หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การวิเคราะห์ปัญหาที่เกิดขึ้นในระบบพัสดุคงคลังปัจจุบัน

ในระบบงานทั่ว ๆ ไปมักจะพบปัญหาสองประการคือ ปัญหาเกี่ยวกับคนและเครื่อง ทั้งคนและเครื่องเป็นกลไกซึ่ง เกี่ยวข้องกับระบบจะต้องมีประสิทธิภาพจึงจะทำให้งานดำเนินไปอย่างมีประสิทธิภาพ ในระบบงานที่ประมวลผลด้วยคอมพิวเตอร์ก็เช่นกัน ถ้าเครื่องมีประสิทธิภาพในการทำงานดีแต่ผู้ใช้ไม่พยายามใช้มันอย่างมีประสิทธิภาพก็ดูจะไม่คุ้มค่า การใช้คอมพิวเตอร์อย่างมีประสิทธิภาพ ผู้ใช้ควรจะทราบถึงขีดความสามารถของเครื่องคอมพิวเตอร์นั้น ๆ เพื่อจะได้วางแผน ออกแบบ และดำเนินการให้เหมาะสมและสอดคล้องกัน คนซึ่งเป็นผู้เกี่ยวข้องกับงาน ในระบบคอมพิวเตอร์ควรจะทำงานด้วยความตั้งใจไม่ว่าตนจะได้รับมอบหมายให้ทำงานในขั้นตอนใดของการประมวลผลก็ตาม ทั้งนี้เพราะถ้างานขั้นตอนใดขั้นตอนหนึ่งบกพร่องนั่นย่อมหมายถึงการประมวลผลของงานนั้นสูญเปล่าไปเนื่องจากรายงานที่ได้ไม่สมบูรณ์และยังไม่อาจนำไปใช้งานได้ ตัวอย่างเช่น การเตรียมข้อมูลก็นับเป็นขั้นตอนที่สำคัญ กล่าวคือถ้าผู้เจาะข้อมูลไม่ตั้งใจเจาะข้อมูลลงสื่อกลางที่จะป้อนคอมพิวเตอร์ เจาะผิดบ้างถูกบ้างโดยไม่เห็นถึงความสำคัญของงานที่ตนได้รับมอบหมาย หรืออาจจะคิดว่าจะมีผู้ตรวจสอบติดตามความผิดพลาดภายหลังอีก ครั้นพอถึงขั้นตอนการตรวจสอบผู้ตรวจสอบความผิดพลาดก็ทำไปโดยสนใจบ้างไม่สนใจบ้างว่าข้อมูลที่ผ่านการตรวจสอบแล้วควรจะเป็นข้อมูลที่ถูกต้องจริง ๆ ครั้นนำข้อมูลนั้นไปประมวลผลพบว่ายังมีข้อผิดพลาดของข้อมูลอยู่ เป็นเหตุให้ต้องแก้ไขข้อมูลเพื่อทำการประมวลผลซ้ำอีก ซึ่งเป็นการสิ้นเปลืองทั้งเวลาและแรงงานและค่าใช้จ่ายโดยเปล่าประโยชน์ จริงอยู่ที่ว่าการทำงานใด ๆ ก็ตามโดยมนุษย์ย่อมต้องมีข้อผิดพลาดบ้างเป็นธรรมดา แต่ถ้าพยายามขจัดข้อผิดพลาดให้เหลือน้อยที่สุดหรือไม่มีเลยได้ก็จะเป็นการควรอย่างยิ่ง

นอกจากข้อผิดพลาดดังกล่าวอันเป็นการสิ้นเปลืองทั้งเวลา และค่าใช้จ่ายอย่างเห็นได้ชัดแล้ว ยังมีการสิ้นเปลืองและการสูญเปล่าแฝงอีกซึ่งถ้าไม่สนใจพิจารณาดูก็อาจจะไม่รู้สึกว่าเป็นการสูญเปล่าตรงไหน แต่ถ้าได้วิเคราะห์ถึงประสิทธิภาพของการทำงานของระบบที่เป็นอยู่ และพยายามปรับปรุงการทำงานของระบบให้มีประสิทธิภาพมากกว่าเดิมจึงจะทราบว่าได้มีการใช้เวลาและค่าใช้จ่ายเกินกว่าที่ควรจะเป็นและนั่นคือสิ่งแอบแฝงที่มักถูกมองข้ามไป ดังนั้นการพัฒนาระบบไม่ว่าจะเป็นระบบงานที่ใช้คอมพิวเตอร์ประมวลผลหรือไม่ก็ตาม จึงเป็นสิ่งจำเป็น ทั้งนี้เพื่อคงไว้ซึ่งประสิทธิภาพของระบบงานให้มากที่สุดสำหรับระบบที่เคยมีประสิทธิภาพดีมาก่อน และปรับปรุงระบบที่มีประสิทธิภาพต่ำให้มีประสิทธิภาพสูงขึ้นกว่าที่เป็นอยู่เท่าที่จะทำได้โดยคำนึงถึงค่าใช้จ่ายที่ตามมาว่าคุ้มค่าหรือไม่หรือควรจะสร้างระบบใหม่เลยจะดีกว่า

การปรับปรุงเปลี่ยนแปลงระบบจะกระทำได้ก็ต่อเมื่อได้ทราบถึงปัญหาหรือสาเหตุที่แท้จริงอันเป็นตัวการสำคัญให้ประสิทธิภาพในการทำงานของระบบลดลง ดังนั้นก่อนทำการแก้ไขและเปลี่ยนแปลงระบบจะต้องศึกษาถึงปัญหาที่เกิดขึ้นเพื่อเป็นแนวทางในการแก้ไขและปรับปรุง โดยพยายามค้นหาถึงปัญหาหลักที่มีอิทธิพลต่อระบบมากที่สุดก่อนและจากปัญหาหลักอาจทำให้ทราบปัญหาที่มีความสำคัญรองลงมาซึ่งจะเป็นส่วนช่วยเสริมให้ระบบทำงานได้คล่องตัวมากยิ่งขึ้นถ้าได้รับการแก้ไข จากการศึกษาระบบปัจจุบันพอจะจำแนกปัญหาหลักของระบบได้ 2 ประการ

3.1 ปัญหาการออกแบบแฟ้มข้อมูล

ในการเรียกใช้แฟ้มข้อมูลเพื่อนำระเบียนภายในแฟ้มออกมาประมวลผลอาจกล่าวได้ว่าจะต้องใช้เวลาซึ่งประกอบด้วย 3 ส่วนใหญ่ ๆ ดังนี้

1. เวลาที่ใช้ในการเข้าถึงระเบียน (record access) ในกรณีที่แฟ้มข้อมูลถูกบันทึกลงบนจานแม่เหล็ก เวลาส่วนนี้ได้แก่เวลาที่ใช้ไปในการเคลื่อนหัวอ่านของจานแม่เหล็กไปยังตำแหน่งที่ต้องการ และเวลาที่เสียไปในการรอให้จานแม่เหล็กหมุนมายังตำแหน่งที่หัวอ่านของจานแม่เหล็กซึ่งพร้อมที่จะอ่านข้อมูลที่ต้องการ

2. เวลาที่ใช้ไปในการส่งข้อมูลจากหน่วยช่วยความจำ เข้าสู่หน่วยความจำของของคอมพิวเตอร์

3 เวลาที่ใช้ไปในการประมวลผลอย่างอื่นนอกเหนือจากข้อ 1 และข้อ 2 ได้แก่เวลาในการประมวลผลในหน่วยประมวลผลกลาง (CPU) เช่น เวลาที่ใช้ไปในการเปรียบเทียบระเบียนที่ต้องการกับระเบียนภายในบล็อคที่ถูกส่งเข้าสู่หน่วยความจำของคอมพิวเตอร์เป็นต้น

แม้ว่า เวลาทั้งหมดที่ใช้ไปจะไม่ใช่ผลบวกของเวลาทั้ง 3 ส่วน ทั้งนี้เนื่องจากอาจมีการทำงานมากกว่า 1 อย่างในขณะเดียวกัน (overlapping) ซึ่งระบบปฏิบัติการ (operating system) เป็นผู้จัดการดูแลในเรื่องนี้ แต่การแยกเวลาออกในลักษณะดังกล่าวข้างต้นเพื่อช่วยในการพิจารณาประสิทธิภาพของการปรับปรุงองค์กรแฟ้มข้อมูล

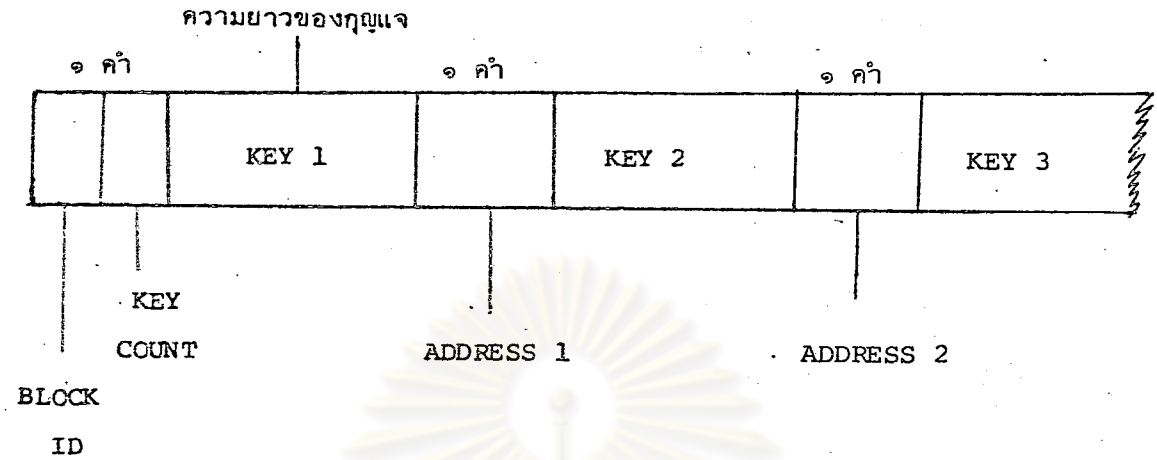
ในระบบคอมพิวเตอร์ UNIVAC 1106 PCIOS (Processor Common Input/Output System) จะทำหน้าที่ควบคุมแฟ้มขอมูลต่าง ๆ รวมทั้งแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนี ซึ่งโครงสร้างแฟ้มข้อมูล [๒] ประกอบด้วย

ก. LABEL INFORMATION จะถูกบันทึกบนเนื้อที่ 4 เซคเตอร์ (sector) แรกของแฟ้มข้อมูล

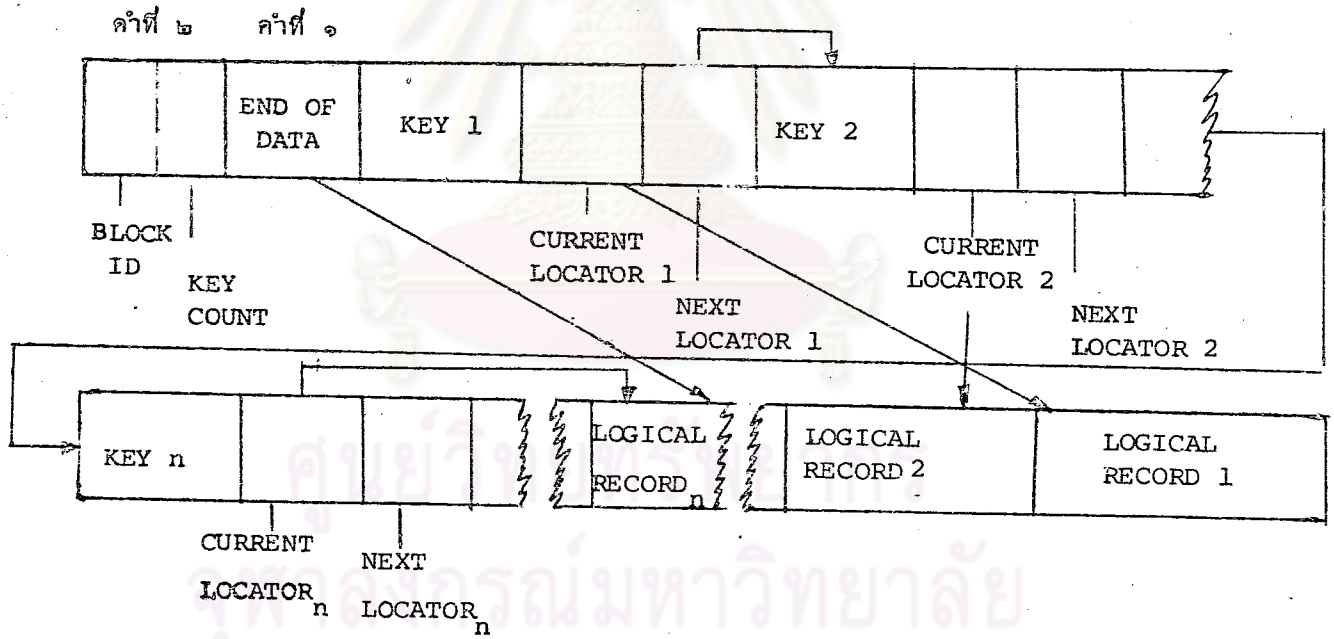
ข. บล็อคดัชนี (indexed block) และบล็อคระเบียน (record block)

ค. บล็อคล้น (overflow block) ซึ่งระบบจะเผื่อไว้อย่างน้อย 1 บล็อค

จะเห็นว่าการปรับปรุงประสิทธิภาพของแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนี ในระบบคอมพิวเตอร์ UNIVAC 1106 จะกระทำโดยการกำหนดขนาดของบล็อคดัชนีและบล็อคระเบียนให้เหมาะสมกับจำนวนข้อมูลที่มีอยู่ทั้งแฟ้ม และมีข้อแม้ว่าบล็อคของดัชนีและบล็อคของระเบียนจะต้องมีขนาดเท่ากันด้วย ดังนั้นเพื่อให้เกิดความเข้าใจในแนวทางการปรับปรุงแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีในระบบคอมพิวเตอร์ UNIVAC 1106 จึงขออธิบายรายละเอียดของบล็อคดัชนีและบล็อคระเบียนดังต่อไปนี้

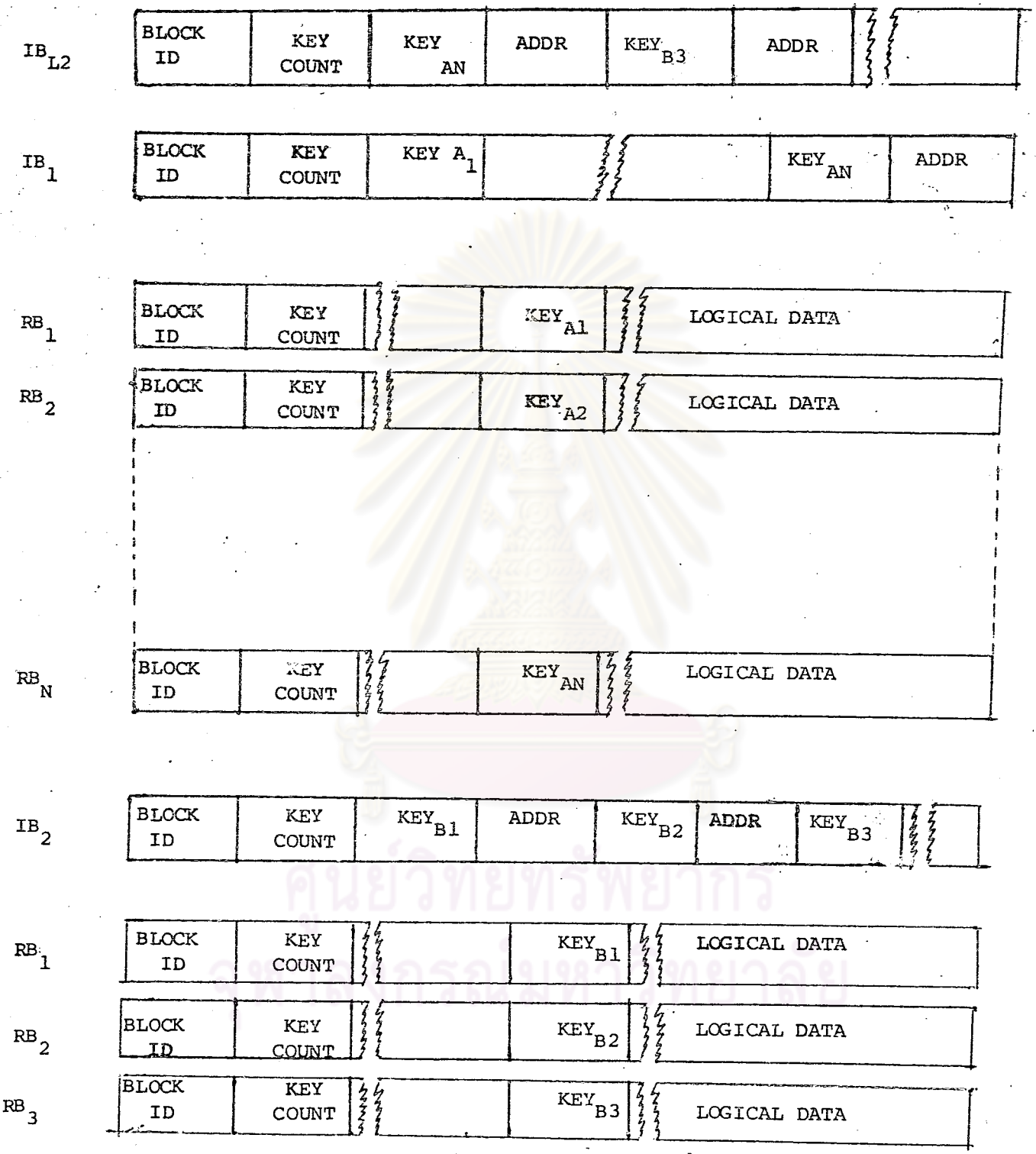


รูปที่ 3.1 โครงสร้างของบล็อคดัชนี (INDEX BLOCK)



รูปที่ 3.2 โครงสร้างของบล็อคระเบียนและบล็อคล้น

(INDEX BLOCK AND OVERFLOW BLOCK)



รูปที่ 3.3 แสดงโครงสร้างภายในของแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนี

กรณีที่มีบล็อคดัชนี 2 ระดับ

โครงสร้างของบล็อคดัชนีได้แสดงไว้ในรูปที่ 3.1 โดยครึ่งคำแรกของบล็อคดัชนีจะใช้สำหรับแสดงบล็อคไอดี (Block ID) และครึ่งหลังของคำแรก จะใช้นับจำนวนคีย์ (key count) ในบล็อคดัชนี ระเบียนหนึ่ง ๆ ในบล็อคดัชนีจะประกอบด้วยคีย์สูงสุด และที่อยู่ (Address) ของบล็อกระเบียน (record block) เนื้อที่สำหรับบันทึกที่อยู่ยาว 1 คำ ดังนั้นระเบียนในบล็อคดัชนีจะมีความยาวเท่ากับความยาวของคีย์ บวกกับอีก 1 คำ บล็อคไอดีจะมีค่าเท่ากับศูนย์ในระบบเลขฐานสอง

โครงสร้างของบล็อคระเบียนได้แสดงไว้ในรูปที่ 3.2 คำแรกของบล็อคยังมีลักษณะและความหมายเช่นเดียวกับกรณีบล็อคดัชนี คือมีไว้สำหรับเป็นบล็อคไอดีและการนับคีย์ คำที่สองของบล็อคแสดงตำแหน่งการสิ้นสุดของข้อมูลซึ่งจะอธิบายต่อไปภายในบล็อคระเบียน ต้นบล็อคจะบันทึกคีย์ของระเบียนโดยเริ่มจากคีย์ต่ำสุดเรียงลำดับจนถึงคีย์สูงสุดของระเบียนในบล็อคนั้น ในขณะที่ระเบียนข้อมูลที่มีคีย์ต่ำสุดในบล็อคจะถูกบันทึกโดยเริ่มจากปลายบล็อคและย้อนหลังขึ้นมาซึ่งระเบียนข้อมูลที่มีคีย์สูงสุดในบล็อคอาจจะอยู่ตรงกลางบล็อคหรือค่อนไปทางต้นหรือปลายบล็อคทั้งนี้ขึ้นอยู่กับปริมาณข้อมูลที่ถูกบันทึกลงในบล็อคระเบียนนั้น ๆ โดยจะทราบตำแหน่งการสิ้นสุดของระเบียนข้อมูลได้จากคำที่สองของบล็อคระเบียน ระเบียนคีย์ในบล็อคระเบียนจะมีความยาวมากกว่าระเบียน คีย์ในบล็อคดัชนีอยู่ 1 คำ กล่าวคือระเบียนคีย์ในบล็อคระเบียนจะประกอบด้วย คีย์ไอ (key i) ตำแหน่งของระเบียนข้อมูลที่มี คีย์ไอ ซึ่งใช้เนื้อที่เท่ากับ 1 คำ และรายการถัดไปมีความยาวเท่ากับ 1 คำ ซึ่งมีไว้สำหรับแสดงตำแหน่งของคีย์ ซึ่งเรียงลำดับถัดไป ในกรณีที่มีการลบข้อมูลออกจากบล็อคระเบียน ที่อยู่ปัจจุบันของระเบียนที่ถูกลบออกจากแฟ้มจะถูกกำหนดให้เป็นศูนย์ (ระบบเลขฐานสอง) และข้อมูลในระเบียนนั้นจะหายไป โดยการเลื่อนข้อมูลระหว่างระเบียนที่ลบทิ้งให้ติดกัน ส่วนในกรณีที่มีการเพิ่มข้อมูลเข้าไปในแฟ้มข้อมูล ข้อมูลใหม่จะถูกบันทึกในบล็อคระเบียนเดิม ถ้ามีที่ว่างพอโดยคีย์ของระเบียนใหม่จะถูกบันทึกต่อจากระเบียนที่แล้วและระเบียนข้อมูลก็จะถูกบันทึกต่อจากระเบียนข้อมูลที่แล้วเช่นกัน ดังนั้นคีย์ของแต่ละระเบียนภายในบล็อคระเบียนหลังจากที่ได้มีการเพิ่มข้อมูลแล้วอาจจะไม่เรียงลำดับกันแต่จะทราบลำดับของคีย์ถัดไปได้จากเนื้อที่ 1 คำ ซึ่งมีไว้สำหรับแสดงตำแหน่งของคีย์ถัดไป (next locator) ตำแหน่งของคีย์ถัดไปอาจจะอยู่

ภายในบล็อคระเบียนเดียวกันหรือคนละบล็อคระเบียนก็ได้ ทั้งนี้เพราะในกรณีที่มีการเพิ่มข้อมูลและมีอาจบันทึกข้อมูลนั้นไว้ในบล็อคระเบียนเดิมได้จึงจำเป็นต้องบันทึกไว้ในบล็อคสัน โครงสร้างบล็อคสันมีลักษณะทำนองเดียวกันกับโครงสร้างของบล็อคระเบียน ในกรณีที่บล็อคระเบียนยังไม่เคยถูกทำให้เป็นปัจจุบันค่าของบล็อคไอดีจะเป็น 1 แต่ถ้าเคยถูกทำให้เป็นปัจจุบันแล้ว ค่าในบล็อคไอดีจะเป็น 2 เช่นเดียวกับ บล็อคสัน

ในการสร้างแฟ้มข้อมูลหลักแบบอนุบรรพเชิงดัชนี นักวิเคราะห์ระบบจะต้องกำหนดจำนวนระเบียนของข้อมูลที่จะบรรจุในหนึ่งบล็อค (บีเอฟ) เพื่อให้ PCIOS ทราบว่าในการอ่านและเขียนข้อมูลในการประมวลผลแต่ละครั้งข้อมูลมีขนาดบล็อคเท่าใดจะได้ทราบว่าขนาดของเนื้อที่ที่จะเตรียมไว้รองรับข้อมูล เมื่อ PCIOS ทราบขนาดของบล็อคและขนาดของคีย์ก็สามารถคำนวณหาว่าในบล็อคดัชนีจะบรรจุคีย์ได้มากน้อยเท่าใด เช่น ถ้าหนึ่งบล็อคบรรจุ ได้ 200 คีย์ แสดงว่าบล็อคดัชนีนั้นจะสามารถควบคุมบล็อคระเบียนได้ 200 บล็อค ในระบบคอมพิวเตอร์ UNIVAC 1106 บล็อคดัชนี จะต้องมีขนาดเท่ากับบล็อคระเบียน ดังนั้นในกรณีที่จำนวนระเบียนมีมากเกินกว่า บล็อคระเบียนที่มีอยู่จะรับได้ PCIOS จะสร้างบล็อคดัชนีที่สองขึ้นดังรูปที่ 3.3 บล็อคดัชนีจะมี 2 ระดับ ระดับสูงสุดของบล็อคดัชนีในรูปที่ 3.3 คือ IB_{L2} ซึ่งจะใช้เพื่อหาว่าคีย์ที่ต้องการจะอยู่ในระเบียนซึ่งควบคุมโดยบล็อคดัชนี IB_1 หรือ IB_2 จึงทำให้การค้นหาระเบียนที่ต้องการใช้เวลาการเข้าถึงระเบียนและการส่งข้อมูลทั้งบล็อคจากหน่วยช่วยความจำไปสู่หน่วยความจำของคอมพิวเตอร์มากขึ้น

จะเห็นว่าถ้าแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีที่ถูกสร้างขึ้นมีบล็อคดัชนีเกินกว่าหนึ่งบล็อคจะต้องมีบล็อคดัชนีอีกระดับหนึ่งถูกสร้างขึ้นเพื่อคุมบล็อคดัชนีดังกล่าวโดยบล็อคดัชนีในระดับสูงนี้จะมีโครงสร้างและขนาดเช่นเดียวกับบล็อคดัชนีระดับต่ำสุด ผิดกันแต่ว่าคีย์ที่บรรจุอยู่ในบล็อคดัชนีระดับสูงกว่านี้จะเป็นคีย์สูงสุดของแต่ละบล็อคดัชนีระดับรองลงมาแทนที่จะบรรจุคีย์สูงสุดของบล็อคระเบียน ดังแสดงในรูปที่ 3.3 เนื่องจากไม่ว่าจะมีการเพิ่มหรือลบข้อมูลในแฟ้ม ระเบียนต่าง ๆ ในบล็อคดัชนีจะไม่มีการเปลี่ยนแปลงดังนั้นการค้นหาข้อมูลภายในบล็อคดัชนีจะถูกค้นหาแบบการค้นหาเชิงคู่ (binary search) ส่วนการค้นหาข้อมูลภายในบล็อคระเบียนถ้ายังไม่มีการทำให้เป็นปัจจุบันจะทำการค้นหาเช่น

เดียวกับบล็อคดัชนีแต่ถ้ามีการทำแฟ้มข้อมูลให้เป็นปัจจุบันเ ได้เพิ่มระเบียนใหม่เข้าไปในบล็อคระเบียนแล้วการค้นหาภายในบล็อคระเบียนจะเป็นแบบอนุบรรพ

ในการประมวลผล เมื่อแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีถูกเรียกใช้ พีซีไอโอเอส จะอ่านบล็อคดัชนีในระดับสูงสุดซึ่งบรรจุคีย์สูงสุดของบล็อคดัชนีเข้าสู่หน่วยความจำของคอมพิวเตอร์ก่อน (ถ้ามี)แล้วจึงทำการค้นหาว่าข้อมูลที่ต้องกา ควรถูกบันทึกอยู่ในบล็อคดัชนีใดในระดับรองลงมาแล้วจึงทำการอ่านบล็อคดัชนีนั้นเข้าสู่หน่วยความจำของคอมพิวเตอร์ เพื่อทำการค้นหาว่าข้อมูลที่ต้องการนำมาประมวลผลนั้นอยู่ในบล็อคระเบียนใด เพื่อจะได้อ่านข้อมูลทั้งบล็อคระเบียนนั้นเข้าสู่หน่วยความจำ เพื่อค้นหารายการที่ต้องการในระหว่างประมวลผลผู้ใช้อาจจะกำหนดให้บล็อคดัชนีระดับสูงสุดให้คงอยู่ในหน่วยความจำของคอมพิวเตอร์ตลอดเวลาของการประมวลผล เพื่อความรวดเร็วในการค้นหา แต่อย่างไรก็ตาม ถ้าแฟ้มข้อมูลถูกออกแบบให้มีบล็อคดัชนีหลาย ๆ ระดับจะทำให้เวลาที่ใช้ในการค้นหาช้ากว่าที่ควรจะเป็นซึ่งเป็นการสิ้นเปลืองเวลา ทั้งนี้เพราะการประมวลผลแต่ละระเบียนจะต้องเสียเวลาในการค้นหาหลายขั้นตอน จนกว่าจะได้ข้อมูลที่ต้องการซึ่งบันทึกอยู่ในบล็อคระเบียน เวลาที่เสียไปดังกล่าวคือเวลาที่ใช้ในการค้นหาบล็อคดัชนีระดับสูงสุดและระดับรอง ๆ ลงมาและบล็อคระเบียนซึ่งการค้นหาข้อมูลในบล็อคดังกล่าวจะต้องเสียเวลาไปกับการหาตำแหน่งของบล็อคที่ต้องการว่าอยู่ ณ ที่ใดบนเนื้อที่ช่วยความจำซึ่งอาจจะเป็นจานแม่เหล็ก แล้วทำการส่งข้อมูลทั้งบล็อคนั้นเข้าสู่หน่วยความจำของคอมพิวเตอร์เพื่อทำการค้นหาระเบียนข้อมูลที่ต้องการ การเข้าหาบล็อคดัชนีหรือบล็อคระเบียนแต่ละครั้งจะใช้เวลาประมาณ 38 ms สำหรับจานแม่เหล็กรุ่น 8434 ซึ่งจะช้ามากเมื่อเปรียบเทียบกับอัตราการส่งข้อมูลซึ่งจะใช้เวลาประมาณ 5.1 μs/Words (36 BIT = 1 WORD)

จะเห็นว่าการกำหนดขนาดของบล็อค (block size) เป็นหัวใจของการออกแบบแฟ้มข้อมูลที่ไม่ควรมองข้ามไป ทั้งนี้เพราะถ้าขนาดของบล็อคเล็กเกินไปจะทำให้เกิดบล็อคดัชนีมากกว่า 1 ระดับ ทำให้ต้องเสียเวลาในการหาตำแหน่งของบล็อคระเบียนที่ต้องการมากถึงแม้ว่าการค้นหาระเบียนภายในบล็อคจะเร็วเนื่องจากมีขนาดเล็ก

ก็ตามและถ้าขนาดของบล็อคใหญ่เกินไปก็จะไม่เป็นการดีเพราะจะทำให้มีที่ว่างเหลืออยู่มากในบล็อคดัชนี ถึงแม้ว่าการค้นหาบล็อคระเบียนจะกระทำได้รวดเร็ว เนื่องจากมีบล็อคดัชนีเพียงบล็อคเดียวแต่การค้นหาระเบียนที่ต้องการในบล็อคจะช้า เนื่องจากบล็อคระเบียนมีขนาดใหญ่ขึ้น ดังนั้นการกำหนดขนาดของบล็อคจะต้องมีขนาดพอดีไม่ใหญ่โตจนเกินความจำเป็นและไม่เล็กจนทำให้ต้องมีดัชนีหลายระดับ จึงจะทำให้การใช้แฟ้มข้อมูลเป็นไปอย่างมีประสิทธิภาพ

ลักษณะการออกแบบแฟ้มข้อมูลหลักที่ใช้ในระบบพัสดุคงคลังปัจจุบัน

แฟ้มข้อมูลหลัก ซึ่งใช้ในการประมวลผลเพื่อออกรายงานประจำวันของคลังพัสดุส่วนกลางและคลังพัสดุส่วนภูมิภาค มีองค์กรแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนี บันทึกข้อมูล 249,989 ระเบียน (ขณะทำการวิจัย) ข้อมูลแต่ละระเบียนมีขนาดคงที่ยาว 62 อักขระ ถูกออกแบบให้บีเอฟมีค่าเท่ากับ 20 ขณะประมวลผลมี User Available core ประมาณ 57 K Words จากการศึกษาการทำงานของระบบพบว่าเวลาที่ใช้ในการประมวลผลนานเกินกว่าที่ควรจะเป็น ทั้งนี้เพราะการออกแบบแฟ้มข้อมูลยังไม่เหมาะสมกับขนาดของแฟ้มข้อมูล และส่วนประกอบอื่น ๆ ที่เกี่ยวข้องกับระบบดังกล่าวแล้วข้างต้น

จากการที่แฟ้มข้อมูลถูกกำหนดให้บีเอฟเท่ากับ 20 ทำให้ พีซีไอโอเอส ซึ่งรับผิดชอบในเรื่องของแฟ้มข้อมูลต้องสร้างบล็อคดัชนีขึ้นถึงสองระดับโดยสามารถคำนวณให้เห็นได้ดังนี้

ในระบบคอมพิวเตอร์ UNIVAC 1106 การคำนวณขนาดบล็อคและขนาดคีย์มีหน่วยเป็นคำ (1 Word เท่ากับ 4 Byte หรือ 36 Bits) และถ้ามีการใช้เนื้อที่ไม่เต็มคำจะปัดเศษนั้นเป็น 1 คำ และขนาดกายภาพของบล็อค (Physical Block) จะถูกปรับให้เป็นทวีคูณของ Disk prep. factor ซึ่ง Disk prep. factor ที่ใช้อยู่ปัจจุบันเท่ากับ 112 Words

ระเบียนข้อมูลเชิงตรรก (Logical Record) มีขนาด 62 byte => 16 words

คีย์ของแต่ละระเบียนมีขนาด 10 byte => 3 words

ภายในบล็อคระเบียนใช้เนื้อที่สำหรับเก็บที่อยู่ของระเบียน (รูปที่ 3.5) = 2 words

ดังนั้น 1 ระเบียนของข้อมูลในบล็อคระเบียนจะต้องใช้เนื้อที่ 16 + 3 + 2 => 21 words

ขนาดกายภาพของบล็อค (Physical Block) คือ $\left\lfloor \frac{21x20+2}{112} \right\rfloor$ 112 = 448 words

และเนื่องจาก PCIOS พยายามจะประหยัดเนื้อที่ในหน่วยช่วยความจำดังนั้นจึงพยายาม

บรรจุระเบียนข้อมูลให้มากที่สุดในที่นี้ 1 บล็อคจุได้ = 21 ระเบียน

และจำนวนกุญแจพร้อมที่อยู่ของบล็อคระเบียนที่จะบรรจุได้เป็นบล็อคดัชนี $= \left\lfloor \frac{448-1}{4} \right\rfloor = 111$

นั่นคือ 1 บล็อคดัชนีจะคุมบล็อคระเบียนได้ = 111 บล็อค

∴ บล็อคดัชนีหนึ่งจะควบคุมข้อมูลได้ = 111x21 ระเบียน

= 2331 ระเบียน

ข้อมูลทั้งหมดภายในแฟ้มมีอยู่ = 249989 ระเบียน

∴ จำเป็นต้องใช้บล็อคดัชนี 249989 ÷ 2331 = 108 บล็อค

ดังนั้นแฟ้มข้อมูลปัจจุบันจำเป็นต้องมีดัชนีถึง 2 ระดับ โดยดัชนีระดับที่ 1 คุมบล็อคระเบียนได้ 111 บล็อคต่อหนึ่งบล็อคดัชนี และ ดัชนีระดับที่ 2 คุมบล็อคดัชนีระดับที่ 1 อยู่ 108 บล็อค ขณะทำการทดลอง ซึ่งแฟ้มข้อมูลมีจำนวนระเบียน 249,989 ระเบียน

สำหรับแฟ้มข้อมูลพัสดุใช้งานของคลังพัสดุส่วนกลางและคลังพัสดุส่วนภูมิภาคปัจจุบัน ถูกออกแบบให้มีเอฟฟีค่าเพียง 10 มีความยาวของระเบียนเท่ากับ 244 อักขระ และคีย์ของแต่ละระเบียนยาว 13 อักขระ ในขณะที่แฟ้มข้อมูลพัสดุใช้งานของคลังพัสดุส่วนกลางมีขนาด 50,015 ระเบียน และแฟ้มข้อมูลหลักพัสดุใช้งานของคลังพัสดุส่วนภูมิภาคมีขนาด 89,519 ระเบียน ดังนั้นแฟ้มข้อมูลหลักพัสดุใช้งานทั้งสองแฟ้มจึงจำเป็นต้องมีบล็อคดัชนีถึงสองระดับ โดยการคำนวณจะกระทำได้ทำนองเดียวกันกับแฟ้มข้อมูลหลักดังได้แสดงให้เห็นแล้วข้างต้น กล่าวคือแฟ้มข้อมูลหลักพัสดุใช้งานของคลังพัสดุกลางจะต้องใช้บล็อคดัชนีถึง 38 บล็อค และแฟ้มข้อมูลหลักพัสดุใช้งานของคลังพัสดุส่วนภูมิภาคจะต้องใช้บล็อคดัชนีถึง

67 บล็อค เพื่อเป็นทางเข้าถึงข้อมูล ดังที่ได้เคยกล่าวมาแล้วว่าแฟ้มข้อมูลที่ถูกออกแบบโดยบีเอฟมีค่าไม่เหมาะสม ทำให้ต้องเสียเวลามากในการค้นหาข้อมูล แต่ถ้าบล็อคดัชนีหลายระดับถูกลดลงให้เหลือเพียงระดับเดียวจะช่วยให้การประมวลผลรวดเร็วขึ้น

3.2 ปัญหาการเลือกวิธีการเข้าถึงข้อมูลที่ต้องการ (ACCESS MODE)

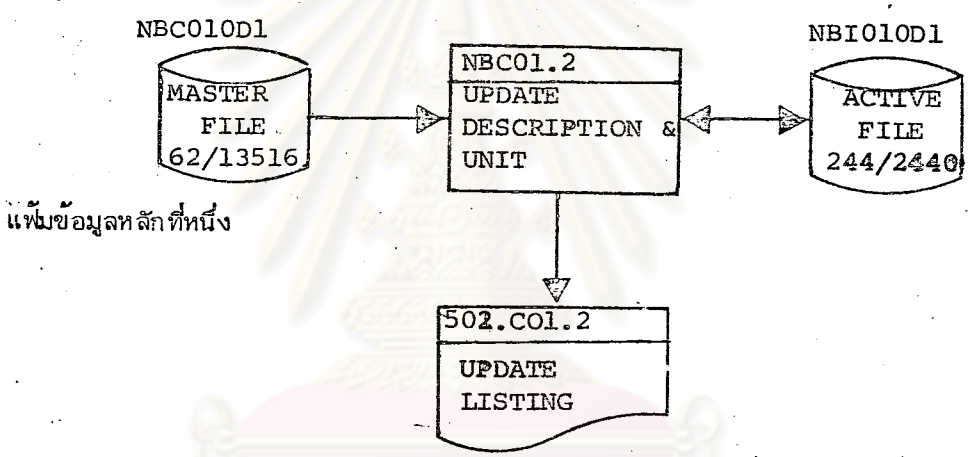
จากการประมวลผลเพื่อทำการแก้ไขชื่อพัสดุในแฟ้มข้อมูลหลักพัสดุใช้งานให้ตรงกับชื่อพัสดุในแฟ้มข้อมูลหลักโดยโปรแกรม NBC 012 โดยระบบปัจจุบันเลือกการเข้าถึงระเบียนที่ต้องการในแฟ้มข้อมูลหลักแบบอนุบรรพทั้ง ๆ ที่จำนวนระเบียนในแฟ้มข้อมูลหลักพัสดุใช้งานมีจำนวนระเบียนน้อยมากเมื่อเทียบกับจำนวนระเบียนของข้อมูลในแฟ้มข้อมูลหลัก ซึ่งไม่เหมาะสมกับการประมวลผลแบบอนุบรรพ นอกจากนี้ระบบปัจจุบันยังต้องทำการเปลี่ยนองค์กรแฟ้มข้อมูลหลักพัสดุใช้งานจากองค์กรแบบอนุบรรพเชิงดัชนีไปเป็นองค์กรแบบอนุบรรพ เพื่อจัดเรียงข้อมูลภายในแฟ้มข้อมูลให้เรียงลำดับไปทางเดียวกับแฟ้มข้อมูลหลัก และในการประมวลผลทุกระเบียนในแฟ้มข้อมูลหลักจะต้องถูกนำมาประมวลผลจนกว่าจะพบคีย์ที่เท่ากันหรือมากกว่าคีย์ของระเบียนที่ต้องการค้นหา ทั้ง ๆ ที่มีข้อมูลบางกลุ่มภายในแฟ้มข้อมูลหลักอาจจะไม่มีปรากฏเลยในแฟ้มข้อมูลพัสดุใช้งาน เมื่อทำการประมวลผลเสร็จแล้วจะต้องนำข้อมูลของแฟ้มพัสดุใช้งานที่ได้ไปจัดเรียงลำดับใหม่เพื่อให้ระเบียนข้อมูลต่าง ๆ เรียงลำดับในลักษณะเดิมแล้วจึงนำเทปที่ได้นั้นไปสร้างแฟ้มข้อมูลพัสดุใช้งานใหม่บนจานแม่เหล็ก โดยมีองค์กรแบบอนุบรรพเชิงดัชนีดังเดิมเพื่อนำไปใช้งานต่อไป จะเห็นว่าขั้นตอนสำคัญของการประมวลผลพอสรุปได้ดังนี้

1. เรียงลำดับข้อมูลระเบียนในแฟ้มข้อมูลหลักพัสดุใช้งานตามรหัสพัสดุ

2. สำหรับระเบียนแต่ละระเบียนในแฟ้มข้อมูลหลักพัสดุใช้งานจะต้องค้นหาจากแฟ้มข้อมูลหลักที่หนึ่งแบบอนุบรรพ โดยแฟ้มข้อมูลทั้งสองเรียงลำดับตามรหัสพัสดุ

3. สร้างแฟ้มข้อมูลหลักพัสดุใช้งานใหม่โดยให้มีองค์กรแบบอนุบรรพเชิงดัชนี

เมื่อพิจารณาขั้นตอนข้างต้นอย่างถี่ถ้วน อาจจะกล่าวได้ว่าขั้นตอนที่ 1 และ 3 ไม่จำเป็น ถ้าเปลี่ยนการค้นหาแฟ้มข้อมูลหลักที่หนึ่ง แบบอนุบรรพมาเป็นการค้นหาแบบตรง ดังลักษณะการประมวลผลในรูปที่ 3.4



รูปที่ 3.4 การประมวลผลเพื่อทำให้แฟ้มข้อมูลหลักที่สองหรือสามให้เป็นปัจจุบัน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย