรมไมโครซอฟท์ฟอร์แทรน กำหนดให้มีการเรียกใช้ ในลักษณะฟังก์ชัน มีชื่อของฟังก์ชันคือ XBTRREE ซึ่งส่งผลกลับมาให้โปรแกรมที่เรียกใช้ เป็นข้อมูลเลขจำนวนเต็มขนาด 2 ไบต์ ข้อมูลที่ส่งกลับมาก็คือ รหัสความผิดพลาดของการทำงาน การพัฒนาโปรแกรมติดต่อให้กับโปรแกรมไมโครซอฟท์ฟอร์แทรน ต้องพัฒนาเป็นโปรแกรมแอสเซมบลีเช่นเดียวกับโปรแกรมปาสคาล โดยมีรูปแบบการเรียกใช้ดังนี้

```
INTEGER * 2 STAT, OPERATION, KEYNUM
```

```
INTEGER * 2 XBTRREE
```

```
STAT = XBTRREE(OPERATION, FILBLK, DATBUF, KEYBUF, KEYNUM)
```

OPERATION เป็นการกำหนดฟังก์ชันจัดการเพิ่มข้อมูลดัชนี ที่ต้องถูกเรียกใช้ มีทั้งหมด 24 ฟังก์ชัน กำหนดเป็นหมายเลขของฟังก์ชันที่ต้องการ ข้อมูลที่ส่งมาทางสแตก โดยโปรแกรมไมโครซอฟท์ฟอร์แทรน จะเป็นข้อมูลตำแหน่งอ้างอิงในลักษณะพาร์พอยต์เตอร์ ที่ระบุตำแหน่งตัวแปรเลขจำนวนเต็มขนาด 2 ไบต์ ซึ่งเก็บค่าหมายเลขฟังก์ชัน หรือระบุตำแหน่งของเนื้อที่ชั่วคราวที่เก็บค่าคงที่ของหมายเลขฟังก์ชันที่ต้องการ ใช้เนื้อที่ภายในสแตกเท่ากับ 4 ไบต์

FILBLK เป็นเนื้อที่บัฟเฟอร์สำหรับบล็อกสารสนเทศเพิ่มของเพิ่มข้อมูลดัชนี การกำหนดเนื้อที่บล็อกสารสนเทศเพิ่มภายในโปรแกรม จะเป็นการกำหนดในรูปแบบแถวลำดับของข้อมูลเลขจำนวนเต็มขนาด 2 ไบต์ (array of integer) ที่มีจำนวนสมาชิกเท่ากับ 64

INTEGER * 2 FILBLK(64)

ข้อมูลที่ส่งมาทางสแตกภายในโปรแกรมฟอร์แทรน จะเป็นข้อมูลตำแหน่งอ้างอิงในลักษณะ ฟาร์พอยต์เตอร์ที่ระบุตำแหน่งบล็อกสารสนเทศเพิ่ม ใช้เนื้อที่ภายในสแตกเท่ากับ 4 ไบต์

DATBUF เป็นเนื้อที่บัฟเฟอร์สำหรับระเบียบข้อมูล ที่ใช้รับและส่งในระหว่างการทำงานของฟังก์ชันจัดการเพิ่มข้อมูลดัชนี การกำหนดเนื้อที่บัฟเฟอร์สำหรับระเบียบข้อมูลภายในโปรแกรมฟอร์แทรน จะกำหนดในรูปแบบแถวลำดับของข้อมูลตัวอักษร (array of character) ซึ่งมีขนาดเท่ากับขนาดระเบียบข้อมูลที่ใช้ แต่ถ้าระเบียบข้อมูลมีขนาดมากกว่า 127 ไบต์ หรือ อาจต้องการกำหนดในลักษณะโครงสร้างระเบียบ ให้ใช้ EQUIVALENCE ช่วยในการกำหนดลักษณะของข้อมูล โดยกำหนดตัวแปรที่ใช้เนื้อที่ร่วมกับบัฟเฟอร์ระเบียบข้อมูลให้มีผลรวมของขนาดเท่ากับ ผลรวมของขนาดเขตข้อมูล(field)ทั้งหมดที่ใช้สำหรับระเบียบข้อมูล ซึ่งอาจเป็นตัวแปรแถวลำดับของข้อมูลตัวอักษรที่มีขนาดไม่เกิน 127 ไบต์ หรือ อาจใช้ตัวแปรแถวลำดับของข้อมูลเลขจำนวนเต็มแทน เพราะสามารถกำหนดขนาดได้มากกว่า 127 ไบต์

CHARACTER DATBUF(40)

CHARACTER FIELD1 * 4, FIELD2 * 30, FIELD * 6

EQUIVALENCE (FIELD1, DATBUF(1)), (FIELD2, DATBUF(5))

EQUIVALENCE (FIELD3, DATBUF(35))

ข้อมูลที่ส่งมาทางสแตกโดยโปรแกรมฟอร์แทรน จะเป็นข้อมูลตำแหน่งอ้างอิงในลักษณะ ฟาร์พอยต์เตอร์ที่ระบุตำแหน่งบัฟเฟอร์นั้น ใช้เนื้อที่ภายในสแตกเท่ากับ 4 ไบต์

KEYBUF เป็นเนื้อที่บัฟเฟอร์สำหรับคีย์ การกำหนดเนื้อที่บัฟเฟอร์สำหรับคีย์ภายในโปรแกรมภาษาฟอร์แทรน จะต้องกำหนดในรูปแบบแถวลำดับ ซึ่งมีขนาดเท่ากับความยาวคีย์ที่ใช้ สำหรับคีย์ประกอบด้วยหลายเซกเมนต์คีย์ ให้กำหนดเนื้อที่บัฟเฟอร์สำหรับคีย์โดยการ ใช้ EQUIVALENCE เช่นเดียวกับ DATBUF ข้อมูลที่ส่งมาทางสแตกโดยโปรแกรมฟอร์แทรน

จะเป็นข้อมูลตำแหน่งอ้างอิงในลักษณะพาร์พอยต์เตอร์ที่ระบุตำแหน่งบัพเพอร์นั้น ใช้เนื้อที่ภายในสแตกเท่ากับ 4 ไบต์

KEYNUM เป็นการกำหนดหมายเลขชุดคีย์ที่ใช้ในการค้นหาคีย์ หมายเลขชุดคีย์กำหนดได้ตั้งแต่ 0 ถึง 23 ขึ้นกับจำนวนชุดคีย์ที่กำหนดในแฟ้ม ข้อมูลที่ส่งมาทางสแตกภายในโปรแกรมฟอร์แทรน จะเป็นข้อมูลตำแหน่งอ้างอิงในลักษณะพาร์พอยต์เตอร์ที่ระบุตำแหน่งตัวแปรเลขจำนวนเต็มขนาด 2 ไบต์ ซึ่งเก็บค่าหมายเลขชุดคีย์ หรือระบุตำแหน่งของเนื้อที่ชั่วคราวที่เก็บค่าคงที่ของหมายเลขชุดคีย์ ใช้เนื้อที่ภายในสแตกเท่ากับ 4 ไบต์

ในการส่งพารามิเตอร์ภายในโปรแกรมไมโครซอฟท์ฟอร์แทรน ข้อมูลที่ส่งมาทางสแตกของทุกพารามิเตอร์ จะถูกกำหนดให้มีรูปแบบเดียวกัน คือ เป็นข้อมูลตำแหน่งอ้างอิงในลักษณะพาร์พอยต์เตอร์

โปรแกรมติดต่อสำหรับโปรแกรมไมโครซอฟท์ซี (CXBT routine)

การพัฒนาโปรแกรมติดต่อสำหรับโปรแกรมไมโครซอฟท์ซี กำหนดให้มีการเรียกใช้ในลักษณะฟังก์ชัน มีชื่อของฟังก์ชันคือ XBTRREE ซึ่งส่งผลกลับมาให้โปรแกรมที่เรียกใช้ เป็นข้อมูลเลขจำนวนเต็มขนาด 2 ไบต์ ข้อมูลที่ส่งกลับมาก็คือ รหัสความผิดพลาดของการทำงาน การพัฒนาโปรแกรมติดต่อให้กับโปรแกรมไมโครซอฟท์ซี สามารถพัฒนาได้ทั้งโปรแกรมแอสเซมบลีและโปรแกรมไมโครซอฟท์ซี เพราะภาษาไมโครซอฟท์ซีได้เตรียมไลบรารีฟังก์ชันสำหรับการใช้คำสั่งอินเทอร์พรีตให้กับผู้เขียนโปรแกรม ลักษณะการกำหนดพารามิเตอร์ของฟังก์ชัน XBTRREE เป็นดังนี้

```
int XBTRREE(OPERATION, FIL_BLK, DAT_BUF, KEY_BUF, KEY_NUM)
    int OPERATRION;
    char * FIL_BLK;
    char * DAT_BUF;
    char * KEY_BUF;
    int KEY_NUM;
```


OPERATION เป็นการกำหนดฟังก์ชันจัดการเพิ่มข้อมูลดัชนี ที่ต้องถูกเรียกใช้ มีทั้งหมด 24 ฟังก์ชัน ข้อมูลที่ส่งมาทางสแตกโดยโปรแกรมไมโครซอฟท์ซี จะเป็นข้อมูล หมายเลขของฟังก์ชันที่ต้องการ ใช้เนื้อที่ภายในสแตกเท่ากับ 2 ไบต์

FIL_BLK เป็นตำแหน่งอ้างอิงที่ระบุตำแหน่งบัพเฟอร์สำหรับบล็อกสารสนเทศเพิ่มเติม การกำหนดเนื้อที่บล็อกสารสนเทศเพิ่มภายในโปรแกรมซี จะกำหนดในรูปแบบแถวลำดับของ ข้อมูลตัวอักษรที่มีขนาด 128 ไบต์

```
char FIL_BLK[128];
```

ข้อมูลที่ส่งมาทางสแตกโดยโปรแกรมซี จะขึ้นกับการกำหนดโมเดลของโปรแกรมที่ผู้เขียนเลือกใช้ ซึ่งลักษณะโมเดลของโปรแกรมไมโครซอฟท์ซีกำหนดได้ 5 โมเดล ได้แก่ small compact medium large และ huge ทั้ง 5 โมเดลมีการส่งตำแหน่งอ้างอิงที่ต่างกัน สำหรับโปรแกรมซีที่ใช้โมเดล small หรือ medium ตำแหน่งอ้างอิงที่ระบุตำแหน่งของบล็อกสารสนเทศเพิ่ม จะเป็นการอ้างอิงในลักษณะเนียร์พอยต์เตอร์ (near pointer) ซึ่งมีเพียงตำแหน่งออฟเซตเท่านั้น และใช้ค่าภายในรีจิสเตอร์ DS ระบุตำแหน่งเซกเมนต์ เนื้อที่ภายในสแตกจึงใช้เพียง 2 ไบต์ สำหรับโปรแกรมซีที่ใช้โมเดล compact หรือ large หรือ huge ตำแหน่งอ้างอิงที่ระบุตำแหน่งบล็อกสารสนเทศเพิ่ม จะเป็นการอ้างอิงในลักษณะฟาร์พอยต์เตอร์ ซึ่งใช้เนื้อที่ภายในสแตกเท่ากับ 4 ไบต์

DAT_BUF เป็นตำแหน่งอ้างอิงที่ระบุตำแหน่งบัพเฟอร์สำหรับระเบียบข้อมูล การกำหนดเนื้อที่บัพเฟอร์ระเบียบข้อมูลภายในโปรแกรมซี จะกำหนดในรูปแบบแถวลำดับซึ่งมีขนาดเท่ากับขนาดระเบียบข้อมูลที่ใช้ หรือกำหนดในรูปแบบ struct สำหรับระเบียบข้อมูลที่มีโครงสร้างระเบียบประกอบด้วยหลายเขตข้อมูล หรืออาจใช้ union สำหรับระเบียบข้อมูลต้องใช้โครงสร้างระเบียบมากกว่าหนึ่งชนิดขึ้นไป ลักษณะของข้อมูลที่ส่งมาทางสแตกโดยโปรแกรมซี จะเป็นลักษณะเดียวกับ FIL_BLK

KEY_BUF เป็นตำแหน่งอ้างอิงที่ระบุตำแหน่งบัพเฟอร์สำหรับคีย์ การกำหนดเนื้อที่บัพเฟอร์ของคีย์ภายในโปรแกรมซี จะกำหนดในรูปแบบแถวลำดับ ซึ่งมีขนาดเท่ากับความ

ยวาคีย์ที่ใช้ หรือกำหนดในรูปแบบ struct สำหรับคีย์ที่มีโครงสร้างระเบียบ ประกอบด้วยหลายเซกเมนต์คีย์ ลักษณะของข้อมูลที่ส่งมาทางสแตกโดยโปรแกรมซี จะเป็นลักษณะเดียวกับ FIL_BLK

KEY_NUM เป็นการกำหนดหมายเลขชุดคีย์ที่ใช้ในการค้นหาคีย์ หมายเลขชุดคีย์กำหนดได้ตั้งแต่ 0 ถึง 23 ขึ้นกับจำนวนชุดคีย์ที่กำหนดในแฟ้ม ข้อมูลที่ส่งมาทางสแตกภายในโปรแกรมซี จะเป็นข้อมูลหมายเลขชุดคีย์ ใช้เนื้อที่ภายในสแตกเท่ากับ 2 ไบต์

สำหรับโปรแกรมติดต่อกับพัฒนาด้วยภาษาไมโครซอฟท์ซี ได้ออกแบบให้โปรแกรมสามารถกำหนดชนิดของตำแหน่งอ้างอิงที่ส่งมาโดยพารามิเตอร์ FIL_BLK, DAT_BUF และ KEY_BUF จากการกำหนดค่าให้กับ FAR_PTR ภายในโปรแกรมติดต่อ

```
#define FAR_PTR 1 /* use far pointer */
```

เป็นการกำหนดโปรแกรมติดต่อให้รับตำแหน่งอ้างอิงในลักษณะฟาร์พอยน์เตอร์ และ

```
#define FAR_PTR 0 /* use near pointer */
```

เป็นการกำหนดโปรแกรมติดต่อให้รับตำแหน่งอ้างอิงในลักษณะเนียร์พอยน์เตอร์

สำหรับโปรแกรมติดต่อกับพัฒนาด้วยภาษาแอสเซมบลี ได้ออกแบบให้โปรแกรมสามารถกำหนดการรับตำแหน่งอ้างอิงในลักษณะเช่นเดียวกับโปรแกรมติดต่อไมโครซอฟท์ซี คือเลือกได้จากการกำหนดค่า FAR_PTR

จุฬาลงกรณ์มหาวิทยาลัย

การพัฒนาโปรแกรมอรรถประโยชน์สำหรับจัดการแฟ้มข้อมูล

ในการวิจัยนี้ได้ทำการพัฒนาโปรแกรมอรรถประโยชน์ เพื่อให้ความสะดวกต่อผู้ใช้ สำหรับการสร้างแฟ้มข้อมูลดัชนี และการทำงานต่าง ๆ ที่ใช้บำรุงรักษาแฟ้มข้อมูลดัชนี (File maintenance) โดยการทำงานของโปรแกรมจะเรียกใช้ฟังก์ชันจัดการแฟ้มข้อมูล ซึ่งต้องบรรจุประจำในหน่วยความจำขณะนั้น

โปรแกรมเมนูหลัก (utilB program)

โปรแกรมเมนูหลักเป็นโปรแกรมที่ควบคุมการเลือกใช้การทำงานต่าง ๆ ที่กำหนดขึ้นในโปรแกรม โดยแสดงหัวข้อการทำงานทั้งหมดทางจอภาพให้ผู้ใช้ได้เลือกใช้ หัวข้อการทำงานเหล่านั้นประกอบด้วย

1. การสร้างแฟ้มข้อมูลดัชนี (Create B-Tree File)
2. การทำสำเนาแฟ้มข้อมูลดัชนี (Copy B-Tree File)
3. การบรรจุข้อมูลลงแฟ้มข้อมูลดัชนี (Load B-Tree File)
4. การเก็บรักษาข้อมูลจากแฟ้มข้อมูลดัชนี (Save B-Tree File)
5. การกู้แฟ้มข้อมูลดัชนีที่เสียหาย (Recover B-Tree File)
6. การรายงานสถานการณ์แฟ้มข้อมูลดัชนี (Report Status of B-Tree File)
7. การยกเลิกฟังก์ชันจัดการแฟ้มข้อมูลดัชนี (Stop B-Tree File Manager)

การทำงานของโปรแกรมเมนูหลัก

1. ผู้ใช้สามารถเลือกการทำงานต่าง ๆ จากข้อความที่ขีดเส้นใต้บนจอภาพ ในขณะนั้นโดยการกดปุ่ม ENTER ทางแป้นพิมพ์ โปรแกรมจะเริ่มการทำงานตามหัวข้อที่กำหนด
2. ผู้ใช้สามารถเปลี่ยนหัวข้อการทำงานได้ จากการกดปุ่มลูกศรขึ้น (Up Arrow) หรือปุ่มลูกศรลง (Down Arrow) เพื่อเป็นการเลื่อนการทำงานในหัวข้อถัดไป และกดปุ่ม ENTER ให้โปรแกรมทำงาน หรือเลือกหัวข้อการทำงานได้โดยตรงจากหมายเลขของหัวข้อการทำงาน
3. ผู้ใช้สามารถจบการทำงานของโปรแกรมเมนูหลักได้ทันทีจากกดปุ่ม ESC

4. ก่อนเริ่มต้นการทำงานใด ๆ จากโปรแกรมเมนูหลัก โปรแกรมจะตรวจสอบการติดตั้งฟังก์ชันจัดการแฟ้มข้อมูลดัชนีในหน่วยความจำ ถ้าหากไม่มีการติดตั้งฟังก์ชันในหน่วยความจำ โปรแกรมจะแสดงข้อความบนจอภาพเพื่อเตือนผู้ใช้ทันที

โปรแกรมสร้างแฟ้มข้อมูลดัชนี (creatB program)

ก่อนการบรรจุข้อมูลในแฟ้มข้อมูลดัชนี ผู้ใช้ต้องสร้างแฟ้มข้อมูลดัชนีนั้นจากคุณสมบัติต่าง ๆ ของแฟ้มที่ได้กล่าวมาแล้ว โดยการทำงานของโปรแกรมได้เตรียมการแสดงข้อความทางจอภาพเพื่อช่วยให้ผู้ใช้เข้าใจถึงข้อมูลที่ต้องป้อนในขณะนั้น

การทำงานของ โปรแกรมสร้างแฟ้มข้อมูลดัชนี

1. โปรแกรมได้แบ่งการป้อนข้อมูลออกเป็น 2 กลุ่ม คือ ข้อมูลที่เป็นคุณสมบัติของแฟ้มและคุณสมบัติของชุดคีย์
2. ในการป้อนคุณสมบัติของแฟ้ม ประกอบด้วย
 - 2.1 ชื่อแฟ้มข้อมูลดัชนี ระบุได้ในรูปแบบ
drive:\path\file name.extension
 - 2.2 ขนาดแผ่นภายในแฟ้ม
 - 2.3 ความยาวระเบียบข้อมูล
 - 2.4 จำนวนชุดคีย์ในแฟ้ม
 - 2.5 การระบุให้สร้างแผ่นแผนที่ข้อมูลในแฟ้ม
3. ในการป้อนคุณสมบัติของชุดคีย์ จะเริ่มต้นจากชุดคีย์ที่ ๑ จนกระทั่งครบตามจำนวนที่กำหนด โดยแยกชุดคีย์ และแต่ละเซกเมนต์ของคีย์ด้วยการกำหนดคุณสมบัติการเป็นเซกเมนต์คีย์ ข้อมูลที่ใช้สำหรับคุณสมบัติของคีย์ ประกอบด้วย
 - 3.1 ตำแหน่งเริ่มต้นภายในระเบียบข้อมูล
 - 3.2 ความยาวของคีย์
 - 3.4 ชนิดของคีย์ได้แก่ คีย์ที่ประกอบด้วยกลุ่มตัวอักษร หรือเป็นคีย์ของเลขจำนวนเต็ม
 - 3.5 การระบุคุณสมบัติการบันทึกคีย์ที่มีค่าซ้ำกันได้
 - 3.6 การระบุคุณสมบัติการใช้คีย์ที่สามารถเปลี่ยนแปลงค่าได้

3.7 การระบุคุณสมบัติการใช้คีย์ ที่ต้องมีชุดการสลับลำดับตัวอักษรเพื่อช่วยในการเปรียบเทียบ

3.8 การระบุคุณสมบัติการใช้คีย์ที่เป็นตัวอักษรภาษาไทย

3.9 การระบุคุณสมบัติการเป็นเซกเมนต์คีย์

4. เมื่อจบการบอคุณสมบัติชุดคีย์ และตรวจพบว่า มีการกำหนดให้ชุดการสลับลำดับตัวอักษรกับคีย์ชุดใดก็ตาม จะกำหนดให้ผู้ใช้บอชื่อเพิ่มชุดการสลับลำดับตัวอักษร โดยระบุได้ในรูปแบบ `drive:\path\file name.extension`

5. เมื่อโปรแกรมสร้างแฟ้มข้อมูลดัชนี และเกิดความผิดพลาดจากการทำงาน โปรแกรมจะแสดงรหัสความผิดพลาด และข้อความที่อธิบายความผิดพลาดการทำงานนั้นทางจอภาพ

โปรแกรมทำสำเนาแฟ้มข้อมูลดัชนี (copyB program)

เป็นโปรแกรมทำสำเนาระเบียงข้อมูลจากแฟ้มข้อมูลหนึ่ง ไปอีกแฟ้มหนึ่ง เพื่อใช้ในกรณีที่ต้องการเปลี่ยนแปลงคุณสมบัติชุดคีย์ที่กำหนดในแฟ้มเดิม เช่น ความยาวของคีย์แต่ละเซกเมนต์ ตำแหน่งเริ่มต้นในระเบียบข้อมูล และคุณสมบัติต่าง ๆ ของคีย์ ให้เป็นลักษณะที่ต้องการในแฟ้มใหม่ โดยการอ่านระเบียบข้อมูลที่ละระเบียบจากแฟ้มต้นฉบับ และเพิ่มเติมในแฟ้มสำเนา

การทำงานของโปรแกรมทำสำเนาแฟ้มข้อมูลดัชนี

1. บอชื่อแฟ้มข้อมูลดัชนี ได้แก่ แฟ้มต้นฉบับ และ แฟ้มสำเนา ในรูปแบบ `drive:\path\file name.extension`
2. เมื่อโปรแกรมตรวจสอบไม่พบแฟ้มสำเนาที่ระบุ จะสอบถามความต้องการสร้างแฟ้มข้อมูลดัชนีใหม่จากผู้ใช้ ถ้าผู้ใช้ต้องการสร้างแฟ้มข้อมูล โปรแกรมจะส่งการทำงานให้กับโปรแกรมสร้างแฟ้มข้อมูลดัชนีแล้วจึงกลับมาที่การทำงานเดิม ถ้าผู้ใช้ไม่ต้องการ โปรแกรมจะยกเลิกการทำงาน และกลับไปที่การทำงานของโปรแกรมเมนูหลัก
3. เมื่อเริ่มทำสำเนาระเบียงข้อมูล การเปลี่ยนแปลงของจำนวนระเบียบข้อมูลที่ทำสำเนาจะแสดงบนจอภาพด้านล่าง
4. ผู้ใช้สามารถยกเลิกการทำสำเนาได้ทันทีจากการกดปุ่ม ESC และจำนวนระเบียบข้อมูลที่ทำสำเนาทั้งหมดจะแสดงบนจอภาพด้านล่าง หลังจากจบการทำงาน หรือถูกยกเลิก โปรแกรมจะกลับไปที่การทำงานของโปรแกรมเมนูหลัก



5. เมื่อเกิดความผิดพลาดจากการทำงาน โปรแกรมจะแสดงรหัสความผิดพลาดและข้อความที่อธิบายความผิดพลาดนั้นทางจอภาพด้านล่าง

โปรแกรมบรรจุข้อมูลลงแฟ้มข้อมูลดัชนี (loadB program)

เป็นโปรแกรมที่จัดเตรียมให้ผู้ใช้ทำการเพิ่มเติมข้อมูลลงในแฟ้มข้อมูลดัชนีได้จากแฟ้มข้อมูลเรียงลำดับ (sequential file) ซึ่งสามารถสร้างแฟ้มข้อมูลนี้ได้จากโปรแกรมบรรณาธิกรณ (text editor) ทั่วไป และเป็นรูปแบบของแฟ้มข้อมูลที่โปรแกรมต่าง ๆ สามารถถ่ายเทข้อมูลมาเป็นแฟ้มข้อมูลชนิดนี้ได้ โปรแกรมบรรจุข้อมูลจะให้ความสะดวกแก่ผู้ใช้ในการเพิ่มเติมระเบียบข้อมูลใหม่ โดยผู้ใช้ไม่จำเป็นต้องพัฒนาโปรแกรมประยุกต์เพื่อใช้ฟังก์ชันการเพิ่มเติมข้อมูลสำหรับการทำงานลักษณะนี้

การทำงานของโปรแกรมบรรจุข้อมูลลงแฟ้มข้อมูลดัชนี

1. ป้อนชื่อแฟ้มข้อมูล ได้แก่ แฟ้มข้อมูลเรียงลำดับ และ แฟ้มข้อมูลดัชนีในรูปแบบ `drive:\path\file name.extension`
2. ลักษณะระเบียบข้อมูลภายในแฟ้มข้อมูลเรียงลำดับ ประกอบด้วยระเบียบข้อมูลที่มีความยาวคงที่ และคั่นแต่ละระเบียบด้วยรหัสควบคุม คือ 0Dh และ 0Ah นอกจากนี้ระเบียบข้อมูลสุดท้ายของแฟ้ม การปิดท้ายด้วยรหัสควบคุม คือ 1Ah
3. เมื่อโปรแกรมไม่พบแฟ้มข้อมูลดัชนีที่ใช้บรรจุข้อมูล จะสอบถามความต้องการสร้างแฟ้มข้อมูลดัชนีใหม่จากผู้ใช้ เช่นเดียวกับการทำงานของโปรแกรมทำสำเนาข้อมูล
4. เมื่อเริ่มทำการบรรจุระเบียบข้อมูล การเปลี่ยนแปลงของจำนวนระเบียบข้อมูลที่ถูกรวบรวมจะแสดงบนจอภาพด้านล่าง และเช่นเดียวกันผู้ใช้สามารถยกเลิกการทำงานได้ทันทีจากการกดปุ่ม ESC
5. เมื่อเกิดความผิดพลาดจากการทำงาน โปรแกรมจะแสดงรหัสความผิดพลาดและข้อความที่อธิบายความผิดพลาดนั้นทางจอภาพด้านล่าง

โปรแกรมเก็บรักษาข้อมูลจากแฟ้มข้อมูลดัชนี (saveB program)

เป็นโปรแกรมที่ทำหน้าที่ตรงข้ามกับการบรรจุข้อมูลลงแฟ้มข้อมูลดัชนี คือทำการอ่านระเบียบข้อมูลจากแฟ้มข้อมูลดัชนี และสร้างแฟ้มข้อมูลเรียงลำดับซึ่งใช้บันทึกระเบียบข้อมูล

เพื่อเป็นการนำระเบียบข้อมูลจากแฟ้มข้อมูลดัชนีไปใช้ในรูปแบบอื่นต่อไป
การทำงานของโปรแกรมเก็บรักษาข้อมูลจากแฟ้มข้อมูลดัชนี

1. ป้อนชื่อแฟ้มข้อมูล ในรูปแบบ

drive:\path\file name.extension

2. ในการระบุแฟ้มข้อมูลเรียงลำดับที่ใช้บันทึกระเบียบข้อมูล ถ้าปรากฏว่ามีการสร้างแฟ้มข้อมูลนั้นแล้วในงานบันทึก การเก็บบันทึกระเบียบข้อมูลจากแฟ้มข้อมูลดัชนี จะทำการลบข้อมูลเดิมทั้งหมดที่มีอยู่ในแฟ้มข้อมูลเรียงลำดับนั้น ก่อนทำการบันทึกระเบียบข้อมูลใหม่ลงไป ลักษณะระเบียบข้อมูลภายในแฟ้ม จะเป็นลักษณะเดียวกับระเบียบข้อมูลที่ใช้ในการบรรจุลงในแฟ้มข้อมูลดัชนี

3. ผู้ใช้ต้องกำหนดหมายเลขชุดคีย์ ที่โปรแกรมต้องอ่านระเบียบข้อมูลจากแฟ้มข้อมูลดัชนี หมายเลขชุดคีย์นั้นมีค่าระหว่าง 0 - 23

4. เมื่อเริ่มทำการอ่านระเบียบข้อมูล การเปลี่ยนแปลงของจำนวนระเบียบข้อมูลที่ถูกรับและเก็บบันทึกจะแสดงบนจอภาพด้านล่าง และเช่นเดียวกันผู้ใช้สามารถยกเลิกการทำงานได้ทันทีจากการกดปุ่ม ESC

5. เมื่อเกิดความผิดพลาดจากการทำงาน โปรแกรมจะแสดงรหัสความผิดพลาดและข้อความที่อธิบายความผิดพลาดนั้นทางจอภาพด้านล่าง

โปรแกรมกู้แฟ้มข้อมูลดัชนีที่เสียหาย (recovB program)

เป็นโปรแกรมที่จัดเตรียมไว้สำหรับการอ่านระเบียบข้อมูลคืนจากแฟ้มข้อมูลดัชนีที่เสียหาย และไม่สามารถใช้ฟังก์ชันการเปิดแฟ้มข้อมูลดัชนีในแบบปกติได้ โดยทำการเปิดแฟ้มในลักษณะการกู้แฟ้มข้อมูล (กำหนดค่าลักษณะการเปิดแฟ้มเท่ากับ 2) และใช้ฟังก์ชันการอ่านระเบียบข้อมูลจากตำแหน่งภายในแผ่นข้อมูลโดยตรง และนำมาเก็บบันทึกในแฟ้มข้อมูลเรียงลำดับ หรือในแฟ้มข้อมูลดัชนีที่สร้างใหม่

การทำงานของโปรแกรมกู้แฟ้มข้อมูลดัชนี

1. ป้อนชื่อแฟ้มข้อมูล ในรูปแบบ

drive:\path\file name.extension

2. ผู้ใช้ต้องกำหนดชนิดของแฟ้มข้อมูลที่ใช้เก็บระเบียบข้อมูลที่สามารถอ่านได้ คือเป็นแฟ้มข้อมูลเรียงลำดับ หรือเป็นแฟ้มข้อมูลดัชนี

3. ในกรณีที่ระบุเป็นแฟ้มข้อมูลดัชนี และตรวจไม่พบแฟ้มข้อมูลนั้น โปรแกรมจะสอบถามความต้องการสร้างแฟ้มข้อมูลใหม่จากผู้ใช้ เช่นเดียวกับการทำสำเนาแฟ้มข้อมูลดัชนี

4. เมื่อเริ่มการทำงาน การเปลี่ยนแปลงของจำนวนระเบียบข้อมูลที่ถูกรื้อถอน และเก็บบันทึก จะแสดงบนจอภาพด้านล่าง และเช่นเดียวกันผู้ใช้สามารถยกเลิกการทำงานได้ทันทีจากการกดปุ่ม ESC

5. เมื่อเกิดความผิดพลาดจากการทำงาน โปรแกรมจะแสดงรหัสความผิดพลาดและข้อความที่อธิบายความผิดพลาดนั้นทางจอภาพด้านล่าง

โปรแกรมแสดงสถานภาพแฟ้มข้อมูลดัชนี (statB program)

เป็นโปรแกรมที่ใช้ในการรายงานสถานภาพแฟ้มข้อมูลดัชนีที่ผู้ใช้ต้องการทราบการทำงานของโปรแกรมแสดงสถานภาพแฟ้มข้อมูลดัชนี

1. บ้อนชื่อแฟ้มข้อมูลดัชนี ในรูปแบบ

drive:\path\file name.extension

2. โปรแกรมจะแสดงคุณสมบัติของแฟ้ม ระเบียบข้อมูล และชุดคีย์ที่กำหนดขึ้น

ได้แก่

2.1 ความยาวระเบียบข้อมูล

2.2 ขนาดแผ่น

2.3 จำนวนชุดคีย์

2.4 การระบุการสร้างแผ่นแผนที่ข้อมูล

2.5 จำนวนระเบียบข้อมูลทั้งหมดในแฟ้ม

2.6 จำนวนที่ว่างจากการลบระเบียบข้อมูล

2.7 จำนวนที่ว่างจากการลบแผ่น

2.8 คุณสมบัติของแต่ละเซกเมนต์คีย์ ได้แก่

2.8.1 หมายเลขชุดคีย์

2.8.2 หมายเลขเซกเมนต์

2.8.3 ตำแหน่งเริ่มต้นในระเบียบข้อมูล

2.8.4 ความยาวของคีย์

2.8.5 ชนิดของคีย์

2.8.6 การระบุคุณสมบัติต่าง ๆ ของเซกเมนต์คีย์นั้น

2.8.7 จำนวนของคีย์ที่มีในชุดคีย์นั้น

3. การแสดงข้อมูลทางจอภาพ จะแสดงรายละเอียดของเซกเมนต์คีย์ได้ครั้งละ 8 เซกเมนต์เท่านั้น ดังนั้นกรณีที่แฟ้มข้อมูลดัชนีมีจำนวนเซกเมนต์คีย์เกินกว่า 8 เซกเมนต์ โปรแกรมจะหยุดการแสดงข้อความเหล่านั้นและรอจนกว่าผู้ใช้กดปุ่มใดปุ่มหนึ่ง จึงแสดงส่วนที่เหลือต่อไป

4. เมื่อเกิดความผิดพลาดจากการทำงาน โปรแกรมจะแสดงรหัสความผิดพลาด และข้อความที่อธิบายความผิดพลาดนั้นทางจอภาพด้านล่าง

โปรแกรมยกเลิกฟังก์ชันจัดการแฟ้มข้อมูลดัชนี (stopB program)

เป็นโปรแกรมที่ทำการยกเลิกฟังก์ชันจัดการแฟ้มข้อมูลดัชนีที่ติดตั้งในหน่วยความจำ รวมทั้งเนื้อที่หน่วยความจำซึ่งจองไว้สำหรับบัฟเฟอร์ และคืนเนื้อที่หน่วยความจำที่ใช้ให้กับระบบ เอ็มเอสดอส โดยก่อนที่โปรแกรมจะยกเลิกการทำงาน จะแสดงข้อความบนจอภาพเพื่อให้ผู้ใช้ตอบรับการยกเลิกนี้ แล้วจึงกลับไปทำการงานของโปรแกรมเมนูหลัก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย