บทที่ 4

ระบบการประมวลผลเอกสารของโปรแกรมซียูไรท์เตอร์

บทนำ

ระบบการประมวลผลเอกสารจะมีขั้นตอนการทำงาน 3 ขั้นตอนคือ การนำข้อมูลเข้าและตรวจสอบการประมวลผลคำ และการแสดงผลลัพธ์ ซึ่งการแสดงผลลัพธ์จะแบ่งออกได้เป็น 2 ประเภทคือ การแสดงผลลัพธ์ทางจอภาพ และการแสดงผลลัพธ์อย่างถาวร เช่น การแสดงผลลัพธ์ทางแผ่นจานบันทึก และ การแสดงผลลัพธ์ทางเครื่องพิมพ์ เป็นต้น

4.1 การประมวลผลคำ

เป็นการจัดรูปแบบของข้อความในขณะป้อนข้อมูลเข้า คำสั่งที่ใช้ในการจัดรูปแบบในโปรแกรมจะสั่งโดยตรงได้ โดยผ่านทางแป้นพิมพ์ และข้อความที่จัดรูปแบบแล้ว จะแสดงให้เห็นทางจอภาพทันที ดังนั้นรูปแบบของข้อความที่ปรากฏให้เห็นบนจอภาพ คือ รูปแบบสุดท้ายของเอกสารที่ได้รับนั้นเอง

สำหรับลักษณะการทำงานของการประมวลผลคำ จะเป็นลักษณะของบรรณาธิกรทั้งจอภาพ (Screen Editor) คือ สามารถแสดงข้อความครั้งละหลายบรรทัด โดยการนำข้อความซึ่งเป็นส่วนหนึ่งที่เก็บในแฟ้มข้อมูลแสดงทางจอภาพ ครั้งละหนึ่งจอภาพ และสามารถแก้ไขเปลี่ยนแปลงข้อความที่ปรากฏบนจอภาพขณะนั้นได้ โดยการใช้เคอร์เซอร์ (Cursor) ซึ่งเป็นตัวชี้บอกตำแหน่งขณะนั้น เลื่อนไปยังตำแหน่งที่ต้องการและทำการแก้ไขต่อไป นอกจากนี้สามารถกำหนดให้ข้อความในส่วนต่างๆของแฟ้มข้อมูลแสดงทางจอภาพได้

อย่างไรก็ตามการบรรณาธิกรทั้งจอภาพ จะต้องมีการใช้บัฟเฟอร์ สำหรับใช้เป็นที่เก็บชั่วคราวสำหรับการแก้ไขหรือสร้างข้อความใหม่ เมื่อแก้ไขเปลี่ยนแปลงข้อความแล้วจะต้องกำหนดให้เก็บข้อความเหล่านั้นไว้ในแฟ้มข้อมูลอีกครั้งหนึ่ง

4.2 ลักษณะการทำงานภายในของการประมวลผลคำในโปรแกรมซียูไรท์เตอร์

การทำงานภายในของการประมวลผลคำในโปรแกรมซียูไรท์เตอร์จะเป็นลักษณะของการประยุกต์ใช้โครงสร้างข้อมูลแบบต่างๆเข้าด้วยกัน เพื่อให้สอดคล้องกับการทำงานที่ซับซ้อนได้ นอกจากนั้นการประมวลผลคำในโปรแกรมซียูไรท์เตอร์ สามารถที่จะรับและแสดงผลข้อความภาษาไทยบนจอภาพได้ ดังนั้นจึงต้องมีการนำระบบภาษาไทยมาใช้งานด้วย

โครงสร้างข้อมูลที่ใช้ภายในของการประมวลผลคำ ในโปรแกรมซียูไรท์เตอร์จะมีส่วนที่สำคัญต่างๆดังนี้คือ

4.2.1 บัฟเฟอร์ (BUFFER)

เป็นที่ใช้ชั่วคราวสำหรับเก็บข้อความขณะทำงานของโปรแกรม โดยเฉพาะงานที่เกี่ยวข้องกับการสร้างและการแก้ไขข้อความภายในบัฟเฟอร์ ก่อนที่จะนำข้อความเหล่านั้นเข้ามาเก็บไว้ในแฟ้มข้อมูลเป็นการถาวรต่อไป จึงอาจจะมองบัฟเฟอร์ได้เสมือนกระดานดำสำหรับเขียน ซึ่งเมื่อเขียนครั้งหนึ่งแล้วสามารถลบออก เพื่อใช้ในครั้งต่อไปได้อีก โดยบัฟเฟอร์จะแบ่งออกตามลักษณะการใช้งานได้ดังนี้คือ

4.2.1.1 โหนดบัฟเฟอร์ (Node Buffer) เป็นบัฟเฟอร์ที่ใช้สำหรับอ่านข้อความตัวอักษรจากแฟ้มข้อมูลมาเก็บไว้ และนำข้อความตัวอักษรจากบัฟเฟอร์ไปเก็บไว้ในแฟ้มข้อมูลได้ โดยการเก็บตัวอักษรภายในโหนดบัฟเฟอร์ จะเก็บอย่างต่อเนื่องอยู่ในบรรทัดเดียวกันตลอด และปิดท้ายข้อความด้วยค่า Null Character (\0) ซึ่งไม่มีการแปลงรหัสตัวอักษรเลย ทำให้โครงสร้างไม่ซับซ้อน จึงจัดการหรือประมวลผลอักขระภาษาไทยได้ง่ายและรวดเร็ว ดังแสดงการเก็บข้อมูลไว้ในโหนดบัฟเฟอร์ในรูปที่ 4.1

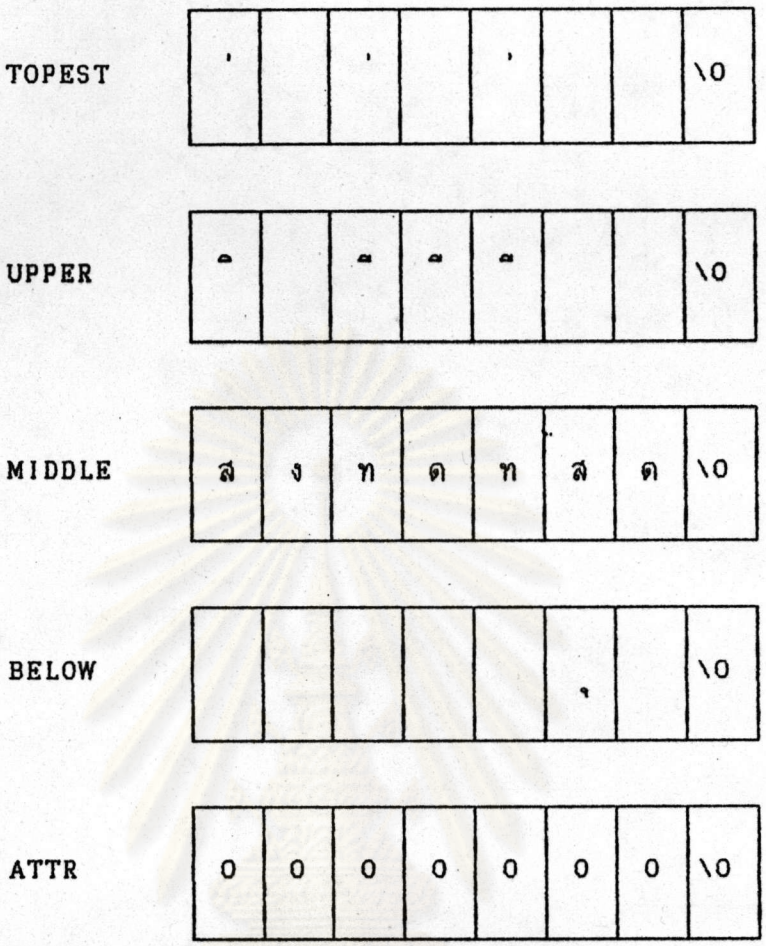
ส	◌ิ	'	ง	ท	◌ี	'	ด	◌ิ	ท	◌ี	'	ส	◌ุ	ด	\0

รูปภาพที่ 4.1 ภาพแสดงการเก็บข้อความตัวอักษรภายในโหนดบัฟเฟอร์

4.2.1.2 บัฟเฟอร์ทำงาน (Work Buffer) เป็นบัฟเฟอร์ที่จะใช้เก็บข้อความเพื่อใช้ทำงานในบรรทัดขณะนั้น โดยจะนำข้อความตัวอักษรจากโหนดบัฟเฟอร์ที่เก็บข้อความอยู่ในบรรทัดเดียวกัน มาแยกไว้ในบัฟเฟอร์ทำงานตามชนิดระดับของตัวอักษร ดังนั้นโครงสร้างของข้อมูลนี้จะประกอบไปด้วยบัฟเฟอร์ 4 ระดับดังนี้คือ

- TOPEST ใช้เก็บตัวอักษรระดับบนสุด ได้แก่ วรรณยุกต์ต่างๆ คือ ่ ้ ๊ ๋
- UPPER ใช้เก็บตัวอักษรระดับบน ได้แก่ ตัวสระต่างๆ
- MIDDLE ใช้เก็บตัวอักษรระดับกลาง ได้แก่ ตัวพยัญชนะทุกตัว และ สระบางตัว
- BELOW ใช้เก็บตัวอักษรระดับล่าง ได้แก่ สระอุ (ุ) และ สระอู (ู)
- ATTR ใช้เก็บรหัสลักษณะพิเศษของตัวอักษร

โดยการทำงานทุกอย่างไม่ว่าจะเป็นการแก้ไข แทรก หรือลบข้อความตัวอักษรที่ใช้งานอยู่ในบรรทัดขณะนั้น จะทำอยู่ในบัฟเฟอร์ทำงานตลอด โดยในรูปที่ 4.2 แสดงแสดงการนำข้อความจากโหนดบัฟเฟอร์ในรูปที่ 4.1 มาเก็บไว้ในบัฟเฟอร์ทำงาน ซึ่งจะเห็นว่าตัวอักษรจะมีการแยกเก็บไว้บัฟเฟอร์ทำงานตามชนิดระดับของตัวอักษร

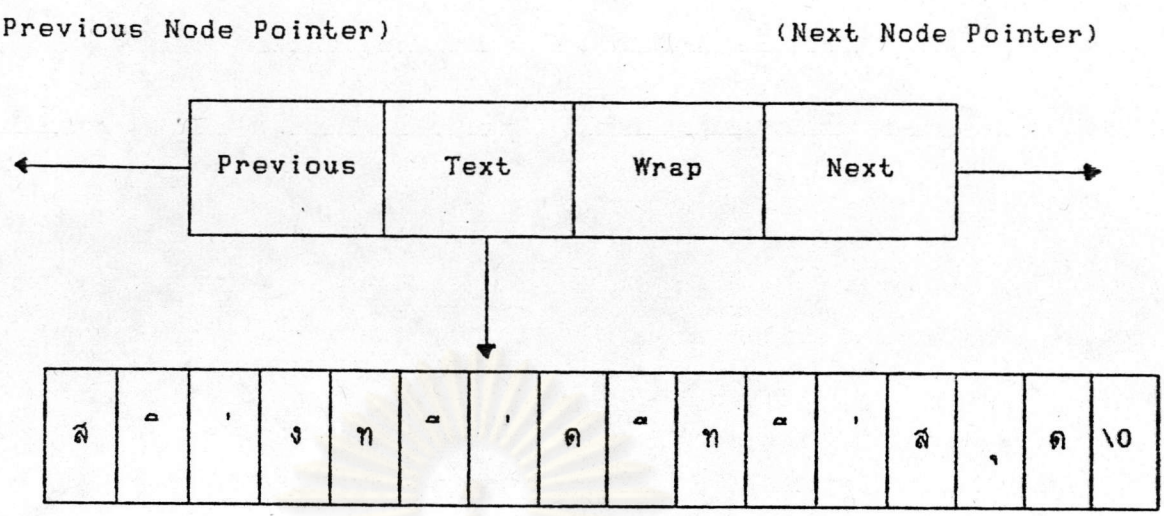


รูปภาพที่ 4.2 แสดงการนำข้อความจากโหนดบัฟเฟอร์ มาเก็บไว้ในบัฟเฟอร์ทำงาน

4.2.1.3 โหนดพอยน์เตอร์ (Node Pointer) เป็นชุดของพอยน์เตอร์ที่จะชี้ไปยังแต่ละโหนดบัฟเฟอร์ โดยโครงสร้างของโหนดพอยน์เตอร์จะใช้เป็นแบบ Double Link List ซึ่งจะประกอบไปด้วยดังนี้คือ

- Previous เป็น Pointer ที่จะชี้ไปยังโหนดพอยน์เตอร์ก่อนหน้า
- Next เป็น Pointer ที่จะชี้ไปยังโหนดพอยน์เตอร์ถัดไป
- Text เป็น Pointer ที่จะชี้ไปยังที่เก็บข้อความตัวอักษร
- Warp เป็น Flag ที่แสดงถึงการตัดคำของแต่ละบรรทัดนั้น

ดังในรูปภาพที่ 4.3 แสดงโครงสร้างของโหนดพอยน์เตอร์



รูปภาพที่ 4.3 แสดงโครงสร้างของโหนดพอยน์เตอร์แบบ Double Link list

หมายเหตุ

ดังนั้นในแต่ละบรรทัด (line) ของการประมวลผลคำ จะประกอบด้วย 2 ส่วนคือ โหนดบัฟเฟอร์และโหนดพอยน์เตอร์ทำงานร่วมกัน โดยโหนดพอยน์เตอร์จะเป็นตัวชี้ให้มีการเชื่อมต่อโหนดบัฟเฟอร์เข้าด้วยกันได้

4.2.2 แฟ้มข้อมูล (File)

แฟ้มข้อมูลที่ใช้เป็นลักษณะของแฟ้มข้อมูลแบบข้อความ (Text File) สำหรับเก็บข้อมูลเรียงตามลำดับการนำเข้า หลังจากแก้ไขข้อความภายในบัฟเฟอร์แล้ว และต้องการเก็บในแฟ้มข้อมูล โดยการนำข้อความของบรรทัดตามลำดับค่าตัวชี้เก็บในแฟ้มข้อมูล แต่ละบรรทัดจะมีตัวอักษร null (\n) กำหนดการสิ้นสุดบรรทัดเป็นตัวสุดท้าย ดังนั้นข้อความของแต่ละบรรทัดจะเก็บเรียงต่อกันในแฟ้มข้อมูล ดังแสดงในรูปภาพที่ 4.4

สถาบันศึกษา \n	โปรแกรมพิมพ์ \n	โหนดบัฟเฟอร์ \n

รูปภาพที่ 4.4 แสดงการเก็บข้อความในแฟ้มข้อมูลแบบข้อความ

4.2.3 ค่าคงที่ทั่วไป (Global Constant)

ค่าคงที่ทั่วไป (Global Constant) จะเป็นค่าคงที่ใช้ได้กับทุกๆโมดูลที่อ้างถึง โดยค่าคงที่ทั่วไปนี้จะถูกกำหนดไว้ตั้งแต่ตอนแรกของโปรแกรม และจะไม่มีการเปลี่ยนแปลงค่าเลย ค่าคงที่ที่สำคัญต่างๆ มีดังนี้คือ

ชื่อค่าคงตัว	ค่าที่เก็บไว้	คำอธิบาย
SHRINK_FACTOR	10	ค่าที่ใช้ปรับช่วงเว้นขอบซ้าย
CENTER_FACTOR	5	ค่าที่ใช้ปรับช่วงเว้นตรงกลาง
MAXCOL	256	ค่ามากสุดของขนาดบัฟเฟอร์
YES	1	ใช่
NO	0	ไม่ใช่
ERROR	1	รหัสความผิดพลาด
TEXT	0	โหมดตัวอักษร
GRAPHIC	1	โหมดกราฟฟิค
TOPEST	3	ระดับบนสุด
UPPER	2	ระดับบน
MIDDLE	0	ระดับกลาง
BELOW	1	ระดับล่าง
BOLDCODE	2	ค่ารหัสควบคุมตัวอักษรตัวเข้ม

ENLARGECODE	5	ค่ารหัสควบคุมตัวอักษรตัวใหญ่
TWOLINECODE	18	ค่ารหัสควบคุมขีดเส้นใต้ 2 เส้น
ONELINECODE	19	ค่ารหัสควบคุมขีดเส้นใต้ 1 เส้น
ITALICCODE	23	ค่ารหัสควบคุมตัวอักษรตัวเอนเอียง
SUPERCODE	20	ค่ารหัสควบคุมตัวอักษรตัวยกขึ้น
SUBCODE	22	ค่ารหัสควบคุมตัวอักษรตัวห้อย
ONELINEATTR	0x01	ลักษณะตัวอักษรขีดเส้นใต้ 1 เส้น
REVERSEATTR	0x02	ลักษณะตัวอักษรพื้นขาวตัวอักษรดำ
BOLDATTR	0x04	ลักษณะตัวอักษรตัวเข้ม
SUPERATTR	0x08	ลักษณะตัวอักษรตัวยกขึ้น
SUBATTR	0x10	ลักษณะตัวอักษรตัวห้อย
ITALICATTR	0x20	ลักษณะตัวอักษรตัวเอน
ENLARGEATTR	0x40	ลักษณะตัวอักษรตัวใหญ่
TWOLINEATTR	0x80	ลักษณะตัวอักษรขีดเส้นใต้ 2 เส้น
WRAPCODE	0x8d	รหัสการตัดคำ
WRAPBLANK	0xa0	รหัสตัวอักษรช่องว่าง
.THAIENG	0	โหมดรับตัวอักษรจากคีย์บอร์ด แบบภาษาไทยและภาษาอังกฤษ
ENGLISH	1	โหมดรับตัวอักษรจากคีย์บอร์ด แบบภาษาอังกฤษเท่านั้น
ENGUPCASE	2	โหมดรับตัวอักษรจากคีย์บอร์ด แบบภาษาอังกฤษตัวใหญ่เท่านั้น
NUMBER	3	โหมดรับตัวอักษรจากคีย์บอร์ด แบบตัวเลขเท่านั้น

นอกจากนี้แล้วยังมีค่าคงที่ทั่วไปอีกเป็นจำนวนมาก ซึ่งจะได้แสดงไว้ในภาคผนวก แล้ว

4.2.4 ตัวแปรทั่วไป (Global Variable)

ตัวแปรทั่วไป (Global Variable) จะเป็นตัวแปรซึ่งใช้ได้กับทุกๆโมดูลที่อ้างถึง ดังนั้นถ้ามีการเปลี่ยนแปลงค่าในตัวแปรเหล่านี้แล้ว ก็จะมีผลกระทบกับการทำงานของทุกๆโมดูลที่เกี่ยวข้องด้วย ตัวแปรที่สำคัญต่างๆ มีดังนี้คือ

4.2.4.1 sentinel curline curpage

เป็นตัวแปรชนิดพอยน์เตอร์ที่จะชี้ไปยังโหนดพอยน์เตอร์ ซึ่งโครงสร้างเป็นแบบ Double Link List ประกอบด้วยข้อมูลดังนี้ คือ

- Previous เป็นพอยน์เตอร์ที่จะชี้ไปยังโหนดพอยน์เตอร์ก่อนหน้า
- Next เป็นพอยน์เตอร์ที่จะชี้ไปยังโหนดพอยน์เตอร์ถัดไป
- Text เป็นพอยน์เตอร์ที่จะชี้ไปยังที่เก็บข้อความตัวอักษร
- Wrap เป็น Flag ที่แสดงถึงการตัดคำของแต่ละบรรทัด

ตัวแปร sentinel จะใช้ชี้ไปยังโหนดพอยน์เตอร์ ซึ่งทำหน้าที่เป็นหัว (Head) ของบรรทัด โดย Previous ของโหนดพอยน์เตอร์นี้ จะชี้ไปยังโหนดพอยน์เตอร์ตัวท้ายสุด ทำให้สามารถเรียกใช้ข้อมูลบรรทัดท้ายสุดได้ และ Next ของโหนดพอยน์เตอร์จะชี้ไปยังโหนดพอยน์เตอร์ตัวแรกสุด ทำให้สามารถเรียกใช้ข้อมูลบรรทัดแรกสุดได้ด้วย

ตัวแปร curline จะชี้ไปยังโหนดพอยน์เตอร์ ซึ่งใช้งานในบรรทัดนั้นๆเพื่อแสดงถึงบรรทัดที่ใช้งานอยู่ขณะนั้นได้

ตัวแปร curpage จะชี้ไปยังโหนดพอยน์เตอร์ ซึ่งใช้งานในบรรทัดนั้นๆเช่นเดียวกับตัวแปร curline แต่จะเปลี่ยนแปลงไปทันทีเมื่อมีการเลื่อนบรรทัดไป

4.2.4.2 thaimode

เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ใช้เพื่อบอกถึงสภาวะการใช้ตัวอักษรในขณะนั้น

- ถ้าเป็นค่า 1 แสดงว่า ให้ขณะนั้นใช้เป็นตัวอักษรภาษาไทย
- ถ้าเป็นค่า 0 แสดงว่า ให้ขณะนั้นใช้เป็นตัวอักษรภาษาอังกฤษ

4.2.4.3 stdcode

เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ใช้เพื่อบอกถึงสภาวะการใช้รหัสตัวอักษรภาษาไทย

- ถ้าเป็นค่า 1 แสดงว่า ขณะนั้นใช้เป็นภาษาไทยรหัส สมอ.
- ถ้าเป็นค่า 0 แสดงว่า ขณะนั้นใช้เป็นภาษาไทยรหัส เกษตร

4.2.4.4 insertmode

เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ใช้เพื่อบอกถึงสภาวะการพิมพ์แบบแทรกตัวอักษร

- ถ้าเป็นค่า 1 แสดงว่า ขณะนั้นสามารถพิมพ์ตัวอักษรแทรกได้
- ถ้าเป็นค่า 0 แสดงว่า ขณะนั้นสามารถพิมพ์ตัวอักษรทับได้

4.2.4.5 wordwrap

เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ใช้เพื่อบอกถึงสภาวะการตัดคำของข้อความ

- ถ้าเป็นค่า 1 แสดงว่า ขณะนั้นสามารถมีการตัดคำของข้อความที่กำลังพิมพ์ตัวอักษรยาวเกินช่วงเว้นขอบขวาไป
- ถ้าเป็นค่า 0 แสดงว่า ขณะนั้นไม่ต้องมีการตัดคำของข้อความที่กำลังพิมพ์ตัวอักษรยาวเกินช่วงเว้นขอบขวาและจะพิมพ์ยาวต่อไปได้อีก

4.2.4.6 changeflag

เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ใช้เพื่อบอกถึงสภาวะที่มีการเปลี่ยนแปลงของข้อความ

- ถ้าเป็นค่า 1 แสดงว่า ขณะนั้นได้มีการเปลี่ยนแปลงของข้อความ ในบรรทัดใดๆแล้ว
- ถ้าเป็นค่า 0 แสดงว่า ขณะนั้นยังไม่มีการเปลี่ยนแปลงของข้อความใดๆเลย

4.2.4.7 replaceflag

เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ใช้เพื่อบอกถึงสภาวะที่มีการค้นหาข้อความเดิมที่ต้องการ แล้วแทนที่ด้วยข้อความใหม่

- ถ้าเป็นค่า 1 แสดงว่า ขณะนั้นเมื่อค้นหาข้อความเดิมที่ต้องการพบแล้ว ให้แทนที่ด้วยข้อความใหม่
- ถ้าเป็นค่า 0 แสดงว่า ขณะนั้นค้นหาข้อความ ที่ต้องการเท่านั้น ไม่ต้องแทนที่ด้วยข้อความใหม่

4.2.4.8 relmargin

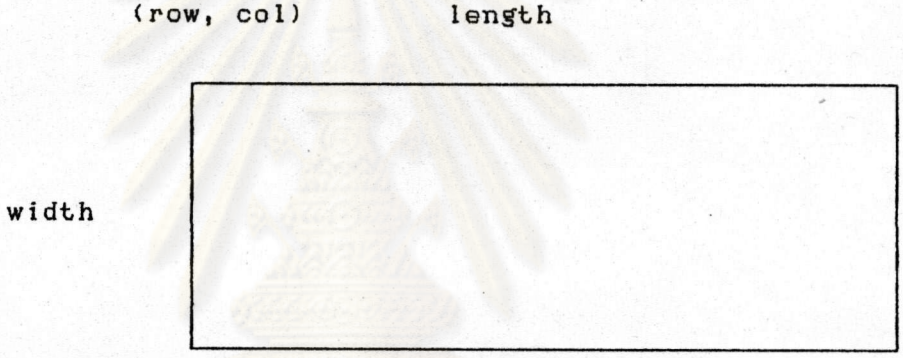
เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ใช้เพื่อบอกถึงสภาวะการปล่อยขอบของช่วงเว้นขอบขวา

- ถ้าเป็นค่า 1 แสดงว่า ขณะนั้นสามารถพิมพ์ตัวอักษรยาวเกินช่วงเว้นขอบขวาไปได้ โดยไม่มีการตัดคำของข้อความ
- ถ้าเป็นค่า 0 แสดงว่า ขณะนั้นไม่สามารถพิมพ์ตัวอักษร ยาวเกินช่วงเว้นขอบขวาไปได้ โดยจะมีการตัดคำของข้อความ ที่ยาวเกินขึ้นบรรทัดใหม่

4.2.4.9 **wind**

เป็นตัวแปรโครงสร้าง ที่ประกอบด้วยชุดของตัวเลขจำนวนเต็ม ใช้สำหรับกำหนดขอบเขตของหน้าต่าง (window) เพื่อแสดงข้อความบนจอภาพได้ ข้อมูลจะประกอบด้วยดังนี้คือ

- row ตำแหน่งของบรรทัดหน้าต่าง ทางมุมบนซ้ายสุด
- col ตำแหน่งของคอลัมน์หน้าต่าง ทางมุมบนซ้ายสุด
- length จำนวนของคอลัมน์ที่จะแสดงข้อความบนจอภาพ
- width จำนวนของบรรทัดต่อหน้าที่จะแสดงข้อความบนจอภาพ



4.2.4.10 **workline**

เป็นตัวแปรโครงสร้าง ที่ประกอบด้วยชุดของข้อมูลตัวอักษร ซึ่งเรียกว่า บัฟเฟอร์ทำงาน ใช้สำหรับเก็บข้อความเพื่อใช้ทำงานในบรรทัดขณะนั้น โดยโครงสร้างจะประกอบไปด้วยข้อมูลดังนี้ คือ

- topest ใช้เก็บตัวอักษรระดับบนสุด
- upper ใช้เก็บตัวอักษรระดับบน
- middle ใช้เก็บตัวอักษรระดับกลาง
- below ใช้เก็บตัวอักษรระดับล่าง
- attr ใช้เก็บรหัสลักษณะพิเศษของตัวอักษร

4.2.4.11 lineno

เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ใช้เก็บหมายเลขบรรทัดของบรรทัดที่ใช้งานอยู่ขณะนั้น จะเริ่มต้นจากค่า 1 เสมอ

4.2.4.12 macro

เป็นตัวแปรชุดชนิดตัวอักษร ขนาด 10 (แถว) x 36 (คอลัมน์) ใช้สำหรับเก็บข้อความของคำย่อ (macro word) ที่จะได้จากการกดคีย์ที่ต้องการ โดยข้อความจะถูกเก็บไว้เป็นชุดๆจำนวน 10 ชุด ในแต่ละชุดจะเก็บได้ 36 ตัวอักษร ดังนั้นจะใช้คีย์กดแทนข้อความได้ 10 คีย์ ในแต่ละคีย์จะเก็บข้อความได้มากสุด 36 ตัวอักษร รวมตัวอักษรที่จะเก็บได้มากสุดถึง 360 ตัวอักษร

4.2.4.13 firstcol

เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ใช้สำหรับเก็บหมายเลขคอลัมน์เริ่มต้น ที่จะใช้แสดงข้อความบนจอภาพ

4.2.4.14 fontused

เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ใช้สำหรับเก็บค่าตัวเลขที่ใช้แสดงลักษณะพิเศษของตัวอักษร (attribute of character) ในขณะที่ใช้งานอยู่นั้น

4.2.4.15 dispblock

เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ใช้เพื่อบอกถึงสภาวะที่มีการแสดงข้อความที่เป็นบล็อก

- ถ้าเป็นค่า 1 แสดงว่า ให้ขณะนั้นมีการแสดงข้อความที่เป็นบล็อกอยู่บนจอภาพ (จะสังเกตุเห็นข้อความมีแสงสว่างทึบคลุมข้อความที่เป็นบล็อก)

- ถ้าเป็นค่า 0 แสดงว่า ให้ขณะนั้นไม่มีการแสดงข้อความที่เป็นบล็อก บนจอภาพ

4.2.4.16 quitprog

เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ใช้เพื่อบอกถึงสภาวะการเลิกทำงานของโปรแกรม

- ถ้าเป็นค่า 1 แสดงว่า ต้องการจะเลิกทำงานจากโปรแกรม
- ถ้าเป็นค่า 0 แสดงว่า ไม่ต้องการจะเลิกทำงานจากโปรแกรม

4.2.4.17 filename

เป็นตัวแปรชนิดตัวอักษร ใช้สำหรับเก็บชื่อของไฟล์ที่ระบุไว้

4.2.4.18 tab

เป็นตัวแปรชุดชนิดตัวเลข ใช้เพื่อบอกถึงสภาวะการแสดงเครื่องหมายแท็ป (TAB) บนจอภาพในแต่ละคอลัมน์

- ถ้าเป็นค่า 1 แสดงว่า ให้แสดงเครื่องหมายแท็ปได้ในคอลัมน์ที่ใช้งานอยู่
- ถ้าเป็นค่า 0 แสดงว่า ไม่ต้องการแสดงเครื่องหมายแท็ป

4.2.4.19 leftmar และ rightmar

เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ใช้เพื่อเก็บหมายเลขตำแหน่งของช่วงเว้นขอบซ้ายและขอบขวาไว้

4.2.4.20 pagecomplete

เป็นตัวแปรชนิดตัวเลข ใช้เพื่อบอกถึงสภาวะการแสดงข้อความบนจอภาพที่สมบูรณ์แบบ

- ถ้าเป็นค่า 1 แสดงว่า มีการแสดงข้อความบนจอภาพสมบูรณ์แบบ โดยไม่มีการขัดจังหวะด้วยการกดคีย์ ในขณะที่กำลังแสดงข้อความ

- ถ้าเป็นค่า 0 แสดงว่า ในขณะที่กำลังแสดงข้อความมีการขัดจังหวะด้วยการกดคีย์ เป็นการแสดงข้อความที่ไม่สมบูรณ์

4.2.4.21 source replace option

เป็นตัวแปรชุดชนิดตัวอักษร ที่มีขนาดเป็น 80 ตัวอักษร เพื่อจะใช้ในการค้นหาและแทนที่ข้อความที่ต้องการ จะมีตัวแปรที่ใช้ได้ทั้งหมด 3 ตัวคือ

source จะใช้เก็บข้อความที่จะต้องการค้นหา

replace จะใช้เก็บข้อความใหม่ที่ใช้แทนที่ข้อความเดิม

option จะใช้เก็บตัวเลือกของวิธีในการค้นหาข้อความ

4.2.4.22 lineperpage

เป็นตัวแปรชนิดตัวเลข ใช้เพื่อเก็บค่าตัวเลขจำนวนบรรทัดที่พิมพ์ต่อหน้า

4.2.4.23 pagebreak

เป็นตัวแปรชนิดตัวเลข ใช้เพื่อบอกสภาวะการแสดงเครื่องหมายแบ่งหน้าบนจอภาพ

- ถ้าเป็นค่า 1 แสดงว่า ต้องการแสดงเครื่องหมายแบ่งหน้าบนจอภาพ เมื่อแสดงข้อความเป็นจำนวนครบ 1 หน้า
- ถ้าเป็นค่า 0 แสดงว่า ในขณะที่กำลังแสดงข้อความ มีการขัดจังหวะด้วยการกดคีย์ เป็นการแสดงข้อความที่ไม่สมบูรณ์

4.2.4.24 blkbegin blkend

เป็นตัวแปรโครงสร้าง ประกอบด้วยชุดของข้อมูลตัวเลขใช้สำหรับเก็บข้อมูลตัวเลขที่เป็นตำแหน่งเริ่มต้นและสุดท้ายของบล็อก โครงสร้างจะประกอบไปด้วยข้อมูลดังนี้ คือ

- lineno ใช้เก็บหมายเลขบรรทัด
- column ใช้เก็บหมายเลขคอลัมน์

4.2.4.25 headdir dirpage

เป็นตัวแปรโครงสร้างชนิดพอยน์เตอร์ ที่ประกอบด้วยชุดของข้อมูลตัวอักษรและพอยน์เตอร์ ใช้สำหรับชี้ไปยังชื่อไฟล์ในไดเรกทอรี่ (Dirctory) โครงสร้างจะประกอบไปด้วยข้อมูลดังนี้ คือ

- filename ใช้เก็บชื่อไฟล์
- previous ใช้เก็บพอยน์เตอร์ที่จะชี้ไปยังชื่อไฟล์ก่อนหน้า
- next ใช้เก็บพอยน์เตอร์ที่จะชี้ไปยังชื่อไฟล์ถัดไป

ตัวแปร headdir เป็นพอยน์เตอร์ที่จะใช้ชี้ไปยังชื่อไฟล์แรก ที่มี previous ชี้ไปยังชื่อไฟล์สุดท้ายในไดเรกทอรี่ และ next ชี้ไปยังชื่อไฟล์ตัวที่สองถัดไป

ตัวแปร dirpage เป็นพอยน์เตอร์ใช้ชี้ไปยังชื่อไฟล์ ที่จะใช้งานขณะนั้น

4.2.4.26 dfont ditalicfont

เป็นตัวแปรชุดชนิดตัวอักษร ที่มีขนาดเป็น 5120 ตัวอักษร ใช้เก็บรูปแบบภาพของตัวอักษร (font) ที่อ่านมาจากไฟล์ NORMAL.FON และ ITALIC.FON โดยตัวแปร dfont จะเก็บรูปแบบภาพของตัวอักษรปกติ และ ตัวแปร ditalicfont จะเก็บรูปแบบภาพของตัวอักษรตัวเอียง

4.2.4.27 cuprint_dir

เป็นตัวแปรชุดชนิดตัวอักษร ที่มีขนาดเป็น 40 ตัวอักษร จะใช้เก็บชื่อไดเรกทอรี่ของโปรแกรม CUPRINT.EXE เพื่อให้สามารถเรียกใช้โปรแกรมนี้ได้

4.2.4.28 cw_dir

เป็นตัวแปรชุดชนิดตัวอักษร ที่มีขนาดเป็น 40 ตัวอักษร จะใช้เก็บชื่อไดเรกทอรี่ของไฟล์ CW.DAT เพื่อให้สามารถเรียกใช้รูปแบบที่กำหนดได้

4.3 หลักการทำงานของการประมวลผลคำในโปรแกรมซียูไรท์เตอร์

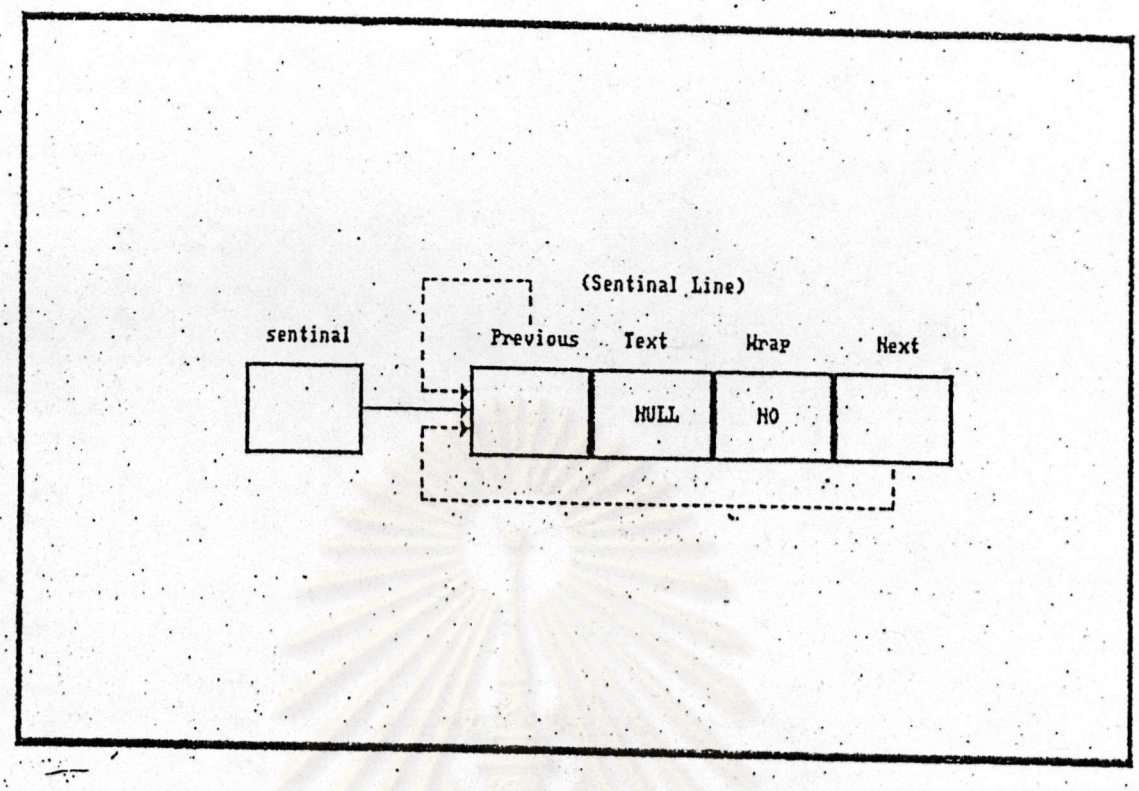
หลักการทำงานภายในของการประมวลผลคำในโปรแกรมซียูไรท์เตอร์เดิม จะมีความซับซ้อนและเกี่ยวพันกันมาก จึงจะต้องมีการปรับปรุงหลักการทำงานภายในให้แยกได้ออกเป็นโมดูลส่วนย่อย (module) เพื่อให้เรียกใช้งานได้สะดวกขึ้น โดยหลักการทำงานภายในมีลักษณะสำคัญต่างๆดังนี้คือ

4.3.1 การจัดเตรียมบัฟเฟอร์

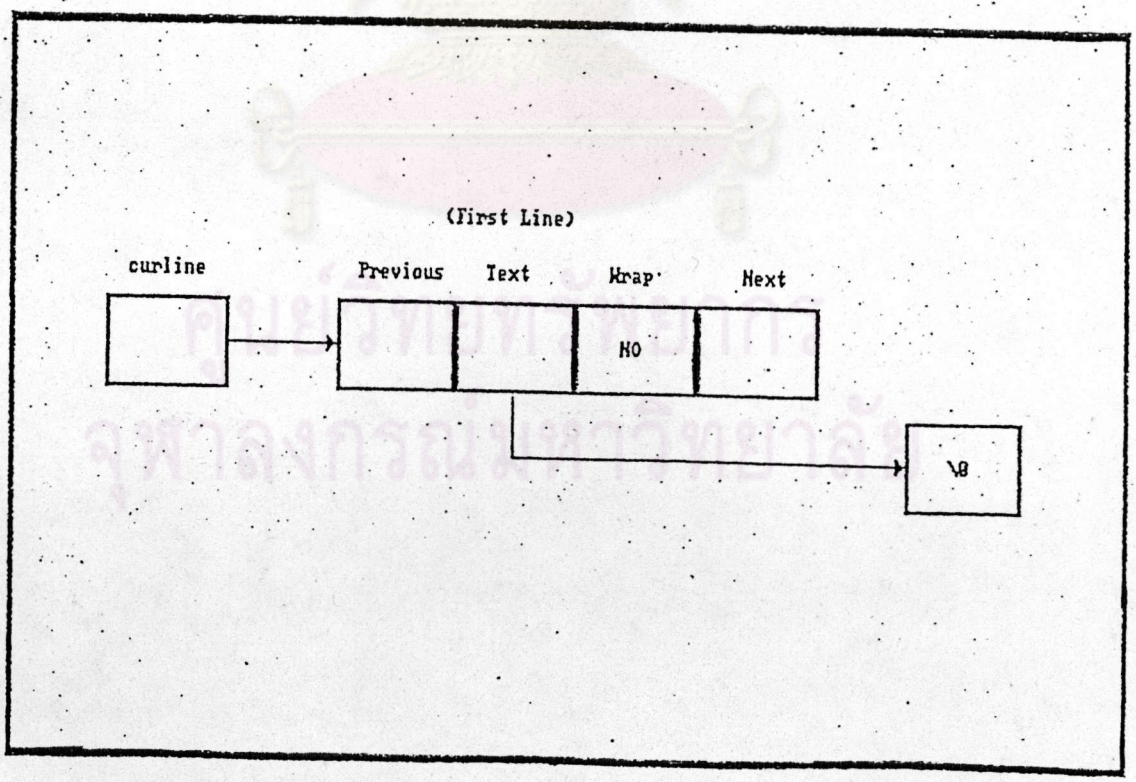
จะเป็นการจัดเตรียมโหนดบัฟเฟอร์และบัฟเฟอร์ทำงานเป็นครั้งแรกซึ่งใช้โหนดพอยน์เตอร์เป็นตัวชี้ไปยังโหนดบัฟเฟอร์ การทำงานจะอยู่ภายใต้โมดูล setupnode

การทำงาน จะมีขั้นตอนดังนี้คือ

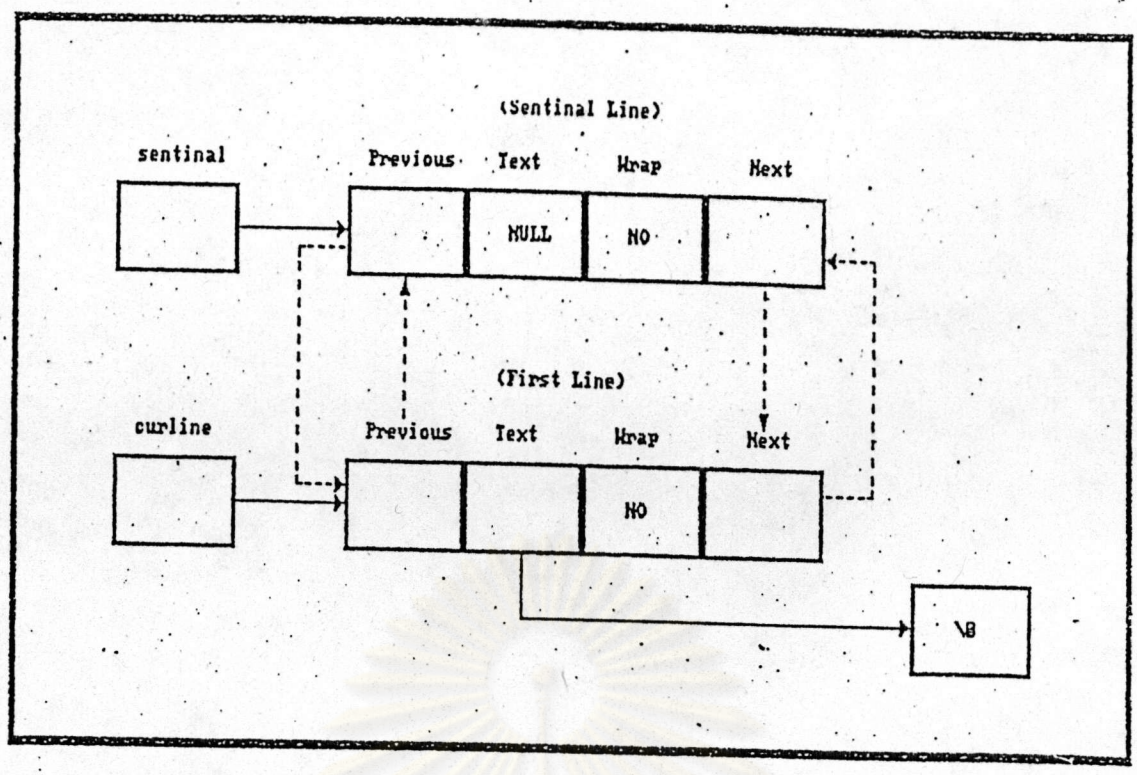
1. โปรแกรมจะจัดเตรียม พอยน์เตอร์ตัวแรกสุดที่เรียกว่า "Sentinal Pointer" เพื่อชี้ไปยัง โหนดพอยน์เตอร์ตัวแรกสุดที่เรียกว่า "Sentinal Line" โดยจะกำหนดให้ Previous Pointer และ Next Pointer ชี้ไปยังโหนดพอยน์เตอร์ของตัวเอง และกำหนดให้ค่าของ Text Pointer เป็น NULL และ Wrap Flag เป็น NO ซึ่งจะได้ผลดังรูปที่ 4.5
2. จะสร้างพอยน์เตอร์ตัวถัดไปที่ชื่อว่า "Curline Pointer" ซึ่งจะกำหนดค่า Text Pointer ให้ชี้ไปยังโหนดบัฟเฟอร์หรือบรรทัดแรกใช้เก็บข้อความไว้ ซึ่งจะเรียกว่า "First Line" และ ให้ค่า Wrap Flag เป็น NO ซึ่งจะได้ผลดังรูปที่ 4.6
3. จะนำโหนดพอยน์เตอร์ทั้งสองตัวมาเชื่อมต่อกัน โดยจะให้ Previous Pointer และ Next Pointer ของ Sentinal Line ชี้ไปยังโหนดพอยน์เตอร์ของ First Line และขณะเดียวกันให้ Previous Pointer และ Next Pointer ของ First line ชี้ไปยังโหนดพอยน์เตอร์ของ Sentinal Line ซึ่งจะได้ผลดังรูปที่ 4.7
4. จะมีการเตรียมบัฟเฟอร์ทำงาน (Work Buffer) เพื่อใช้เก็บข้อความจากโหนดบัฟเฟอร์ โดยจะให้บัฟเฟอร์ทำงานเก็บค่าช่องว่าง ไว้ก่อน ซึ่งจะได้ผลดังรูปที่ 4.8



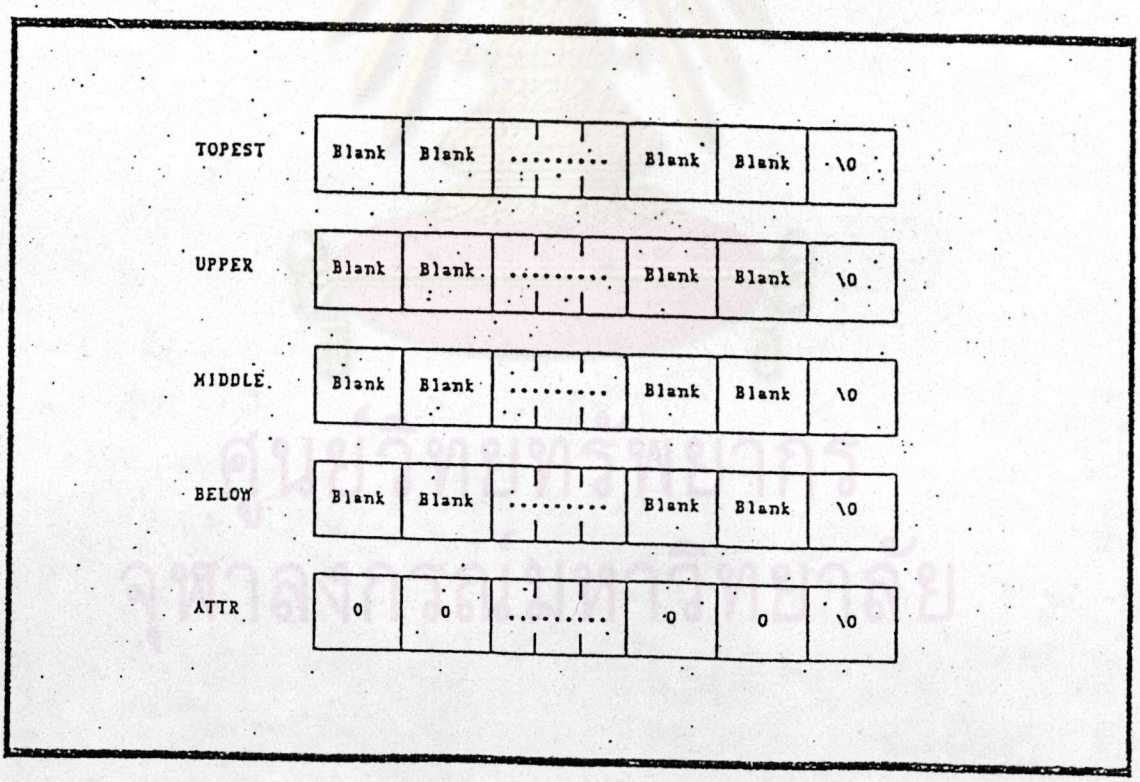
รูปภาพที่ 4.5 แสดงการทำงานของ sentinal pointer และ sentinal line



รูปภาพที่ 4.6 แสดงการทำงานของ curline pointer และ first line



รูปภาพที่ 4.7 แสดงการเชื่อมต่อของ sentinal line และ first line



รูปภาพที่ 4.8 แสดงการเตรียมบัฟเฟอร์ทำงานให้เป็นช่องว่างทั้งหมด

ดังนั้นจะเห็นว่า Sentinal Line จะทำหน้าที่เป็นตัวชี้ไปยังบรรทัดแรกด้วย Next Pointer และ เป็นตัวชี้ไปยังบรรทัดสุดท้ายด้วย Previous Pointer รวมทั้ง First Line จะทำหน้าที่เป็นตัวชี้ไปยังโหนดบัฟเฟอร์บรรทัดแรกด้วย

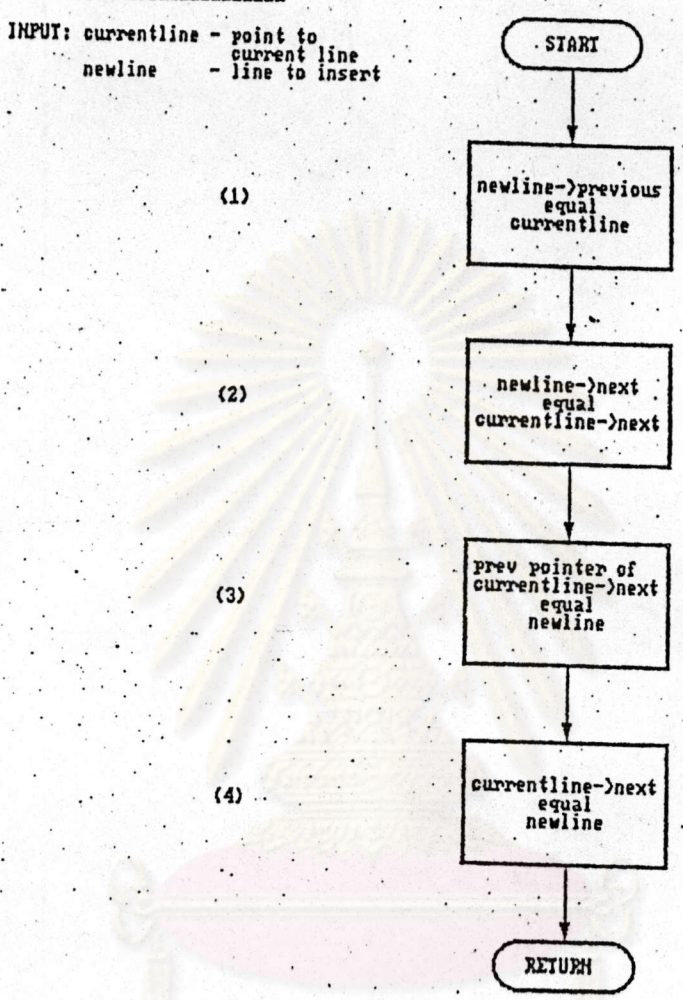
4.3.2 การแทรกบรรทัดของโหนดบัฟเฟอร์

จะเป็นลักษณะของการเก็บข้อความที่ต้องการแทรกลงไปในบรรทัดว่างที่สร้างขึ้นโดยบรรทัดว่างนี้ จะเปลี่ยนให้ตัวพอยน์เตอร์ ชี้ไปยังบรรทัดอื่นๆเพื่อให้เกิดความสัมพันธ์กัน ดังนั้นจึงไม่จำเป็นต้องเคลื่อนย้ายข้อความในบรรทัดของโหนดบัฟเฟอร์เลย การทำงานจะอยู่ภายใต้โมดูล insert_line โดยกำหนดโหนดพอยน์เตอร์ curline ให้ชี้ไปยังบรรทัดของโหนดบัฟเฟอร์ ที่จะต้องการแทรกบรรทัด และให้ newline ชี้ไปยังบรรทัดของโหนดบัฟเฟอร์ที่จะนำมาเชื่อมต่อกัน

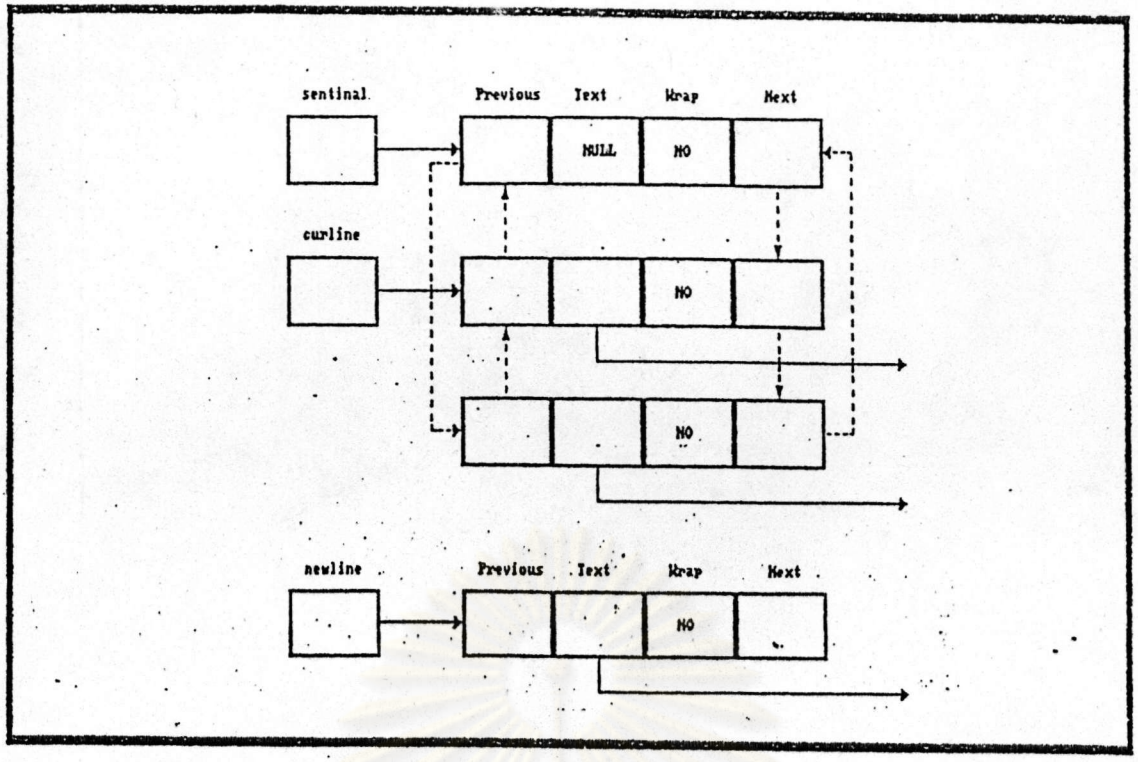
การทำงาน จะมีขั้นตอนดังนี้คือ

1. เปลี่ยนโหนดพอยน์เตอร์ของบรรทัดใหม่ ที่ถูกชี้ด้วยโหนดพอยน์เตอร์ newline โดยให้ previous pointer ชี้ไปยังบรรทัดที่เดียวกับ curline
2. เปลี่ยน next pointer ของบรรทัดที่ถูกชี้ด้วยโหนดพอยน์เตอร์ newline ให้ชี้ไปยังบรรทัดที่เดียวกับ next pointer ของโหนดพอยน์เตอร์ curline
3. เปลี่ยน previous pointer ของบรรทัดที่ถูกชี้ด้วย next pointer ของโหนดพอยน์เตอร์ curline ให้ชี้ไปยังบรรทัดที่เดียวกับ newline
4. เปลี่ยน next pointer ของโหนดพอยน์เตอร์ curline ให้ชี้ไปยังบรรทัดที่เดียวกับ newline

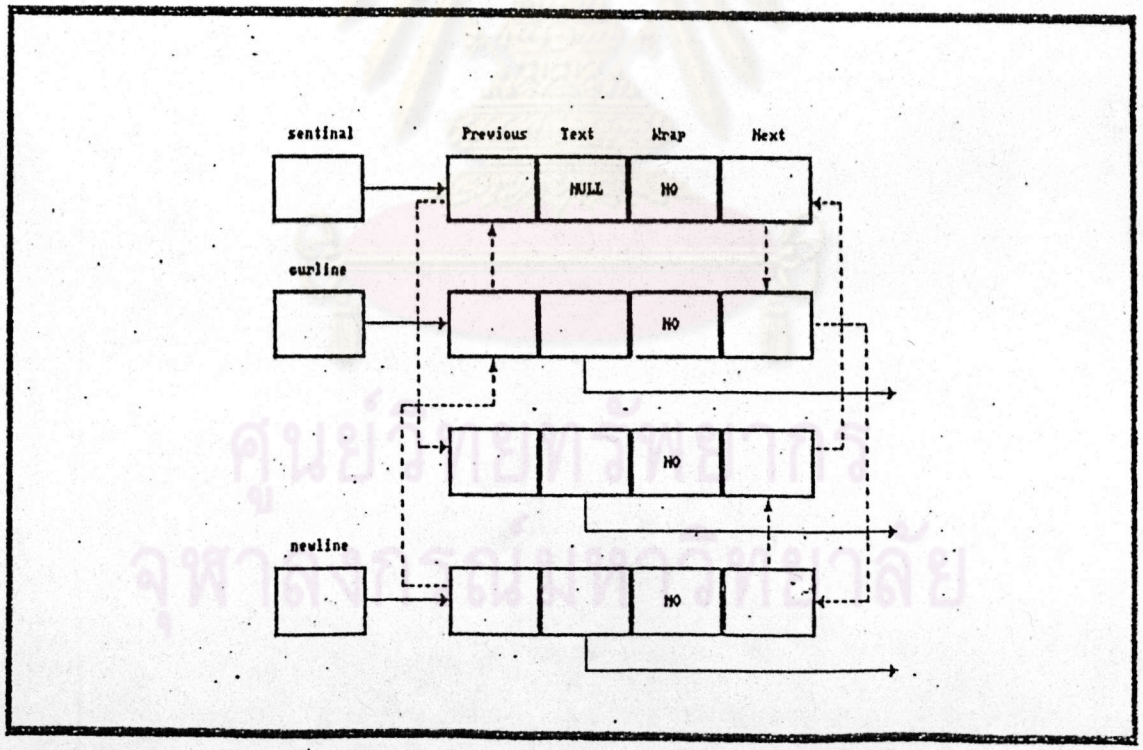
สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปที่ 4.9 แล้ว และในรูปที่ 4.10 แสดงบรรทัดก่อนการแทรกบรรทัด โดยในขณะนั้นมีบรรทัดของโหนดบัฟเฟอร์ที่ถูกชี้ด้วย curline ซึ่งเป็นบรรทัดที่ต้องการแทรกบรรทัด และมีโหนดบัฟเฟอร์จำนวนหนึ่งบรรทัดที่ถูกชี้ด้วย newline ซึ่งเป็นโหนดบัฟเฟอร์ใหม่ที่จะนำมาแทรกบรรทัด โดยเมื่อมีการแทรกบรรทัดแล้ว จะแสดงผลดังรูปที่ 4.11 โดยเราจะแทรกบรรทัดใหม่ที่ถูกชี้ด้วย newline เข้าไปอยู่ระหว่างบรรทัดทั้งสองนั้นได้



รูปภาพที่ 4.9 ผังงานแสดงขั้นตอนการทำงานของโมดูล insert_line



รูปภาพที่ 4.10 แสดงบรรทัดของโหนดบัฟเฟอร์ ก่อนการแทรกบรรทัด



รูปภาพที่ 4.11 แสดงบรรทัดของโหนดบัฟเฟอร์ หลังการแทรกบรรทัด

4.3.3 <u>การแทรกตัวอักษร</u>

จะเป็นการแทรกตัวอักษร 1 ตัว ในบัฟเฟอร์ทำงานของบรรทัดที่ใช้งานอยู่ขณะนั้นโดยจะต้องมีการเคลื่อนย้ายข้อความในบัฟเฟอร์ทำงาน ที่ตำแหน่งต้องการแทรก เพื่อให้เกิดช่องว่างที่จะนำตัวอักษรมาบรรจุใส่ได้นั้นเอง การทำงานจะอยู่ภายใต้โมดูล insert_char โดยจะกำหนดตัวอักษร c และตำแหน่ง x และ y ที่จะแทรกตัวอักษร

<u>การทำงาน</u> จะมีขั้นตอนต่างๆดังนี้คือ

1. ตรวจสอบว่า ขณะนั้นตำแหน่งคอลัมน์ x อยู่ในตำแหน่งที่เกินคอลัมน์สูงสุดของบรรทัดหรือไม่

   - ถ้าอยู่เกินคอลัมน์แล้วจะไม่มีการแทรกตัวอักษร และเลิกทำงานโดยให้ค่า NO
   - ถ้ายังไม่เกินคอลัมน์ จะทำงานตามขั้นตอนดังนี้คือ

   1.1 ตรวจสอบว่า ขณะนั้นสภาวะการตัดคำ (wordwrap) เป็นค่า YES และอยู่ในตำแหน่งที่เกินช่วงเว้นขอบขวา (rightmar) หรือไม่

   - ถ้าเป็นจริงแล้ว จะตรวจสอบสภาวะการปล่อยขอบของช่วงเว้นขอบขวา (relmargin) ว่าเป็นค่า NO หรือไม่ โดยถ้าเงื่อนไขเป็นจริง ก็จะไปทำงานในโมดูล "autowrap" เพื่อที่จะตัดคำในบรรทัดนั้น ให้ขึ้นบรรทัดใหม่อย่างอัตโนมัติ

   1.2. ทำงานในโมดูล "whatlevel" เพื่อตรวจระดับตัวอักษรที่ต้องการแทรกว่า เป็นตัวอักษรในระดับใด

   - ถ้าเป็นตัวอักษรระดับกลาง MIDDLE ก็จะทำการเคลื่อนย้ายข้อความในบัฟเฟอร์ ทำงานตั้งแต่คอลัมน์ x ที่ระบุไว้ทั้งบัฟเฟอร์ทำงาน 4 ระดับคือ TOPEST BUFFER UPPER BUFFER, MIDDLE BUFFER และ BELOW BUFFER รวมทั้ง ATTR BUFFER ด้วย โดยจะขยับไปทางขวามือ 1 ตำแหน่ง เพื่อให้เกิดช่องว่าง 1 ช่องที่จะนำตัวอักษรนั้นไปใส่ไว้ใน MIDDLE BUFFER

- ถ้าเป็นตัวอักษรระดับอื่นที่ไม่ใช่ตัวอักษรระดับกลาง MIDDLE จะไม่มีการเคลื่อนย้ายข้อความในบัฟเฟอร์ทำงาน แต่จะนำตัวอักษรนั้นไปใส่ในบัฟเฟอร์ทำงานตามค่าระดับของตัวอักษร โดยใส่ก่อนหน้าในตำแหน่งที่ระบุไว้ 1 ตำแหน่ง

1.3 ตรวจสอบตำแหน่งคอลัมน์ x ว่าขณะนั้นเกินจากค่าตำแหน่งขวาสุดของจอภาพหรือไม่ (ค่า length ของ wind)

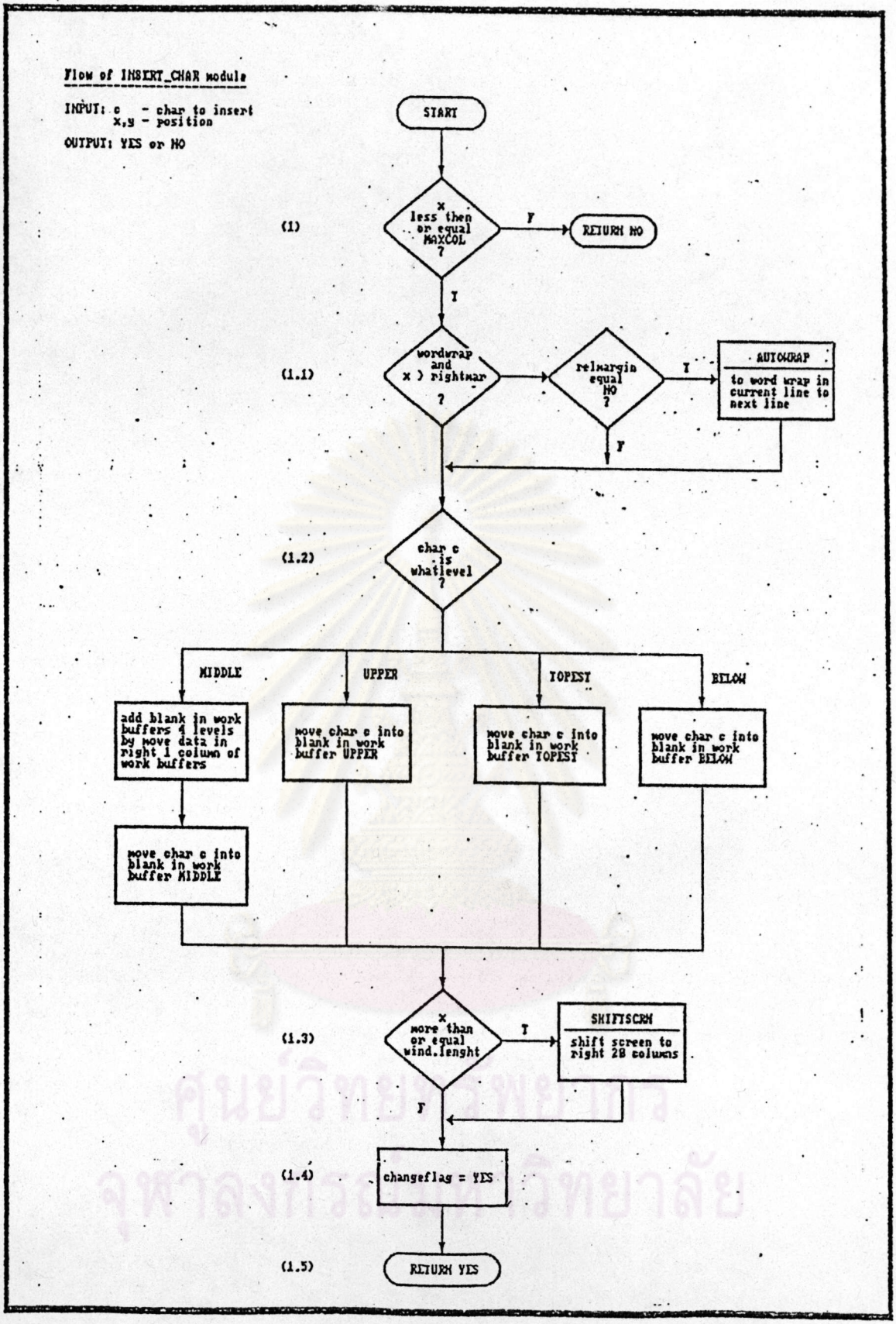
- ถ้าเกินตำแหน่งขวาสุดของจอภาพแล้ว จะทำงานในโมดูล "whatlevel" เพื่อตรวจระดับตัวอักษรว่าเป็นตัวอักษรระดับกลาง (MIDDLE) หรือไม่ ถ้าเป็นจริงแล้วจะไปทำงานในโมดูล "shiftscrn" เพื่อที่จะขยับจอภาพไปทางขวามือได้

1.4 กำหนดค่าของ changflag ให้เป็น YES เพื่อแสดงว่ามีการแก้ไขข้อความแล้ว

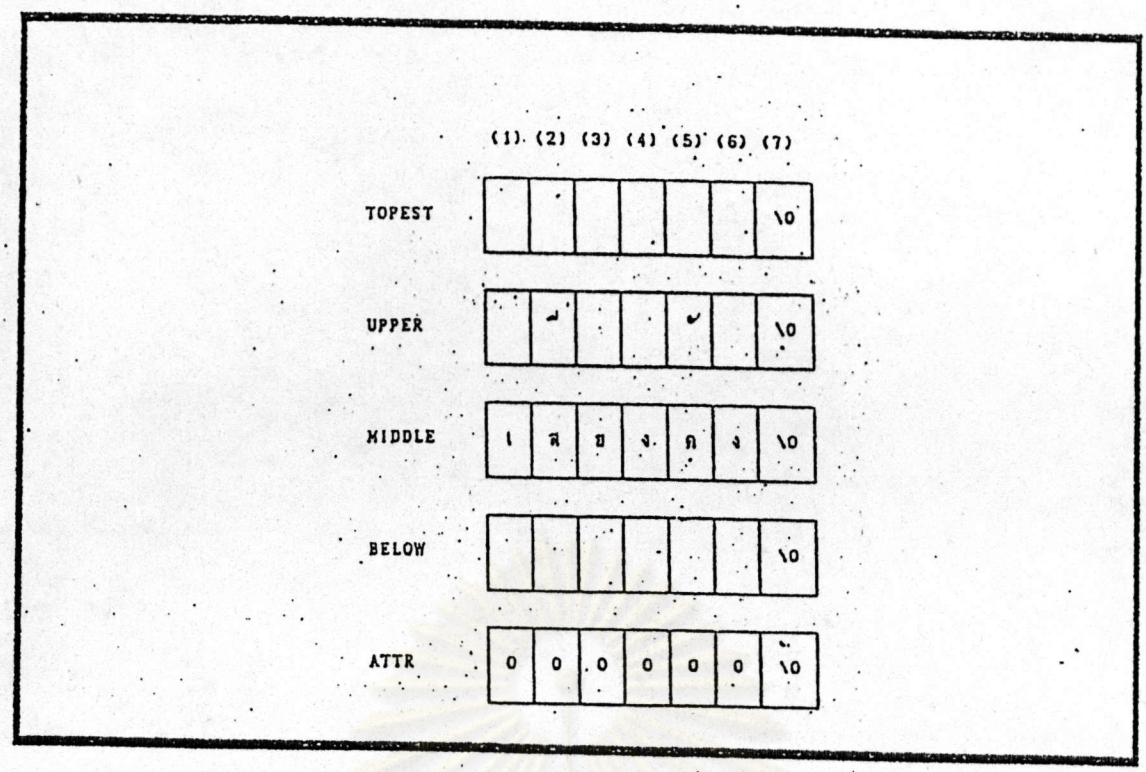
1.5. เลิกทำงาน โดยให้ค่า YES

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.12 แล้ว โดยสมมุติว่าจะมีการเก็บข้อความว่า "เสียงดัง" ไว้ในบัฟเฟอร์ทำงาน ดังแสดงในรูปภาพที่ 4.13 จะต้องการแทรกคำว่า "กู่" ไว้ในตำแหน่งที่ 5 ของบัฟเฟอร์ทำงาน จะต้องมีการทำทีละตัวอักษร โดยเริ่มต้นเคลื่อนย้ายข้อความตั้งแต่ตำแหน่งที่ 5 ของบัฟเฟอร์ทั้งหมดทางขวามือเพื่อให้เกิดช่องว่าง 1 ช่อง แล้วจะนำตัวอักษร "ก" มาใส่ในตำแหน่งที่ 5 ของบัฟเฟอร์ในระดับ MIDDLE ดังแสดงในรูปภาพที่ 4.14

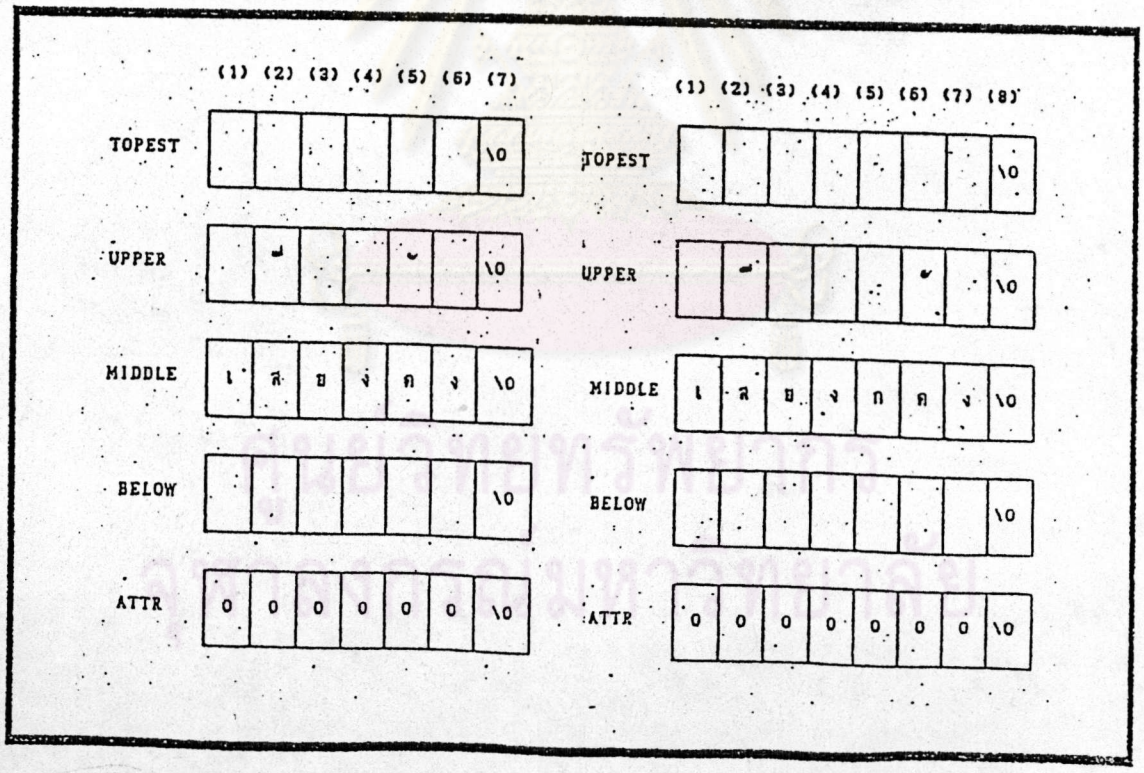
หลังจากนั้นแล้วก็จะนำตัวอักษร " ู " เพื่อจะนำตัวอักษรไปใส่ในบัฟเฟอร์ ในระดับ BELOW ของตำแหน่งที่ 5 ดังแสดงได้ดังรูป 4.15 และสุดท้าย ก็จะนำตัวอักษร " ' " ไปใส่ในบัฟเฟอร์ระดับ TOPEST ในตำแหน่งคอลัมน์ที่ 5 (ไม่เลื่อนตำแหน่ง) ดังแสดงได้ดังรูปภาพที่ 4.16



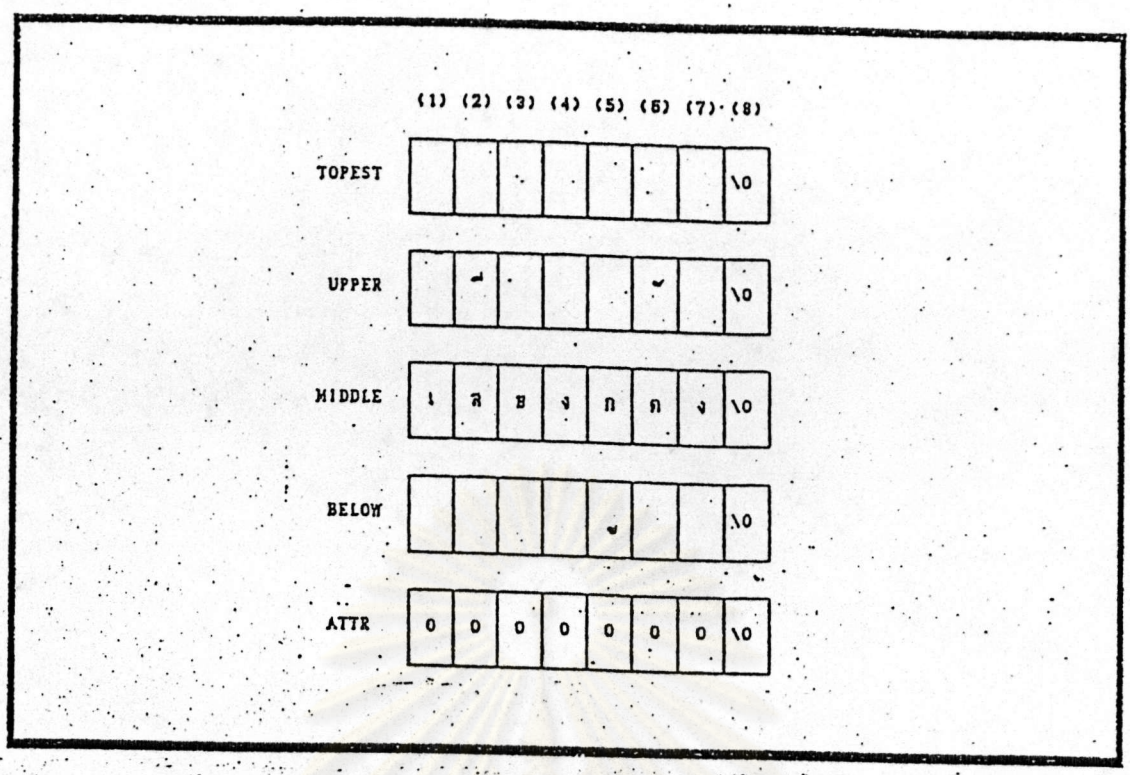
รูปภาพที่ 4.12 ผังงานแสดงขั้นตอนการทำงานของโมดูล insert_char



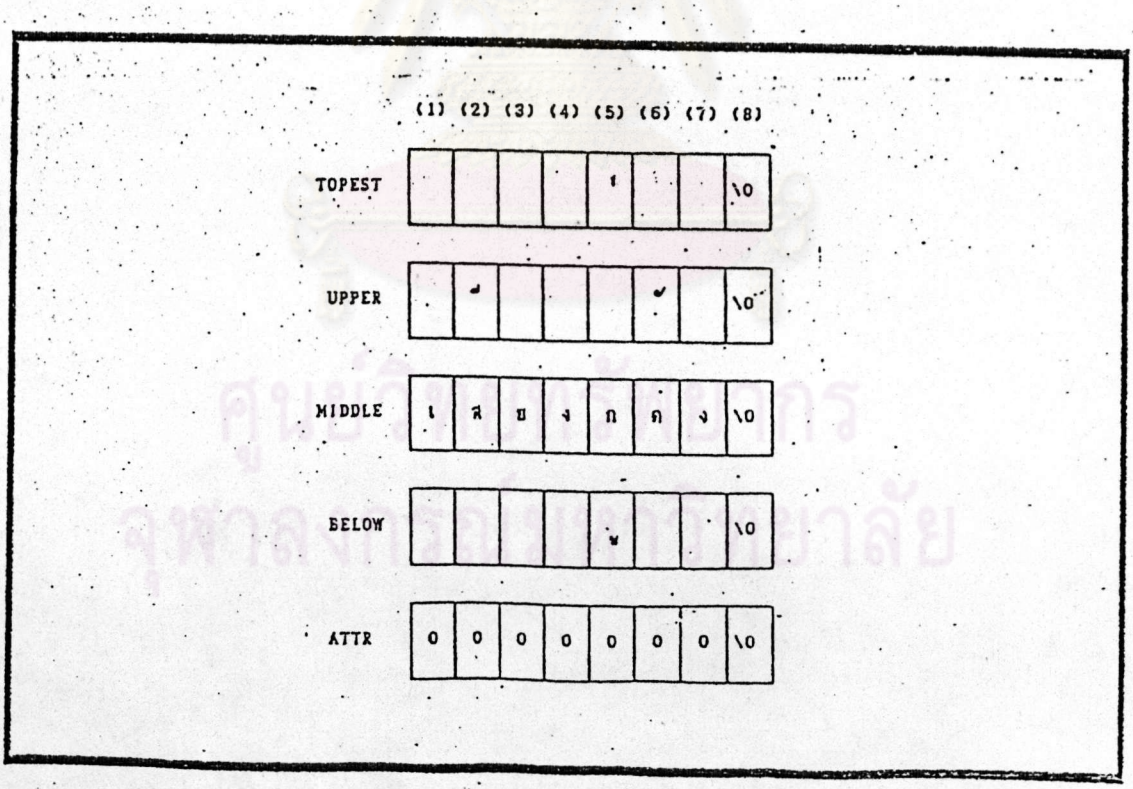
รูปภาพที่ 4.13 แสดงข้อความที่เก็บไว้ในบัฟเฟอร์ทำงานก่อนการแทรกตัวอักษร



รูปภาพที่ 4.14 แสดงข้อความการแทรกตัวอักษร 'ก' เข้าไปไว้ในบัฟเฟอร์ทำงาน MIDDLE BUFFER



รูปภาพที่ 4.15 แสดงข้อความการแทรกตัวอักษร ' ' เข้าไปไว้ในบัฟเฟอร์ทำงาน BELOW BUFFER



รูปภาพที่ 4.16 แสดงข้อความการแทรกตัวอักษร ' ' เข้าไปไว้ในบัฟเฟอร์ทำงาน TOP BUFFER

สำหรับการแทนที่ตัวอักษร จะมีการทำงานเช่นเดียวกันกับการแทรกตัวอักษร แต่จะไม่มีการเคลื่อนย้ายข้อความในบัฟเฟอร์ทำงานเลย และจะนำตัวอักษรที่ต้องการนั้นแทนที่ตัวอักษรเดิม ในตำแหน่งที่ระบุไว้ การทำงานจะอยู่ภายใต้โมดูล overwrite_char

4.3.4 การตัดข้อความในตำแหน่งเคอร์เซอร์ไปขึ้นบรรทัดใหม่

จะเป็นการตัดข้อความในตำแหน่งที่เคอร์เซอร์อยู่ขณะนั้น ให้เลื่อนลงไปขึ้นบรรทัดใหม่อีกบรรทัดหนึ่ง เมื่อเรากดคีย์ Enter หรือ Ctrl + M ในขณะอยู่ในโหมดพิมพ์แทรก (insert mode) การทำงานนี้จะอยู่ภายใต้โมดูล insert_ret โดยจะกำหนดตำแหน่งคอลัมน์ x ที่จะตัดข้อความ

การทำงาน จะมีขั้นตอนดังนี้คือ

1. ไปทำงานในโมดูล "storeline" เพื่อจะเรียกข้อมูลจากบัฟเฟอร์ทำงานมาเก็บไว้ในโหนดบัฟเฟอร์ที่ใช้งานอยู่ขณะนั้น
2. ไปทำงานในโมดูล "linearcolumn" เพื่อที่จะค้นหาตำแหน่งคอลัมน์ที่แท้จริงของข้อความในโหนดบัฟเฟอร์ที่ใช้งานอยู่ ซึ่งจะถูกชี้ด้วยโหนดพอยน์เตอร์ curline จากตำแหน่งคอลัมน์ x ที่กำหนดไว้ แล้วนำค่าตำแหน่งนั้นไปเก็บไว้ในตัวแปร i
3. สร้างโหนดพอยน์เตอร์ line2 ขึ้นใหม่ เพื่อจะชี้ไปยังโหนดบัฟเฟอร์ใหม่ซึ่งจะเก็บข้อความในตำแหน่งเริ่มต้น i ไปจนสุดข้อความในโหนดบัฟเฟอร์ที่ใช้งานอยู่
4. เปลี่ยนข้อความในโหนดบัฟเฟอร์ที่ใช้งานอยู่ โดยให้เก็บข้อความตั้งแต่ตำแหน่งแรกไปจนถึงก่อนตำแหน่งที่ i ของข้อความเดิม
5. ไปทำงานในโมดูล "insert_line" เพื่อที่จะเชื่อมโหนดพอยน์เตอร์ line2 เข้ากับโหนดพอยน์เตอร์ curline ที่ใช้งานอยู่
6. ไปทำงานในโมดูล "loadtoline" เพื่อจะเรียกข้อมูลจากโหนดบัฟเฟอร์มาเก็บในบัฟเฟอร์ทำงาน
7. กำหนดค่าให้ตัวแปร pagecomplete เป็น NO และตัวแปร changeflag เป็น YES เพื่อบอกถึงว่ามีการเปลี่ยนแปลงตัวอักษร

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.17 แล้ว และสมมุติว่าขณะนั้นมีการเก็บข้อความไว้ในโหนดบัฟเฟอร์ทั้งหมด 3 บรรทัด ดังแสดงไว้ในรูปภาพที่ 4.18 โดยในบรรทัดที่ถูกชี้ด้วย curline จะตัดข้อความตั้งแต่ในคอลัมน์ที่ 4 ให้ไปขึ้นอีกบรรทัดหนึ่ง ซึ่งจะถูกชี้ด้วย line2 ดังแสดงไว้ในรูปภาพที่ 4.19

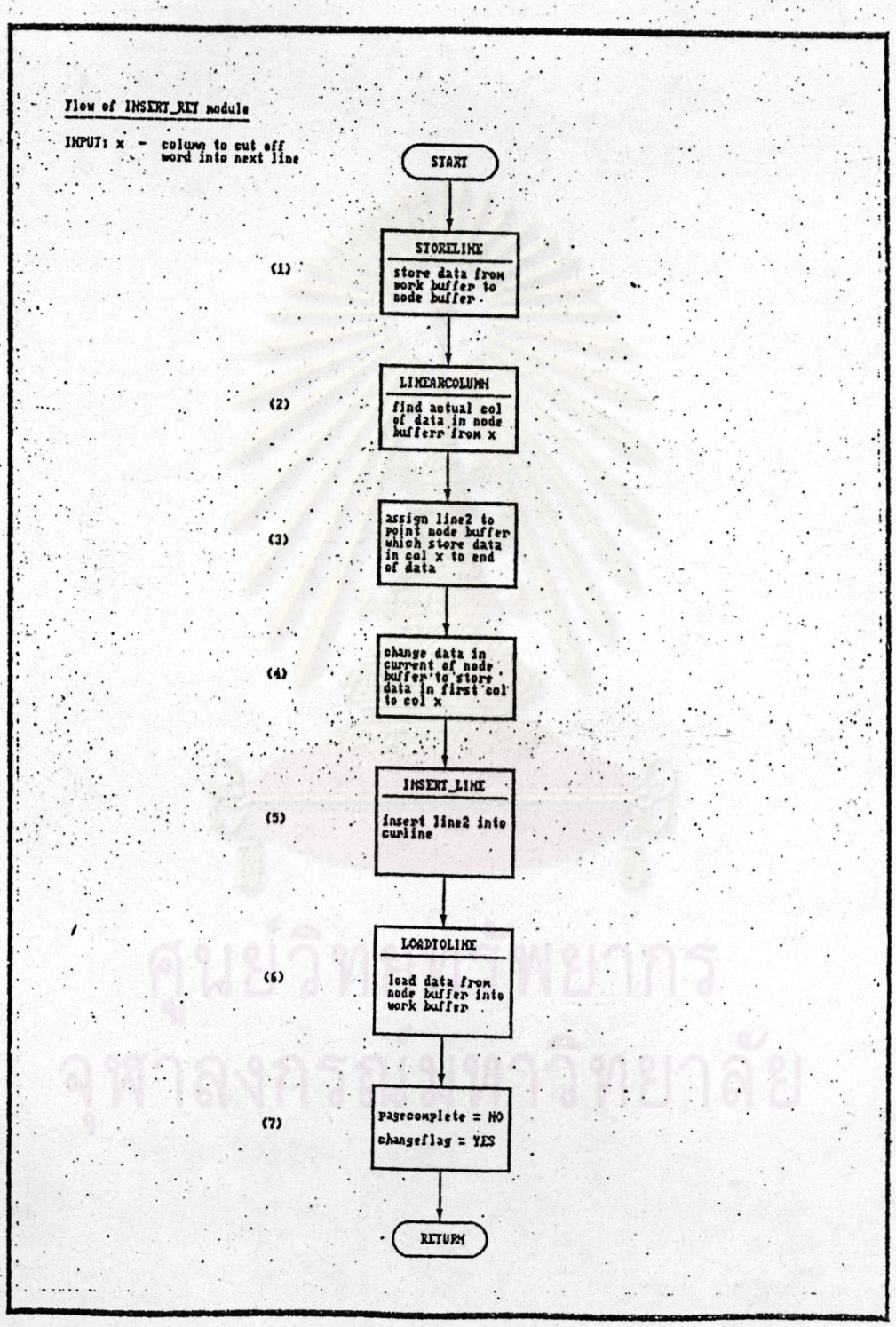
4.3.5 การเลื่อนตัวเคอร์เซอร์ไปขึ้นบรรทัดใหม่

จะเป็นการทำให้ตัวเคอร์เซอร์เลื่อนลงไปขึ้นบรรทัดใหม่อีกบรรทัดหนึ่งที่อยู่ในคอลัมน์แรกของบรรทัด เมื่อเรากดคีย์ Enter หรือ Ctrl + M การทำงานนี้จะอยู่ภายใต้โมดูล returnkey โดยจะต้องกำหนดตำแหน่งคอลัมน์ (x) และบรรทัด (y) ที่ใช้งานอยู่

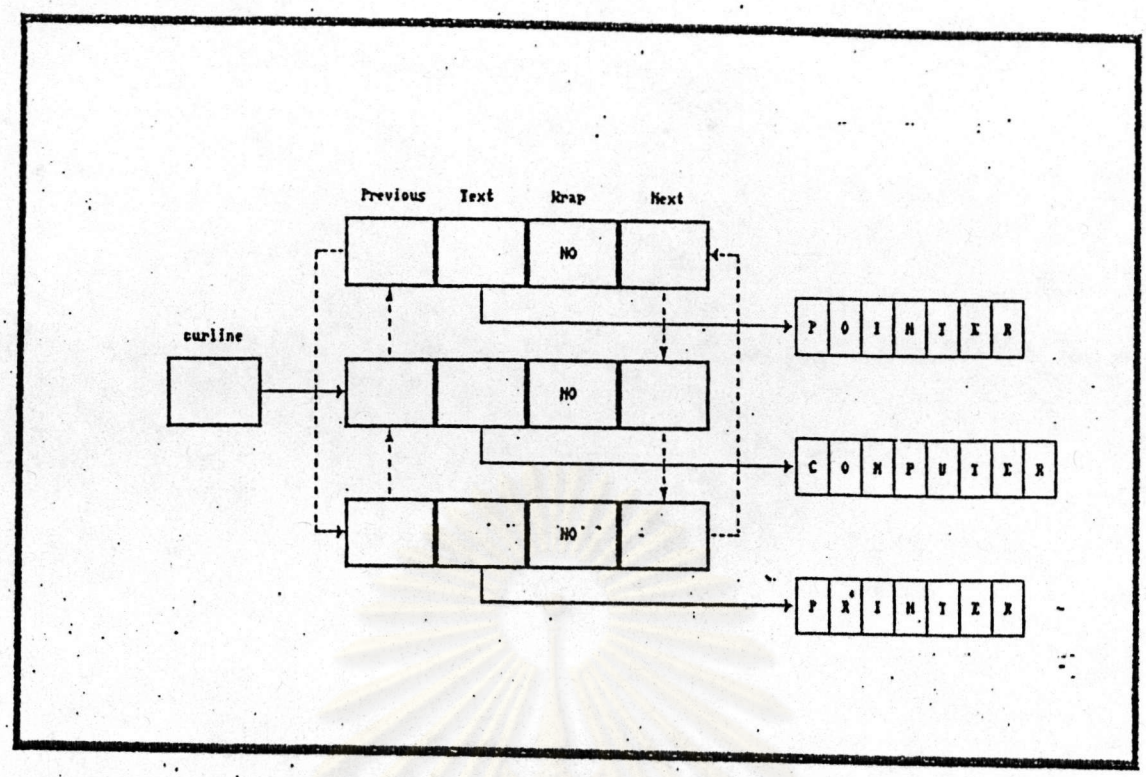
การทำงาน จะมีขั้นตอนดังนี้คือ

1. กำหนดค่าตัวแปร firstcol ให้เป็นศูนย์และตัวแปร pagecomplete เป็น NO
2. ตรวจสอบว่า บรรทัดถัดไปจะหมดข้อมูลหรือไม่
   - ถ้าหมดข้อมูลจริง จะเลิกทำงาน
   - ถ้ายังไม่หมดข้อมูล จะให้ทำงานดังนี้คือ
     - 2.1 ไปทำงานในโมดูล "storeline" เพื่อจะเรียกข้อมูลจากบัฟเฟอร์ทำงานมาเก็บไว้ในโหนดบัฟเฟอร์ที่ใช้งานอยู่ขณะนั้น
     - 2.2 กำหนดค่าให้ตัวแปร lineno ใช้เก็บค่าหมายเลขบรรทัดเพิ่มขึ้นอีก 1
     - 2.3. เลื่อนโหนดพอยน์เตอร์ curline ให้ชี้ไปยังอีกบรรทัดถัดไป
     - 2.4. ไปทำงานในโมดูล "loadtoline" เพื่อจะเรียกข้อมูลจากโหนดบัฟเฟอร์มาเก็บไว้ในบัฟเฟอร์ทำงาน

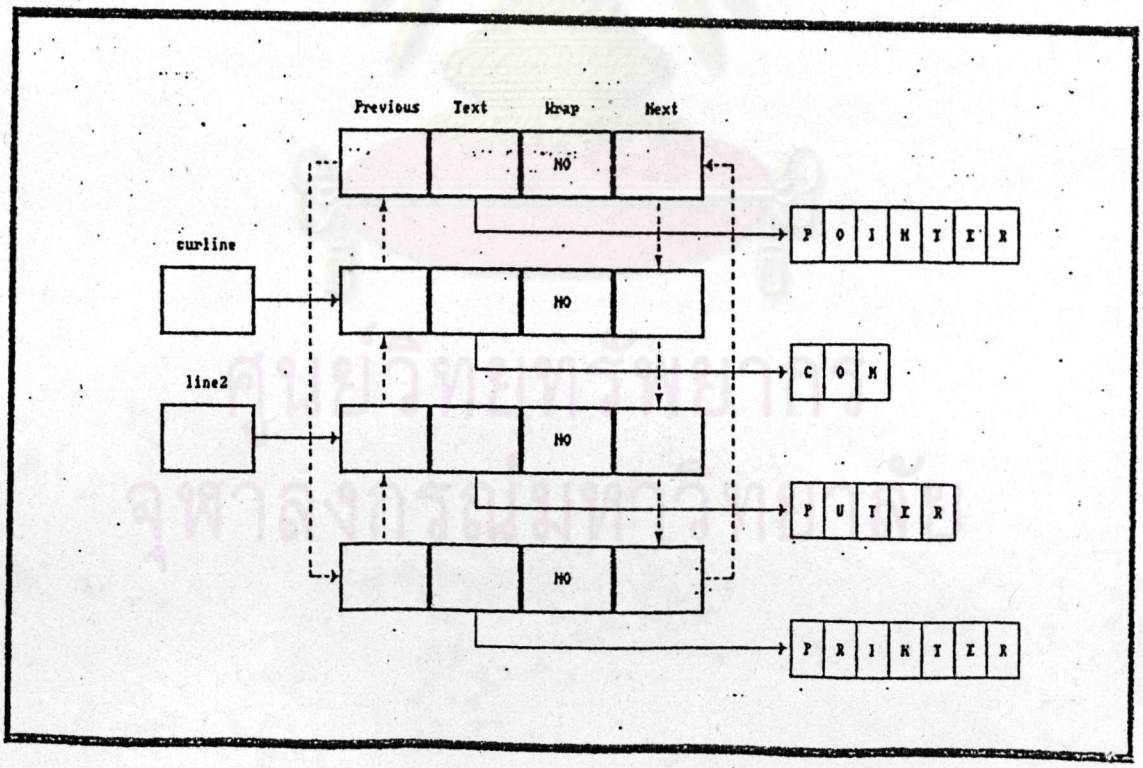
สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.20 แล้ว



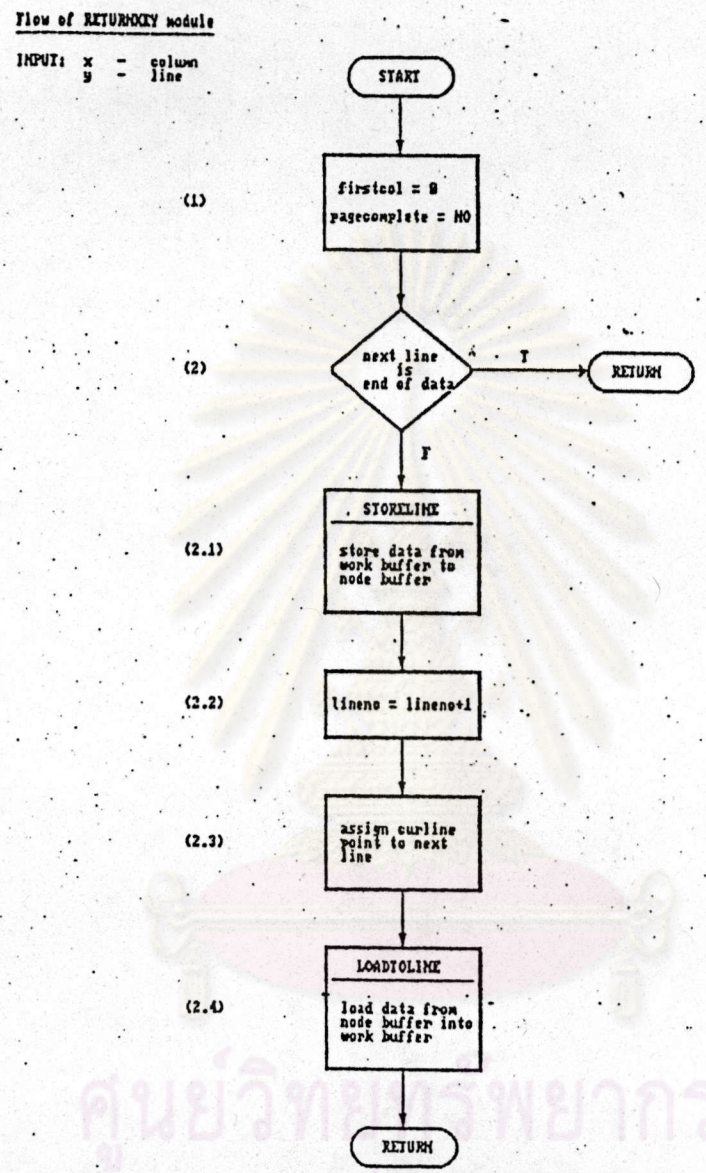
รูปภาพที่ 4.17 ผังงานแสดงขั้นตอนการทำงานของโมดูล insert_ret



รูปภาพที่ 4.18 แสดงข้อความที่เก็บไว้ในบัฟเฟอร์ทำงานก่อนการตัดข้อความ



รูปภาพที่ 4.19 แสดงข้อความที่เก็บไว้ในบัฟเฟอร์ทำงานหลังการตัดข้อความ



รูปภาพที่ 4.20 ผังงานแสดงขั้นตอนการทำงานของโมดูล returnkey

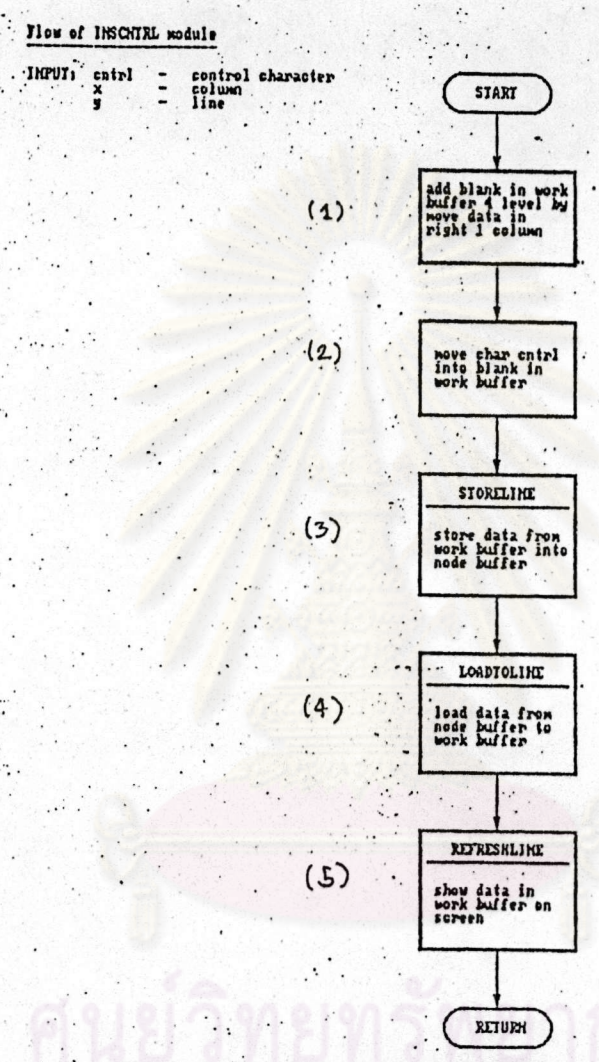
4.3.6 การแทรกรหัสควบคุมตัวอักษร

จะทำงานคล้ายๆกับการแทรกตัวอักษร โดยใช้เมื่อกดคีย์ Ctrl + P เพื่อให้มีการแทรกรหัสควบคุมตัวอักษรในบัฟเฟอร์ทำงานของบรรทัดที่ใช้งานอยู่ขณะนั้น ซึ่งการทำงานอยู่ภายใต้โมดูล inscntrl โดยจะต้องกำหนดรหัสควบคุมตัวอักษร cntrl คอลัมน์ x และบรรทัด y ที่จะแทรกตัวอักษร

การทำงาน จะมีขั้นตอนต่างๆดังนี้คือ

1. จะทำการเคลื่อนย้ายข้อความในบัฟเฟอร์ทำงานตั้งแต่ตำแหน่งคอลัมน์ x ที่ระบุไว้ทั้งบัฟเฟอร์ 4 ระดับคือ TOPEST BUFFER, UPPER BUFFER, MIDDLE BUFFER และ BELOW BUFFER รวมทั้ง ATTR BUFFER ด้วย โดยจะขยับไปทางขวามือ 1 ตำแหน่ง เพื่อให้เกิดช่องว่าง 1 ช่อง
2. นำรหัสควบคุมตัวอักษรที่จะต้องการแทรกนั้น ไปใส่ไว้ใน MIDDLE BUFFER และใส่ค่าช่องว่างให้กับ UPPER BUFFER, TOPEST BUFFER และ BELOW BUFFER รวมทั้ง ATTR BUFFER จะให้เก็บค่ารหัสลักษณะพิเศษนั้น
3. ไปทำงานในโมดูล "storeline" เพื่อจะเรียกข้อมูลจากบัฟเฟอร์ทำงานมาเก็บไว้ในโหนดบัฟเฟอร์ที่ใช้งานอยู่ขณะนั้น
4. ไปทำงานในโมดูล "loadtoline" เพื่อจะเรียกข้อมูลจากโหนดบัฟเฟอร์มาเก็บในบัฟเฟอร์ทำงาน
5. แสดงข้อความในบัฟเฟอร์ทำงานขณะนั้น เพื่อให้เห็นลักษณะพิเศษของตัวอักษรใหม่อีกครั้ง โดยจะไปทำงานในโมดูล "refreshline"

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.21 แล้ว



รูปภาพที่ 4.21 ผังงานแสดงขั้นตอนการทำงานของโมดูล inscntrl

4.3.7 การลบบรรทัดของโหนดบัฟเฟอร์

จะเป็นการลบข้อความในโหนดบัฟเฟอร์ของบรรทัด ที่ระบุไว้ให้หายไป การลบบรรทัดนี้จะทำได้โดยการเปลี่ยนตัวพอยน์เตอร์ของบัฟเฟอร์พอยน์เตอร์ก่อนหน้า และพอยน์เตอร์ของบัฟเฟอร์ถัดไปจากบรรทัดที่ต้องการลบให้มีความสัมพันธ์กันแล้วทำการยกเลิกโหนดบัฟเฟอร์และบัฟเฟอร์พอยน์เตอร์ที่ต้องการลบให้ทิ้งไป การทำงานจะเกิดขึ้นเมื่อกดคีย์ Ctrl + Y ซึ่งจะทำงานอยู่ภายใต้โมดูล deleteline โดยจะต้องกำหนดค่าตัวแปรพอยน์เตอร์ line ซึ่งจะชี้ไปยังโหนดบัฟเฟอร์ที่จะต้องการลบนั้น

การทำงาน จะมีขั้นตอนต่างๆดังนี้คือ

1. ตรวจสอบบรรทัดถัดไปของโหนดพอยน์เตอร์ line ว่าหมดข้อมูลหรือไม่
   - ถ้ายังไม่หมดข้อมูล จะทำงานตามขั้นตอนดังนี้คือ
     - 1.1 เปลี่ยนค่า next pointer ของบัฟเฟอร์พอยน์เตอร์ที่อยู่ในบรรทัดก่อนหน้าของบรรทัด line ให้ชี้ไปที่บัฟเฟอร์พอยน์เตอร์บรรทัดถัดไปของโหนดพอยน์เตอร์ line
     - 1.2 เปลี่ยนค่า previous pointer ของบัฟเฟอร์พอยน์เตอร์ ที่อยู่ในบรรทัดถัดไปของบรรทัด line ให้ชี้ไปยังบัฟเฟอร์พอยน์เตอร์บรรทัดก่อนหน้าของโหนดพอยน์เตอร์ line นั้นเช่นกัน
     - 1.3 ยกเลิกบัฟเฟอร์พอยน์เตอร์และโหนดบัฟเฟอร์ ของบรรทัดที่พอยน์เตอร์ line ชี้อยู่ ทำให้ข้อความในบัฟเฟอร์หายไปด้วย
   - ถ้าหมดข้อมูลแล้ว จะทำงานตามขั้นตอนดังนี้คือ
     - 1.4 ยกเลิกโหนดบัฟเฟอร์ของบรรทัดที่โหนดพอยน์เตอร์ line ชี้อยู่เท่านั้น ทำให้ข้อความในบัฟเฟอร์หายไปด้วย

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.22 แล้ว โดยสมมุติว่า ขณะนั้นมีการเก็บข้อความไว้ในโหนดบัฟเฟอร์ทั้งหมด 3 บรรทัด ซึ่งได้กำหนด curline pointer ให้เป็นตัวชี้ไปยังบัฟเฟอร์พอยน์เตอร์ที่จะลบ ในที่นี้ใช้โหนดบัฟเฟอร์บรรทัดที่ 2 ดังแสดงไว้ในรูปภาพที่ 4.23 และในภาพที่ 4.24 แสดงการเปลี่ยน

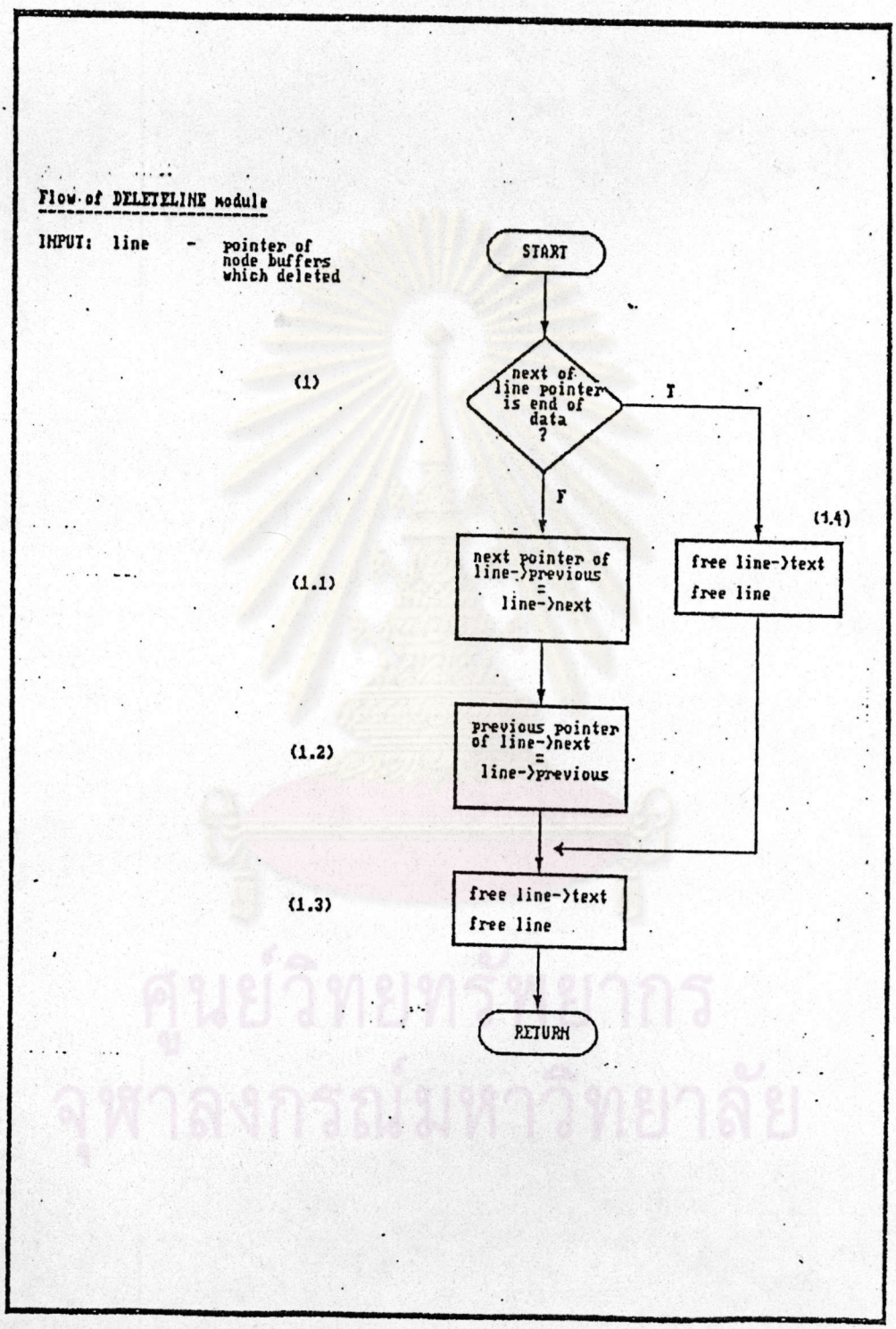
ค่าพอยน์เตอร์ โดยให้ next pointer ของบัฟเฟอร์พอยน์เตอร์บรรทัดที่ 1 ชี้ไปยังบัฟเฟอร์พอยน์เตอร์บรรทัดที่ 3 และ previous pointer ของบัฟเฟอร์พอยน์เตอร์บรรทัดที่ 3 ไปชี้ที่บัฟเฟอร์พอยน์เตอร์บรรทัดที่ 1 แล้วทำการยกเลิกบัฟเฟอร์พอยน์เตอร์และโหนดบัฟเฟอร์บรรทัดที่ 2 ไปซึ่งในที่สุดจะให้ผลดังรูปภาพที่ 4.25

4.3.8 การลบรหัสรีเทริ์นของบรรทัด

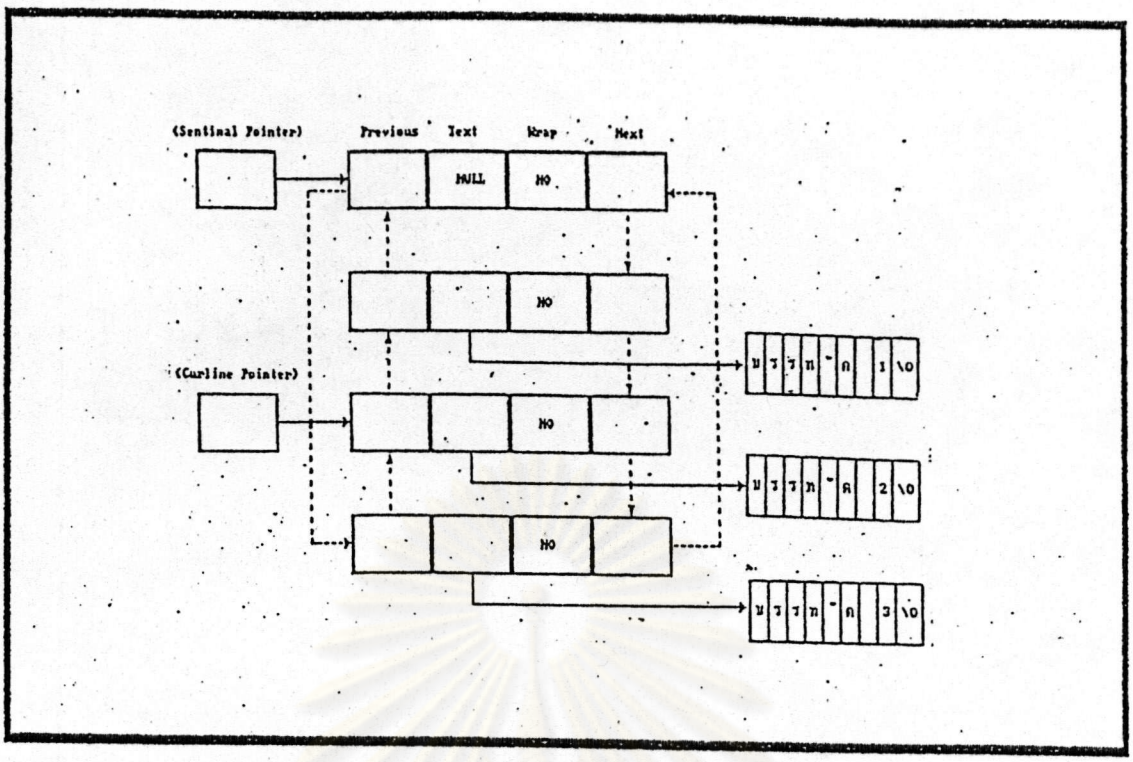
จะเป็นการลบรหัสรีเทริ์นของบรรทัดเพื่อจะนำข้อความในบรรทัดถัดไปมาเชื่อมต่อกับบรรทัดขณะนั้น ซึ่งจะทำงานนี้ได้เมื่อกดคีย์ Del หรือ Ctrl + G ขณะที่ตำแหน่งเคอร์เซอร์อยู่ท้ายสุดของบรรทัดนั้น การทำงานนี้จะอยู่ภายใต้โมดูล del_return โดยจะต้องกำหนดค่าพอยน์เตอร์ line ที่จะชี้ไปยังบรรทัดที่ลบตัวอักษรนั้น

การทำงาน จะมีขั้นตอนต่างๆดังนี้คือ

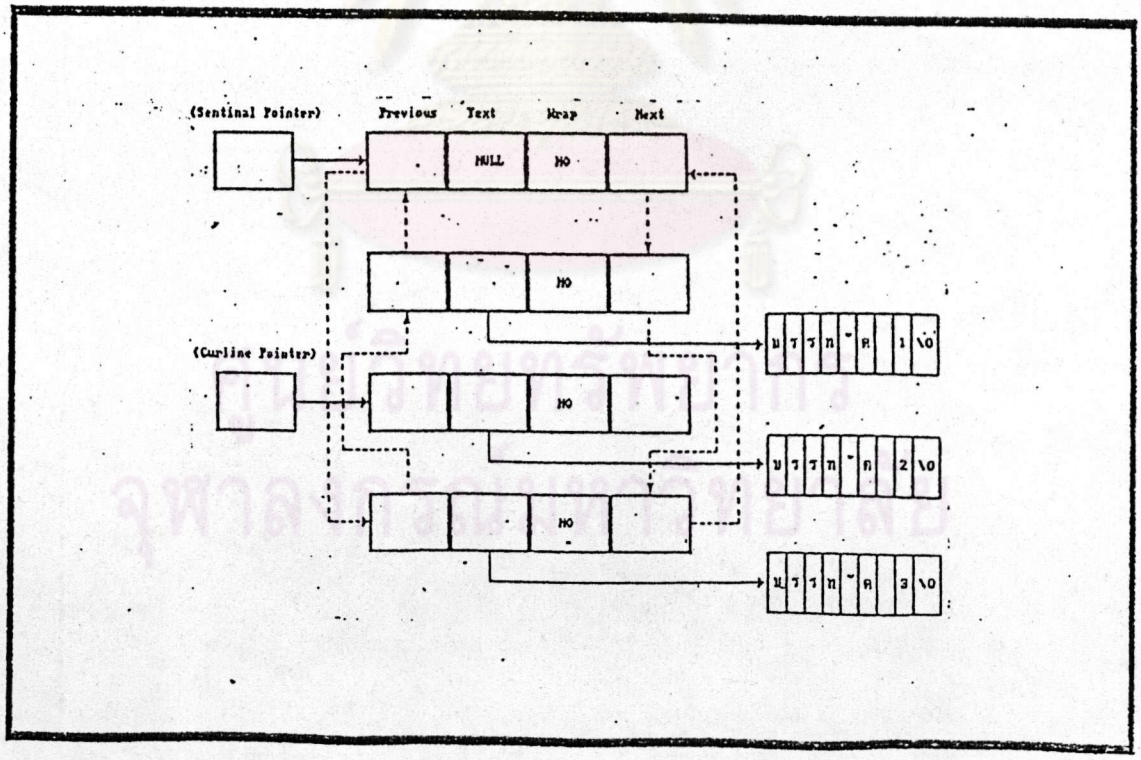
1. ไปทำงานในโมดูล "storeline" เพื่อจะเรียกข้อมูลจากบัฟเฟอร์ทำงานมาเก็บไว้ในโหนดบัฟเฟอร์ที่ใช้งานอยู่ขณะนั้น (ซึ่งถูกชี้ด้วยพอยน์เตอร์ line)
2. ตรวจสอบว่า บรรทัดถัดไปจากบรรทัดที่พอยน์เตอร์ line ชี้อยู่หมดข้อมูลหรือไม่ ถ้ายังไม่หมดข้อมูลจะทำงานดังต่อไปนี้คือ
   - 2.1 กำหนดตัวแปรพอยน์เตอร์ linedeleted ให้ชี้ไปยังบรรทัดถัดไปจากบรรทัดที่พอยน์เตอร์ line ชี้อยู่
   - 2.2 กำหนดตัวแปรพอยน์เตอร์ temp ให้ชี้ไปยังบัฟเฟอร์โหนดใหม่ที่สร้างขึ้น
   - 2.3 คัดลอกข้อมูลจากบัฟเฟอร์โหนดที่ถูกชี้ด้วยพอยน์เตอร์ line ไปยังบัฟเฟอร์โหนดใหม่ที่ถูกชี้ด้วยพอยน์เตอร์ temp
   - 2.4 เชื่อมข้อความในบัฟเฟอร์โหนดที่ถูกชี้ด้วยพอยน์เตอร์ temp กับข้อความในบัฟเฟอร์โหนดของบรรทัดที่ถูกชี้ด้วยพอยน์เตอร์ linedeleted แล้วเก็บผลลัพธ์ไว้ในบัฟเฟอร์โหนดที่ถูกชี้ด้วยพอยน์เตอร์ line ที่กำหนดไว้
   - 2.5 เปลี่ยนค่า next pointer ของบรรทัด line ให้ชี้ไปยังบรรทัดถัดไปจากบรรทัดที่พอยน์เตอร์ linedeleted ชี้อยู่



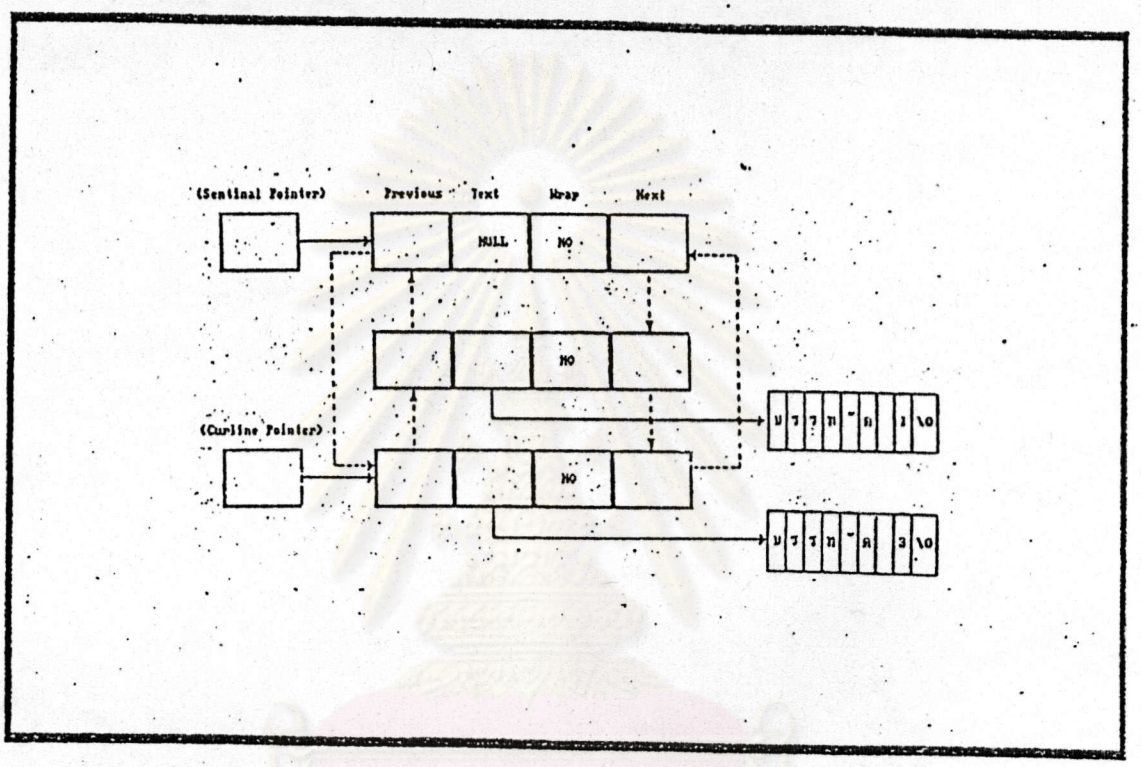
รูปภาพที่ 4.22 ผังงานแสดงขั้นตอนการทำงานของโมดูล deleteline



รูปภาพที่ 4.23 แสดงการเก็บข้อมูลของโหนดัฟเฟอร์ ก่อนการทำงานในโมดูล deleteline



รูปภาพที่ 4.24 แสดงการเปลี่ยนค่าของพอยน์เตอร์ ในระหว่างการทำงานของโมดูล deleteline



รูปภาพที่ 4.25 แสดงการเก็บข้อมูลของโหนดบัฟเฟอร์ ก่อนการทำงานในโมดูล deleteline

2.6 เปลี่ยนค่า previous pointer ของบรรทัดถัดไป จากบรรทัดที่พอยน์เตอร์ linedeleted ชี้อยู่ ให้ชี้ไปยังบรรทัดเดียวกันกับที่พอยน์เตอร์ line ชี้อยู่

2.7 ยกเลิกการใช้งานของบรรทัดที่พอยน์เตอร์ linedeleted ชี้อยู่

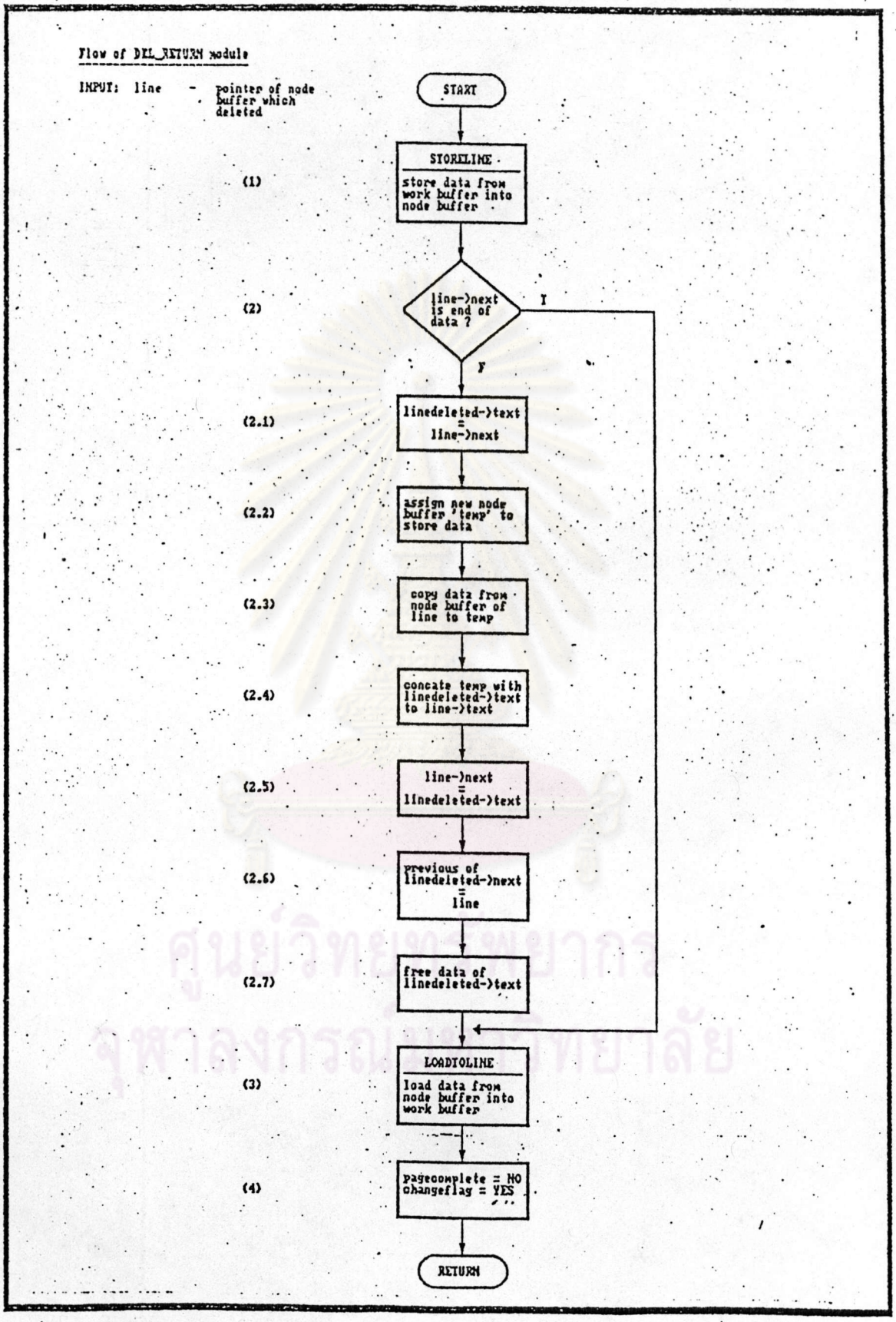
3. ไปทำงานในโมดูล "loadtoline" เพื่อจะเรียกข้อมูลจากบรรทัด ที่ถูกชี้ด้วยพอยน์เตอร์ line มาเก็บไว้ในบัฟเฟอร์ทำงาน

4. กำหนดค่าให้ตัวแปร pagecomplete เป็น NO และ ตัวแปร changeflag ให้เป็น YES

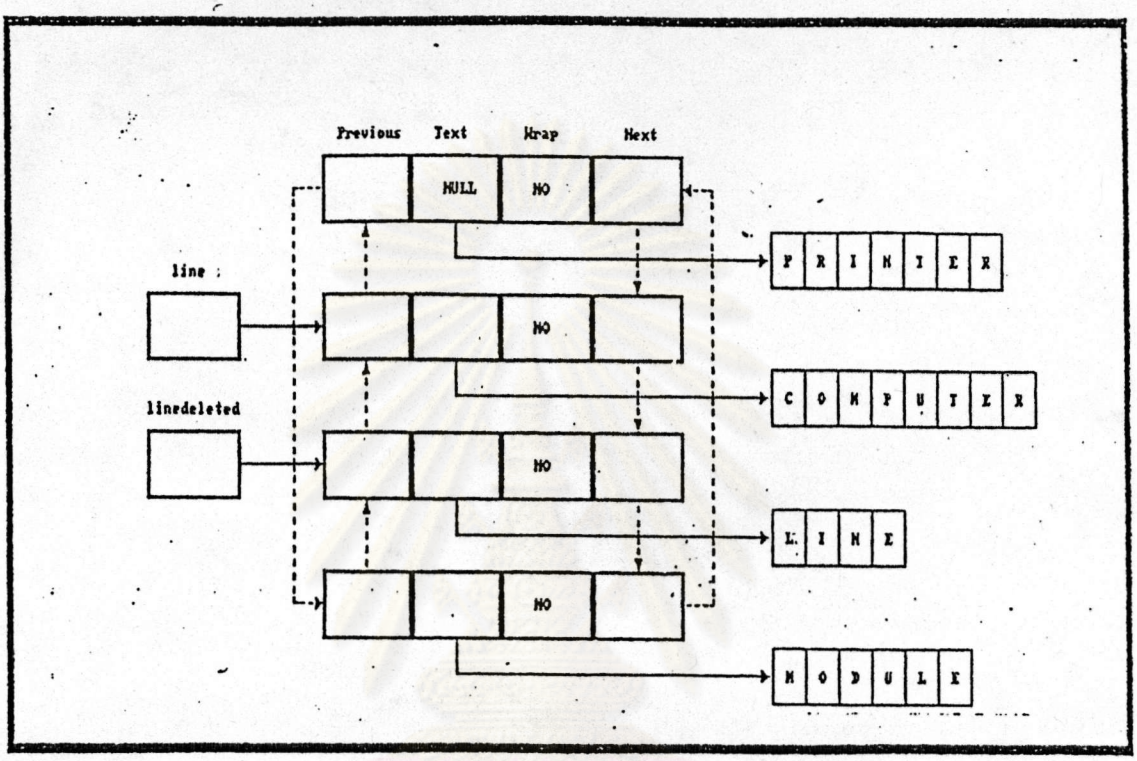
สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.26 แล้ว โดยสมมติว่าขณะนั้นมีการเก็บข้อความไว้ในโหนดบัฟเฟอร์ทั้งหมด 4 บรรทัด โดยได้กำหนดพอยน์เตอร์ line ให้เป็นตัวชี้ไปบรรทัดที่จะเริ่มต้นใช้งาน และ กำหนดให้พอยน์เตอร์ linedeleted เป็นตัวชี้ไปยังบรรทัดถัดไปจากบรรทัดที่พอยน์เตอร์ line ชี้อยู่ ดังแสดงไว้ในรูปภาพที่ 4.27 และ ในระหว่างการทำงานจะมีการเชื่อมข้อความของบรรทัดที่ 2 (เก็บข้อความ "COMPUTER") และข้อความของบรรทัดที่ 3 (เก็บข้อความ "LINE") เข้าด้วยกัน รวมทั้งเปลี่ยนค่าพอยน์เตอร์บางตัวด้วย ดังแสดงไว้ในรูปภาพที่ 4.28 จากนั้นจะคัดลอกข้อความที่เชื่อมต่อกัน มาเก็บไว้ในโหนดบัฟเฟอร์ของบรรทัดที่พอยน์เตอร์ line ชี้อยู่ และลบบรรทัด linedeleted ทิ้งไป ผลจะได้ดังแสดงไว้ในภาพที่ 4.29

4.3.9 <u>การลบตัวอักษร</u>

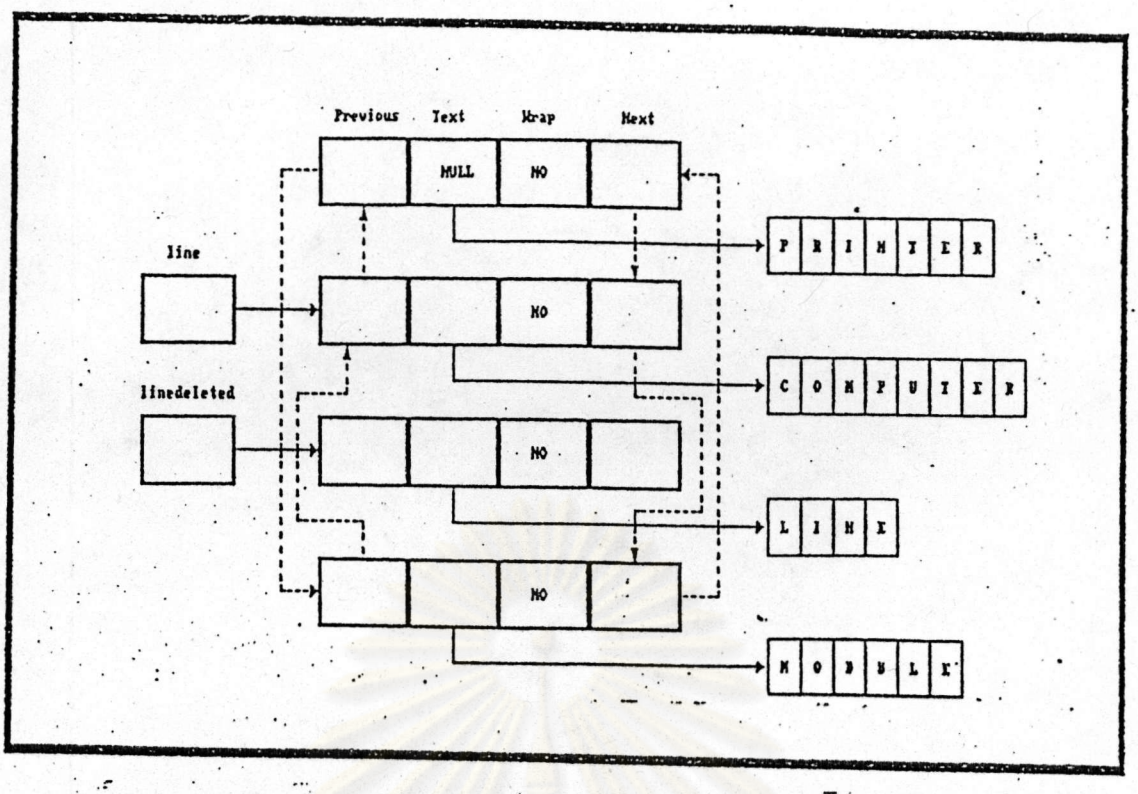
เป็นการลบตัวอักษร 1 ตัวในบัฟเฟอร์ทำงานของบรรทัดที่ใช้งานอยู่ โดยจะต้องมีการเคลื่อนย้ายข้อความในบัฟเฟอร์ทำงาน เพื่อจะนำตัวอักษรที่อยู่ตำแหน่งถัดไปมาแทนที่ตัวอักษร ในตำแหน่งที่ต้องการลบ การทำงานจะเกิดขึ้นเมื่อกดคีย์ Ctrl + G หรือคีย์ Del โดยจะอยู่ภายใต้โมดูล delete_char ซึ่งจะรับค่าจากตัวแปร x ที่เป็นค่าตำแหน่งคอลัมน์ของเคอร์เซอร์ การลบตัวอักษรนี้จะถือว่าเป็นการลบตัวอักษรในตำแหน่งที่ต้องการ



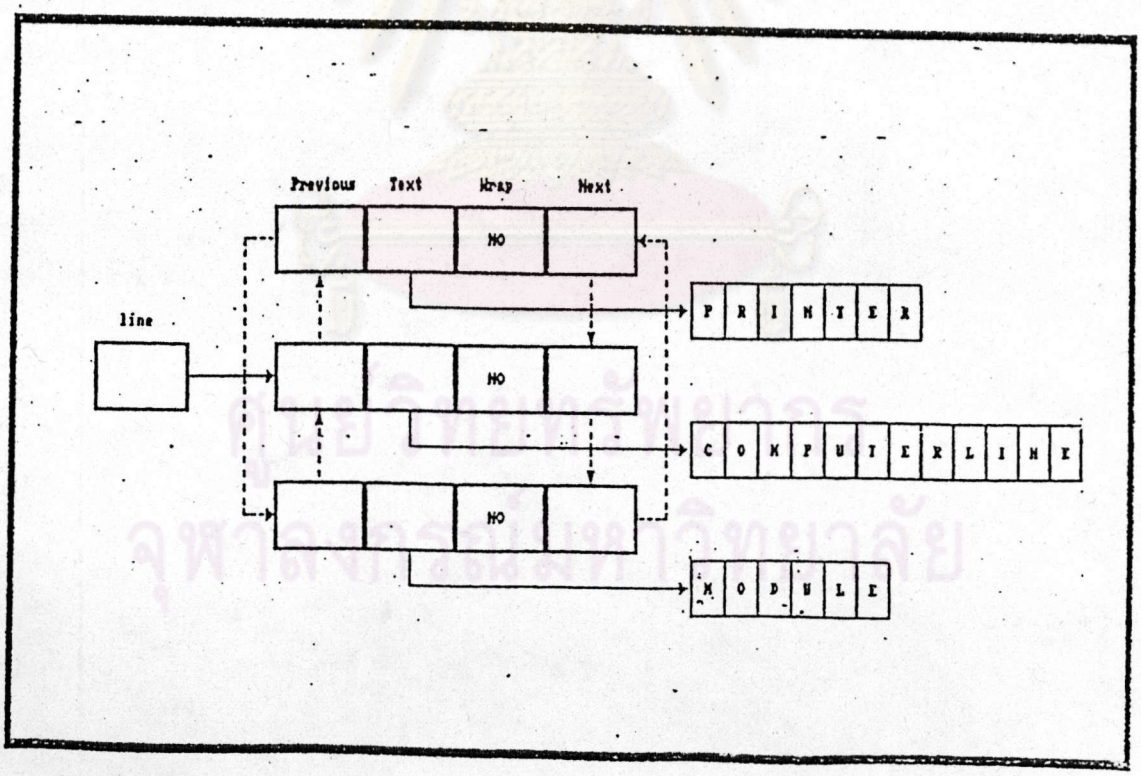
รูปภาพที่ 4.26 ผังงานแสดงขั้นตอนการทำงานของโมดูล del_return



รูปภาพที่ 4.27 แสดงสภาพของบรรทัดก่อนการทำงานของโมดูล del_return



รูปภาพที่ 4.28 แสดงสภาพของบรรทัดระหว่างการทำงานของโมดูล del_return.



รูปภาพที่ 4.29 แสดงสภาพของบรรทัดหลังการทำงานของโมดูล del_return

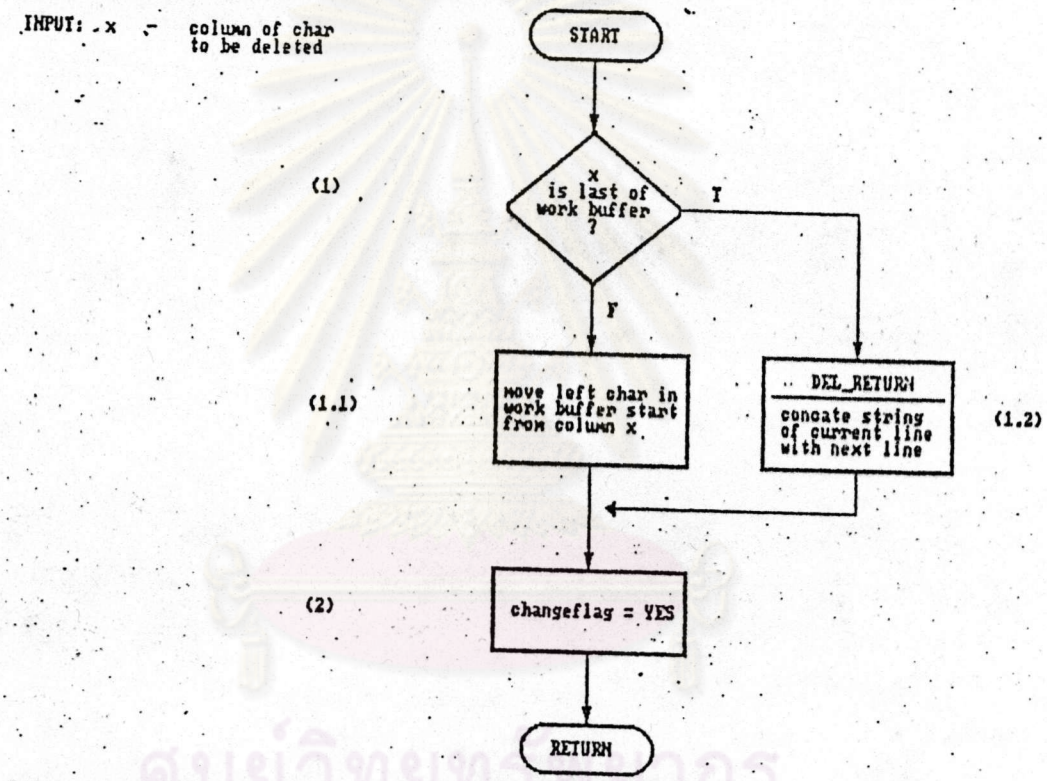
<u>การทำงาน</u> จะมีขั้นตอนต่างๆดังนี้คือ

1. ตรวจสอบตำแหน่งคอลัมน์ x ที่ต้องการลบว่าอยู่ท้ายสุดของบัฟเฟอร์ทำงานหรือไม่
   - ถ้าไม่ได้อยู่ตำแหน่งท้ายสุดของบัฟเฟอร์ทำงานแสดงว่าต้องเป็นตำแหน่งภายในนั้น จะทำงานดังนี้คือ
     1.1 จะทำการเคลื่อนย้ายตัวอักษรในบัฟเฟอร์ทำงาน ไปทางซ้ายมือทีละตัว นับตั้งแต่ตำแหน่ง x ถัดไปที่ระบุไว้ ไปจนหมดข้อความในบัฟเฟอร์ โดยจะทำในบัฟเฟอร์ทำงานทั้งหมด ซึ่งจะทำให้ตัวอักษรในตำแหน่ง x ที่ระบุไว้ถูกแทนที่หายไป
   - ถ้าอยู่ท้ายสุดของบัฟเฟอร์ทำงาน จะทำงานดังนี้คือ
     1.2 จะไปทำงานในโมดูล "del_return" เพื่อจะนำข้อความในบรรทัดถัดไป มาเชื่อมต่อกับบรรทัดที่ใช้งานอยู่ขณะนั้น
2. กำหนดค่าให้ตัวแปร changeflag เป็น YES เพื่อแสดงว่าได้มีการเปลี่ยนแปลงตัวอักษรแล้ว

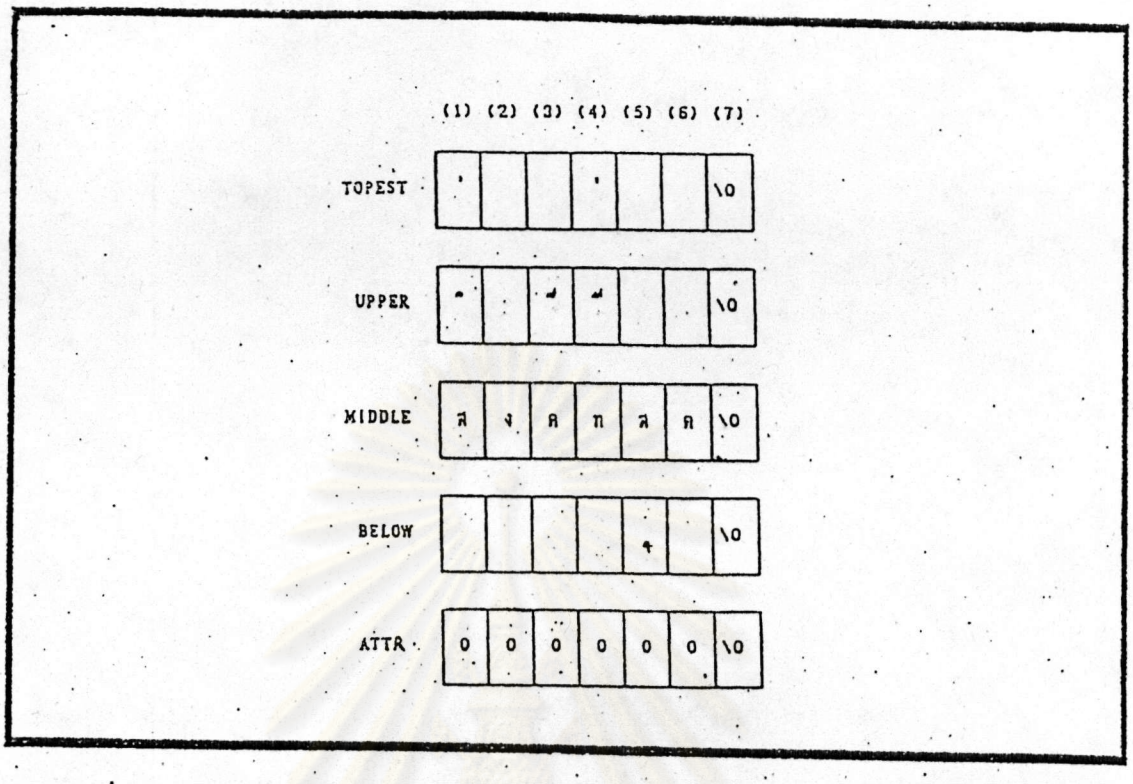
สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.30 แล้ว โดยสมมุติว่าขณะนั้นมีบัฟเฟอร์ทำงานแสดงในรูปภาพที่ 4.31 ซึ่งเก็บข้อความว่า "สิ่งดีที่สุด" แยกตามระดับของชนิดตวอักษร จะต้องการลบคำว่า "ที่" ในตำแหน่งที่ 4 ของบัฟเฟอร์ทำงาน จึงต้องมีการเคลื่อนย้ายตัวอักษรตั้งแต่ตำแหน่งที่ 5 มาทางซ้ายมือทีละตัวอักษรจนหมดข้อความในบัฟเฟอร์ ซึ่งจะได้ผลดังรูปที่ 4.32

4.3.10 <u>การลบข้อความเป็นคำ</u>

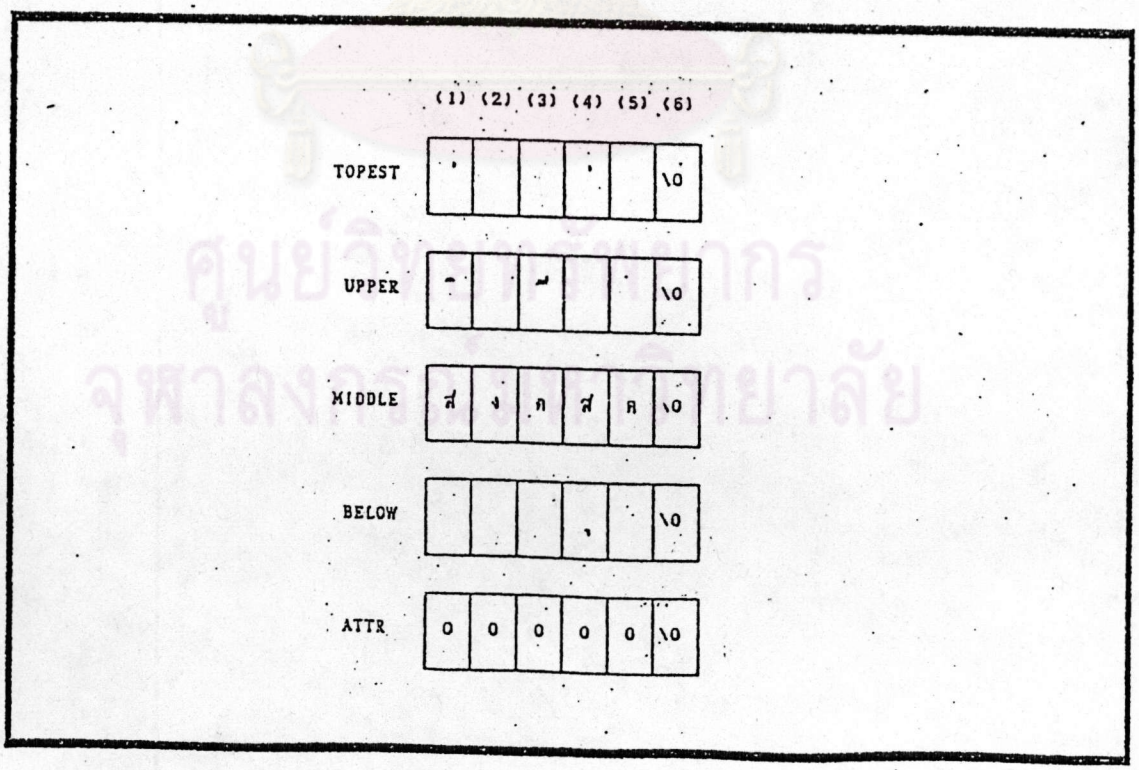
จะเป็นการลบข้อความเป็นคำโดยจะใช้ช่องว่างเป็นตัวคั้นระหว่างคำ การทำงานนี้จะเกิดขึ้นเมื่อกดคีย์ Ctrl + T ซึ่งจะทำการลบในตำแหน่งที่เคอร์เซอร์อยู่ การทำงานจะอยู่ภายใต้โมดูล delete_word ซึ่งจะรับค่าจากตัวแปร x ที่เป็นค่าตำแหน่งคอลัมน์ของเคอร์เซอร์อยู่ขณะนั้น



รูปภาพที่ 4.30 ผังงานแสดงขั้นตอนการทำงานของโมดุล delete_char



รูปภาพที่ 4.31 แสดงการเก็บข้อมูลในบัฟเฟอร์ทำงานก่อนการทำงานของโมดูล delete_char



รูปภาพที่ 4.32 แสดงการเก็บข้อมูลในบัฟเฟอร์ทำงานหลังการทำงานของโมดูล delete_char

<u>การทำงาน</u> จะมีขั้นตอนต่างๆดังนี้คือ

1. ตรวจสอบตัวอักษรในบัฟเฟอร์ทำงานระดับ MIDDLE ตำแหน่งที่ x ว่าหมดข้อมูลหรือไม่
   - ถ้ายังไม่หมดข้อมูล จะทำงานในโมดูล "delete_char" เพื่อที่จะลบตัวอักษรในตำแหน่งที่เคอร์เซอร์อยู่ทีละตัวอักษร จนกระทั่งตรวจสอบพบว่าตัวอักษรนั้นเป็นค่าช่องว่าง (blank)
   - ถ้าหมดข้อมูลแล้ว จะทำงานในโมดูล "del_return" เพื่อที่จะเชื่อมข้อความในบรรทัดถัดไปให้เข้ากับข้อความในบรรทัดขณะนั้น

4.3.11 <u>การลบข้อความไปจนบรรทัด</u>

จะเป็นการลบข้อความจากตำแหน่งที่เคอร์เซอร์อยู่ไปจนหมดข้อความของบรรทัดที่ใช้งาน การทำงานนี้จะเกิดขึ้นเมื่อกดคีย์ Ctrl + Q + Y โดยจะอยู่ภายใต้โมดูล deltoendline ซึ่งจะรับค่าจากตัวแปร x และ y ที่เป็นค่าตำแหน่งคอลัมน์และบรรทัดของเคอร์เซอร์อยู่ขณะนั้น

<u>การทำงาน</u> จะมีขั้นตอนต่างๆดังนี้คือ

1. กำหนดให้ตัวแปร i เป็นค่าจำนวนตัวอักษรทั้งหมดในบัฟเฟอร์โหนด (MAXCOL)
2. กำหนดให้ตัวแปร j เป็นค่าจากตัวแปร x ที่เป็นค่าตำแหน่งคอลัมน์
3. จะตรวจสอบค่าของ i ว่าน้อยกว่าหรือเท่ากับค่าของ j หรือไม่
   - ถ้าเป็นจริงแล้ว จะกำหนดค่าให้กับบัฟเฟอร์ทำงานทั้ง 4 ระดับ ให้มีค่าเป็นช่องว่าง (blank) รวมทั้งกำหนดค่าบัฟเฟอร์ทำงาน ATTR ให้เป็นศูนย์ด้วย จากนั้นจะลดค่าของ i อีก 1 แล้วไปทำงานซ้ำๆกัน จนกระทั่งค่าของ i มีค่ามากกว่า j
   - ถ้าไม่เป็นจริงแล้ว ให้ทำงานในหัวข้อถัดไป
4. ทำงานในโมดูล "refreshline" ในตำแหน่ง x และ y เพื่อแสดงข้อความในบัฟเฟอร์ทำงานบนจอภาพอีกครั้ง

5. กำหนดค่าให้ตัวแปร changeflag ให้เป็น YES เพื่อแสดงว่ามีการเปลี่ยนแปลงข้อความเกิดขึ้น

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.33 แล้ว

4.3.12 การทำงานเกี่ยวกับคำย่อ (Macro Word)

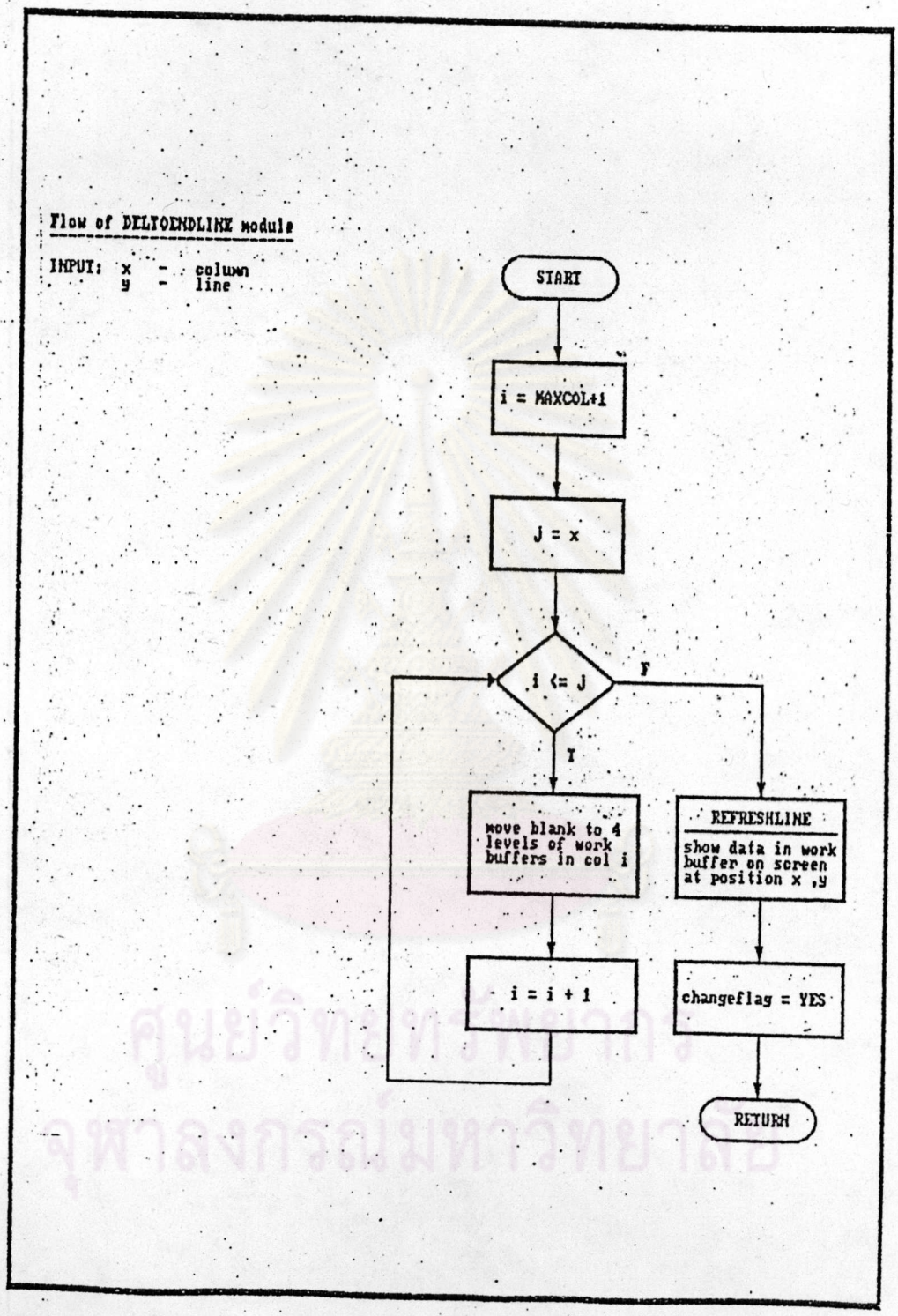
จะเป็นการกำหนดข้อความที่ต้องการไว้ให้ตรงกับคีย์ Ctrl + F1 ถึง Ctrl + F10 เพื่อให้สามารถเรียกข้อความที่กำหนดไว้ ออกมาแทรกได้ในตำแหน่งนั้นตลอดเวลา เมื่อกดคีย์ต่างๆเหล่านั้นซึ่งเหมาะสมกับการพิมพ์เอกสารที่ต้องมีการใช้ข้อความบางอย่างที่ซ้ำๆกันได้ โดยเราจะใช้คีย์ Alt + M เพื่อใช้กำหนดข้อความเหล่านั้นได้ ข้อความเหล่านั้นจะถูกเก็บไว้ในตัวแปรชุดชนิดตัวอักษร ชื่อว่า macro ที่มีขนาดเป็น 10 (แถว) x 36 (คอลัมน์) โดยจะเก็บข้อความเป็นจำนวน 10 ชุด แต่ละชุดจะเก็บได้ 36 ตัวอักษร

4.3.12.1 การกำหนดคำย่อ (Define Macro Word)

จะเป็นขั้นตอนของการกำหนดคำที่ต้องการไปเก็บไว้ ในตัวแปร macro โดยจะเริ่มต้นทำงานได้เมื่อกดคีย์ Alt + M การทำงานนี้จะอยู่ภายใต้โมดูล editmacro

การทำงาน จะมีขั้นตอนต่างๆดังนี้คือ

1. จะไปทำงานในโมดูล "dispmacro" เป็นจำนวน 10 ครั้ง เพื่อแสดงข้อความที่เป็นคำอธิบายของ คีย์ Ctrl + F1 ถึง Ctrl + F10
2. กำหนดตัวแปร i เพื่อจะใช้ควบคุมหมายเลขชุดของตัวแปร macro ขณะนั้น โดยจะกำหนดให้เป็นศูนย์ ซึ่งถือว่าเป็นแถวแรก
3. เริ่มต้นเข้ามาทำงานซ้ำๆกัน ดังนี้คือ
   3.1 รับข้อความจากแป้นพิมพ์ เพื่อป้อนข้อความที่จะให้เป็นคำย่อเก็บไว้ในตัวแปรชุด macro หมายเลขชุดที่ i โดยการทำงานนี้จะอยู่ภายใต้โมดูล "getstring" ซึ่งจะให้ค่าแป้นพิมพ์ของคีย์พิเศษออกมาให้ด้วย



รูปภาพที่ 4.33 ผังงานแสดงขั้นตอนการทำงานของโมดูล deltoendline

3.2 ตรวจสอบค่าแป้นพิมพ์ที่ได้

3.2.1 ถ้าเป็นค่า UPKEY (คีย์ลูกศรขึ้น) จะไปแสดงข้อความคำอธิบายของคีย์ Ctrl ตัวที่ i โดยจะทำงานในโมดูล "dispmacro" อีกครั้งแล้วจะไปตรวจสอบค่า ของ i ว่าเป็นค่าศูนย์หรือไม่ ถ้าเป็นศูนย์ (แสดงว่าอยู่ที่ชุดแรกแล้ว) ก็จะกำหนดค่าตัวแปร i เป็นค่า 9 (เพื่อเลื่อนลงมาที่ชุดสุดท้าย) แต่ถ้าไม่เป็นศูนย์จะลดค่าของ i อีก 1

3.2.2 ถ้าเป็นค่า DNKEY (คีย์ลูกศรลง) จะไปแสดงข้อความคำอธิบายของคีย์ Ctrl ตัวที่ i โดยจะทำงานในโมดูล "dispmacro" อีกครั้งแล้วจะไปตรวจสอบค่าของ i ว่าเป็นค่า 9 หรือไม่ ถ้าเป็น 9 (แสดงว่าอยู่ที่ชุดสุดท้าย) ก็จะให้กำหนดค่าตัวแปร i เป็นค่าศูนย์ (เพื่อเลื่อนขึ้นไปที่ชุดแรก) แต่ถ้าไม่เป็นศูนย์จะเพิ่มค่าของ i อีก 1

3.2.3 ถ้าเป็นค่า ESCKEY จะเลิกทำงานไป

3.2.4 ถ้าเป็นคีย์อื่นๆ จะกลับไปทำงานใหม่

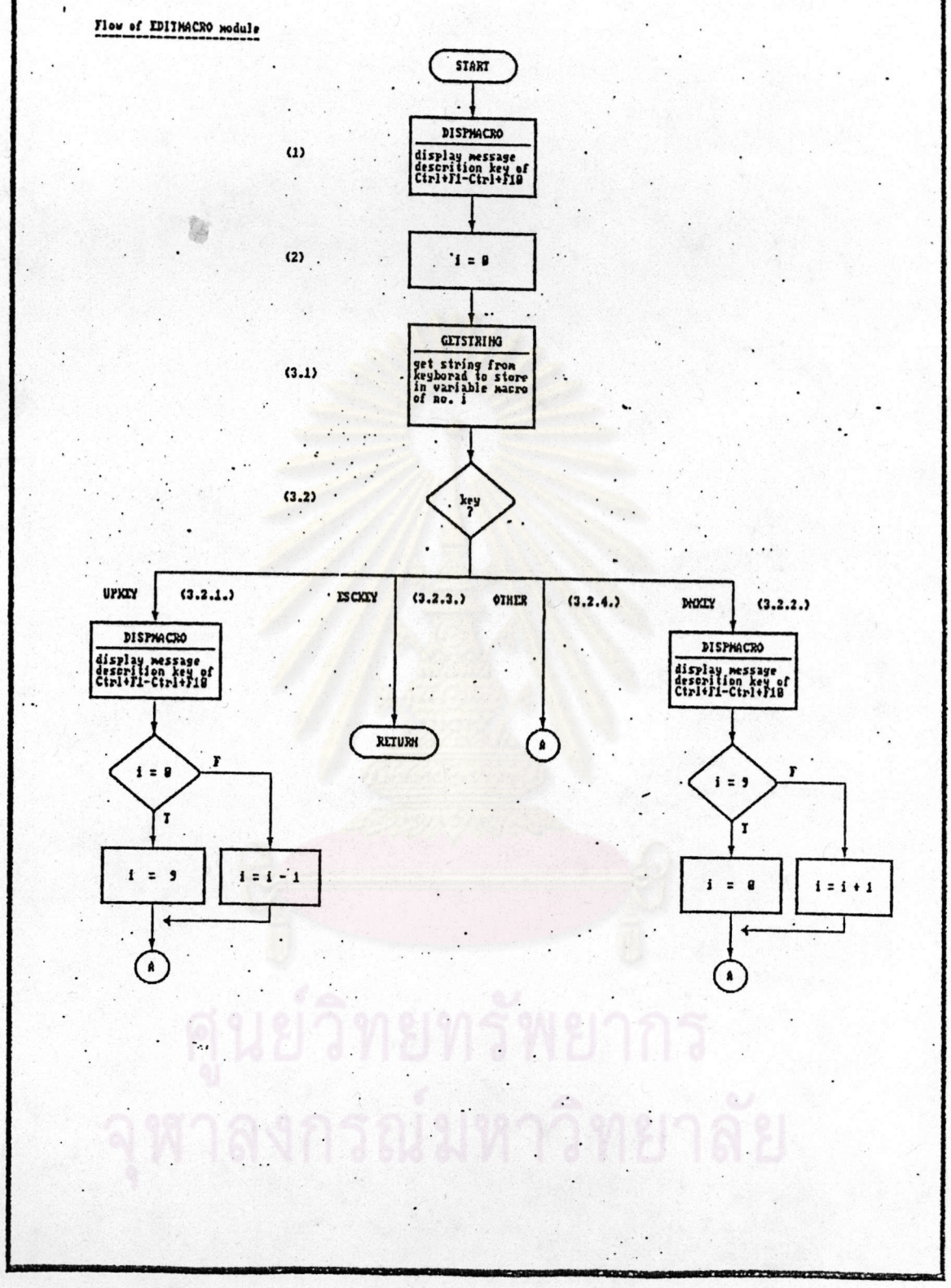
สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.30 แล้ว

4.3.12.2 การแทรกคำย่อ (Insert Macro Word)

จะเป็นขั้นตอนของการเรียกใช้ข้อความที่ได้ เก็บไว้ในตัวแปร macro มาแทรกในระหว่างตัวอักษรที่ตำแหน่งขณะนั้น โดยการกดคีย์ Ctrl + F1 ถึง Ctrl + F10 การทำงานนี้จะอยู่ภายในโมดูล insertmacro

<u>การทำงาน</u> จะมีขั้นตอนต่างๆดังนี้คือ

1. ตรวจสอบตัวอักษรในตัวแปร macro ที่ใช้งานอยู่ว่าหมดข้อมูลหรือยัง ถ้ายังไม่หมดข้อมูลจะไปนำตัวอักษรในตัวแปร macro มาทำงานในโมดูล "insert_char" เพื่อแทรกตัวอักษรนั้นในตำแหน่งที่ได้ใช้งานอยู่ ของบัฟเฟอร์ทำงาน แล้วจะเลื่อนไป



รูปภาพที่ 4.34 ผังงานแสดงขั้นตอนการทำงานของโมดูล editmacro

ตัวอักษรถัดไป ทำงานซ้ำๆกันเช่นนี้จนกระทั่งหมดข้อมูลในตัวแปร

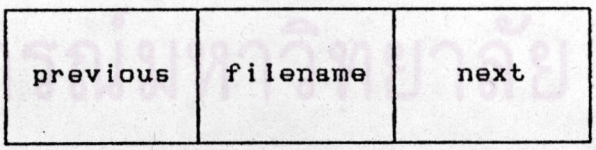
2. ทำงานในโมดูล "refreshline" เพื่อแสดงข้อความในบัฟเฟอร์ทำงานบนจอภาพอีกครั้ง

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.35 แล้ว

4.3.13 การเลือกใช้ชื่อไฟล์แบบฟลูสกรีนไดเรคทอรี (Full Screen Directory)

การเลือกชื่อไฟล์เราจะสามารถทำได้ โดยกำหนดชื่อไฟล์ที่ต้องการโดยตรง แต่กรณีที่จำชื่อไฟล์ไม่ได้ เราสามารถใช้การเลือกชื่อไฟล์แบบฟลูสกรีนไดเรคทอรีมาช่วยได้ โดยวิธีนี้จะปรากฏให้เห็นชื่อไฟล์ในไดเรคทอรีบนจอภาพ แล้วให้เราสามารถใช้คีย์ลูกศรซ้ายขวาและขึ้นลง เพื่อเลือกชื่อไฟล์ที่จะต้องการเลือกบนจอภาพมาใช้งานได้ โปรแกรมประยุกต์สมัยใหม่ ก็นิยมเลือกใช้ชื่อไฟล์ด้วยวิธีนี้เช่นกัน เช่น โปรแกรมโลตัสเอดิเตอร์ของเทอร์โบซี เป็นต้น

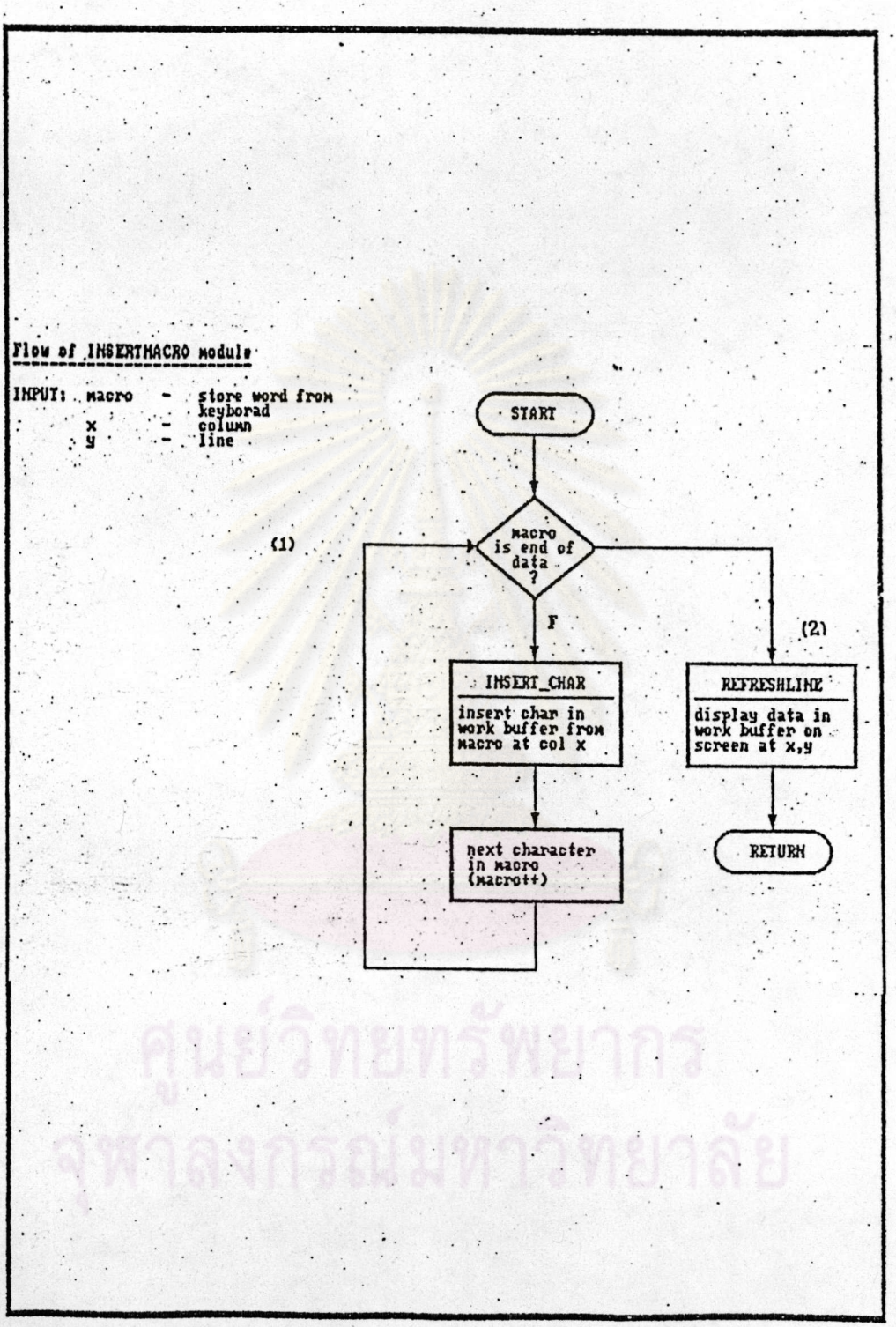
การออกแบบโปรแกรมในลักษณะนี้ จะต้องมีตัวแปรที่ใช้เก็บรายชื่อไฟล์ไว้ และเนื่องจากจะต้องมีการเลื่อนไปยังชื่อไฟล์อื่นๆได้ใน 2 ทิศทาง คือก่อนหน้าและถัดไป จึงออกแบบโครงสร้างตัวแปรให้เป็นแบบ Double Link List ซึ่งในที่นี้จะเรียกว่า "ไดเรคทอรีโหนด" โดยจะมีองค์ประกอบที่สำคัญดังนี้คือ



โดย previous จะเป็นพอยน์เตอร์ ที่จะชี้ไปยังโหนดก่อนหน้า

next จะเป็นพอยน์เตอร์ ที่จะชี้ไปยังโหนดถัดไป

filename จะเป็นที่เก็บชื่อไฟล์ไว้



รูปภาพที่ 4.35 ผังงานแสดงขั้นตอนการทำงานของโมดูล insertmacro

และจะมีตัวแปรพอยน์เตอร์อีก 2 ตัว ซึ่งจะทำหน้าที่ชี้ไปยังโหนดต่างๆคือ headdir และ dirpage โดย headdir จะทำหน้าที่ชี้ไปยังโหนดที่ใช้เป็นหัวเริ่มต้น (Head) ของชื่อไฟล์ และ dirpage จะทำหน้าที่ชี้ไปยังโหนดอื่นๆใหม่ เมื่อมีการเคลื่อนที่ผ่านไปยังโหนดต่างๆ การทำงานภายในโปรแกรมฟลูสกรีนไดเรคทอรี จะมีลักษณะที่สำคัญหลายอย่างดังนี้คือ

4.3.13.1 การสร้างไดเรคทอรีลิสต์ (Creating of Directory List)

จะเป็นขั้นตอนแรกสุดของการทำงาน ซึ่งจะต้องมีการสร้างไดเรคทอรีลิสต์เพื่อเก็บรายชื่อไฟล์ต่างๆไว้ใช้งานได้ การทำงานนี้จะอยู่ภายใต้โมดูล createdir

<u>การทำงาน</u> จะมีขั้นตอนต่างๆดังนี้คือ

1. สร้างไดเรคทอรีโหนดใหม่ตัวแรกเพื่อจะใช้เป็นหัวเริ่มต้น โดยให้ previous และ next ของโหนดนั้นชี้ไปยังตัวมันเอง และ กำหนดให้ headdir และ dirpage ชี้ไปยังโหนดตัวนั้น
2. เริ่มต้นอ่านชื่อไฟล์ตัวแรก จากไดเรคทอรี
3. ตรวจสอบว่าชื่อไฟล์ในไดเรคทอรี หมดข้อมูลหรือยัง
   - ถ้ายังไม่หมดข้อมูล จะทำงานดังนี้คือ
     3.1 ทำการสร้างไดเรคทอรีโหนดตัวใหม่โดยให้ next ของโหนดตัวเก่าก่อนหน้า ชี้ไปยังไดเรคทอรีโหนดตัวใหม่นั้น
     3.2 นำชื่อไฟล์ที่อ่านได้ไปเก็บไว้ใน filename ของไดเรคทอรีโหนดตัวใหม่นั้น และทำการเชื่อมพอยน์เตอร์กันระหว่างไดเรคทอรีโหนดตัวใหม่และไดเรคทอรีโหนดตัวเก่าก่อนหน้า โดยจะทำดังนี้คือ
        - ให้ previous ของโหนดใหม่ ชี้ไปยังโหนดตัวเก่า
        - ให้ dirpage ชี้ไปยังโหนดตัวใหม่
        - ให้ next ของโหนดตัวใหม่ ชี้ไปยังโหนดตัวเก่า
        - ให้ previous ของโหนดเก่า ชี้ไปยังโหนดตัวใหม่

- ถ้ายังหมดข้อมูลแล้ว จะให้ค่าเป็น YES แล้วเลิกทำงาน

4. อ่านชื่อไฟล์ตัวถัดไปจากไดเรคทอรี แล้วกลับไปทำงานใหม่ ในหัวข้อ 3 อีกครั้ง

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.36 แล้ว โดยสมมุติว่าขณะนั้นได้สร้างไดเรคทอรีโหนดใหม่ที่พอยน์เตอร์ headdir และ dirpage ชี้อยู่ ซึ่งพอยน์เตอร์ previous และ next ของไดเรคทอรีโหนด จะชี้ไปยังตัวมันเองด้วยและชื่อไฟล์ไดเรคทอริมีจำนวน 3 ไฟล์ แสดงผลในรูปภาพที่ 4.37 หลังจากนั้นจะอ่านชื่อไฟล์แรกจากไดเรคทอรีมาเก็บไว้ในไดเรคทอรีโหนด และเชื่อมพอยน์เตอร์ระหว่างไดเรคทอรีโหนดทั้งสองเข้าด้วยกัน ดังแสดงในภาพที่ 4.38 แล้วจะอ่านชื่อไฟล์ถัดไปในไดเรคทอรีมาเก็บไว้ในไดเรคทอรีโหนดที่สร้างใหม่ และเชื่อมต่อพอยน์เตอร์เข้าด้วยกันโดยทำซ้ำๆกันจนกว่าหมดชื่อไฟล์ในไดเรคทอรี ซึ่งจะให้ผลแสดงในภาพที่ 4.39

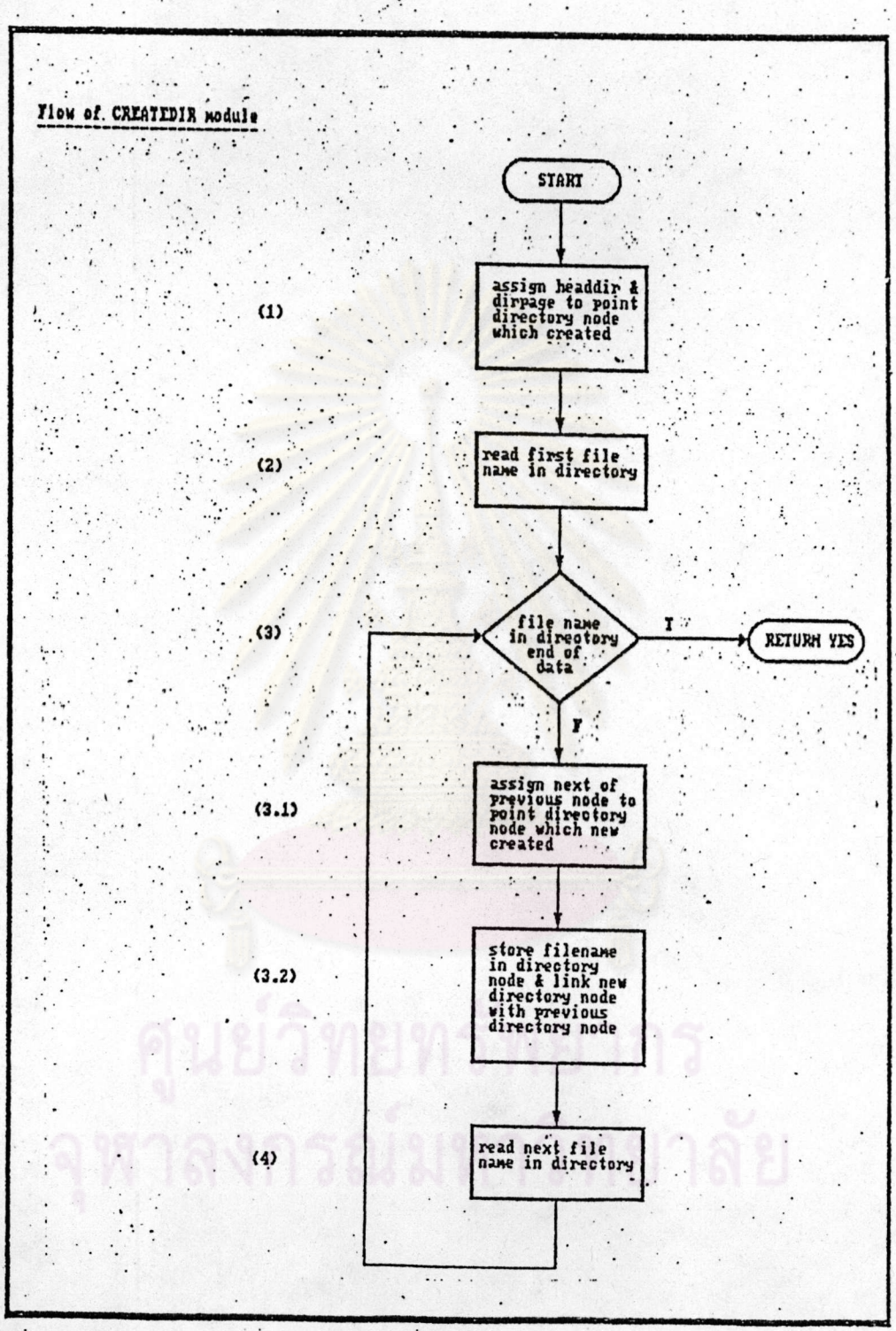
4.3.13.2 การแสดงชื่อไฟล์บนจอภาพ (Show File Name on Screen)

จะเป็นการนำชื่อไฟล์ในไดเรคทอรีลิสต์ มาแสดงบนจอภาพ แต่จะมีการจำกัดเนื้อที่ในการแสดงชื่อไฟล์บนจอภาพด้วย ดังนั้นถ้าชื่อไฟล์ในไดเรคทอรีลิสต์ มีอยู่เป็นจำนวนมาก ก็จะถูกแสดงได้เพียงบางส่วนของชื่อไฟล์ในไดเรคทอรีลิสต์เท่านั้น การทำงานนี้จะอยู่ภายใต้โมดูล showpagedir

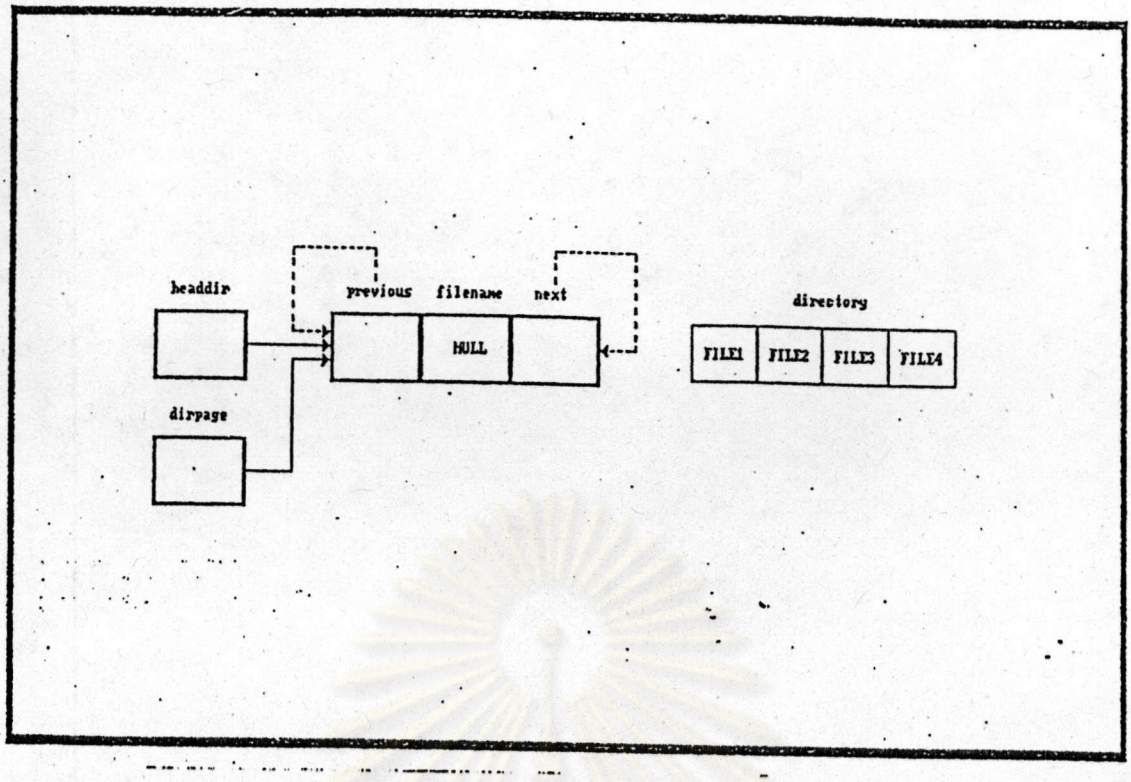
<u>การทำงาน</u> จะมีขั้นตอนต่างๆดังนี้คือ

1. กำหนดตัวแปร row และ col ให้มีค่าเริ่มต้นเป็นศูนย์
2. ตรวจสอบค่า row และ col ว่ามีค่าเกิน DIRCOLMAX ซึ่งเป็นขอบเขตคอลัมน์ที่กำหนดไว้หรือไม่ ซึ่งถ้าเกินขอบเขตแล้ว จะเลิกทำงานไป
3. นำค่า row และ col มาคำนวณค่าออกมาเป็นตำแหน่งบนจอภาพ ที่จะแสดงชื่อไฟล์ในไดเรคทอรีลิสต์โดยจะเริ่มต้นจากตำแหน่งของโหนดที่ dirpage ชี้อยู่ แล้วเลื่อนไปยังโหนดถัดไป เป็นจำนวนครั้งตามที่คำนวณได้ แล้วนำไปแสดงชื่อไฟล์นั้น
4. เพิ่มค่า row และ col อีกหนึ่ง แล้วกลับไปทำงานในข้อ 2

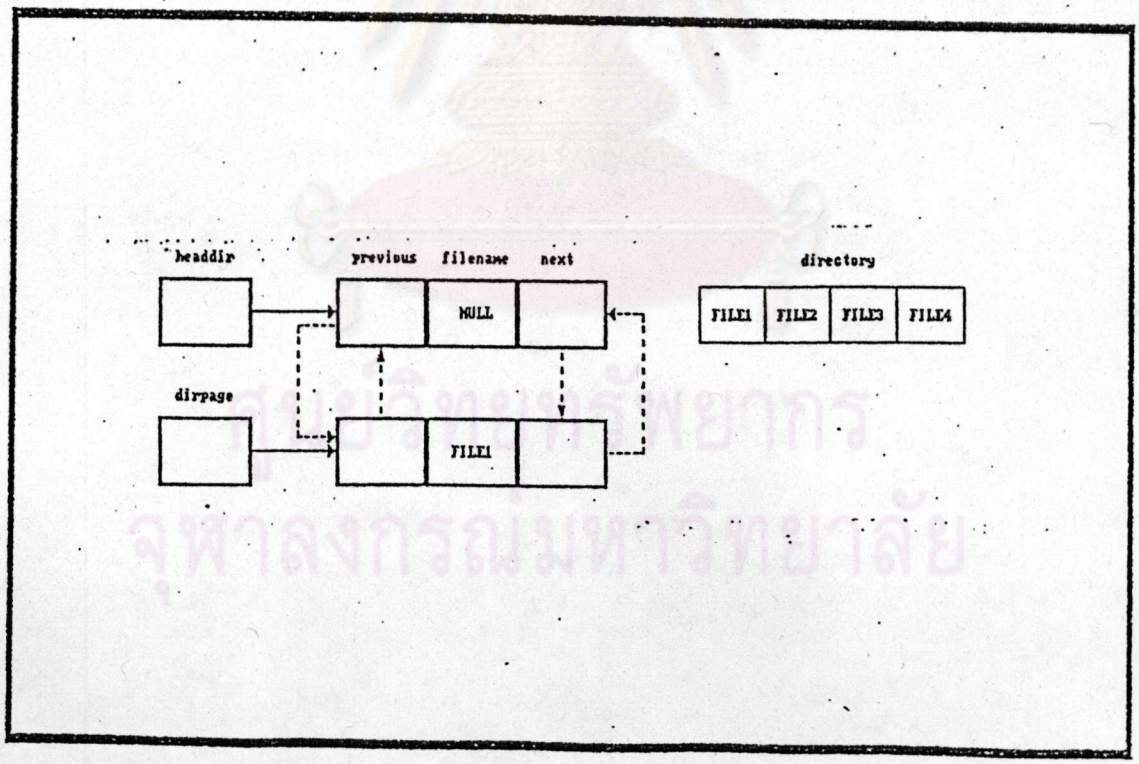
สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.40 แล้ว



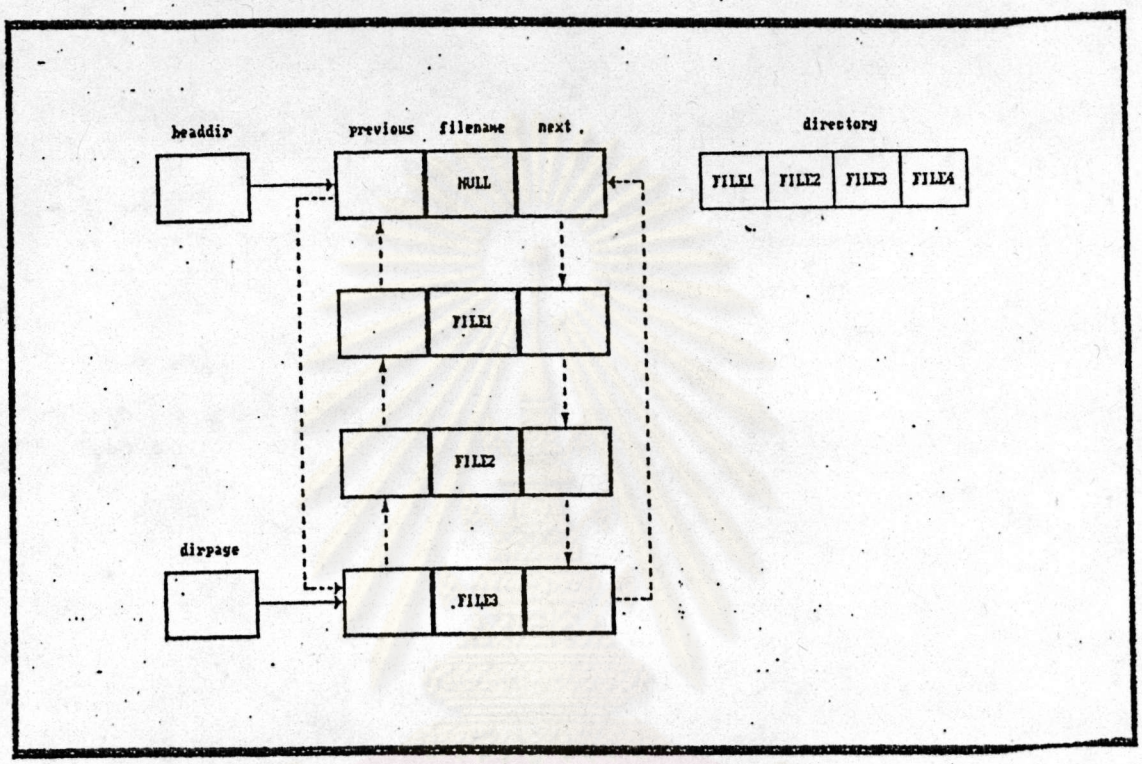
รูปภาพที่ 4.36 ผังงานแสดงขั้นตอนการทำงานของโมดูล creatdir



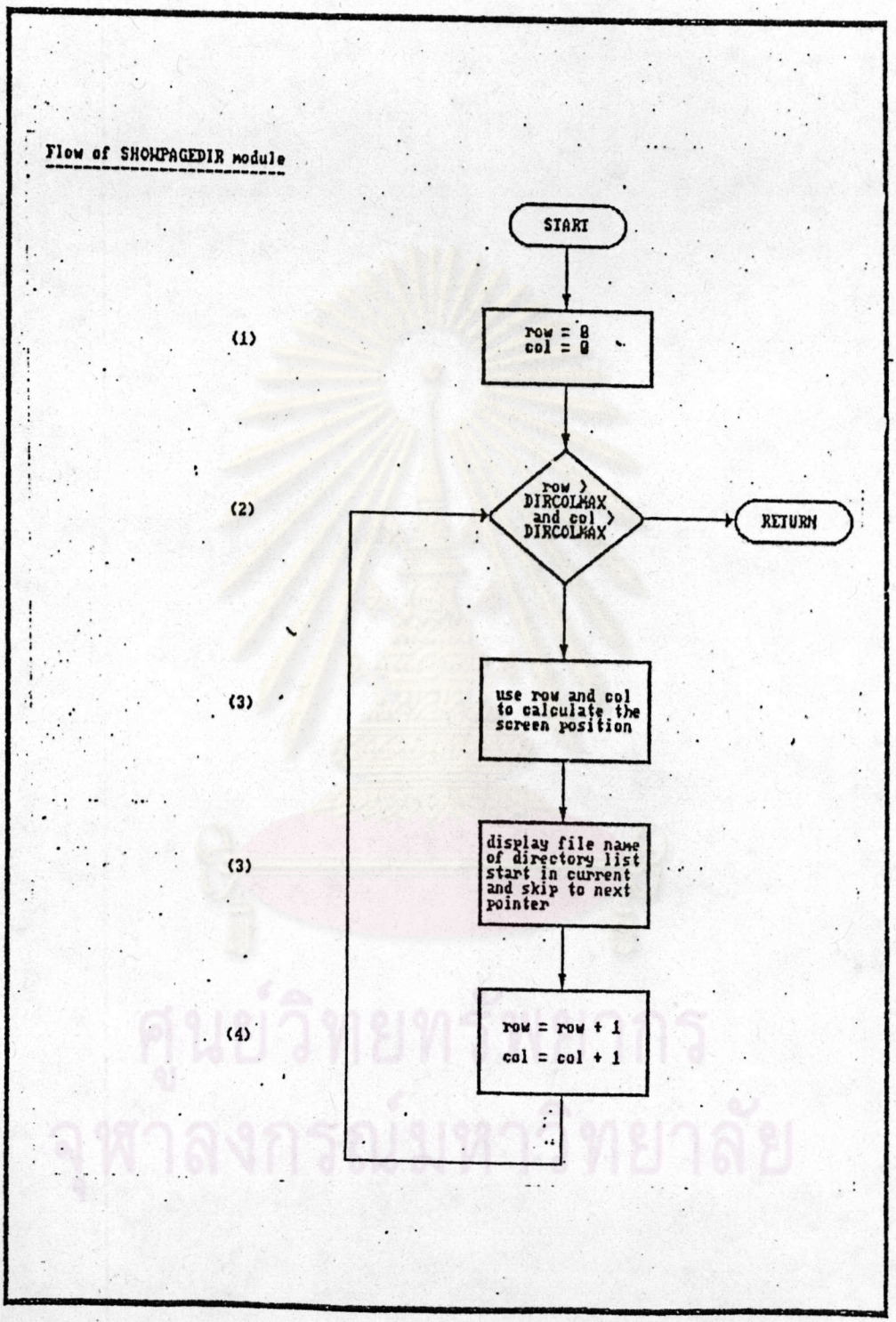
รูปภาพที่ 4.37 แสดงการเก็บข้อมูลของไดเรคทอรี่โหนดก่อนการทำงานของโมดูล creatdir



รูปภาพที่ 4.38 แสดงการเก็บข้อมูลของไดเรคทอรี่โหนดระหว่างการทำงานของโมดูล creatdir



รูปภาพที่ 4.39 แสดงการเก็บข้อมูลของไดเรคทอรี่โหนดหลังการทำงานของโมดูล createDir



รูปภาพที่ 4.40 ผังงานแสดงขั้นตอนการทำงานของโมดูล showpagedir

4.3.13.3 การเคลื่อนผ่านไปตามโหนด (Travel in Directory List)

จะเกิดขึ้นจากการที่เรากดคีย์ลูกศร เพื่อเลือกชื่อไฟล์ที่ต้องการ ทำให้เราจะต้องมีการเคลื่อนที่ไปตามโหนดต่างๆในทิศทางของคีย์ลูกศรที่กดไปด้วยคีย์ลูกศรที่จะใช้ได้มีทั้งหมด 6 อย่าง คือ คีย์ลูกศรซ้าย (Left) คีย์ลูกศรขวา (Right) คีย์ลูกศรขึ้น (Up) คีย์ลูกศรลง (Down) คีย์เลื่อนหน้าขึ้น (PgUp) และคีย์เลื่อนหน้าลง (PgDn) การทำงานนี้จะอยู่ภายใต้โมดูล dirleft dirright dirup dirdown dirpgup และ dirpgdn

<u>การทำงาน</u> จะมีขั้นตอนต่างๆดังนี้คือ

1. กำหนดตัวแปร row และ col ให้มีค่าเริ่มต้นเป็นค่าที่เหมาะสม ของแต่ละตัวแปร
2. ตรวจสอบ row และ col ว่าจะมีค่าเป็นศูนย์หรือไม่ ซึ่งถ้าเป็นศูนย์แล้ว ก็จะเลิกทำงาน
3. กำหนดตำแหน่งเริ่มต้นของโหนดที่ dirpage ชี้อยู่ แล้วเลื่อนไปยังโหนดอื่นๆตามทิศทางของคีย์ลูกศรเป็นจำนวนครั้งตามที่คำนวณได้ แล้วนำไปแสดงชื่อไฟล์นั้น

4.3.13.4 การยกเลิกไดเรคทอรีลิสต์ (Abort Directory List)

จะเป็นการทำให้รายชื่อไฟล์ต่างๆทั้งหมด ที่อยู่ในไดเรคทอรีลิสต์หายไป โดยการปลดปล่อยไดเรคทอรีโหนดให้เป็นอิสระ จะใช้เมื่อเราเลิกทำงานจากวิธีเลือกใช้ชื่อไฟล์แบบฟูลสกรีนไดเรคทอรี ด้วยวิธีการกดคีย์ Esc การทำงานนี้จะอยู่ภายใต้การทำงานของโมดูล freedir

<u>การทำงาน</u> จะมีขั้นตอนต่างๆดังนี้คือ

1. กำหนดตัวแปรพอยน์เตอร์ tempdir ให้ชี้ไปยังไดเรคทอรีโหนดตัวแรก ที่ใช้เก็บชื่อไฟล์ไว้
2. ตรวจสอบค่าของตัวแปรพอยน์เตอร์ tempdir ว่าเท่ากับ headdir หรือไม่
   - ถ้าเท่ากัน tempdir และ headdir ชี้ไปที่ไดเรคทอรีโหนดตัวเดียวกัน แสดงว่า มีการปลดปล่อยไดเรคทอรีโหนดให้เป็นอิสระจนหมดแล้ว แล้วจะเลิกทำงาน

- ถ้าไม่เท่ากัน จะทำงานดังนี้คือ

2.1 กำหนดตัวแปรพอยน์เตอร์ tempdir2 ให้ชี้ไปยังตำแหน่งเดียวกันกับที่พอยน์เตอร์ tempdir ชี้อยู่

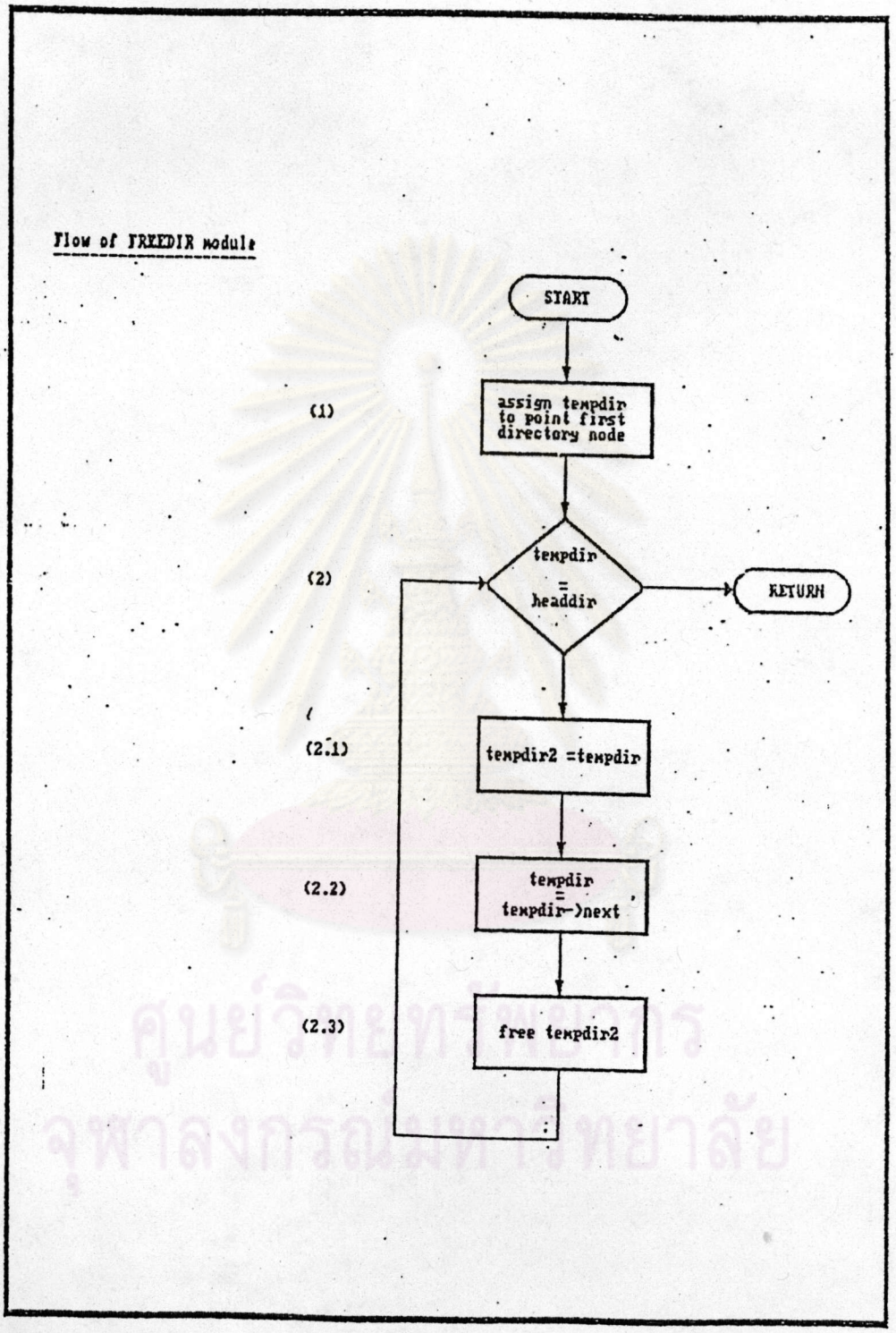
2.2 เปลี่ยนค่าให้ tempdir ชี้ไปยังไดเรคทอรีโหนดตัวถัดไปและทำให้ไดเรคทอรีโหนดที่ tempdir2 ชี้อยู่เป็นอิสระ

2.3 กลับไปทำงานในข้อ 2

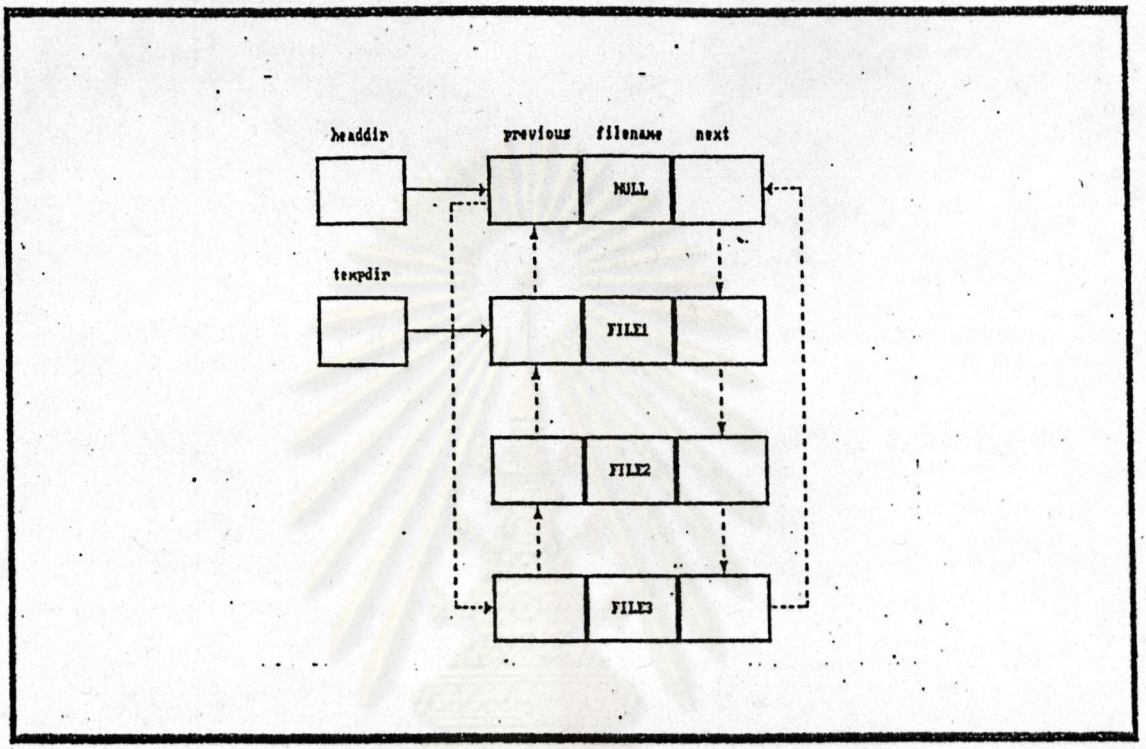
สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.41 แล้ว สมมติว่าขณะนั้นมีไดเรคทอรีลิสต์ ที่ประกอบด้วยไดเรคทอรีโหนดจำนวน 3 โหนด โดยจะมีพอยน์เตอร์ headdir ชี้ไปที่ส่วนหัวเริ่มต้นของไดเรคทอริลิสต์ และพอยน์เตอร์ tempdir ชี้ไปยังไดเรคทอรีแรก แสดงดังรูปภาพที่ 4.42 จากนั้นจะกำหนดให้พอยน์เตอร์ tempdir2 ชี้ไปยังไดเรคทอรีโหนดตำแหน่งเดียวกับ tempdir และเปลี่ยนค่าพอยน์เตอร์ tempdir ให้ชี้ไปยังไดเรคทอรีโหนดถัดไป แล้วปลดปล่อยไดเรคทอรีโหนดที่พอยน์เตอร์ tempdir2 ชี้อยู่ให้เป็นอิสระ ดังแสดงในรูปภาพที่ 4.43 และทำซ้ำเช่นนี้ไปเรื่อยๆจนกระทั่งพอยน์เตอร์ tempdir มาชี้ที่ส่วนหัวเริ่มต้นของไดเรคทอรีลิสต์ อีกครั้งหนึ่ง จึงจะเลิกทำงานดังแสดงผลการทำงานในรูปภาพที่ 4.44

4.3.14 <u>การแปลงรหัสควบคุมให้เป็นรหัสลักษณะพิเศษ</u>

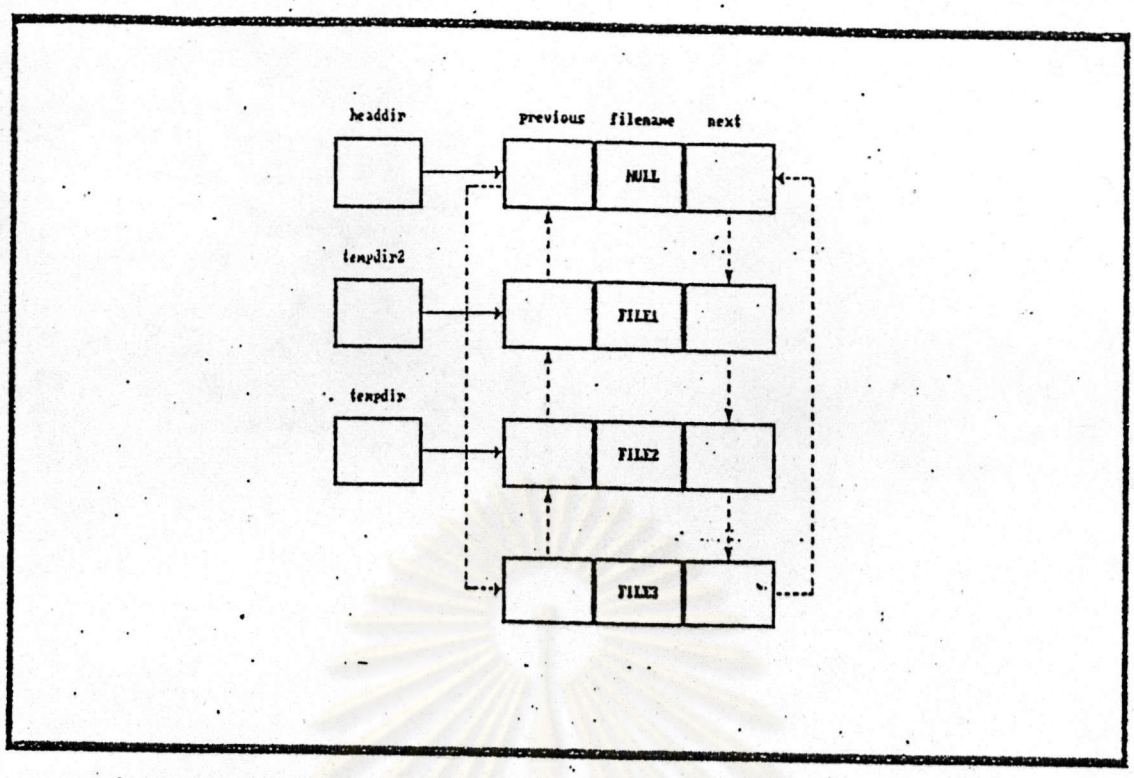
โดยปกติรหัสควบคุมจะเป็นตัวอักษรรหัส ASCII ที่มีค่าอยู่ระหว่าง 1 (ฐานสิบ) หรือ 01 (ฐานสิบหก) ไปจนถึงค่าของรหัส 31 (ฐานสิบ) หรือ 1F (ฐานสิบหก) จะใช้สำหรับควบคุมลักษณะพิเศษของการแสดงหรือพิมพ์ข้อความตามค่าในรหัสลักษณะพิเศษนั้นๆ ซึ่งรหัสควบคุมจะถูกเก็บไว้เป็นตัวอักษรเช่นเดียวกันกับข้อความในไฟล์ ดังนั้นเมื่อต้องการนำมาใช้งาน จะต้องมีการแปลงให้กลายเป็นรหัสลักษณะพิเศษก่อน เพื่อให้ได้ลักษณะพิเศษต่างๆของการแสดงหรือพิมพ์ข้อความ สำหรับรหัสควบคุม จะมีค่าต่างๆได้ดังนี้คือ



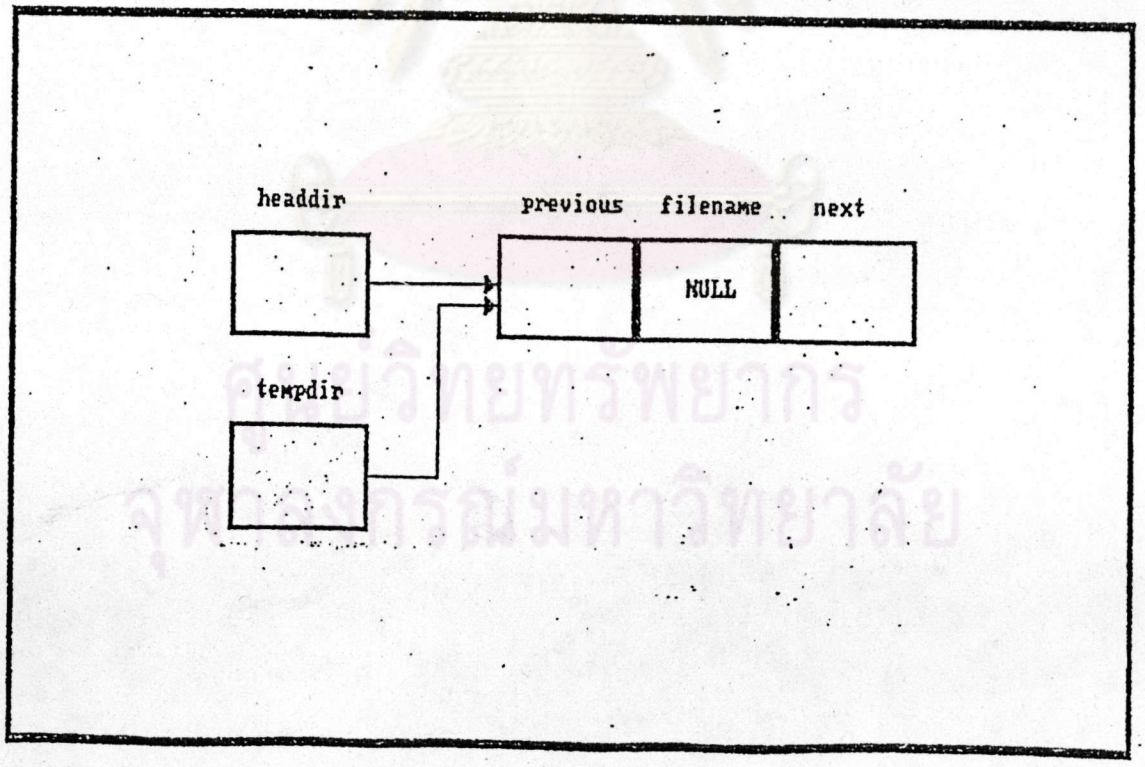
รูปภาพที่ 4.41 ผังงานแสดงขั้นตอนการทำงานของโมดูล freedir



รูปภาพที่ 4.42 แสดงไดเรคทอรี่ลิสต์ ก่อนการทำงานของโมดูล freedir



รูปภาพที่ 4.43 แสดงไดเรคทอรี่ลิสต์ ระหว่างการทำงานของโมดูล freedir



รูปภาพที่ 4.44 แสดงไดเรคทอรี่ลิสต์ หลังการทำงานของโมดูล freedir

ONELINECODE (รหัสควบคุมขีดเส้นใต้ 1 เส้น)
ค่าฐานสิบคือ 19 สัญญลักษณ์ ASCII คือ !!

BOLDCODE (รหัสควบคุมตัวเข้ม)
ค่าฐานสิบคือ 2 สัญญลักษณ์ ASCII คือ ☻

SUPERCODE (รหัสควบคุมตัวยกกำลัง)
ค่าฐานสิบคือ 20 สัญญลักษณ์ ASCII คือ ¶

SUBCODE (รหัสควบคุมตัวห้อย)
ค่าฐานสิบคือ 22 สัญญลักษณ์ ASCII คือ _

ITALICCODE (รหัสควบคุมตัวเอน)
ค่าฐานสิบคือ 23 สัญญลักษณ์ ASCII คือ ↨

ENLARGECODE (รหัสควบคุมตัวใหญ่)
ค่าฐานสิบคือ 5 สัญญลักษณ์ ASCII คือ ♣

TWOLINECODE (รหัสควบคุมขีดเส้นใต้ 2 เส้น)
ค่าฐานสิบคือ 18 สัญญลักษณ์ ASCII คือ ↕

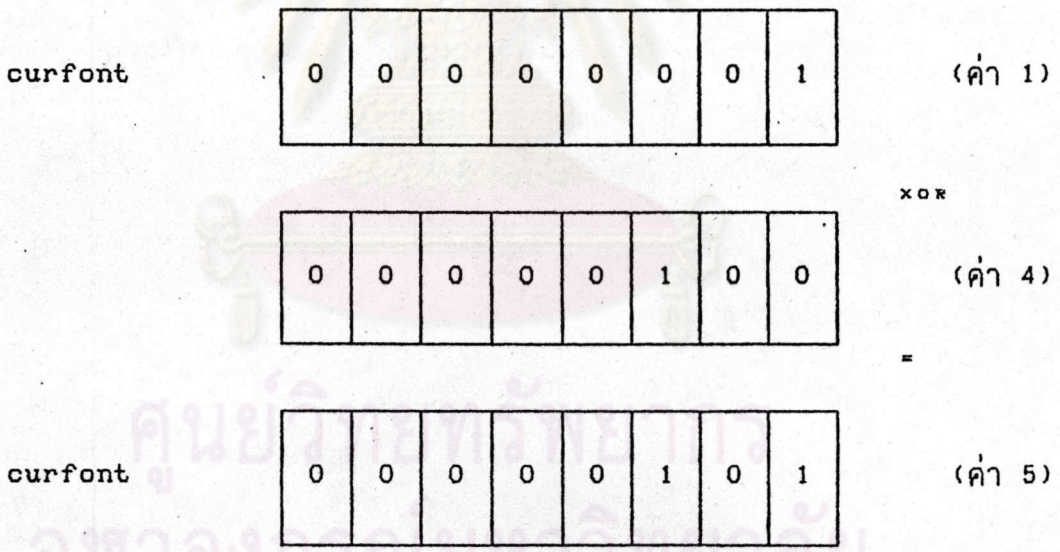
และจะมีรหัสลักษณะพิเศษ ที่ใช้คู่กันกับรหัสควบคุม ซึ่งจะมีค่าต่างๆได้ดังนี้คือ

ONELINEATTR	(ลักษณะพิเศษขีดเส้นใต้ 1 เส้น)	ค่าฐานสิบคือ	1
BOLDATTR	(ลักษณะพิเศษตัวเข้ม)	ค่าฐานสิบคือ	4
SUPERATTR	(ลักษณะพิเศษตัวยกกำลัง)	ค่าฐานสิบคือ	8
SUBATTR	(ลักษณะพิเศษตัวห้อย)	ค่าฐานสิบคือ	16
ITALICATTR	(ลักษณะพิเศษตัวเอน)	ค่าฐานสิบคือ	32
ENLARGEATTR	(ลักษณะพิเศษตัวใหญ่)	ค่าฐานสิบคือ	64
TWOLINEATTR	(ลักษณะพิเศษขีดเส้นใต้ 2 เส้น)	ค่าฐานสิบคือ	128

สำหรับการแปลงรหัสจะทำงานอยู่ภายใต้โมดูล togglefont โดยกำหนดค่าตัวแปร code เก็บค่ารหัสควบคุมที่ต้องการ และค่าตัวแปร curfont เก็บค่ารหัสลักษณะพิเศษ

<u>การทำงาน</u> จะมีขั้นตอนต่างๆดังนี้คือ

1. จะตรวจสอบค่าในตัวแปร code ให้ตรงกับชนิดรหัสควบคุม แล้วจะไปทำการกำหนดบิทของค่าในตัวแปร curfont ให้ ON หรือ OFF (XOR หรือ exclusive OR) ตามค่าลักษณะพิเศษที่ใช้คู่กับรหัสควบคุม เช่น ค่าในตัวแปร code มีค่าเป็น 2 ซึ่งตรงกับรหัสควบคุม BOLDCODE และค่าในตัวแปร curfont มีค่าเป็น 1 ซึ่งหมายถึงลักษณะแสดงเป็นตัวอักษรขีดเส้นใต้ 1 เส้น ดังนั้นจะนำค่าลักษณะพิเศษ BOLDATTR (มีค่าเป็น 4) มา XOR กับค่าในตัวแปร curfont ให้ผลมีค่าเป็น 5 ซึ่งหมายถึงลักษณะแสดงเป็นตัวอักษรขีดเส้นใต้ 1 เส้น และตัวเข้มจะแสดงผลการทำงาน ดังนี้คือ



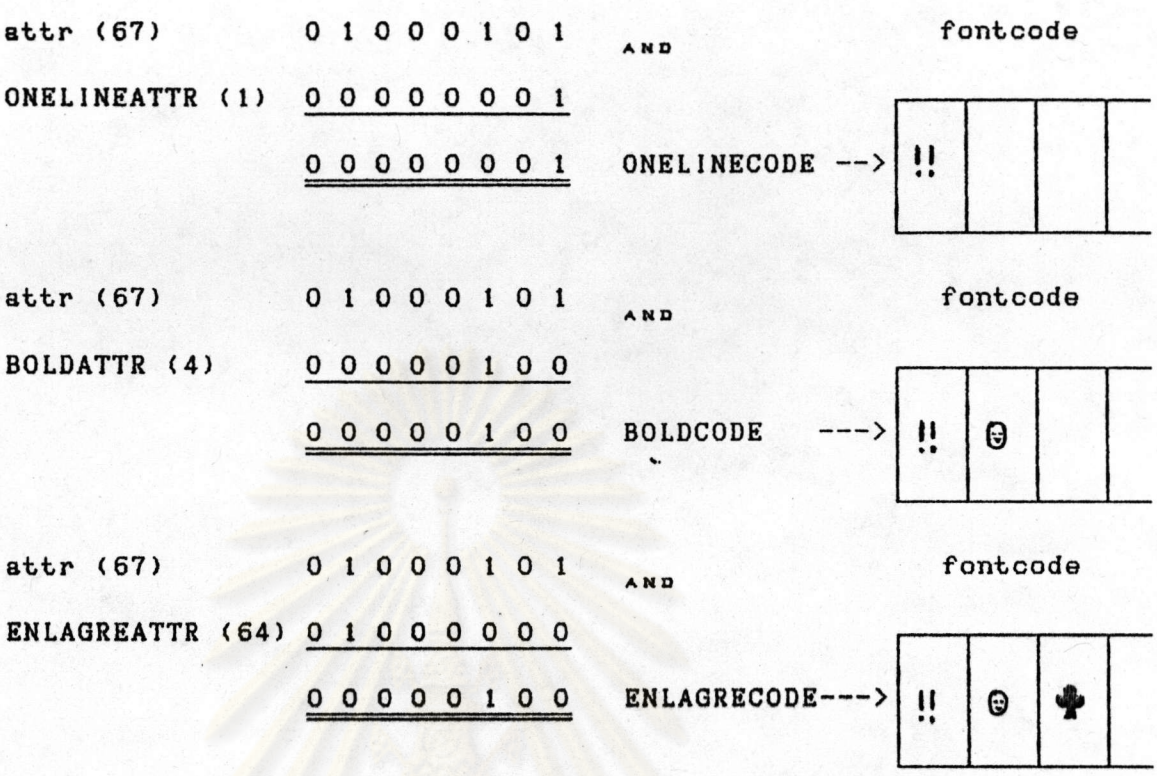
รูปภาพที่ 4.45 แสดงการแปลงรหสควบคุมให้เป็นรหัสลักษณะพิเศษ

4.3.15 <u>การแปลงรหัสลักษณะพิเศษให้เป็นรหัสควบคุม</u>

จะเป็นการแปลงรหัสลักษณะพิเศษให้กลายเป็น รหัสควบคุมเก็บไว้ได้ การทำงานจะอยู่ภายในโมดูล findstrcode โดยต้องกำหนดค่าของตัวแปร attr เพื่อเก็บค่าของรหัสลักษณะพิเศษที่ต้องการ และค่าของตัวแปร fontcode เพื่อเก็บค่าของรหัสควบคุมที่เกิดขึ้น ซึ่งจะเป็นการทำงานที่ตรงข้ามกับโมดูล togglefont

<u>การทำงาน</u> จะมีขั้นตอนต่างๆดังนี้คือ

1. จะตรวจสอบผลของการกำหนดบิทด้วย AND ระหว่างค่าในตัวแปร attr กับชนิดรหัสลักษณะพิเศษทุกรหัสว่ามีค่าไม่เป็นศูนย์หรือไม่ ถ้าไม่เป็นศูนย์จริงแสดงว่าค่าในตัวแปร attr จะแปลงเป็นรหัสควบคุมได้ โดยจะกำหนดค่ารหัสควบคุมที่ตรงกับชนิดรหัสลักษณะพิเศษนั้น ไปเก็บไว้ในตัวแปร fontcode ดังนั้นค่าในตัวแปร attr 1 ค่า อาจจะทำให้เกิดรหัสควบคุมหลายๆตัวได้ เช่น ค่าในตัวแปร attr มีค่าเป็น 69 (ฐานสิบ) ซึ่งเป็นลักษณะพิเศษแสดงตัวอักษรตัวใหญ่เข้มและขีดเส้นใต้ 1 เส้น ด้วย จากนั้นจะนำค่าในตัวแปร attr ไปทำบิท AND กับชนิดรหัสลักษณะพิเศษทุกรหัส ผลที่ได้คือ มี 3 รหัสลักษณะพิเศษที่กระทำแล้วไม่เป็นศูนย์ คือ ONELINEATTR BOLDATTR และ ENLARGEATTR แล้วจะกำหนดค่ารหัสควบคุม ONELINECODE BOLDCODE และ ENLARGECODE ไปเก็บไว้ในตัวแปร fontcode ซึ่งจะแสดงผลการทำงานดังนี้คือ



รูปภาพที่ 4.46 แสดงการแปลงรหัสลักษณะพิเศษให้เป็นรหัสควบคุม

4.3.16 การเรียกข้อมูลจากโหนดบัฟเฟอร์ไปไว้ในบัฟเฟอร์ทำงาน

เป็นการนำข้อมูลจากโหนดบัฟเฟอร์ที่ถูกชี้ด้วยโหนดพอยน์เตอร์ขณะนั้นมาแยกเก็บไว้ในบัฟเฟอร์ทำงานทั้ง 4 ระดับ เพื่อให้สามารถจัดการข้อมูลที่อยู่ในบัฟเฟอร์ทำงานได้โดยตรง เช่น การแทรกตัวอักษร การลบตัวอักษร เป็นต้น การทำงานจะอยู่ภายใต้โมดูล loadtoline โดยจะต้องกำหนดโหนดบัฟเฟอร์ที่ใช้งานอยู่ขณะนั้น

การทำงาน จะมีขั้นตอนดังนี้คือ

1. จะมีการเตรียมบัฟเฟอร์ทำงาน (Work Buffer) โดยจะให้บัฟเฟอร์ทำงานทั้ง 4 ระดับเป็นค่าช่องว่าง (Blank) ทั้งหมด และกำหนดค่าตัวแปร curfont ให้เป็นศูนย์ โดยตัวแปรนี้จะใช้สำหรับเก็บค่ารหัสลักษณะพิเศษของตัวอักษรขณะนั้น ซึ่งเป็นค่าศูนย์ จะหมายถึง ตัวอักษรลักษณะปกติ

2. นำข้อมูลในโหนดบัฟเฟอร์ทีละตัวอักษรมาใช้งาน โดยจะตรวจสอบว่าหมดข้อความในโหนดบัฟเฟอร์แล้วหรือไม่

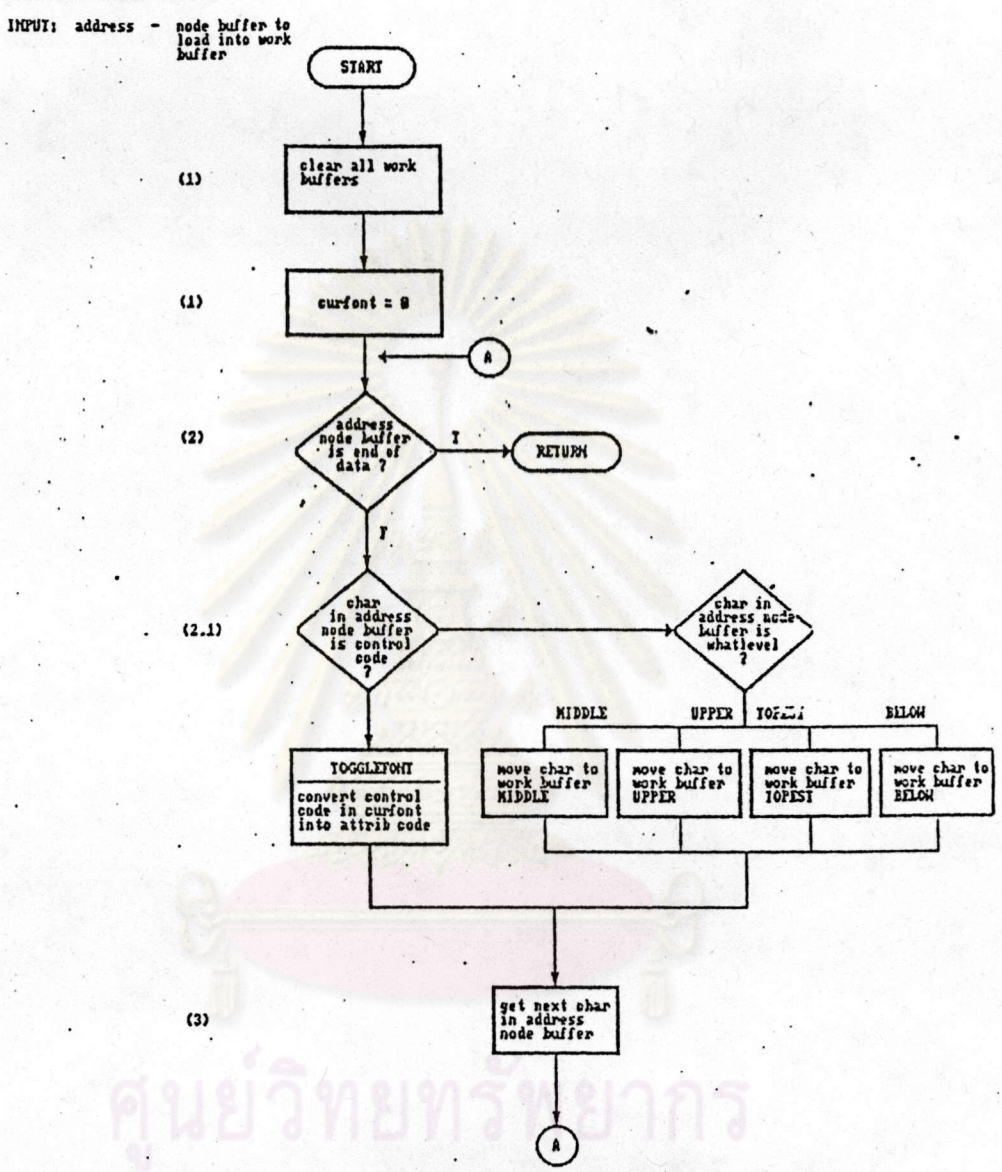
- ถ้าตัวอักษรหมดข้อความจริง ก็จะเลิกทำงาน
- ถ้ายังไม่หมดข้อความ จะทำงานตามขั้นตอนดังนี้คือ

2.1 ตรวจสอบตัวอักษรที่รับมา ว่าเป็นตัวรหัสควบคุมหรือไม่

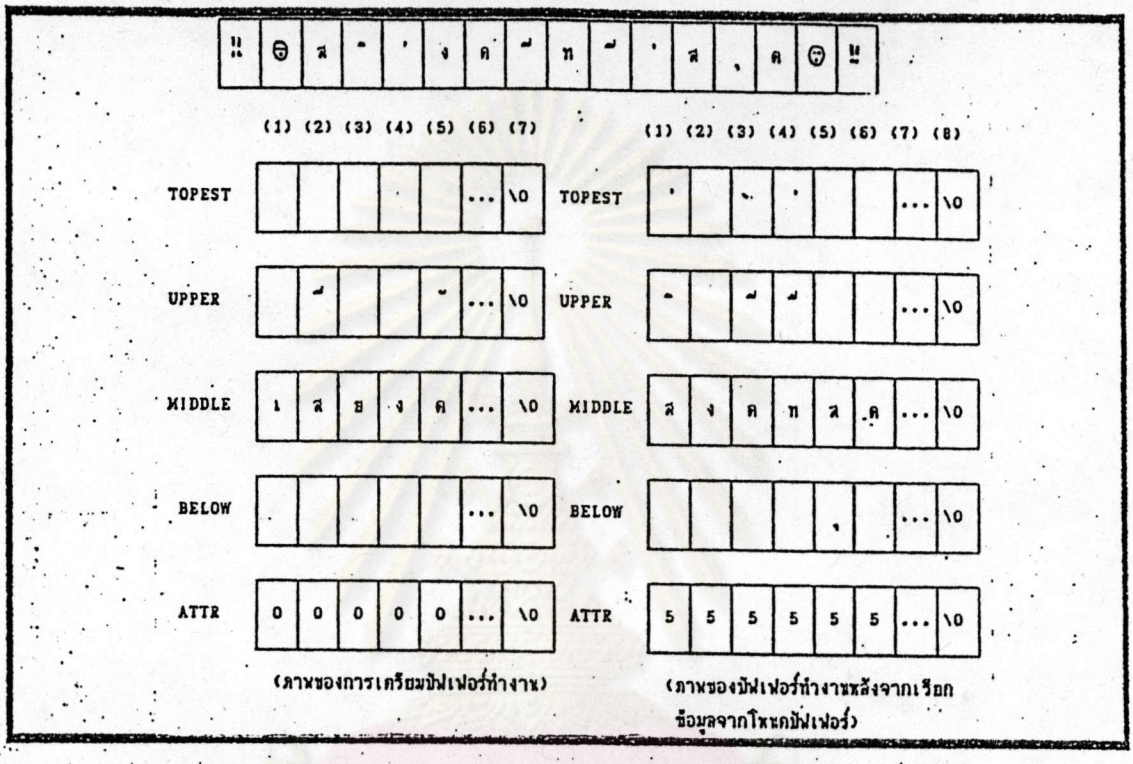
- ถ้าเป็นรหัสควบคุมจริง ก็จะไปทำงานในโมดูล "togglefont" เพื่อจะเปลี่ยนแปลงค่ารหัสลักษณะพิเศษที่เก็บไว้ใน curfont ให้เป็นไปตามค่ารหัสควบคุมนั้นๆ โดยการกำหนดบิทต่างๆของค่ารหัสลักษณะพิเศษนั้น
- ถ้าไม่ใช่ตัวรหัสควบคุมแล้ว แสดงว่าเป็นตัวอักษรทั่วไป ก็จะไปทำงานในโมดูล "whatlevel" เพื่อจะตรวจสอบระดับการจัดวางตัวอักษร โดยตัวอักษรที่เป็นระดับกลาง ก็จะถูกนำไปเก็บไว้ในบัฟเฟอร์ทำงาน MIDDLE และ กำหนดค่าบัฟเฟอร์ทำงาน ATTR ให้เป็นไปตามค่า curfont แต่ตัวอักษรที่เป็นระดับอื่นๆ ก็จะถูกนำไปเก็บไว้ในบัฟเฟอร์ทำงานอื่นๆ คือ UPPER TOPEST และ BELOW ตามความเหมาะสม

3. กลับไปรับตัวอักษรถัดไปจากโหนดบัฟเฟอร์ โดยไปทำงานในข้อ 2. ต่อไป

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.47 แล้ว สมมติว่า ในขณะนั้นมีโหนดบัฟเฟอร์อยู่จำนวนหนึ่งบรรทัด และบัฟเฟอร์ทำงานที่เตรียมไว้ ซึ่งถูกกำหนดให้เป็นช่องว่างทั้งหมด และ ผลจากการทำงานจะได้ตัวอักษรถูกแยกเก็บไว้ในแต่ละระดับของบัฟเฟอร์ทำงานดังแสดงผลในรูป 4.48 โดยรหัสควบคุม 2 ตัว ถูกแปลงให้เป็นรหัสลักษณะพิเศษ 1 ค่า ซึ่งมีค่าเป็น 5 (ฐานสิบ)



รูปภาพที่ 4.47 ผังงานแสดงขั้นตอนการทำงานของโมดูล loadtoline



รูปภาพที่ 4.48 แสดงการทำงานภายในบัฟเฟอร์ทำงานของโมดูล loadtoline

4.3.17 การเรียกข้อมูลจากบัฟเฟอร์ทำงานไปไว้ในโหนดบัฟเฟอร์

เป็นการนำข้อมูลจากบัฟเฟอร์ทำงานทั้ง 4 ระดับ เก็บรวมไว้เป็นบรรทัดเดียวกันในโหนดบัฟเฟอร์ที่ใช้งานอยู่ การทำงานจะภายใต้โมดูล storeline โดยจะต้องกำหนด curline เพื่อเก็บข้อมูลจากบัฟเฟอร์ทำงานไปไว้ใน โหนดบัฟเฟอร์ ซึ่งจะทำงานตรงข้ามกับโมดูล loadtoline

การทำงาน จะมีขั้นตอนดังนี้คือ

1. กำหนดค่าตัวแปร oldfont ให้เป็นศูนย์โดยตัวแปรนี้จะใช้สำหรับเก็บค่ารหัสลักษณะพิเศษของตัวอักษร ซึ่งเป็นค่าศูนย์ จะหมายถึง ตัวอักษรลักษณะปกติ และ ค่าตัวแปร fontcode ให้เป็นค่าช่องว่าง โดยตัวแปรนี้จะใช้สำหรับเก็บค่าควบคุมของตัวอักษร ซึ่งเป็นค่าช่องว่าง จะหมายถึง เป็นรหัสควบคุมของตัวอักษรปกติ
2. นำข้อมูลในบัฟเฟอร์ทำงาน MIDDLE ทีละตัวอักษรมาใช้งาน โดยจะตรวจสอบว่าหมดข้อความในบัฟเฟอร์ทำงานแล้วหรือไม่
   - ถ้าตัวอักษรหมดข้อความจริงก็จะเลิกทำงาน
   - ถ้ายังไม่หมดข้อความ ก็จะทำงานตามขั้นตอนดังนี้คือ
     - 2.1 ตรวจสอบค่าบัฟเฟอร์ทำงาน ATTR กับค่าในตัวแปร oldfont ว่ามีค่าเท่ากันหรือไม่ ถ้าไม่เท่ากันแสดงว่า มีการเปลี่ยนแปลงค่าในบัฟเฟอร์ทำงาน ATTR ดังนั้นจะนำค่าในบัฟเฟอร์ทำงาน ATTR นั้นไปทำงานโมดูล "findstrcode" เพื่อใช้แปลงค่ารหัสลักษณะพิเศษให้กลายเป็นค่ารหัสควบคุมของตัวอักษรได้ แล้วนำไปเก็บไว้โหนดบัฟเฟอร์ที่ใช้งานนั้น
     - 2.2 กำหนดค่าตัวแปร oldfont ให้มีค่าเท่ากับบัฟเฟอร์ทำงาน ATTR
     - 2.3 นำค่าในบัฟเฟอร์ทำงาน MIDDLE ไปเก็บไว้ในโหนดบัฟเฟอร์ถัดไป
     - 2.4 ตรวจสอบค่าบัฟเฟอร์ทำงานอื่นๆ คือ BELOW UPPER และ TOPEST ว่าเป็นค่าช่องว่างหรือไม่ ถ้าไม่เป็นช่องว่างจะนำค่าบัฟเฟอร์ทำงานนั้นไปเก็บไว้ในโหนดบัฟเฟอร์ถัดไป

2.5 รับตัวอักษรในบัฟเฟอร์ทำงาน MIDDLE ตัวถัดไป โดยกลับไปทำงานในข้อ 2.

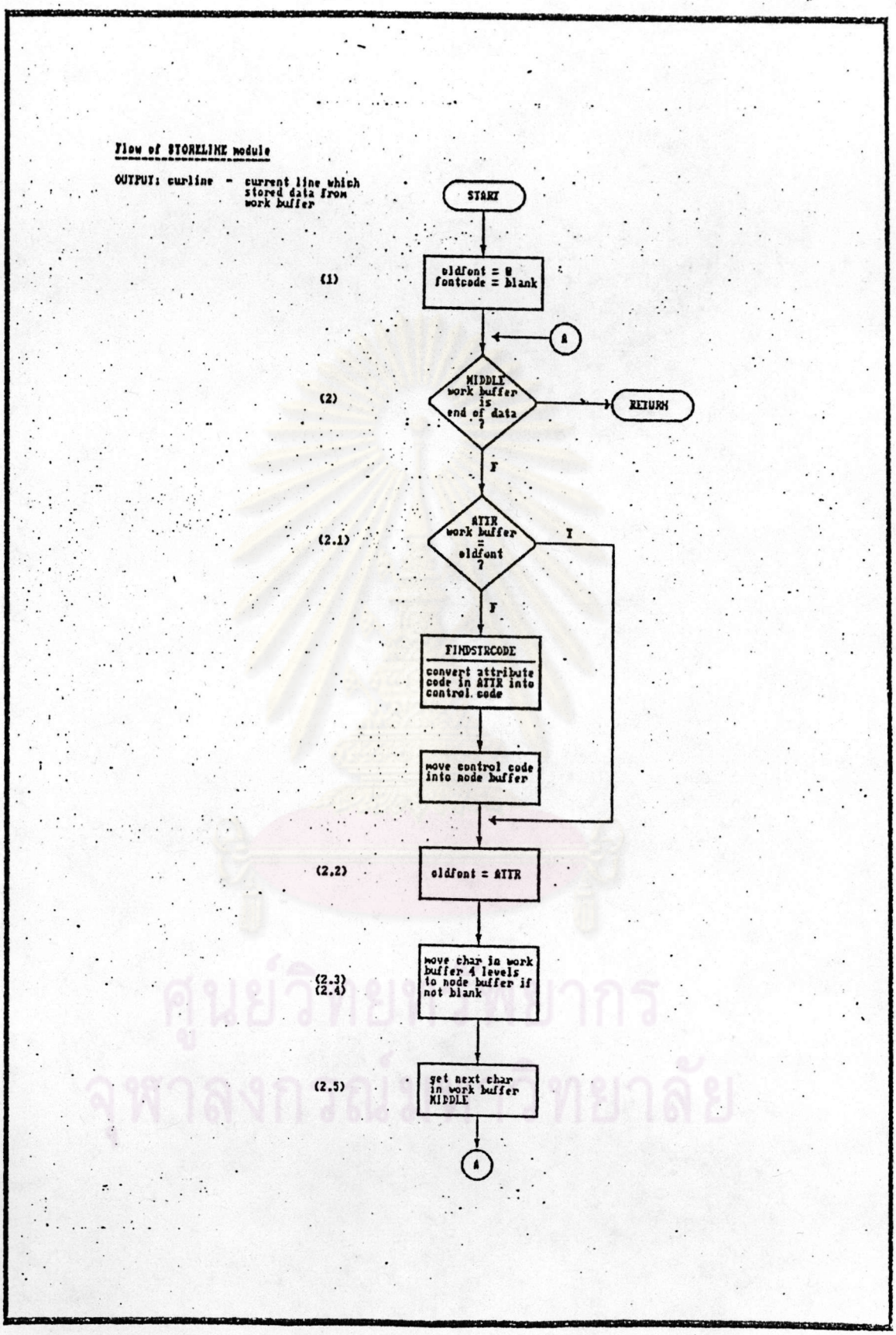
สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.49 แล้ว และในรูปภาพที่ 4.50 แสดงผลของการทำงาน ทำให้ตัวอักษรที่แยกในต่ละระดับของบัฟเฟอร์ทำงาน มาเก็บรวมกันเป็นบรรทัดเดียวในโหนดบัฟเฟอร์ โดยจะแปลงรหัสลักษณะพิเศษซึ่งในที่นี้เป็นค่า 5 ให้กลายเป็นรหัสควบคุม เก็บไว้โหนดบัฟเฟอร์ด้วย

4.3.18 <u>การแสดงข้อมูลในบัฟเฟอร์บนจอภาพ</u>

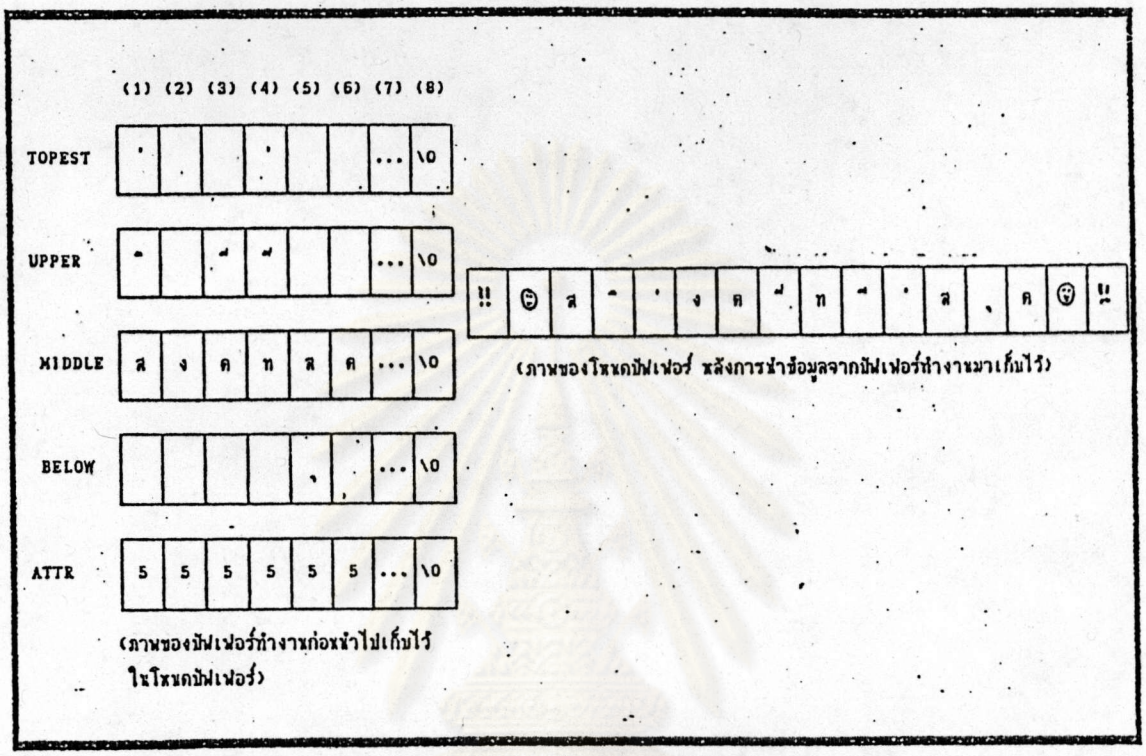
จะเป็นการนำข้อความในโหนดบัฟเฟอร์ มาแสดงบนจอภาพ โดยจะอาศัยโมดูล "prchar" ที่จะนำข้อความไปแสดงผลภาษาไทยบนจอภาพได้ ซึ่งในระหว่างการแสดงข้อความจะมีการตรวจสอบการกดคีย์ด้วย เพื่อให้สามารถรับคีย์ที่กดได้ทันกับการแสดงได้ การแสดงข้อความนี้จะเริ่มต้นจากโหนดบัฟเฟอร์ที่ใช้งานอยู่ขณะนั้น ซึ่งจะถูกชี้ด้วยโหนดพอยน์เตอร์ curpage ไปจนหมดข้อมูลของโหนดบัฟเฟอร์ หรือไปจนสุดขอบล่างของจอภาพที่กำหนดไว้ การทำงานนี้จะอยู่ภายใต้โมดูล showpage

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. กำหนดตัวแปร temppage เพื่อเก็บค่าของโหนดบัฟเฟอร์ curpage ที่จะทำให้ temppage ชี้ไปยังโหนดบัฟเฟอร์ที่เดียวกับ curpage ชี้อยู่และ กำหนดตัวแปร linenum เพื่อเก็บค่าของ lineno ซึ่งเป็นค่าหมายเลขบรรทัดขณะนั้นและตัวแปร y เพื่อใช้เก็บค่าจำนวนบรรทัดที่แสดง
2. ตรวจสอบค่าของตัวแปร y ว่าไม่เท่ากับค่าของตัวแปร wind.width ซึ่งเก็บจำนวนบรรทัดที่จะแสดงได้มากสุด และค่าตัวแปรพอยน์เตอร์ temppage ไม่เท่ากับค่าของ sentinel หรือไม่
   - ถ้าเป็นเท็จจะเลิกทำงาน โดยจะกำหนดให้ตัวแปร pagecomplete เป็น YES ซึ่งจะหมายความว่า การแสดงข้อความได้อย่างสมบูรณ์แล้วโดยไม่มีการขัดจังหวะด้วยการกดคีย์เลย



รูปภาพที่ 4.49 ผังงานแสดงขั้นตอนการทำงานของโมดูล storeline



รูปภาพที่ 4.50 แสดงการทำงานระหว่างบัฟเฟอร์กับโหนดบัฟเฟอร์ของโมดูล storeline

- ถ้าเป็นจริงทั้งสองกรณี แสดงว่า การแสดงข้อความยังไม่เกินสุดขอบล่างของจอภาพและข้อมูลในโหนดบัฟเฟอร์ยังไม่หมด จะมีขั้นตอนการทำงานดังนี้คือ

  2.1 ตรวจสอบว่า ขณะนั้นมีการกดคีย์หรือไม่ ด้วยโมดูล "keypressed" ถ้ามีการกดคีย์จริงแล้ว แสดงว่าขณะที่กำลังแสดงข้อความได้มีการกดคีย์ด้วย ซึ่งจะไปกำหนดค่าของตัวแปร pagecomplete ให้เป็นค่า NO แล้วเลิกทำงานไป

  2.2 ตรวจสอบค่าของตัวแปร temppage ว่าขณะนั้น ชี้ไปยังโหนดบัฟเฟอร์ที่เดียวกันกับตัวแปร curline หรือไม่

  - ถ้าตรงกัน จะไปทำงานในโมดูล "refreshline" เพื่อนำเอาข้อความในบัฟเฟอร์ทำงานมาแสดงบนจอภาพ
  - ถ้าไม่ตรงกัน จะไปทำงานในโมดูล "displine" เพื่อจะนำเอาข้อความในโหนดบัฟเฟอร์ มาแสดงบนจอภาพ

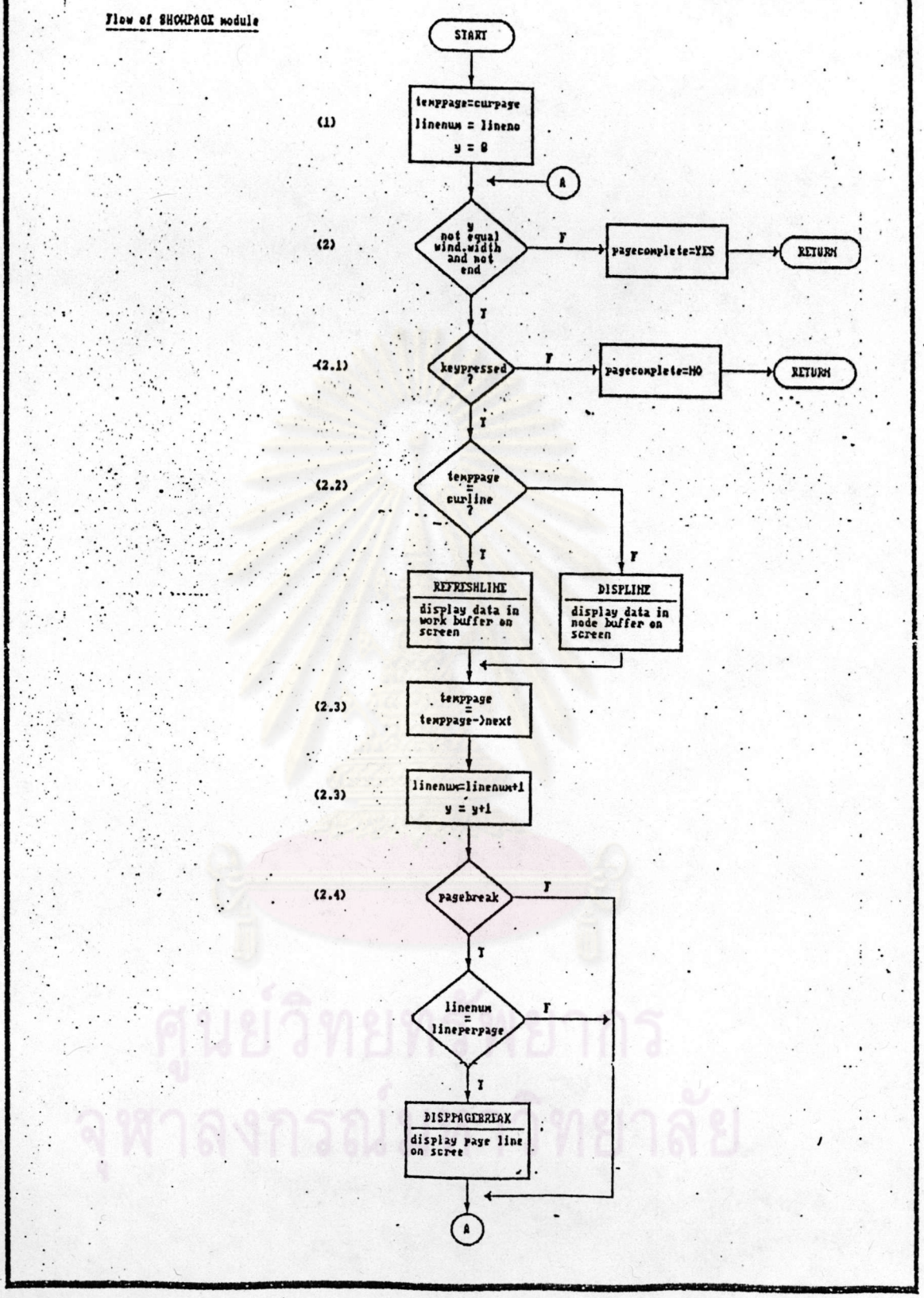
  2.3 เปลี่ยนค่าให้ตัวแปร temppage ไปชี้ยังโหนดบัฟเฟอร์ตัวถัดไป และเพิ่มค่าของตัวแปร linenum และ y ให้เพิ่มขึ้นอีก 1

  2.4 ตรวจสอบค่าในตัวแปร pagebreak ว่าเป็น YES หรือไม่ ถ้าเป็น YES จะหมายถึงว่า ต้องการให้มีการแสดงเส้นแบ่งหน้าบนจอภาพและตรวจสอบค่าในตัวแปร linenum กับ lineperpage ว่าเท่ากันหรือไม่

  - ถ้าเท่ากันแสดงว่า ขณะนั้นอยู่ในบรรทัดที่ครบหนึ่งหน้าแล้วจะไปทำงานในโมดูล "disppagebreak" เพื่อแสดงเส้นแบ่งหน้าบนจอภาพ

  2.5 กลับไปทำงานซ้ำในข้อ 2

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.51 แล้ว



รูปภาพที่ 4.51 ผังงานแสดงขั้นตอนการทำงานของโมดูล showpage

4.3.19 การทำงานเกี่ยวกับบล็อก (Block Operation)

ลักษณะของบล็อกจะเป็นการกำหนดข้อความให้เกิดเป็นกลุ่มข้อมูลเพื่อจะนำไปใช้งานบางอย่างตามที่ต้องการได้ โดยจะต้องกำหนดตำแหน่งเริ่มต้นและสุดท้ายของข้อความที่จะให้เป็นบล็อกไว้ การทำงานนี้ จะมีตัวแปรสำคัญที่จะใช้งานดังนี้คือ

- dispblock ใช้เป็นสภาวะการแสดงข้อความในบล็อก ให้เห็นเด่นชัดขึ้น
  - ถ้ามีค่า YES แสดงว่าขณะนั้นมีการแสดงข้อความในบล็อก ให้แตกต่างจากข้อความอื่นๆบนจอภาพ โดยจะแสดงในลักษณะแบบ Reverse (ตัวอักษรดำพื้นขาว)
  - ถ้ามีค่า NO แสดงว่าขณะนั้นต้องการแสดงข้อความในบล็อกให้เหมือนกับข้อความอื่นๆบนจอภาพ

- blkbegin และ blkend ใช้เก็บตำแหน่งเริ่มต้นและสุดท้ายของบล็อกโดยจะประกอบด้วยตัวแปร line เก็บหมายเลขบรรทัด และ ตัวแปร column เก็บหมายเลขคอลัมน์

ลักษณะการทำงานเกี่ยวกับบล็อกที่สำคัญ จะมีดังนี้คือ

4.3.19.1 การกำหนดบล็อก (Define Block)

จะเป็นการกำหนดให้ข้อความที่ต้องการ สามารถทำเป็นบล็อกได้ โดยการเลื่อนเคอร์เซอร์ไปยังตำแหน่งที่ต้องการให้เป็นจุดเริ่มต้นของบล็อก จากนั้นจะใช้คีย์ ^KB เพื่อบอกตำแหน่งเริ่มต้นของบล็อก แล้วเลื่อนเคอร์เซอร์ไปยังตำแหน่งที่ต้องการให้เป็นท้ายบล็อก และใช้คำสั่ง ^KK เพื่อบอกตำแหน่งท้ายของบล็อก การทำงานนี้จะอยู่ภายใต้โมดูล markbegin และ markend โดยจะต้องกำหนดตำแหน่งคอลัมน์เริ่มต้น (x) และสุดท้ายของบล็อก (y) ให้ด้วย

การทำงานของ markbegin จะมีขั้นตอนต่างๆดังนี้คือ

1. กำหนดค่าตัวแปร lineno ของ blkbegin ให้เป็นค่าหมายเลขบรรทัดเริ่มต้นที่ใช้งานขณะนั้น
2. กำหนดค่าตัวแปร column ของ blkbegin ให้เป็นค่าตำแหน่งคอลัมน์เริ่มต้นของบล็อก (x)
3. ตรวจสอบค่าตัวแปร lineno ของ blkbegin ว่ามีมากกว่าค่าตัวแปร lineno ของ blkend หรือไม่

   ถ้ามีค่ามากกว่า แสดงว่ากำหนดตำแหน่งบรรทัดเริ่มต้นของบล็อกอยู่หลังตำแหน่งบรรทัดสุดท้ายของบล็อก ซึ่งจะทำงานดังนี้คือ

   3.1 กำหนดค่าตัวแปร lineno ของ blkend ให้เท่ากับค่าในตัวแปร lineno ของ blkbegin

   3.2. กำหนดค่าตัวแปร column ของ blkend ให้เท่ากับค่าในตัวแปร column ของ blkbegin

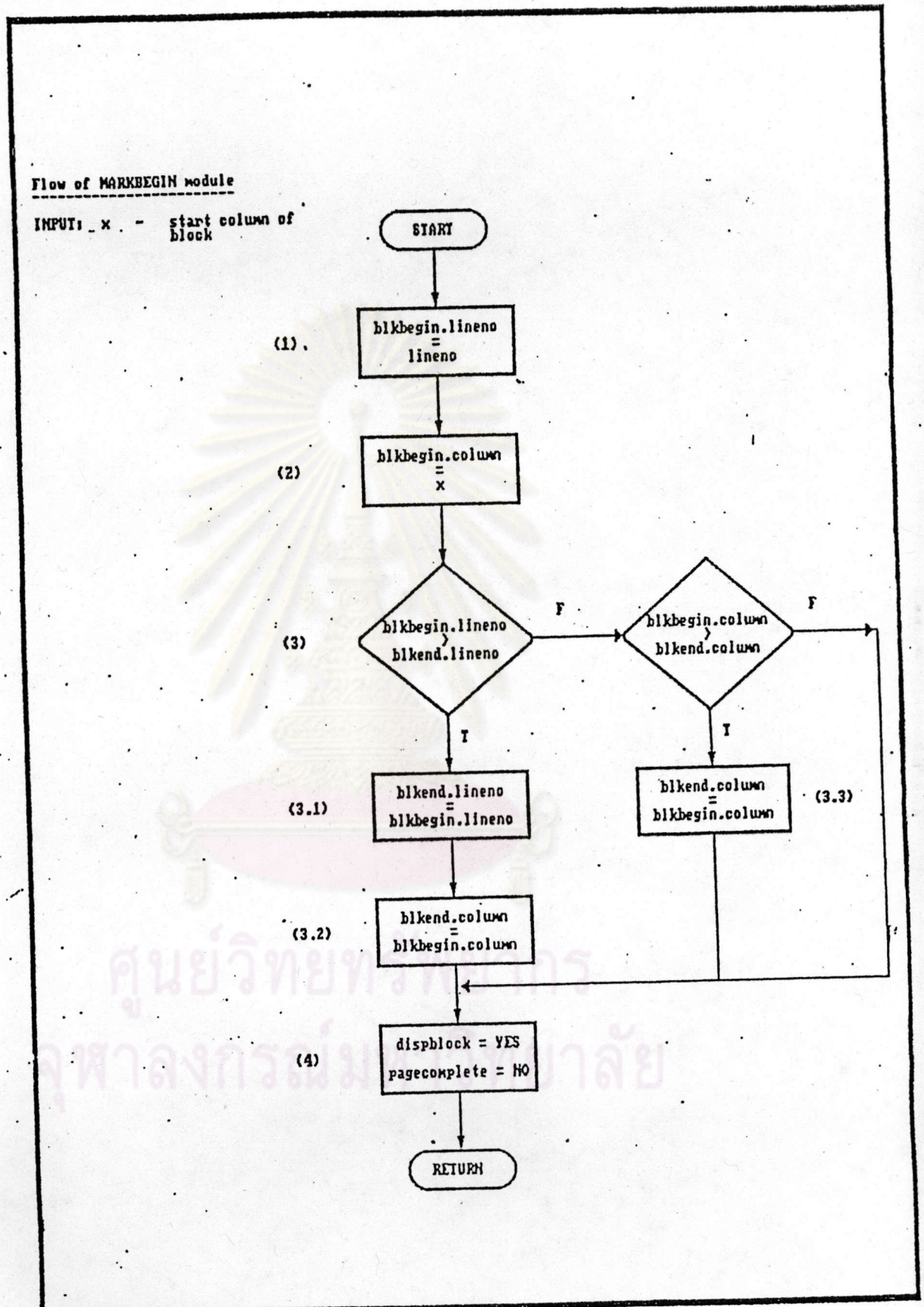
   ถ้ามีค่าเท่ากัน แสดงว่า กำหนดตำแหน่งบรรทัดเริ่มต้นของบล็อกให้อยู่บรรทัดเดียวกันกับตำแหน่งบรรทัดสุดท้ายของบล็อก ซึ่งก็จะทำงานดังนี้คือ

   3.3. ตรวจสอบค่าตัวแปร column ของ blkbegin ว่ามีมากกว่าค่าตัวแปร column ของ blkend หรือไม่

   - ถ้ามีค่ามากกว่า แสดงว่า กำหนดตำแหน่งคอลัมน์เริ่มต้นของบล็อกอยู่หลังตำแหน่งคอลัมน์สุดท้ายของบล็อก ก็จะกำหนดค่าตัวแปร column ของ blkend ให้เท่ากับค่าในตัวแปร column ของ blkbegin

4. กำหนดค่าตัวแปร dispblock เป็น YES และตัวแปร pagecomplete เป็น NO

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.52 แล้ว



รูปภาพที่ 4.52 ผังงานแสดงขั้นตอนการทำงานของโมดูล markbegin

<u>การทำงานของ markend</u> จะมีขั้นตอนต่างๆดังนี้คือ

1. กำหนดค่าตัวแปร lineno ของ blkend ให้เป็นค่าหมายเลขบรรทัดสุดท้ายที่ใช้งานขณะนั้น
2. กำหนดค่าตัวแปร column ของ blkend ให้เป็นค่าตำแหน่งคอลัมน์สุดท้ายของบล็อก (y)
3. ตรวจสอบค่าตัวแปร lineno ของ blkbegin ว่ามีมากกว่าค่าตัวแปร lineno ของ blkend หรือไม่

ถ้ามีค่ามากกว่า แสดงว่ากำหนดตำแหน่งบรรทัดเริ่มต้นของบล็อกอยู่หลังตำแหน่งบรรทัดสุดท้ายของบล็อก ซึ่งจะทำงานดังนี้คือ

3.1 กำหนดค่าตัวแปร lineno ของ blkbegin ให้เท่ากับค่าในตัวแปร lineno ของ blkend

3.2 กำหนดค่าตัวแปร column ของ blkbegin ให้เท่ากับค่าในตัวแปร column ของ blkend

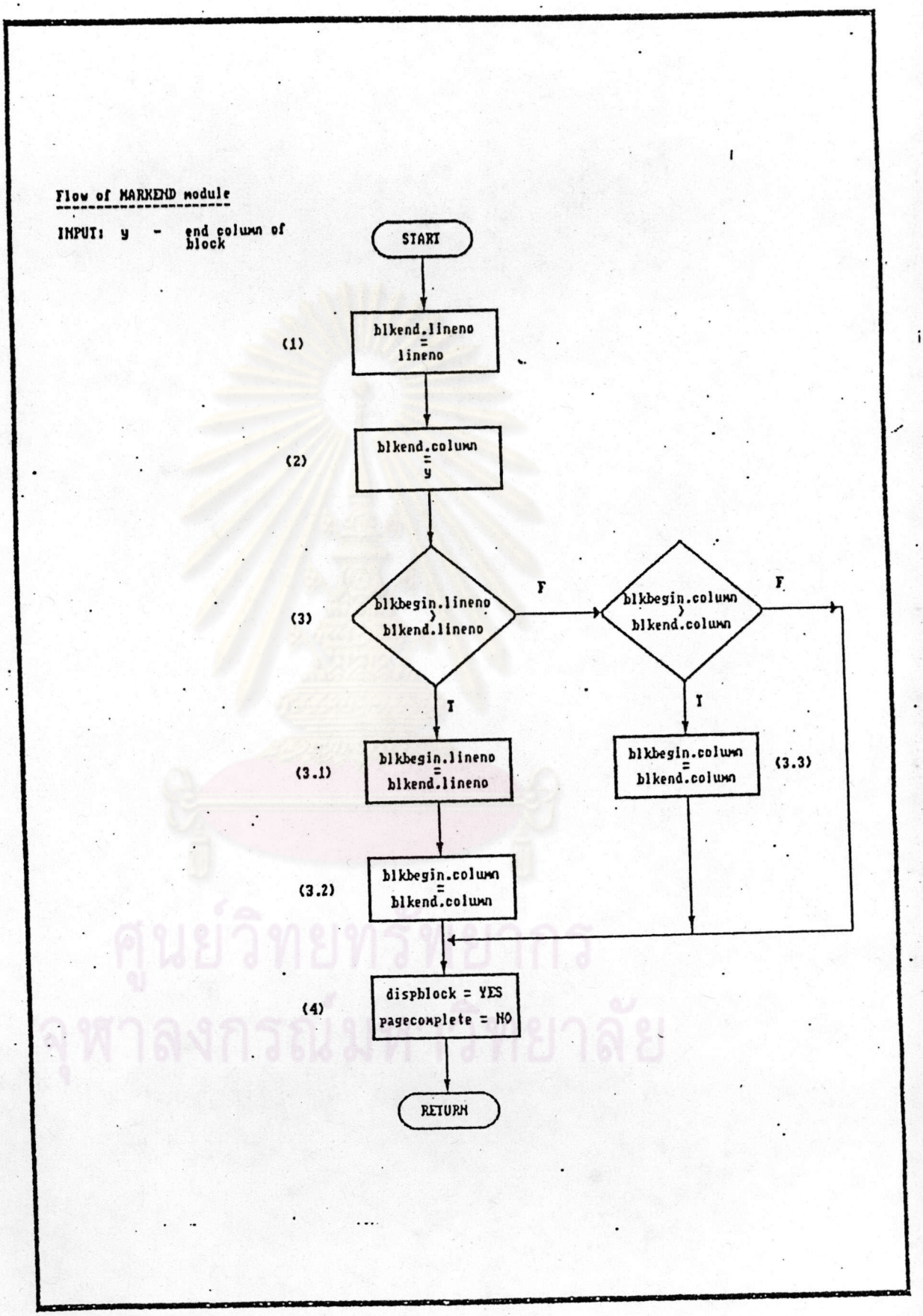
ถ้ามีค่าเท่ากัน แสดงว่า กำหนดตำแหน่งบรรทัดเริ่มต้นของบล็อกให้อยู่บรรทัดเดียวกันกับตำแหน่งบรรทัดสุดท้ายของบล็อก ซึ่งก็จะทำงานดังนี้คือ

3.3 ตรวจสอบค่าตัวแปร column ของ blkbegin ว่ามีมากกว่าค่าตัวแปร column ของ blkend หรือไม่

- ถ้ามีค่ามากกว่า แสดงว่า กำหนดตำแหน่งคอลัมน์เริ่มต้นของบล็อกอยู่หลังตำแหน่งคอลัมน์สุดท้ายของบล็อก ก็จะกำหนดค่าตัวแปร column ของ blkbegin ให้เท่ากับค่าในตัวแปร column ของ blkend

4. กำหนดค่าตัวแปร dispblock เป็น YES และตัวแปร pagecomplete เป็น NO

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.53 แล้ว



รูปภาพที่ 4.53 ผังงานแสดงขั้นตอนการทำงานของโมดูล markend

4.3.19.2 การตรวจสอบบล็อก (Check Block)

จะเป็นการตรวจสอบว่า ขณะนี้ได้มีการกำหนดบล็อกขึ้นมาหรือไม่ การทำงานนี้จะอยู่ภายใต้โมดูล haveblock

<u>การทำงาน</u> จะมีขั้นตอนต่างๆดังนี้คือ

1. ตรวจสอบค่าตัวแปร lineno และ column ของ blkbegin ว่าเท่ากับค่าตัวแปร lineno และ column ของ blkend หรือไม่
   - ถ้าเท่ากัน แสดงว่า ยังไม่ได้มีการกำหนดบล็อกมาก่อน จะให้ค่าผลลัพธ์เป็น NO
   - ถ้าไม่เท่ากัน แสดงว่า มีการกำหนดบล็อกแล้ว จะให้ค่าผลลัพธ์เป็น YES

4.3.19.3 การตรวจสอบตำแหน่งในบล็อก (Position in Block)

จะเป็นการตรวจสอบว่า ตำแหน่งบรรทัด (linenum) และคอลัมน์ (column) ที่กำหนดให้ อยู่ภายในบล็อกหรือไม่ การทำงานนี้จะอยู่ภายใต้โมดูล inblock

<u>การทำงาน</u> จะมีขั้นตอนต่างๆ ดังนี้คือ

1. ตรวจสอบค่าของบรรทัดที่กำหนดให้ (linenum) ว่าอยู่ระหว่างค่าตัวแปร lineno ของ blkbegin และ ค่าตัวแปร lineno ของ blkend หรือไม่
   1.1 ถ้าไม่อยู่ในระหว่างค่าตัวแปรทั้งสอง แสดงว่าตำแหน่งที่กำหนดให้ไม่ได้อยู่ภายในบล็อกแล้ว จะให้ค่าผลลัพธ์เป็น NO
   1.2 ถ้าอยู่ในระหว่างค่าตัวแปรทั้งสองแล้วจะทำงานดังนี้คือ
      - ตรวจสอบตำแหน่งบรรทัดที่กำหนดให้ ว่าอยู่บรรทัดเดียวกันกับค่าในตัวแปร lineno ของ blkbegin หรือไม่ โดยถ้าอยู่บนบรรทัดเดียวกันจริงแล้ว จะตรวจสอบตำแหน่งคอลัมน์ที่กำหนดให้ (column) ว่าน้อยกว่าค่าตัวแปร column ของ blkbegin หรือไม่
         ถ้าเป็นจริง แสดงว่าตำแหน่งที่กำหนดให้ไม่ได้อยู่ภายในบล็อก จะให้ผลลัพธ์เป็น NO
         ถ้าไม่จริง แสดงว่าตำแหน่งที่กำหนดให้อยู่ภายในบล็อกจะให้ผลลัพธ์เป็น YES

- ตรวจสอบตำแหน่งบรรทัดที่กำหนดให้ ว่าอยู่บรรทัดเดียวกันกับค่าในตัวแปร lineno ของ blkend หรือไม่ โดยถ้าอยู่บนบรรทัดเดียวกันจริงแล้วจะตรวจสอบตำแหน่งคอลัมน์ที่กำหนดให้ว่า มากกว่าหรือเท่ากับค่าตัวแปร column ของ blkend หรือไม่

  ถ้าเป็นจริง แสดงว่าตำแหน่งที่กำหนดให้ไม่ได้อยู่ภายในบล็อก จะให้ผลลัพธ์เป็น NO

  ถ้าไม่จริง แสดงว่าตำแหน่งที่กำหนดให้อยู่ภายในบล็อกจะให้ผลลัพธ์เป็น YES

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.54 แล้ว

4.3.19.4 การคัดลอกข้อมูลมาเก็บในที่ว่าง (Copy Data to Space)

จะเป็นการคัดลอกข้อมูลจากโหนดบัฟเฟอร์ ของบรรทัดและคอลัมน์เริ่มต้นจนถึงบรรทัดและคอลัมน์สุดท้ายที่กำหนดให้ ไปเก็บไว้ในที่ว่าง (space) ชั่วคราวเพื่อจะใช้ทำงานต่อไป การทำงานจะอยู่ภายใต้โมดูล copytospace โดยจะรับค่า fline และ lline ที่เป็นค่าบรรทัดเริ่มต้นและบรรทัดสุดท้าย และค่า fcol และ lcol ที่เป็นค่าคอลัมน์เริ่มต้นและคอลัมน์สุดท้าย ซึ่งให้ผลลัพธ์เป็นค่าพอยน์เตอร์ ซึ่งจะชี้ไปยังบรรทัดแรกของที่ว่างนั้น

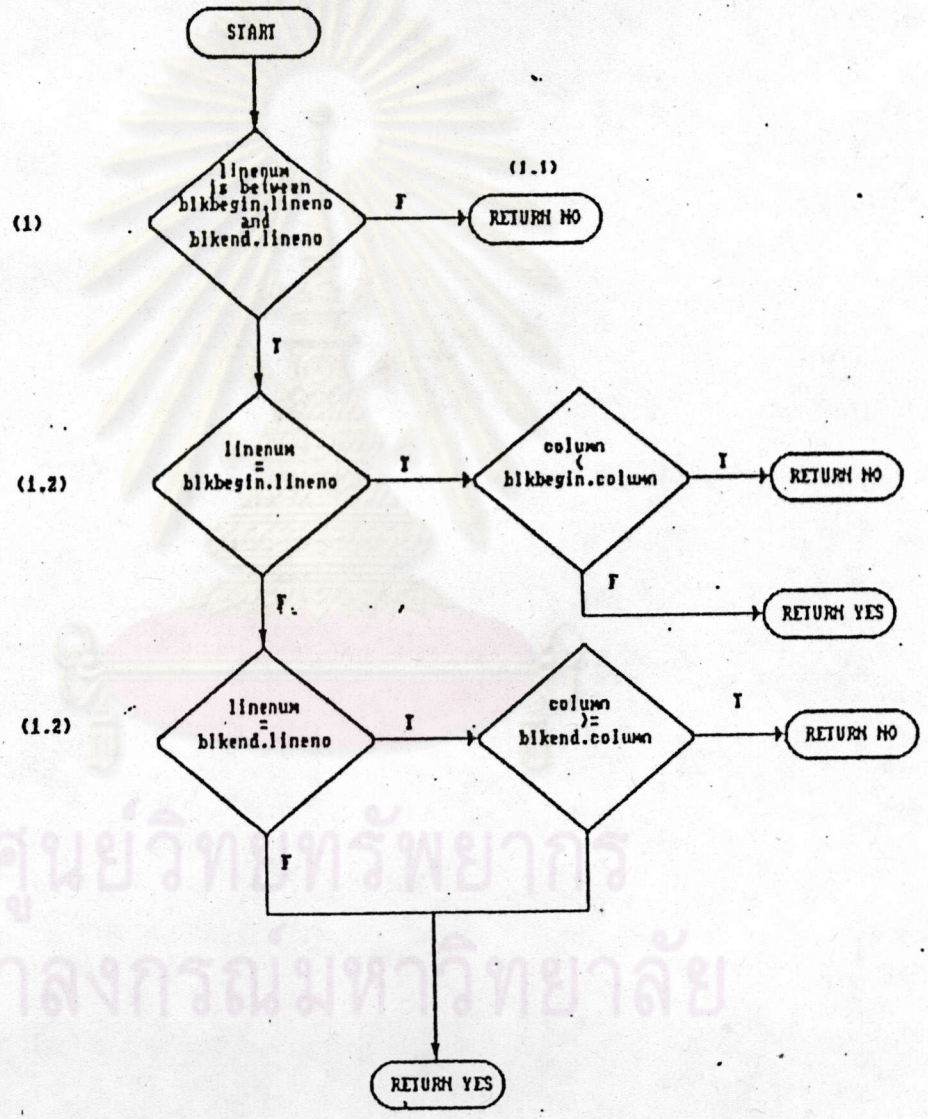
<u>การทำงาน</u> จะมีขั้นตอนต่างๆ ดังนี้คือ

1. กำหนดพอยน์เตอร์ firstline ที่จะใช้ชี้ไปยังโหนดพอยน์เตอร์ที่สร้างขึ้นใหม่ โดยจะกำหนดโหนดพอยน์เตอร์ให้มีค่าต่างๆ ดังนี้คือ
   - o previous Pointer และ next Pointer ให้ชี้ไปยังโหนดพอยน์เตอร์ตัวเองก่อน
   - o กำหนดค่า wrap ให้เท่ากับค่า wrap ของ fline ที่ชี้อยู่บรรทัดเริ่มต้น
   - o คัดลอกข้อมูลจากบัฟเฟอร์โหนด ตั้งแต่ตำแหน่งที่ fcol ของบรรทัดที่ถูกชี้ด้วย fline ไปยังบัฟเฟอร์โหนดที่สร้างใหม่ ซึ่งจะกำหนดให้ถูกชี้ด้วย text

Flow of INBLOCK module

INPUT: linenum - line no to check
colnum - column no to check

OUTPUT: YES - position is in the block
NO - position is not in the block



รูปภาพที่ 4.54 ผังงานแสดงขั้นตอนการทำงานของโมดูล inblock

2. ตรวจสอบค่าของตัวแปร fline และ lline ว่ามีค่าเท่ากันหรือไม่

o ถ้าเท่ากัน แสดงว่า fline และ lline อยู่ในบรรทัดเดียวกัน ก็จะคำนวณจำนวนของตัวอักษรที่จะคัดลอกจากบัฟเฟอร์โหนดของบรรทัดที่ถูกชี้ด้วย fline และ lline จากค่าของ fcol และ lcol ไปยังบัฟเฟอร์โหนดที่สร้างใหม่ ซึ่งจะกำหนดให้ถูกชี้ด้วย text

o ถ้าไม่เท่ากันแสดงว่า fline และ lline อยู่คนละบรรทัด จะทำงานดังนี้คือ กำหนดตัวแปรพอยน์เตอร์ templine ให้ชี้ไปที่เดียวกันกับ firstline และทำงานซ้ำๆกันจนกว่าตัวแปร fline เท่ากับ lline โดยทำตามขั้นตอนดังนี้

- เปลี่ยนค่าตัวแปร fline ให้ชี้ไปยังบรรทัดถัดไป
- กำหนดพอยน์เตอร์ newline ใช้ชี้ไปยังโหนดพอยน์เตอร์ที่สร้างขึ้นใหม่ เพื่อจะใช้เป็นพอยน์เตอร์สำหรับคัดลอกข้อความมาเก็บไว้ในที่ว่าง
- ตรวจสอบค่าตัวแปรของ fline กับ lline

  ถ้า fline ไม่ เท่ากับ lline แสดงว่า fline ยังไม่ถึงบรรทัดสุดท้าย จะมีการกำหนดดังนี้คือ

  - กำหนดค่า wrap ให้เท่ากับค่า wrap ของ fline ที่ชี้อยู่บรรทัดเริ่มต้น
  - คัดลอกข้อมูลจากบัฟเฟอร์โหนดทั้งหมดของบรรทัด ที่ถูกชี้ด้วย fline ไปยังบัฟเฟอร์โหนดที่สร้างใหม่ ซึ่งจะกำหนดให้ถูกชี้ด้วย text ของ newline
  - ไปทำงานในโมดูล "insert_line" เพื่อนำบรรทัดที่ถูกชี้ด้วย templine และ newline มาเชื่อมต่อกัน
  - เปลี่ยนค่าตัวแปร templine ให้ชี้ไปยังบรรทัดถัดไป

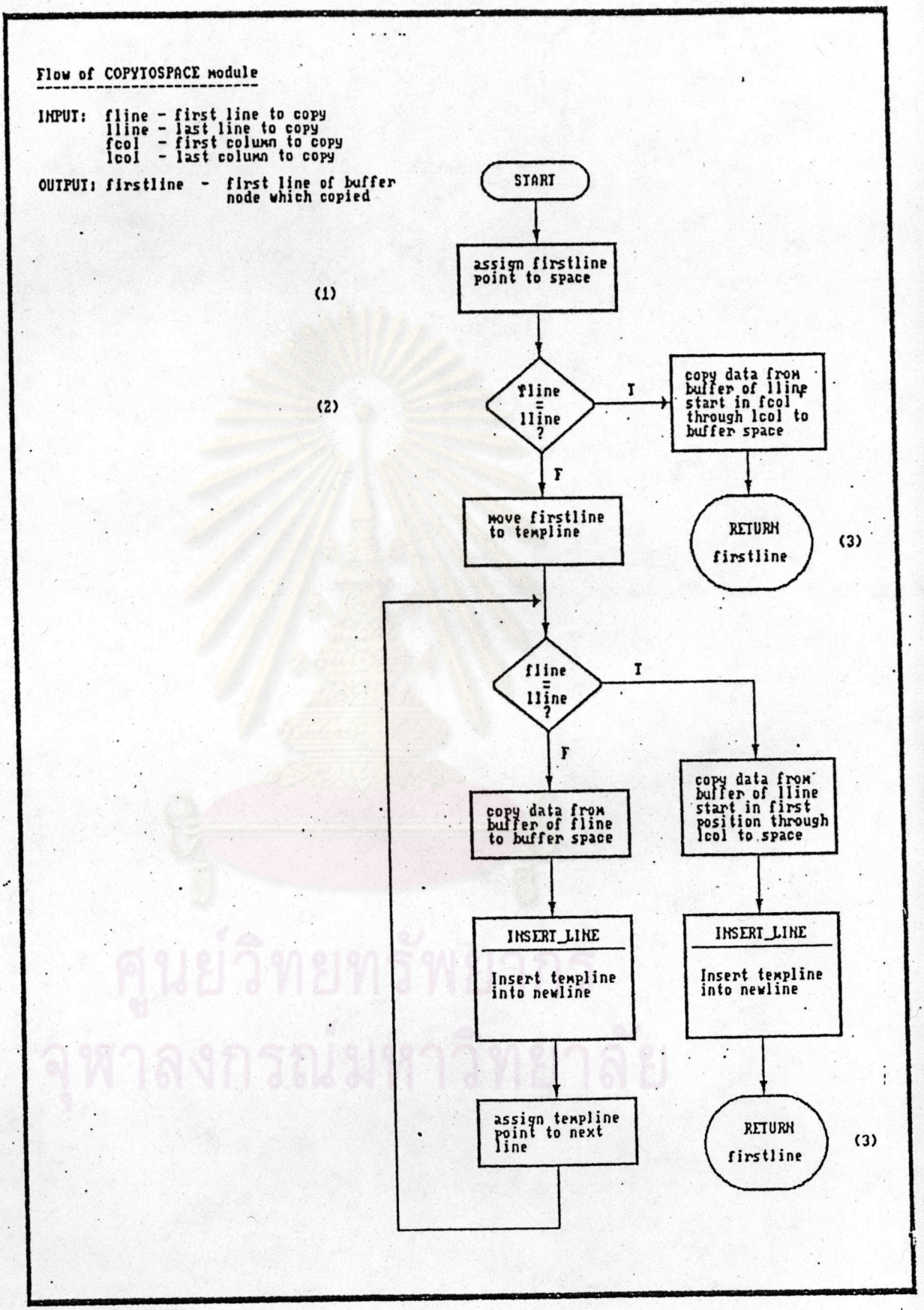
ถ้าค่าตัวแปร fline เท่ากับ lline แสดงว่า fline มาถึงบรรทัดสุดท้ายแล้ว จะมีการกำหนดดังนี้คือ

- กำหนดค่า wrap ให้เท่ากับค่า wrap ของ fline ที่ชี้อยู่บรรทัดเริ่มต้น
- คัดลอกข้อมูลจากบัฟเฟอร์โหนด ตั้งแต่ตำแหน่งแรก จนถึงตำแหน่งที่ lcol ของบรรทัดที่ถูกชี้ด้วย lline ไปยังบัฟเฟอร์โหนดที่สร้างใหม่ ซึ่งจะกำหนดให้ถูกชี้ด้วย text ของ newline
- ไปทำงานในโมดูล "insert_line" เพื่อนำบรรทัดที่ถูกชี้ด้วย templine และ newline มาเชื่อมต่อกัน

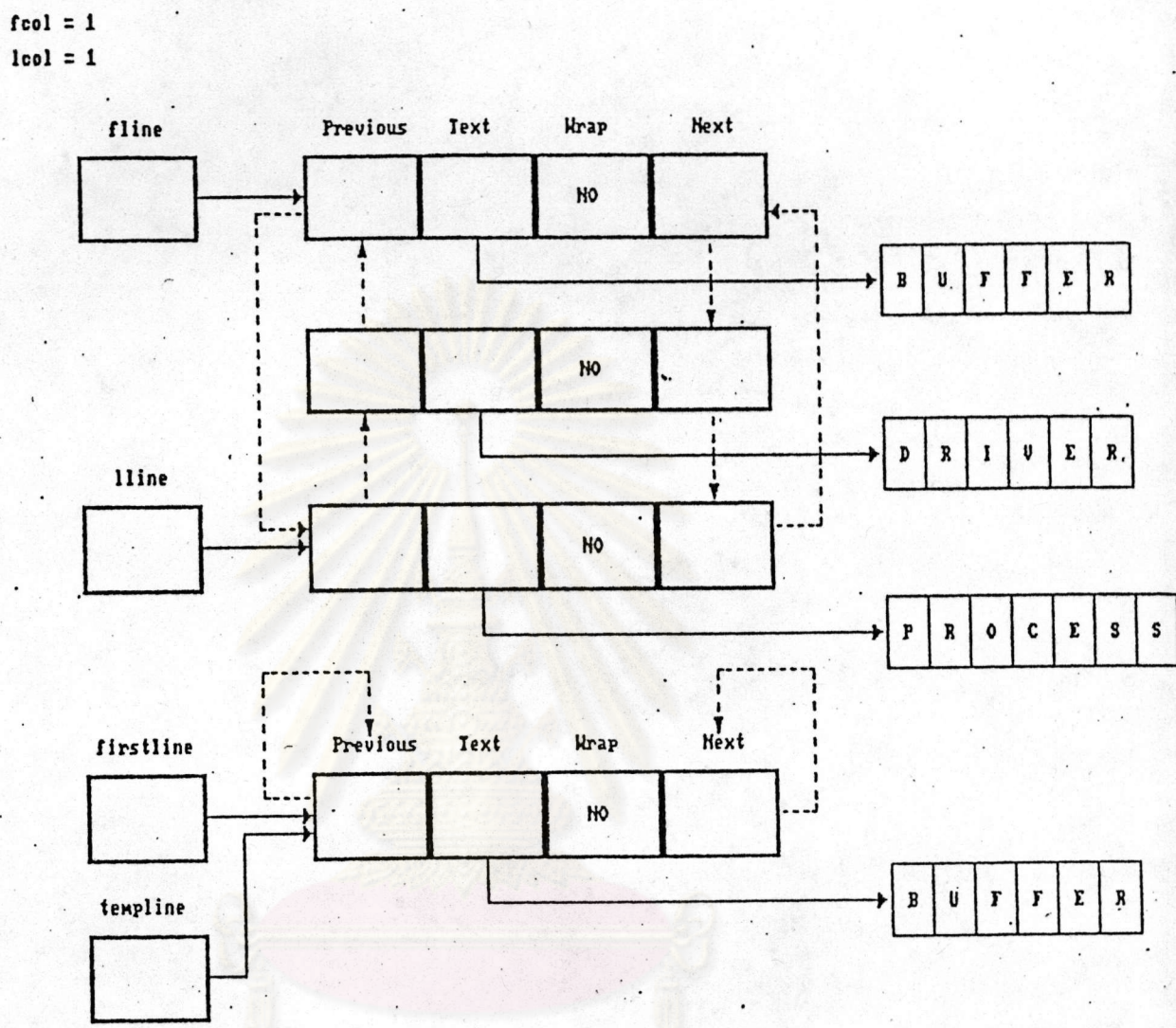
3. ให้ผลลัพธ์เป็นค่าของตัวแปร firstline ซึ่งเป็นพอยน์เตอร์ที่จะชี้ไปยังบรรทัดแรกของข้อมูลที่คัดลอกมาไว้ในที่ว่าง ซึ่งสร้างเตรียมไว้

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.55 แล้ว โดยการทำงานเราจะแบ่งออกเป็น 2 กรณี คือ ในกรณีที่บรรทัดต้นแบบมีข้อมูลมากกว่า 1 บรรทัด และกรณีที่บรรทัดต้นแบบมีข้อมูลเพียง 1 บรรทัด

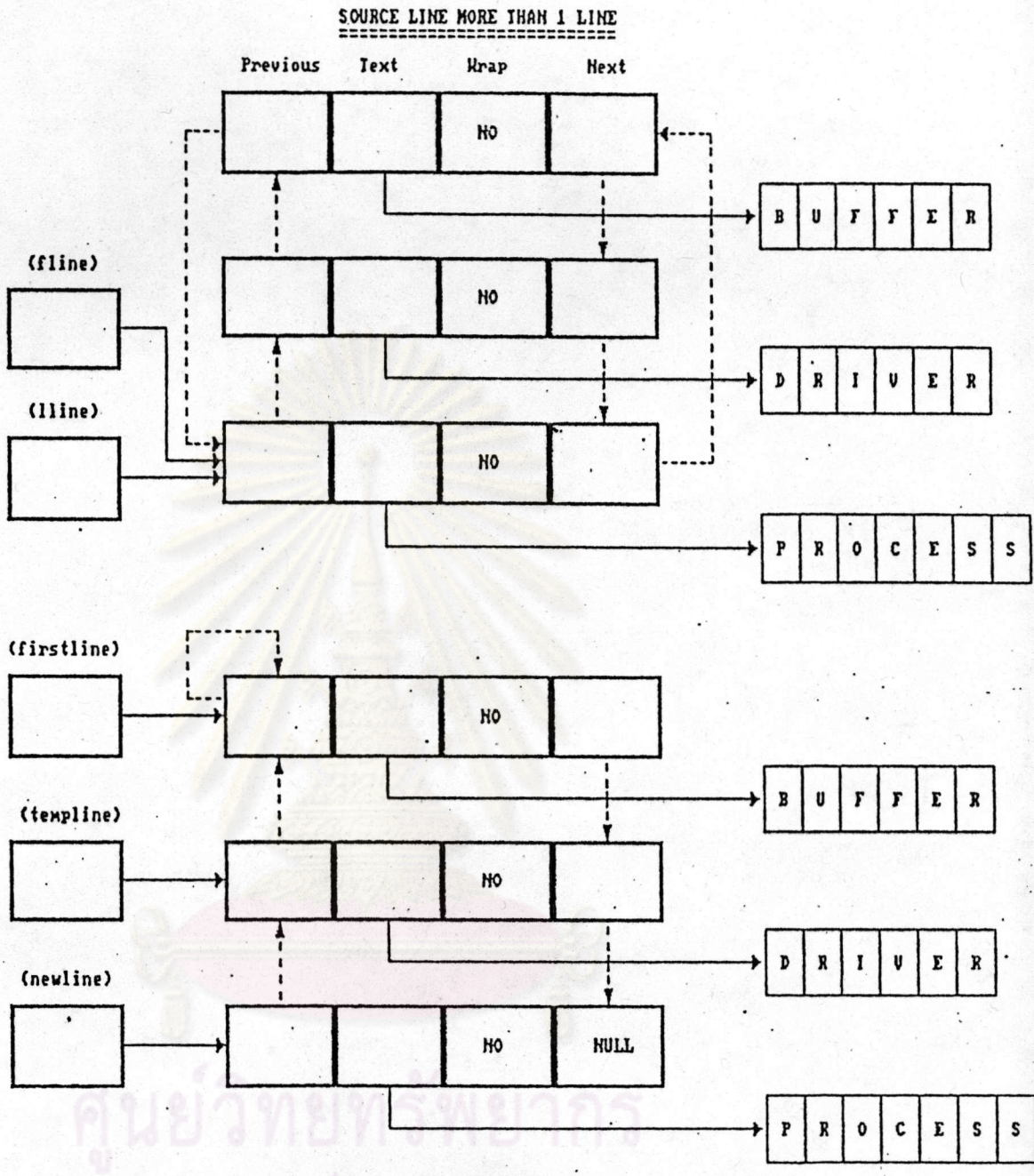
สำหรับในภาพที่ 4.56 แสดงการเก็บข้อมูลของบรรทัดต้นแบบที่มากกว่า 1 บรรทัด ซึ่งจะมีพอยน์เตอร์ fline ชี้ไปที่บรรทัดแรก และ lline ชี้ไปที่บรรทัดสุดท้ายรวมทั้งโหนดบัฟเฟอร์ที่ได้สร้างใหม่ สำหรับคัดลอกข้อมูลจากบรรทัดต้นแบบในคอลัมน์ fcol ของบรรทัดแรก ไปจนถึงคอลัมน์ lcol ของบรรทัดสุดท้าย (ในที่นี้จะกำหนดให้ fcol และ lcol เป็น 1) ซึ่งจะถูกชี้ด้วยพอยน์เตอร์ firstline และ templine โดยในรูปภาพจะเป็นการเริ่มต้นทำงานที่คัดลอกข้อมูลบรรทัดแรกของต้นแบบ มาไว้ที่โหนดบัฟเฟอร์ว่างตัวแรก จากนั้นก็จะคัดลอกข้อมูลบรรทัดถัดไปของต้นแบบ มาเก็บไว้ในโหนดบัฟเฟอร์ที่สร้างใหม่ แล้วไปเชื่อมต่อกับโหนดบัฟเฟอร์ที่อยู่ก่อนหน้า ทำซ้ำๆกันเช่นนี้ สุดท้ายจะได้ผลดังรูปที่ 4.57 โดยจะให้ค่าของพอยน์เตอร์ firstline ซึ่งชี้ไปยังโหนดบัฟเฟอร์ใหม่ตัวแรก



รูปภาพที่ 4.55 ผังงานแสดงขั้นตอนการทำงานของโมดูล copytospace



รูปภาพที่ 4.56 แสดงการเก็บข้อมูลของบรรทัดต้นแบบที่มากกว่า 1 บรรทัด ก่อนการทำงานของโมดูล copytospace



รูปภาพที่ 4.57 แสดงการเก็บข้อมูลของบรรทัดต้นแบบที่มากกว่า 1 บรรทัด หลังการทำงานของโมดูล copytospace

สำหรับในภาพที่ 4.58 แสดงการเก็บข้อมูลของบรรทัดต้นแบบที่มีเพียง 1 บรรทัด โดยคัดลอกข้อมูลจากบรรทัดต้นแบบในคอลัมน์ fcol จนถึงคอลัมน์ lcol (ในที่นี้จะกำหนดให้ fcol เป็น 4 และ lcol เป็น 7) ของบรรทัดเดียวกัน มาเก็บไว้โหนดบัฟเฟอร์ แล้วจะให้ค่าของพอยน์เตอร์ firstline ด้วยเช่นกัน

4.3.19.5 การเชื่อมต่อชุดของบรรทัดข้อมูลเข้าด้วยกัน (Insert Link List)

จะเป็นการนำเอาชุดของบรรทัดข้อมูลที่กำหนดให้ มาเชื่อมต่อกันกับบรรทัดข้อมูลผลลัพธ์ที่ใช้งานอยู่ การทำงานนี้จะอยู่ภายใต้โมดูล insertlinklist ซึ่งจะต้องกำหนดพอยน์เตอร์ sourceline ที่จะชี้ไปยังบรรทัดเริ่มต้น และ พอยน์เตอร์ destline ที่จะชี้ไปยังบรรทัดผลลัพธ์ ซึ่งจะนำบรรทัดข้อมูลที่ถูกชี้ด้วย sourceline มาเชื่อมกันในตำแหน่งคอลัมน์ destcol

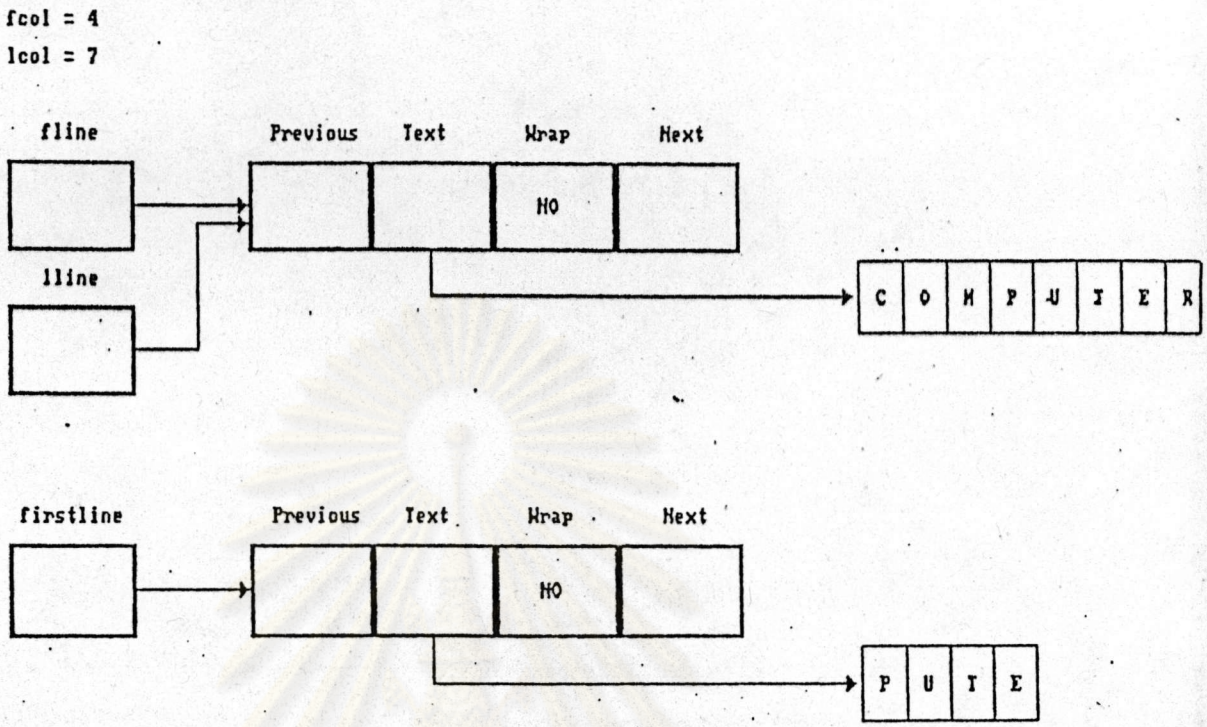
<u>การทำงาน</u> จะมีขั้นตอนต่างๆ ดังนี้คือ

1. กำหนดตัวแปรพอยน์เตอร์ lastline ให้ชี้ไปยังบรรทัดสุดท้ายของชุดข้อมูล sourceline โดยให้กำหนดค่าพอยน์เตอร์ previous ของ sourceline ให้กับตัวแปรพอยน์เตอร์ lastline
2. ตรวจสอบชุดของบรรทัด sourceline ว่ามีอยู่เพียง 1 บรรทัดหรือไม่ โดยตรวจสอบค่าของบรรทัดถัดไปของ sourceline (sourceline->next) กับค่าของ sourceline ว่าไม่เท่ากัน หรือไม่

   2.1 ถ้าไม่เท่ากัน แสดงว่า sourceline มีมากกว่า 1 บรรทัด จะทำงานตามขั้นตอนต่างๆดังนี้คือ

   - o ทำงานในโมดูล "insertreturn" เพื่อตัดข้อความของ destline ตั้งแต่ในคอลัมน์ destcol ให้ไปแทรกขึ้นบรรทัดใหม่ต่อจาก destline
   - o ทำการเชื่อมบรรทัด sourceline เข้ากับบรรทัด destline โดยจะมีขั้นตอนดังนี้คือ

SOURCE LINE EQUAL 1 LINE



รูปภาพที่ 4.58 แสดงการเก็บข้อมูลของบรรทัดต้นแบบที่มีเพียง 1 บรรทัด หลังการทำงานของโมดูล copytospace

- กำหนดให้ previous ของ destline บรรทัดถัดไป ชี้ไปที่ตำแหน่งเดียวกันกับ lastline
- กำหนดให้ บรรทัดถัดไปของ lastline ชี้ไปที่ตำแหน่งเดียวกันกับบรรทัดถัดไปของ destline
- กำหนดให้ บรรทัดถัดไปของ destline ชี้ไปที่ตำแหน่งเดียวกันกับ sourceline
- เชื่อมข้อความในบรรทัด sourceline เข้ากับ destline

o ลบข้อมูลในบรรทัด sourceline และ lastline

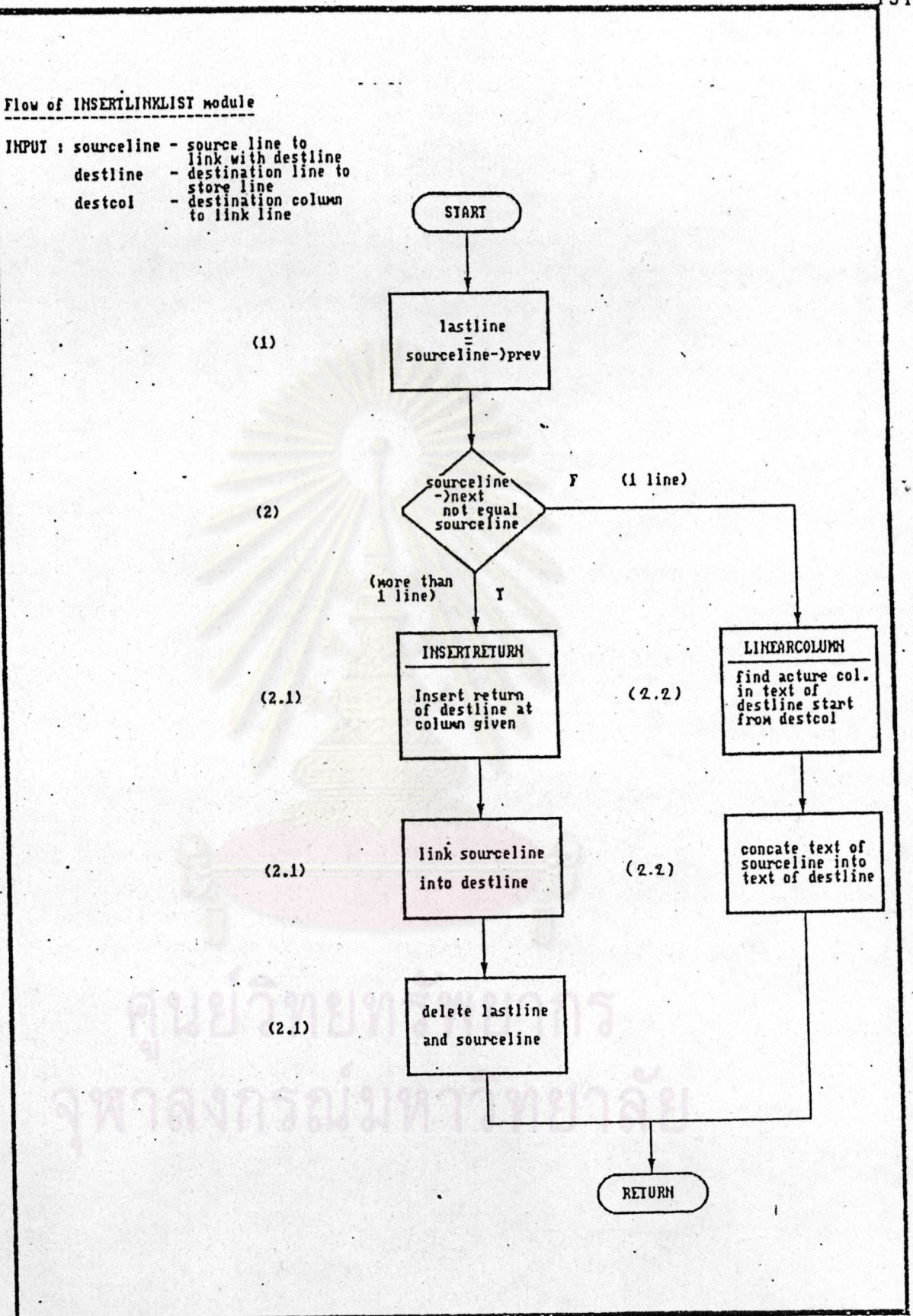
2.2 ถ้าเท่ากัน แสดงว่า sourceline มีเพียง 1 บรรทัด จะทำตามขั้นตอนดังนี้คือ

o ทำงานในโมดูล "linesrcolumn" เพื่อหาคอลัมน์ที่แท้จริงของข้อความใน destline จากค่าของคอลัมน์ของภาษาไทย destcol แล้วเก็บค่าไว้ในตัวแปร i

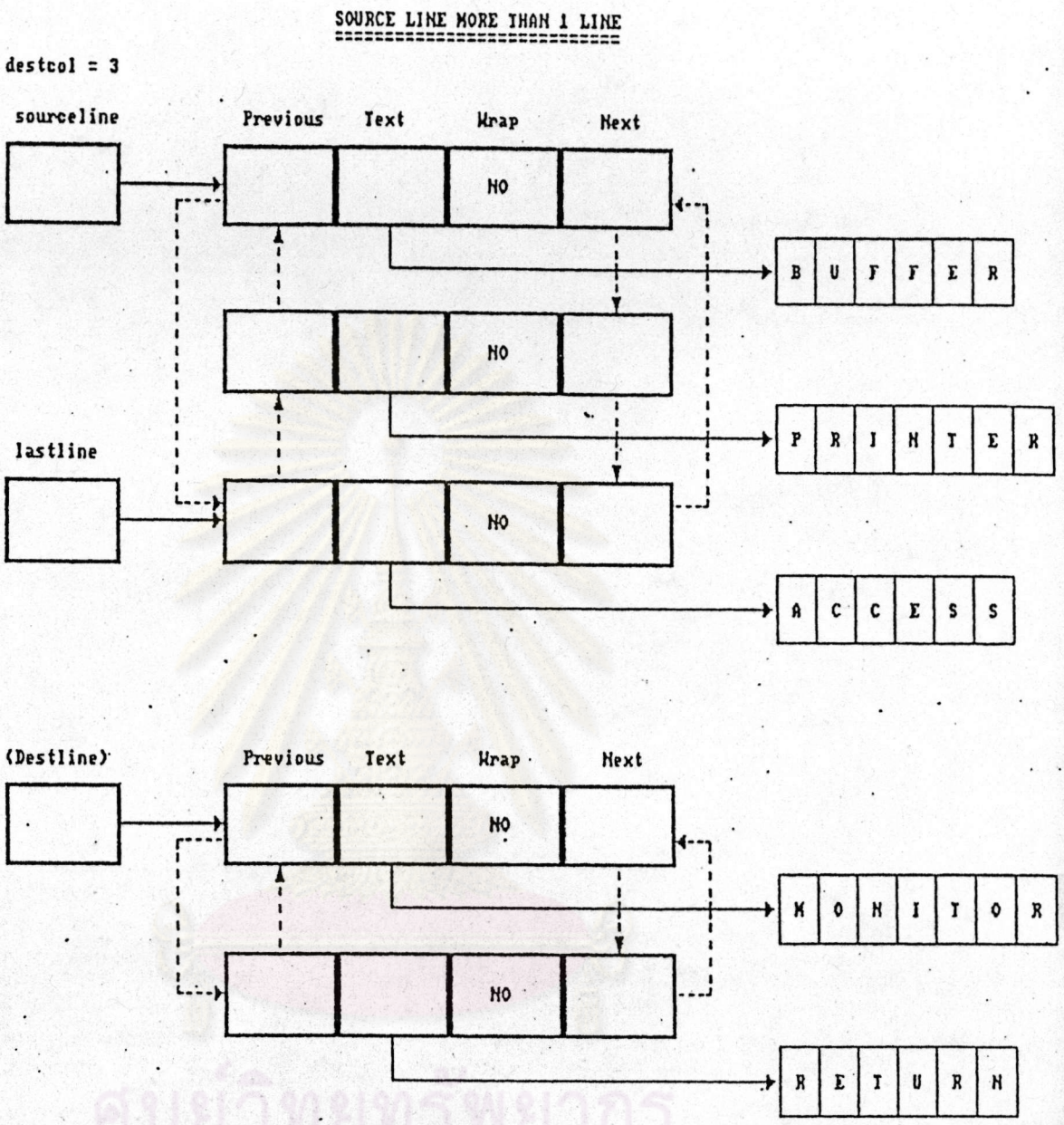
o คัดลอกข้อความจากบรรทัด sourceline มาแทรกไว้ตรงกลางข้อความของบรรทัด destline ในตำแหน่งที่ i

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.59 แล้ว โดยการทำงานเราจะแบ่งออกเป็น 2 กรณี คือกรณีบรรทัดที่จะนำมาเชื่อมต่อ มีข้อมูลมากกว่า 1 บรรทัด และกรณีบรรทัดที่จะนำมาเชื่อมต่อ มีข้อมูลเพียง 1 บรรทัด

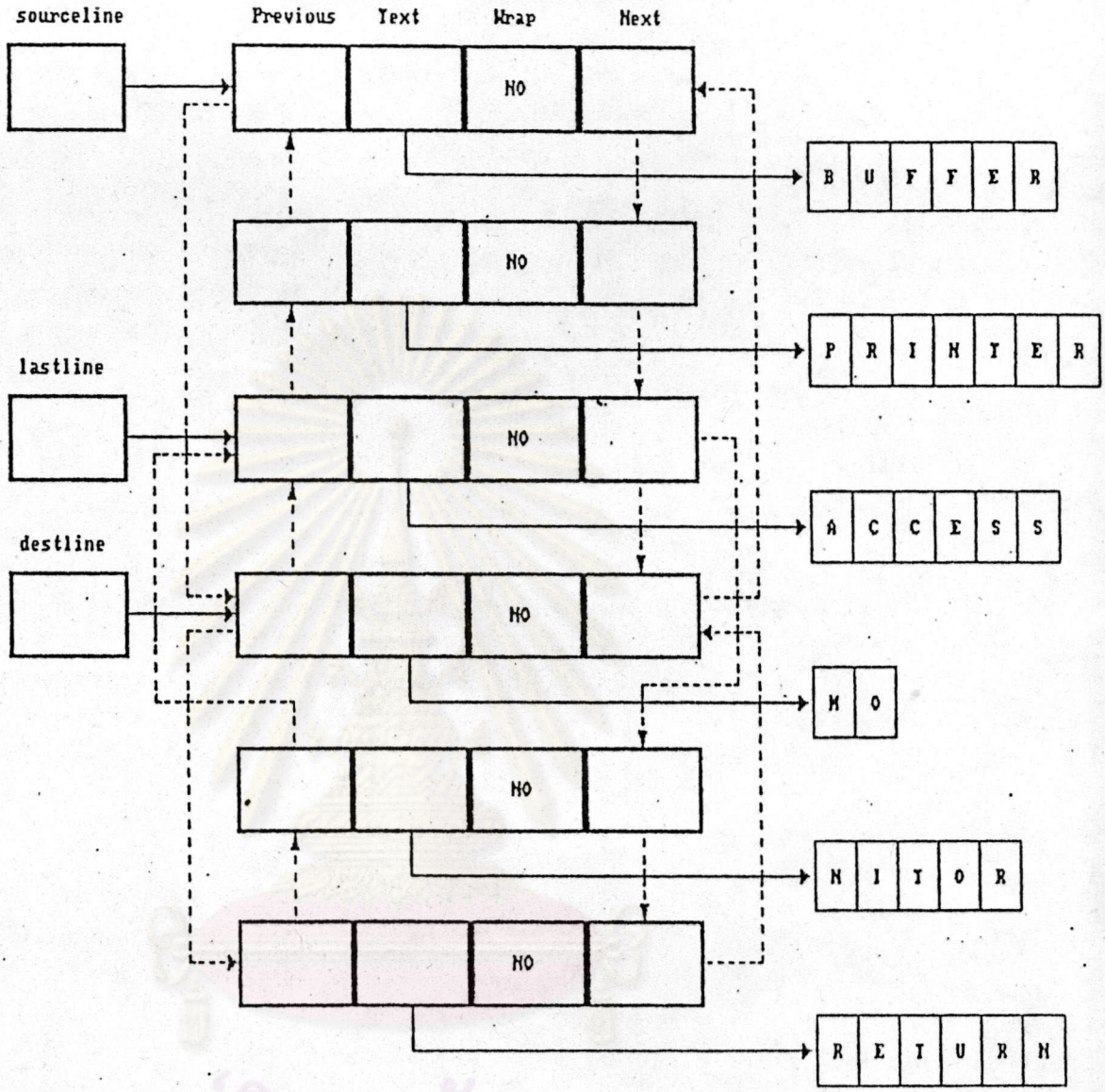
สำหรับในภาพที่ 4.60 แสดงการเก็บข้อมูลของบรรทัดที่นำมาเชื่อมต่อมีมากกว่า 1 บรรทัด ซึ่งจะถูกชี้ด้วยพอยน์เตอร์ sourceline และบรรทัดที่จะถูกเชื่อมต่อ ซึ่งจะชี้ด้วยพอยน์เตอร์ destline จากนั้นก็จะนำมาเชื่อมต่อบรรทัดเข้าด้วยกันโดยจะนำข้อมูลในบรรทัดที่ถูกชี้ด้วยพอยน์เตอร์ sourceline ไปเชื่อมต่อเข้าในบรรทัด ที่ถูกชี้ด้วยพอยน์เตอร์ destline ในตำแหน่งคอลัมน์ destcol ในที่นี้ได้กำหนดให้ destcol มีค่าเป็น 3 ตรงกับตัวอักษร N ของข้อความว่า MONITOR ซึ่งในรูปภาพที่ 4.61 จะแสดงให้เห็นการเชื่อมต่อของพอยน์เตอร์ทั้งสองเข้าด้วยกัน และให้ผลหลังการทำงานแสดง



รูปภาพที่ 4.59 ผังงานแสดงขั้นตอนการทำงานของโมดูล insertlinklist



รูปภาพที่ 4.60 แสดงการเก็บข้อมูลของบรรทัดที่จะนำมาเชื่อมต่อและบรรทัดที่จะถูกเชื่อมต่อกันก่อนการทำงานของโมดูล insertlinklist สำหรับกรณีบรรทัดที่จะมาเชื่อมต่อมีมากกว่า 1 บรรทัด



รูปภาพที่ 4.61 แสดงการเก็บเชื่อมต่อของบรรทัด ในระหว่างการทำงานในโมดูล insertlinklist สำหรับกรณีบรรทัดที่จะมาเชื่อมต่อมีมากกว่า 1 บรรทัด

ดังในรูปภาพที่ 4.62

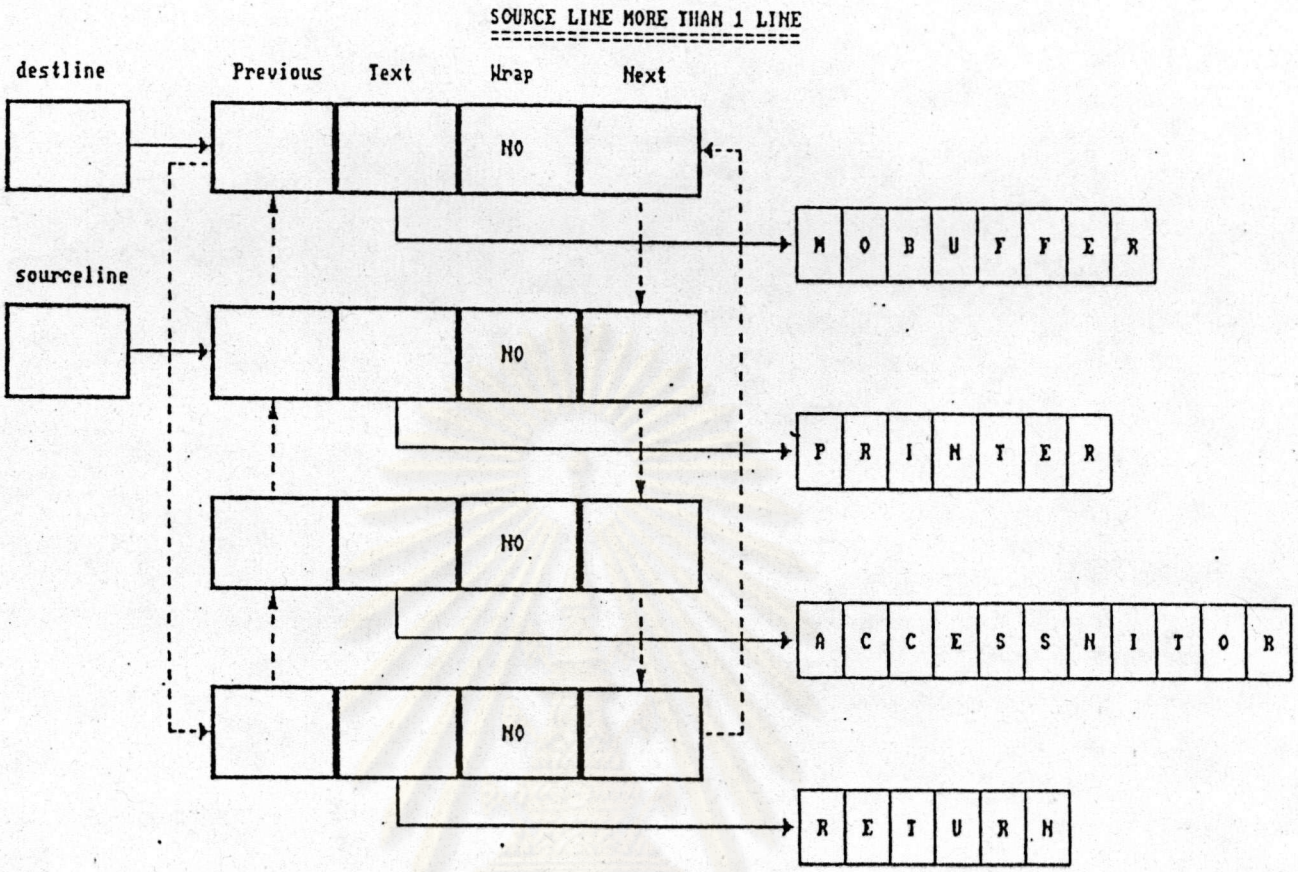
สำหรับในภาพที่ 4.63 แสดงการเก็บข้อมูลของบรรทัดที่จะนำมาเชื่อมต่อ ซึ่งมีเพียง 1 บรรทัด โดยคัดลอกข้อมูลจากบรรทัดถูกชี้ด้วยพอยน์เตอร์ sourceline ไปไว้ในบรรทัด ซึ่งจะถูกชี้ด้วยพอยน์เตอร์ destline ก่อนการทำงาน และผลจากการทำงานจะแสดงดังในภาพที่ 4.64

4.3.19.6 การลบชุดของบรรทัดข้อมูล (Delete Link List)

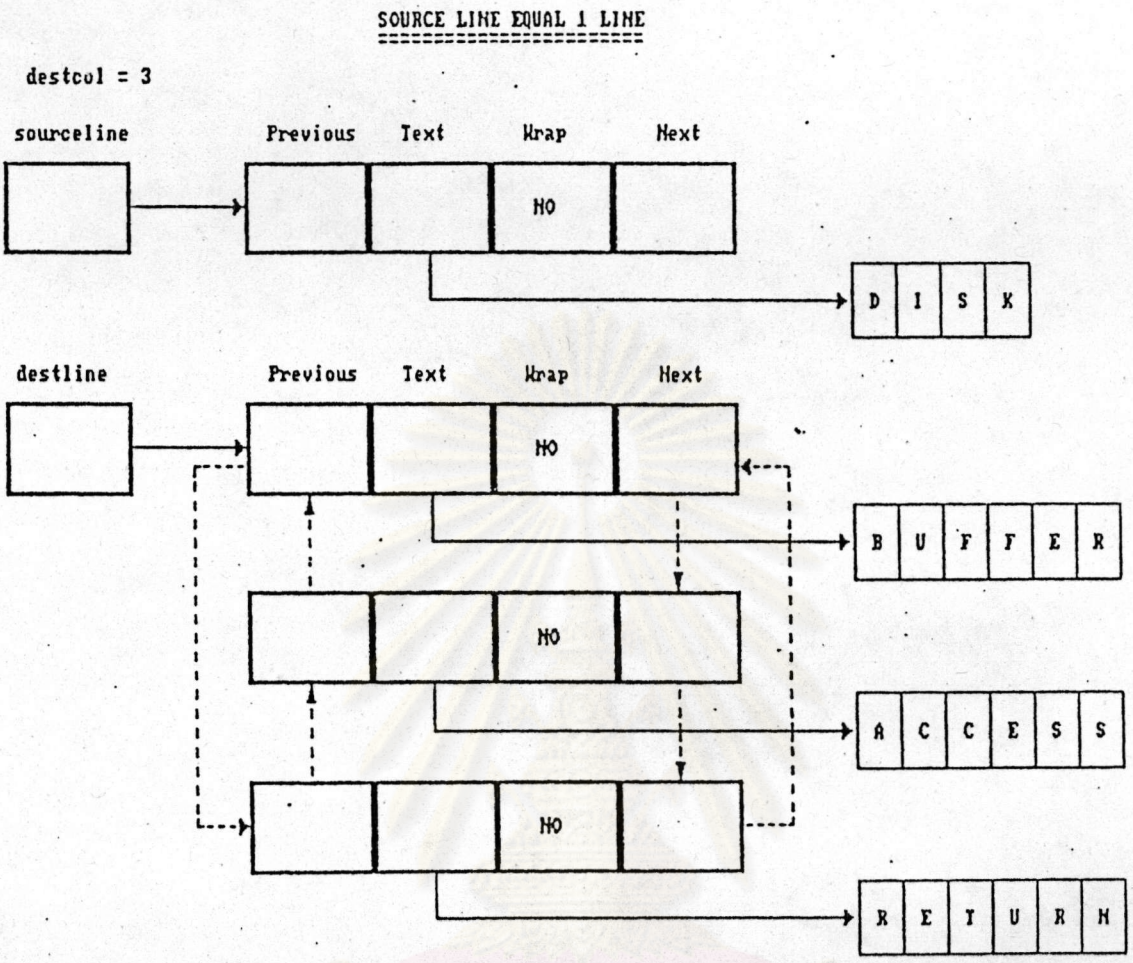
จะเป็นการลบชุดของบรรทัดข้อมูลที่ต้องการ จากชุดของบรรทัดข้อมูลทั้งหมดที่ใช้งานอยู่ขณะนั้น การทำงานนี้จะอยู่ภายใต้โมดูล deletelinklist ซึ่งจะต้องกำหนดพอยน์เตอร์ fline และ lline ที่จะชี้ไปยังบรรทัดเริ่มต้นและสุดท้ายที่จะลบ รวมทั้งกำหนดค่า fcol และ lcol ที่เป็นค่าคอลัมน์เริ่มต้นและคอลัมน์สุดท้าย

<u>การทำงาน</u> จะมีขั้นตอนต่างๆ ดังนี้คือ

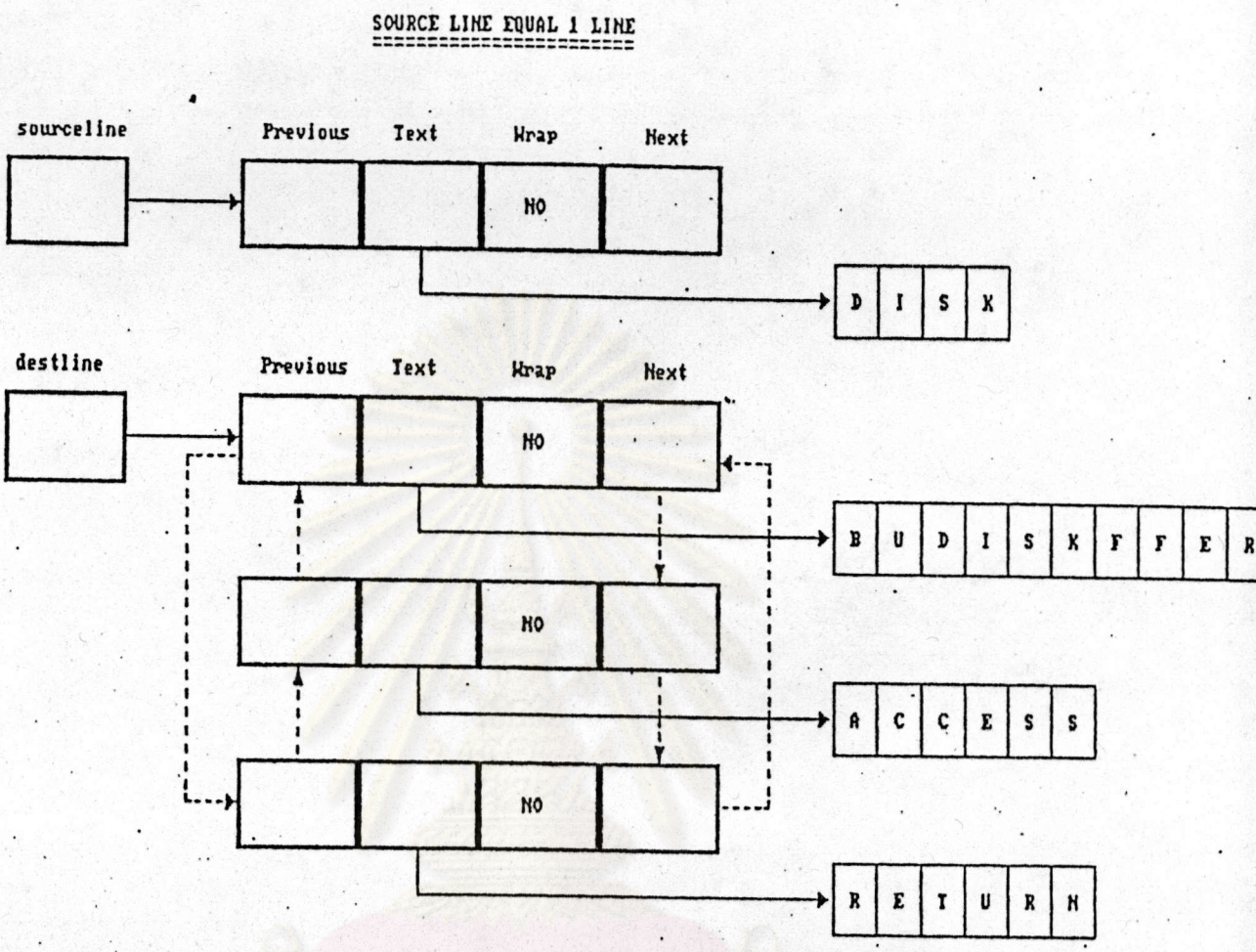
1. ทำงานในโมดูล "insertreturn" เพื่อตัดข้อความของ lline ตั้งแต่ในคอลัมน์ lcol ให้ไปแทรกขึ้นบรรทัดใหม่ต่อจาก lline
2. เปลี่ยนค่าตัวแปร lline ให้ชี้ไปยังบรรทัดถัดไป
3. ทำงานในโมดูล "insertreturn" เพื่อตัดข้อความของ fline ตั้งแต่ในคอลัมน์ fcol ให้ไปแทรกขึ้นบรรทัดใหม่ต่อจาก fline
4. กำหนดตัวแปร templine ให้ชี้ไปยังบรรทัดถัดไปจาก fline
5. กำหนดตัวแปรพอยน์เตอร์ next ของ fline ให้ชี้ไปยังบรรทัด lline
6. กำหนดตัวแปรพอยน์เตอร์ previous ของ lline ให้ชี้ไปยังบรรทัด fline
7. ทำงานซ้ำๆกัน จนกว่าค่าของพอยน์เตอร์ templine เท่ากับ lline โดยขณะที่ทั้งสองยังไม่เท่ากันก็จะทำงานตามขั้นตอนดังนี้คือ
   - กำหนดตัวแปรพอยน์เตอร์ freeline ให้ชี้ไปยังบรรทัดเดียวกันกับพอยน์เตอร์ templine ที่ชี้อยู่ในบรรทัดนั้น
   - ยกเลิกข้อมูลในบรรทัดที่พอยน์เตอร์ freeline ชี้อยู่
   - กำหนดตัวแปร templine ให้ชี้ไปยังบรรทัดถัดไปจากบรรทัดเดิมนั้น



รูปภาพที่ 4.62 แสดงบรรทัดที่เชื่อมต่อแล้ว หลังการทำงานของโมดูล insertlinklist สำหรับกรณีบรรทัดที่จะมาเชื่อมต่อมีมากกว่า 1 บรรทัด



รูปภาพที่ 4.63 แสดงการเก็บข้อมูลของบรรทัดที่จะนำมาเชื่อมต่อและบรรทัดที่จะถูกเชื่อมต่อกัน ก่อนการทำงานของโมดูล insertlinklist สำหรับกรณีบรรทัดที่จะมาเชื่อมต่อมีเพียง 1 บรรทัด



รูปภาพที่ 4.64 แสดงบรรทัดที่เชื่อมต่อแล้ว หลังการทำงานของโมดูล insertlinklist สำหรับกรณีบรรทัดที่จะมาเชื่อมต่อมีเพียง 1 บรรทัด

8. ทำงานในโมดูล "deletereturn" เพื่อที่จะเชื่อมข้อความในบรรทัดที่ lline ชี้อยู่ ให้เข้ากับข้อความในบรรทัดที่ fline ชี้อยู่

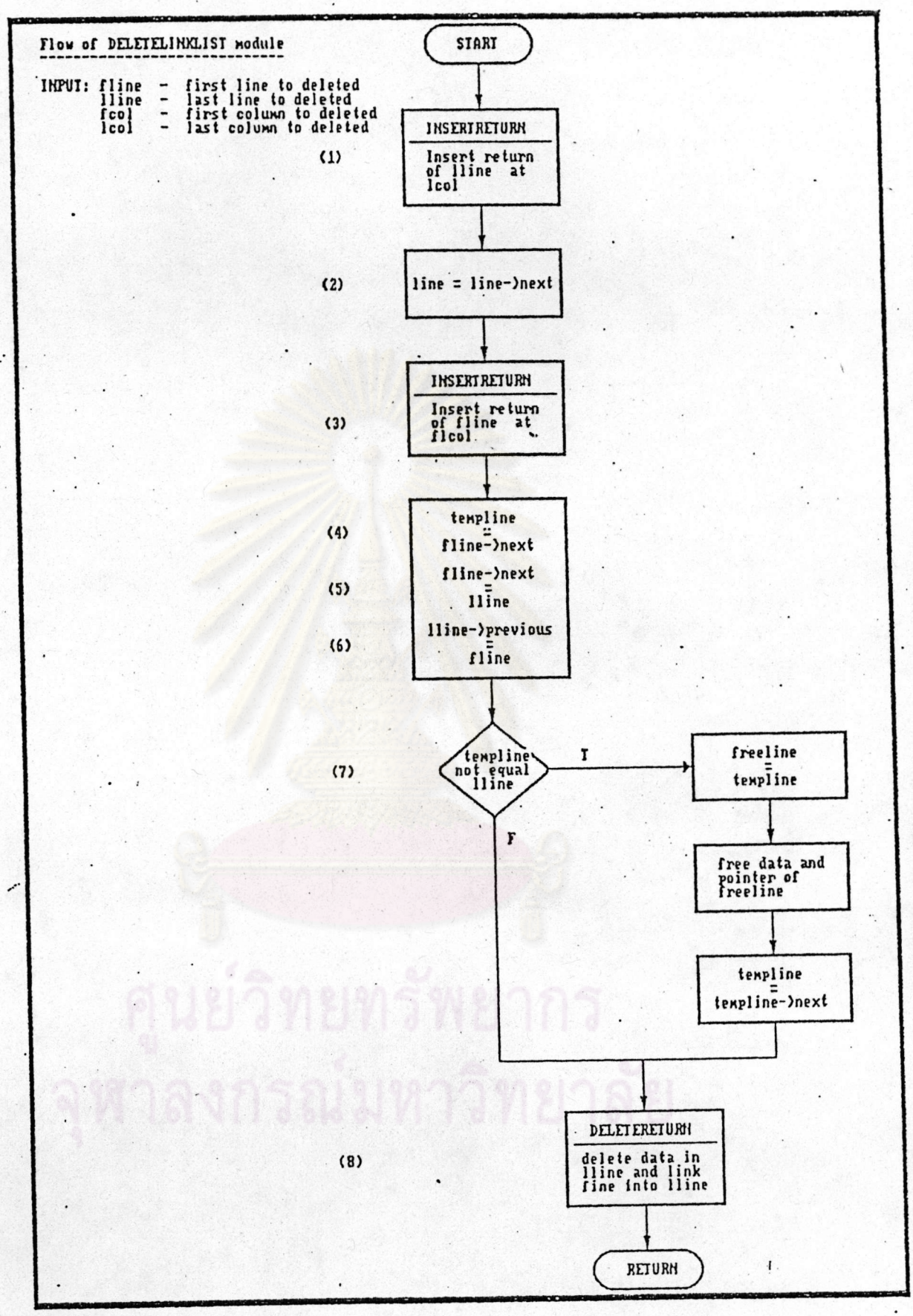
สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.65 แล้ว โดยจะแสดงการเก็บข้อมูลก่อนการทำงานของแต่ละบรรทัด ในรูปภาพที่ 4.66 พอยน์เตอร์ fline และ lline ชี้ไปยังบรรทัดแรกและบรรทัดสุดท้ายที่จะต้องการลบ รวมทั้งกำหนดตำแหน่ง fcol และ lcol เพื่อเป็นตำแหน่งคอลัมน์เริ่มต้นและสุดท้ายที่จะต้องการลบ (ในที่นี้กำหนดให้ fcol เป็น 5 และ lcol เป็น 8) โดยจะมีการตัดข้อความในบรรทัดที่พอยน์เตอร์ fline ชี้อยู่ ตั้งแต่ในคอลัมน์ 5 และตัดข้อความในบรรทัดที่พอยน์เตอร์ lline ชี้อยู่ ตั้งแต่ในคอลัมน์ 8 ให้ขึ้นบรรทัดใหม่ รวมทั้งกำหนดพอยน์เตอร์ templine ให้ชี้ไปยังบรรทัดถัดไป ต่อจากบรรทัดที่พอยน์เตอร์ fline ชี้อยู่ เพื่อกำหนดบรรทัดเริ่มต้นที่จะลบบรรทัด ดังแสดงในรูปภาพที่ 4.67 และจะทำการลบชุดบรรทัดข้อมูล ตั้งแต่บรรทัด templine ชี้อยู่จนถึงบรรทัดที่ lline แล้วจะนำข้อความในบรรทัด lline ซึ่งเก็บข้อความ "ING" นั้นไปเชื่อมต่อกับบรรทัด fline ซึ่งเก็บข้อความ "DISK" กลายเป็นข้อความ "DISKING" จะให้ผลหลังการทำงานเป็นดังรูปภาพที่ 4.68

4.3.19.7 การอ่านข้อมูลจากไฟล์ (Read file data)

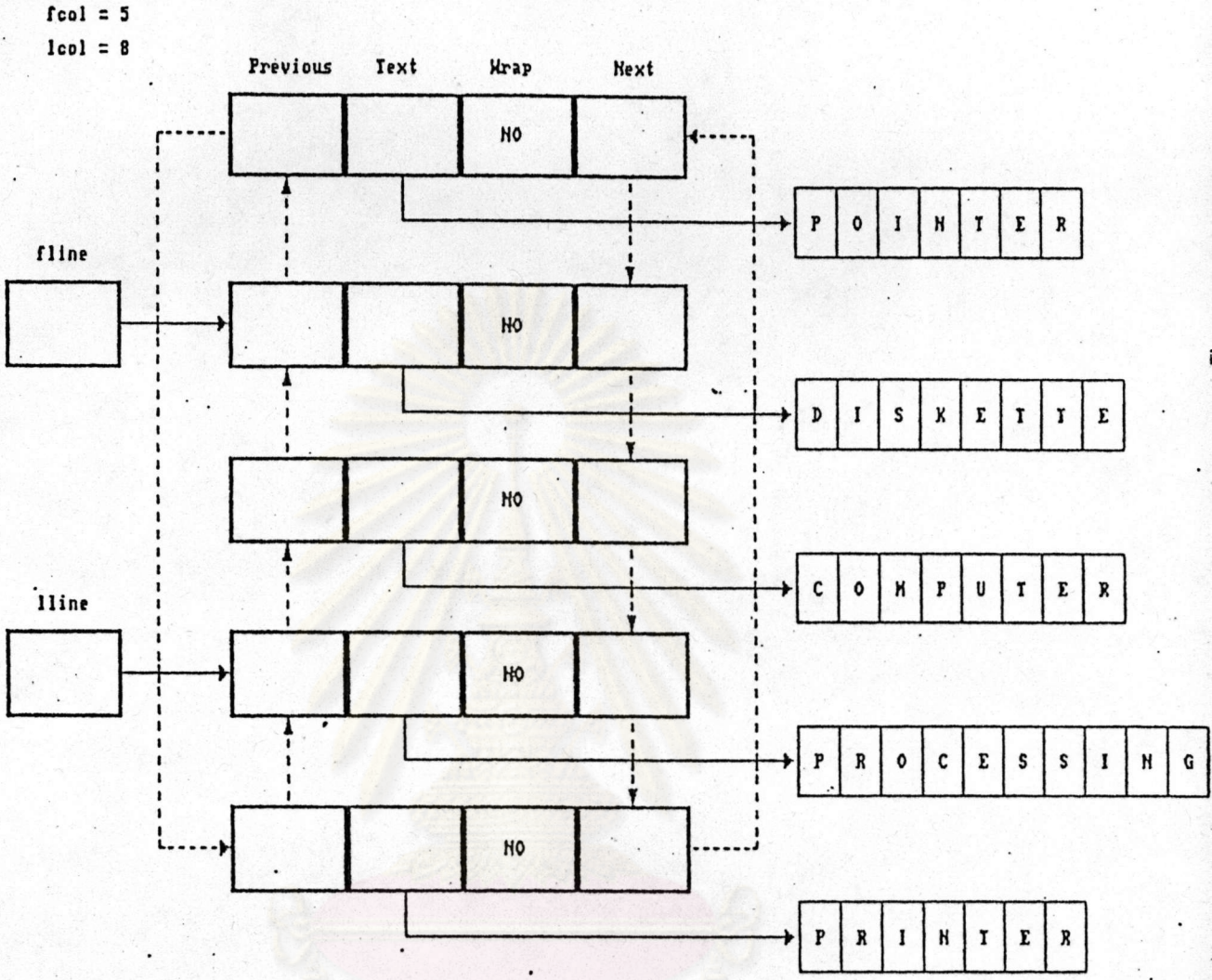
จะเป็นการอ่านข้อมูลจากไฟล์ที่ได้ระบุชื่อไว้ มาเก็บไว้ในโหนดบัฟเฟอร์ที่ได้เตรียมไว้ ซึ่งเป็นการทำงานภายใต้โมดูล rdfiletospace โดยจะต้องกำหนดชื่อไฟล์ที่จะอ่านไว้ใน file_name และจะถูกนำไปใช้งานในโมดูล readblk เพื่ออ่านข้อมูลจากไฟล์ มาเชื่อมต่อกับบรรทัดที่ใช้งานอยู่

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

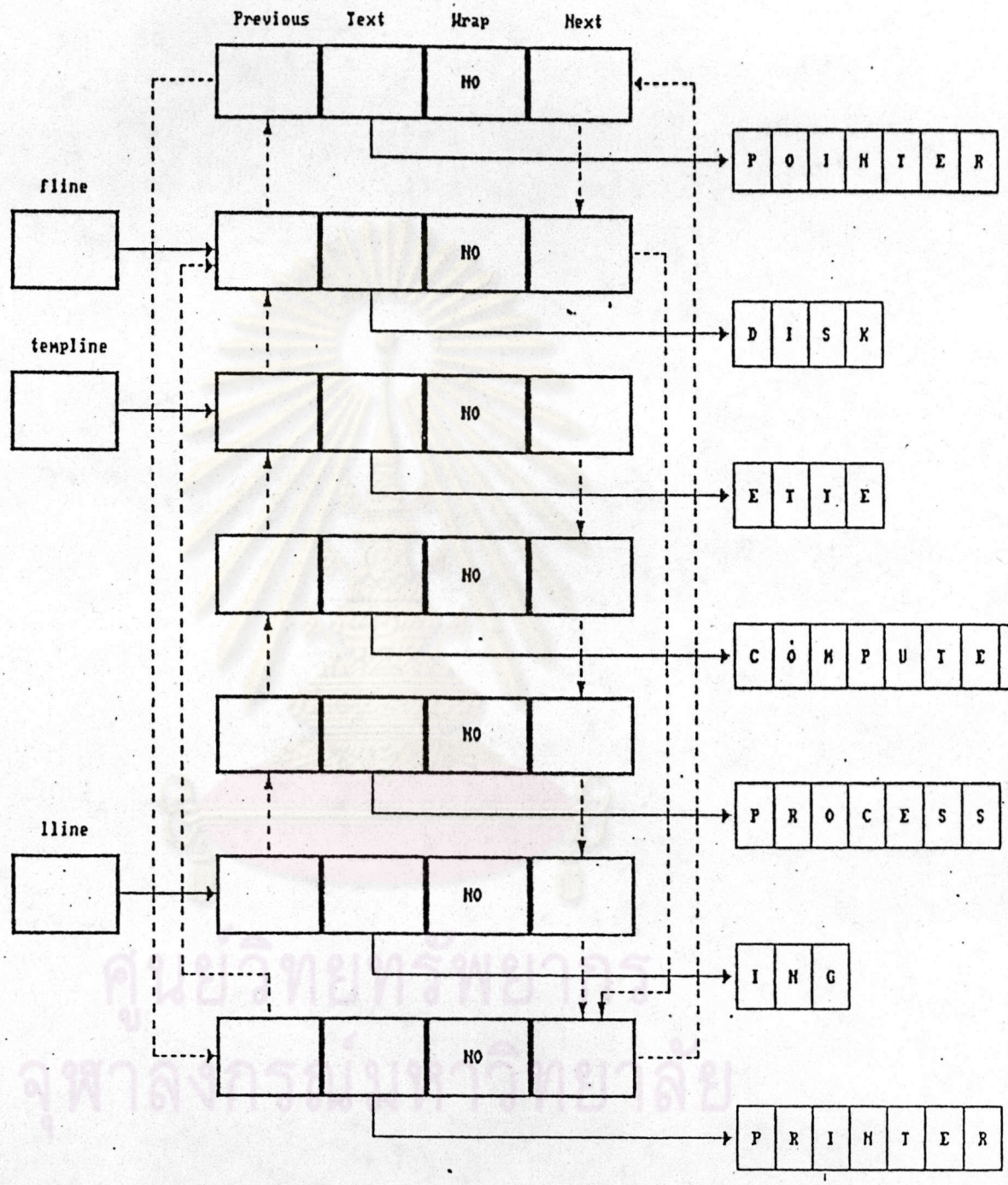
1. เปิดใช้ไฟล์ข้อมูลที่ระบุชื่อไว้ใน file_name
2. อ่านข้อมูลเรคอร์ดแรกมาไว้ในโหนดบัฟเฟอร์ที่กำหนดไว้ โดยข้อมูลแต่ละเรคอร์ดจะต้องถูกคั้นด้วยตัวอักษร NULL (\n)
3. สร้างตัวโหนดพอยน์เตอร์ newline ให้ชี้ไปยังโหนดบัฟเฟอร์ที่เก็บข้อมูลไว้นั้น



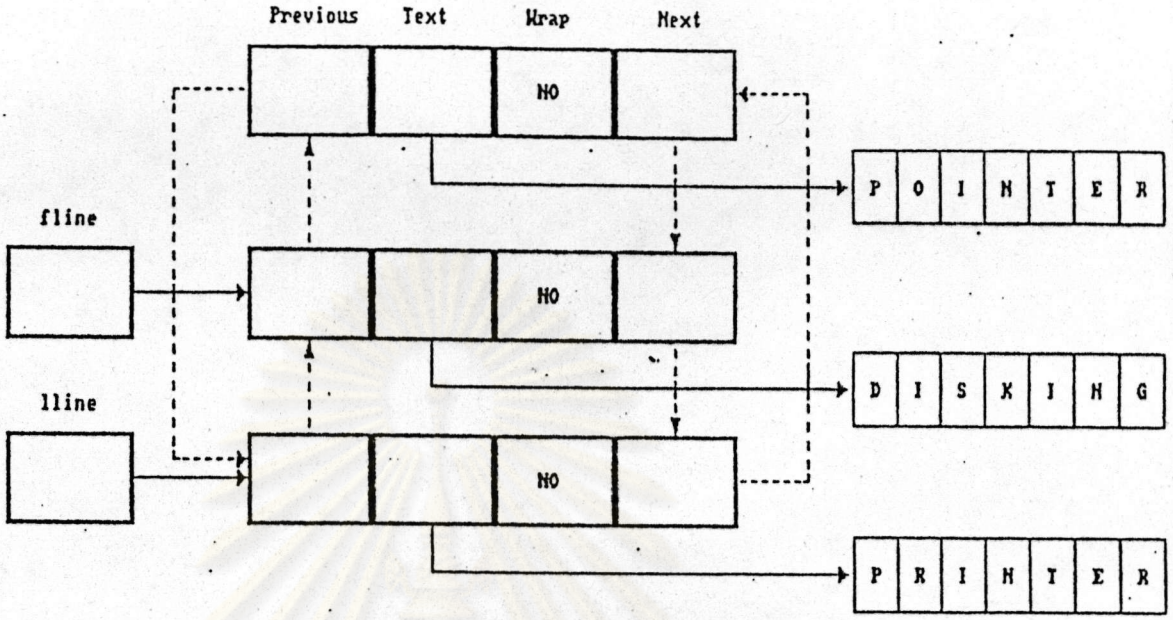
รูปภาพที่ 4.65 ผังงานแสดงขั้นตอนการทำงานของโมดูล deletelinklist



รูปภาพที่ 4.66 แสดงการเก็บข้อมูลของแต่ละบรรทัด ก่อนการทำงานของโมดูล deletelinklist



รูปภาพที่ 4.67 แสดงการเก็บข้อมูลของแต่ละบรรทัด ในระหว่างการทำงานของโมดูล deletelinklist



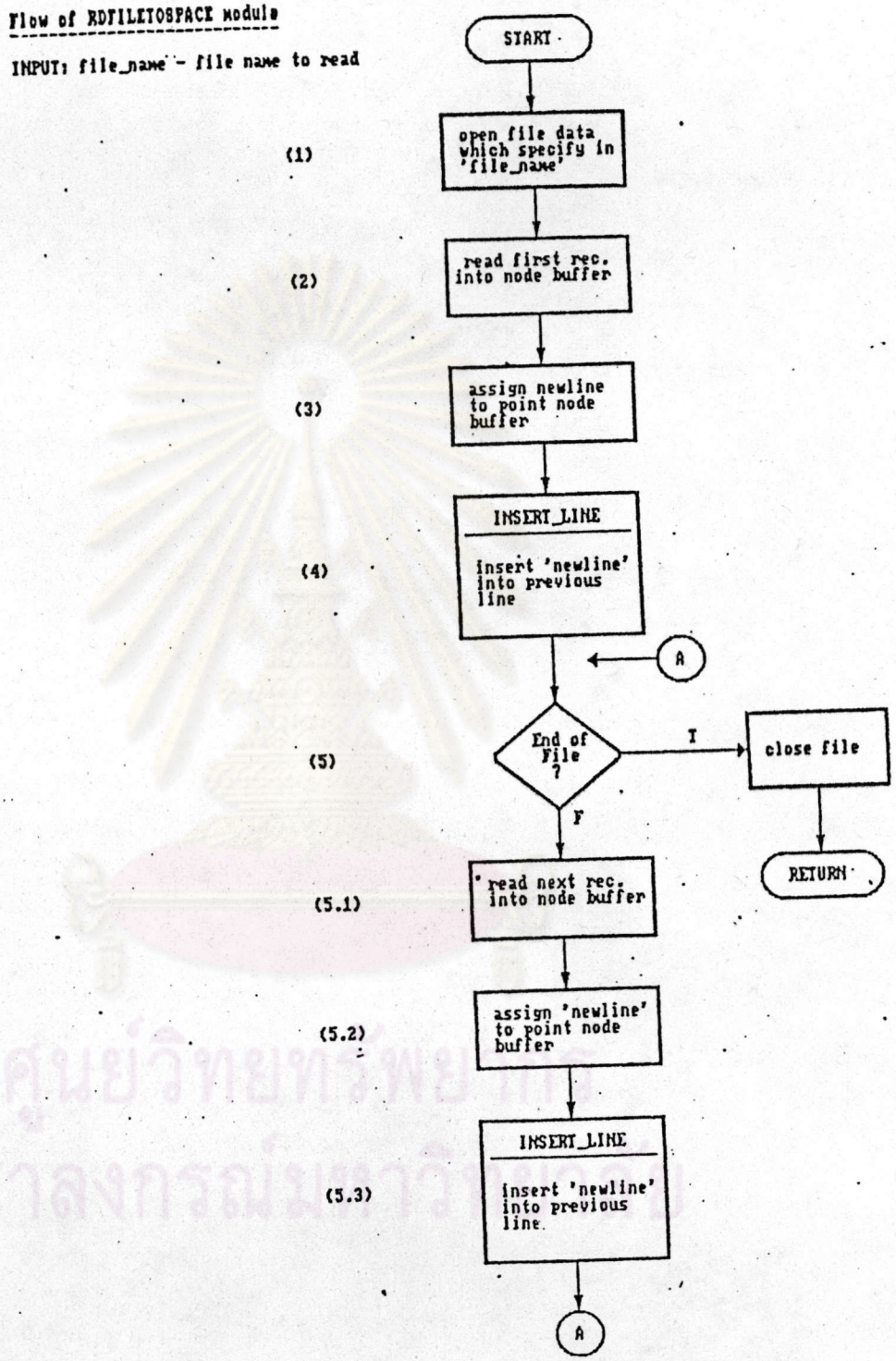
รูปภาพที่ 4.68 แสดงการเก็บข้อมูลของแต่ละบรรทัด หลังการทำงานของโมดูล deletelinklist

4. ทำงานในโมดูล "insert_line" เพื่อนำบรรทัดที่ถูกชี้ด้วย newline ไปเชื่อมต่อกับบรรทัดก่อนหน้าซึ่งถูกชี้ด้วย sentinal line
5. ตรวจสอบว่า ขณะนั้นเรคอร์ดข้อมูลหมดไฟล์ หรือไม่
   - ถ้าหมดไฟล์ข้อมูลแล้ว จะปิดไฟล์ข้อมูลนั้น และเลิกทำงาน
   - ถ้ายังไม่หมดไฟล์ข้อมูล จะให้ทำงานตามขั้นตอนดังนี้ คือ
     5.1 อ่านข้อมูลเรคอร์ดถัดไป มาเก็บในโหนดบัฟเฟอร์ที่กำหนดไว้
     5.2 สร้างตัวโหนดพอยน์เตอร์ newline ให้ชี้ไปยังโหนดบัฟเฟอร์ที่เก็บข้อมูลไว้นั้น
     5.3 ทำงานในโมดูล "insert_line" เพื่อนำบรรทัดที่ถูกชี้ด้วย newline ไปเชื่อมต่อกับบรรทัดก่อนหน้า
     5.4 กลับไปที่หัวข้อ 5 อีกครั้ง

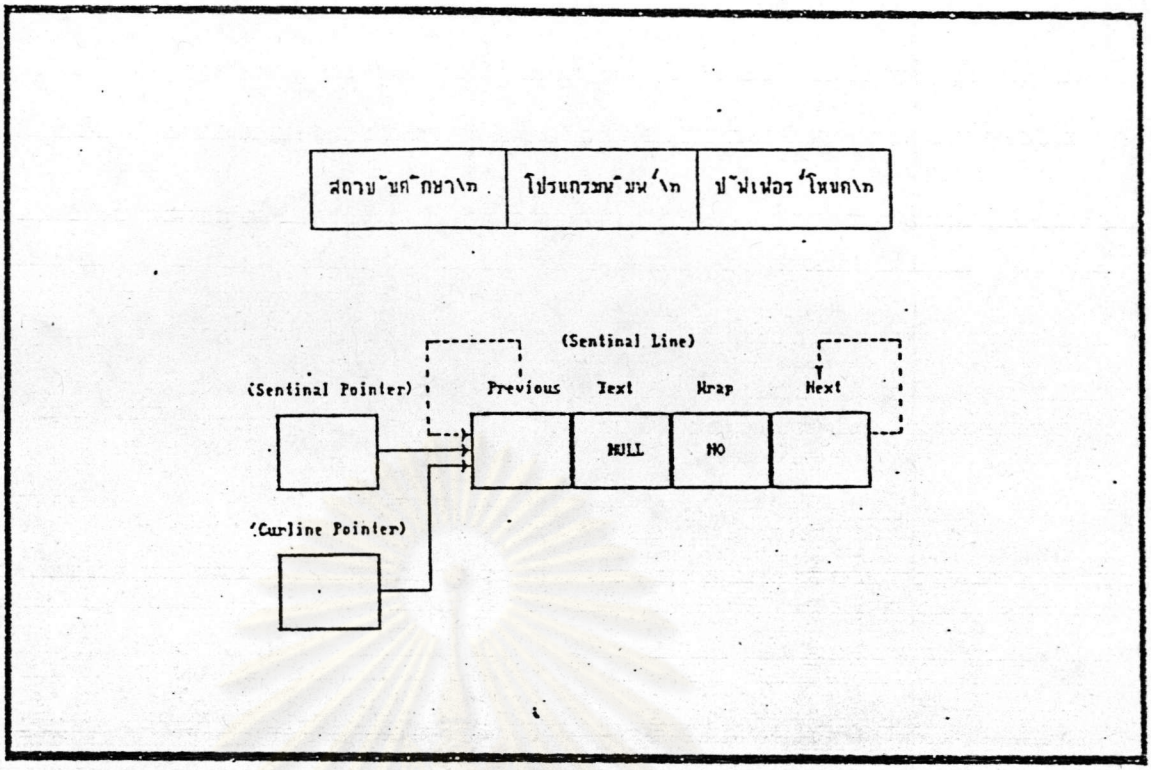
สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.69 แล้ว โดยสมมุติว่าขณะนั้นเก็บข้อมูลในเรคอร์ดเป็นจำนวน 3 เรคอร์ด และสร้างตัวโหนดพอยน์เตอร์ใหม่ ที่ถูกชี้ด้วยพอยน์เตอร์ sentinal และ curline เพื่อเตรียมอ่านข้อมูลจากเรคอร์ด ดังรูปในภาพที่ 4.70 และหลังจากนั้นจะทำการอ่านข้อมูลในเรคอร์ด มาเก็บไว้ในโหนดบัฟเฟอร์และนำโหนดพอยน์เตอร์มาเชื่อมต่อกันได้ ซึ่งจะได้ผลการทำงานดังรูปภาพที่ 4.71

4.3.19.8 การเก็บบรรทัดข้อมูลไว้ในไฟล์ (Write file data)

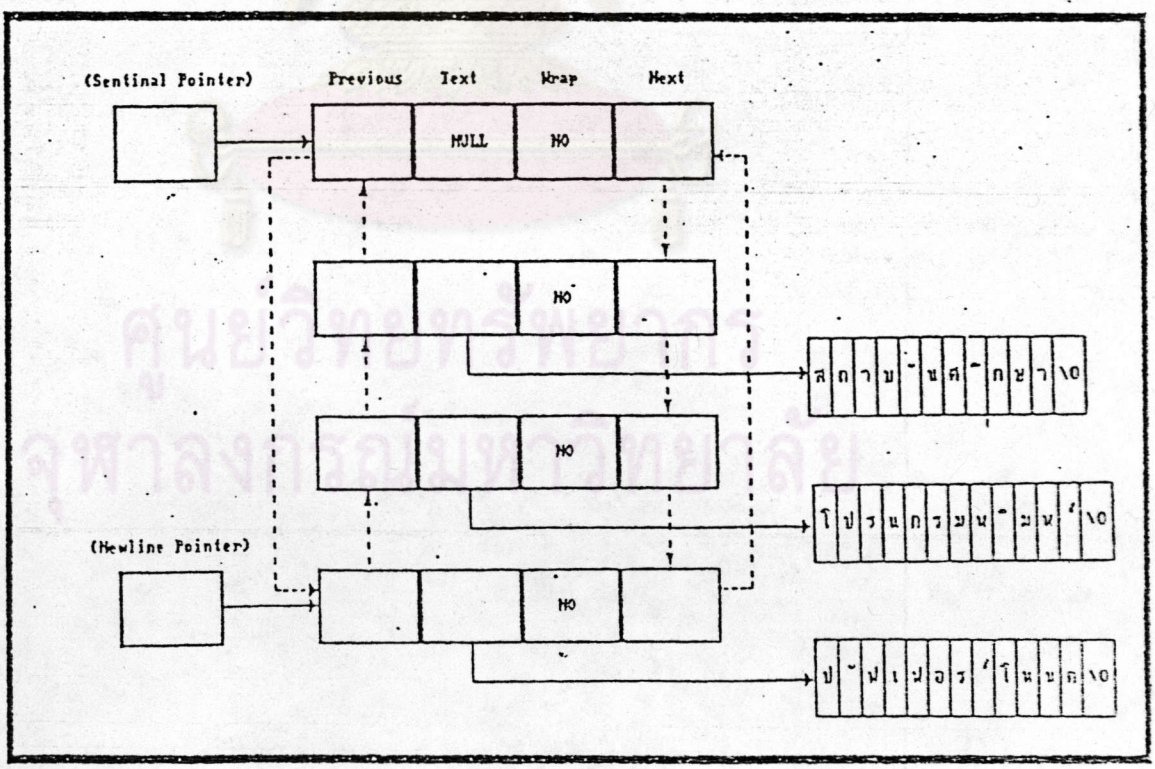
จะเป็นการนำข้อมูลจากโหนดบัฟเฟอร์มาเก็บไว้อย่างถาวรในไฟล์ข้อมูล (จะทำงานตรงกันข้ามกับการอ่านข้อมูลจากไฟล์) ซึ่งจะอยู่ภายใต้การทำงานของโมดูล writeblk โดยจะต้องกำหนดชื่อไฟล์ที่จะเก็บข้อมูล (file_name) ตำแหน่งบรรทัดเริ่มต้นและสุดท้ายของข้อมูล (linebegin และ lineend) และ ตำแหน่งคอลัมน์เริ่มต้นและสุดท้ายของข้อมูล (colbegin และ colend) และจะถูกนำไปใช้งานในโมดูล writeblock เพื่อเขียนข้อมูลในบรรทัดมาลงไฟล์



รูปภาพที่ 4.69 ผังงานแสดงขั้นตอนการทำงานของโมดูล rdfiletospace



รูปภาพที่ 4.70 แสดงข้อมูลที่เก็บไว้ในไฟล์ที่จะอ่านไปไว้ในโหนดบัฟเฟอร์ ก่อนการทำงานในโมดูล rdfiletospace



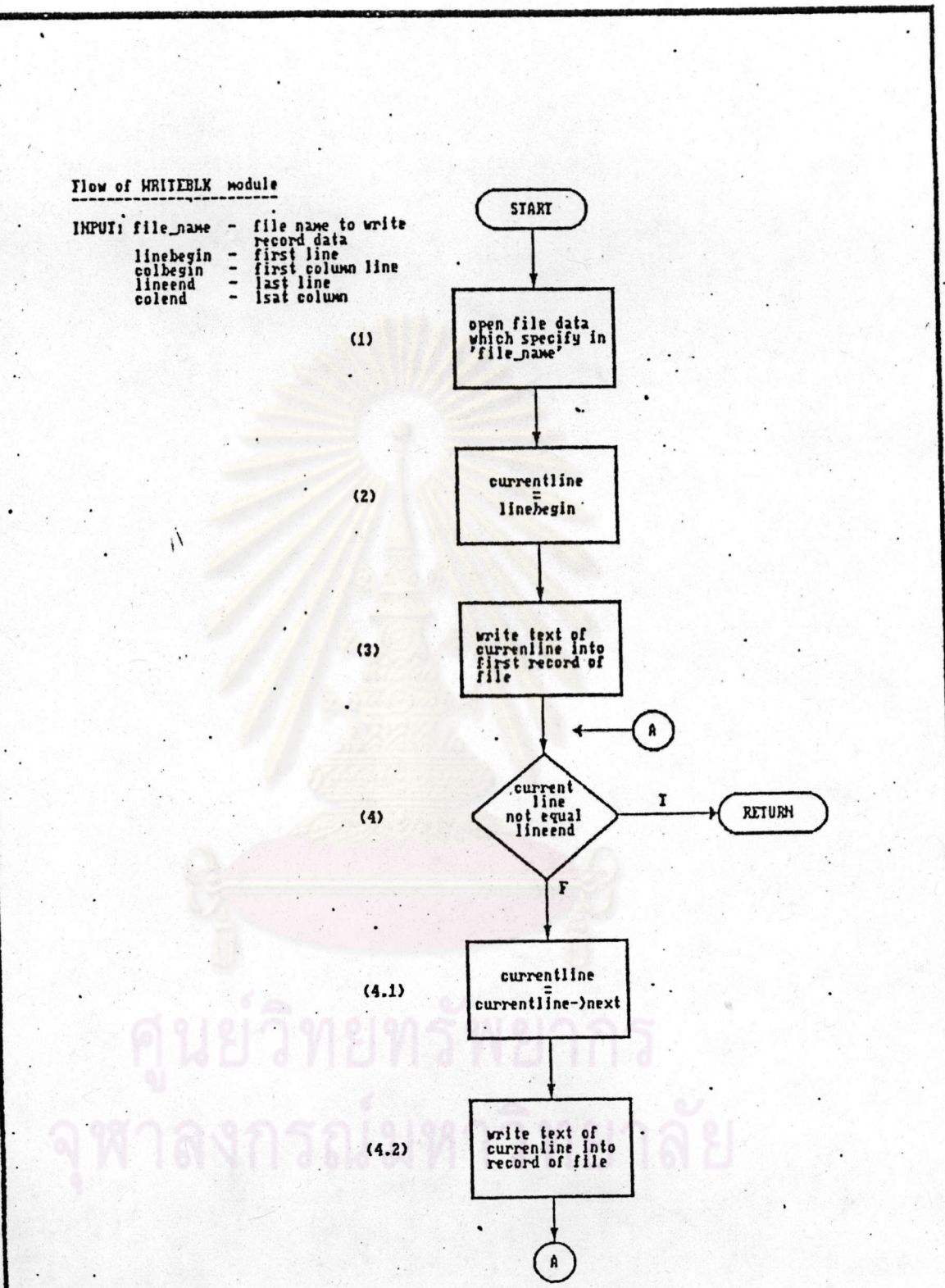
รูปภาพที่ 4.71 แสดงข้อมูลที่เก็บไว้ในโหนดบัฟเฟอร์ หลังการทำงานในโมดูล rdfiletospace

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

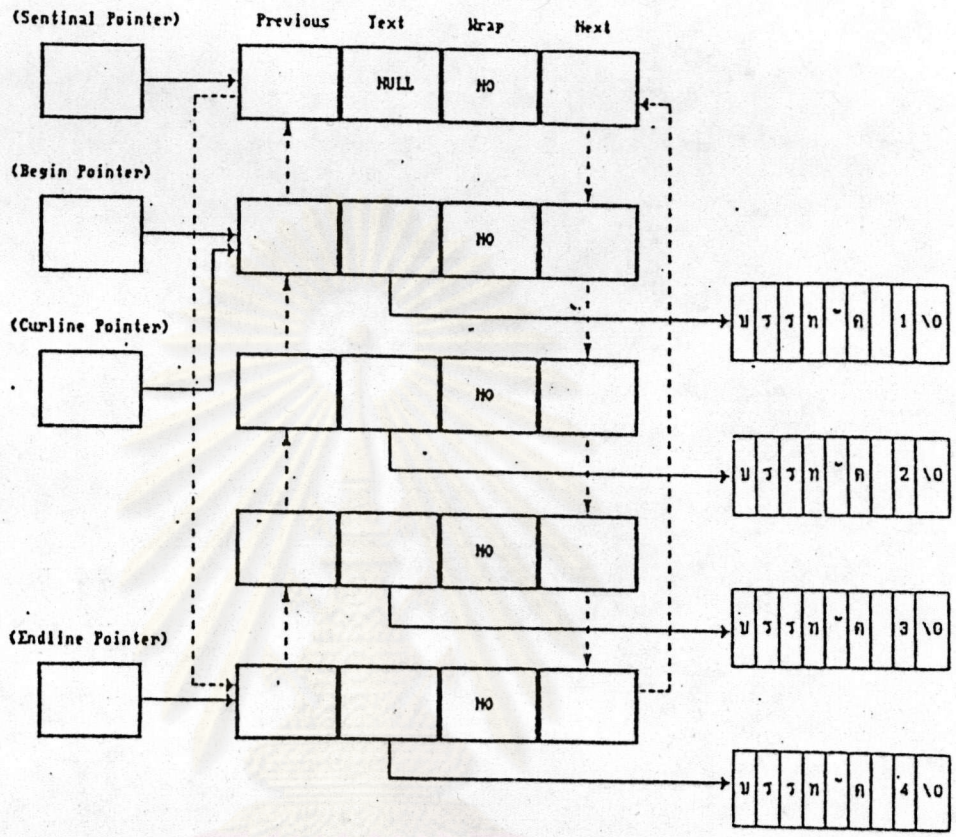
1. เตรียมเปิดไฟล์ข้อมูลตามชื่อที่ได้ระบุไว้ใน file_name
2. กำหนดโหนดพอยน์เตอร์ currentline ให้เท่ากับ linebegin เพื่อให้ชี้ไปยังบรรทัดเริ่มต้น ซึ่งเป็นบรรทัดเดียวกันกับที่โหนดพอยน์เตอร์ linebegin ชี้อยู่
3. จะนำข้อความตัวอักษรในโหนดบัฟเฟอร์ที่ถูกชี้ด้วยโหนดบัฟเฟอร์ linebegin ตั้งแต่ในคอลัมน์ colbegin ไปเก็บไว้ในไฟล์
4. ตรวจสอบค่าโหนดบัฟเฟอร์ currentline กับ lineend
   - ถ้าเท่ากัน แสดงว่า ทำงานจนถึงบรรทัดสุดท้ายแล้วจะปิดไฟล์ข้อมูลและเลิกทำงาน
   - ถ้าไม่เท่ากัน แสดงว่า ทำงานยังไม่ถึงบรรทัดสุดท้าย จะให้ทำขั้นตอนดังนี้คือ
     - 4.1 เปลี่ยนโหนดพอยน์เตอร์ currentline ให้ชี้ไปยังบรรทัดถัดไป
     - 4.2 นำข้อความในบรรทัดที่ currentline ชี้อยู่ไปเก็บเรคอร์ดไว้ในไฟล์
     - 4.3 กลับไปตรวจสอบเงื่อนไขในหัวข้อ 4 อีกครั้ง

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.72 แล้ว สมมุติว่าขณะนั้นมีข้อมูลในโหนดบัฟเฟอร์ของแต่ละบรรทัดที่ถูกชี้ด้วยพอยน์เตอร์ linebegin และ currentline ไปยังโหนดบัฟเฟอร์ตัวที่หนึ่ง และ lineend ชี้ไปยังโหนดบัฟเฟอร์ตัวที่สี่ และกำหนดให้ใช้ชื่อไฟล์ข้อมูลว่า LINE ดังแสดงไว้ในรูปภาพที่ 4.73 หลังจากนั้นจะทำการเก็บข้อมูลบางส่วนจากโหนดบัฟเฟอร์ตัวที่หนึ่งถึงตัวที่สาม ไปเก็บไว้ในไฟล์ที่ชื่อว่า LINE เป็นจำนวน 3 เรคอร์ด ซึ่งจะได้ผลดังรูปภาพที่ 4.74

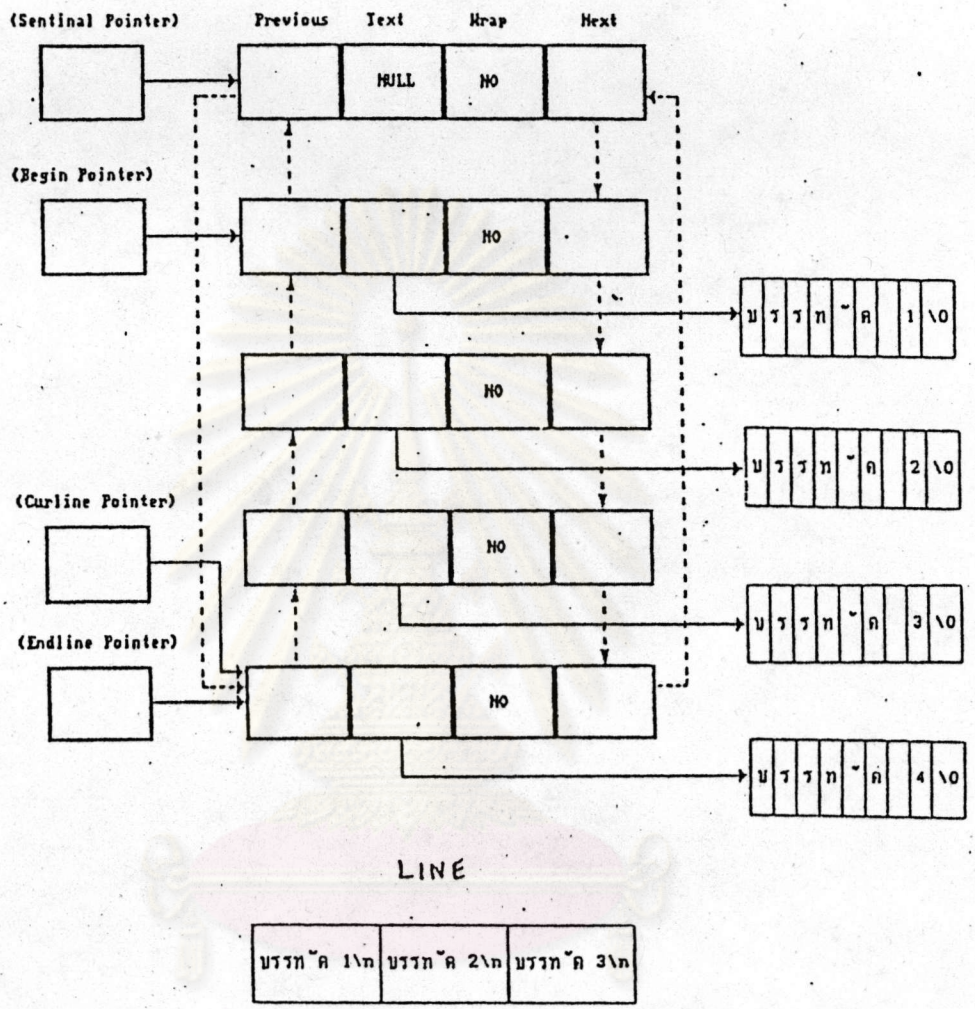
สำหรับในกรณีที่ต้องการเก็บข้อมูลทั้งหมดเราจะให้ linebegin ชี้ไปที่โหนดบัฟเฟอร์บรรทัดแรกและให้ lineend ชี้ไปที่โหนดบัฟเฟอร์ตัวท้ายสุด



รูปภาพที่ 4.72 ผังงานแสดงขั้นตอนการทำงานของโมดูล writeblk



รูปภาพที่ 4.73 แสดงการเก็บข้อมูลของแต่ละบรรทัด ซึ่งจะถูกชี้ด้วย linebegin และ lineend เพื่อจะไปเขียนข้อมูลไว้ในไฟล์ข้อมูลชื่อว่า LINE



รูปภาพที่ 4.74 แสดงผลการเขียนข้อมูลลงไปเก็บไว้ในไฟล์ข้อมูลชื่อว่า LINE หลังจากการทำงานในโมดูล writeblk

4.3.19.9 การคัดลอกบล็อก (Copy Block)

จะเป็นการคัดลอกข้อมูลในบล็อกที่กำหนดไว้ไปยังตำแหน่งอื่นๆของบรรทัดขณะนั้นได้ ซึ่งจะเป็นการทำงานภายใต้โมดูล copyblk โดยจะต้องกำหนดตำแหน่งคอลัมน์ x ที่จะคัดลอกไป

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. ตรวจสอบว่าขณะนั้นได้มีการกำหนดบล็อกหรือไม่

- ถ้าเป็นจริง จะให้ทำงานตามขั้นตอนดังนี้คือ

1.1 ทำงานในโมดูล linepointer เพื่อหาตำแหน่งของโหนดพอยน์เตอร์ fline และ lline จากค่าตำแหน่งบรรทัด lineno ของ blkbegin และ blkend ที่เป็นตำแหน่งเริ่มต้นและสุดท้ายของบล็อก

1.2 ทำงานในโมดูล copytospace เพื่อคัดลอกข้อมูลในบรรทัดที่อยู่ระหว่าง fline และ lline จากค่าตำแหน่ง column ของ blkbegin และ blkend ไปไว้ในบัฟเฟอร์โหนดว่าง ที่ถูกชี้ด้วยโหนดพอยน์เตอร์ space

1.3 ทำงานในโมดูล insertlinklist เพื่อนำข้อมูลในบัฟเฟอร์โหนดไปเชื่อมต่อกับข้อมูลในบรรทัด curline ที่คอลัมน์ x ซึ่งเป็นบรรทัดที่ใช้งานอยู่

1.4 เปลี่ยนค่าตำแหน่งบรรทัด lineno และตำแหน่ง column ของ blkbegin และ blkend ดังนี้คือ

- กำหนดค่า lineno ของ blkbegin ให้เท่ากับค่าหมายเลขบรรทัด lineno ของบรรทัดที่ใช้งานอยู่
- กำหนดค่า lineno ของ blkend ให้เท่ากับผลบวกของค่าหมายเลขบรรทัด lineno กับผลต่างของค่าตำแหน่งบรรทัด lineno ของ blkend และ blkbegin
- กำหนดค่า column ของ blkbegin ให้เท่ากับคอลัมน์ x ที่กำหนดให้

- กำหนดค่า column ของ blkend ให้เท่ากับผลบวกของคอลัมน์ x กับ ผลต่างของค่า column ของ blkend และ blkbegin ก็ต่อเมื่อค่าของตำแหน่งบรรทัด lineno ของ blkend และ blkbegin มีค่าเท่ากัน

1.5 เปลี่ยนค่าในตัวแปรทั่วไปดังนี้คือ

- pagecomplete ให้เป็น NO
- changflag และ dispblock ให้เป็น YES

- ถ้าเป็นเท็จ ก็จะเลิกทำงานไป

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.75 แล้ว

4.3.19.10 การเคลื่อนย้ายบล็อก (Move Block)

จะเป็นการเคลื่อนย้ายข้อมูลในบล็อกที่กำหนดไว้ ไปยังตำแหน่งอื่นๆของบรรทัดขณะนั้นได้ ซึ่งจะเป็นการทำงานภายใต้โมดูล moveblk โดยจะต้องกำหนดตำแหน่งคอลัมน์ x ที่จะเคลื่อนย้าย

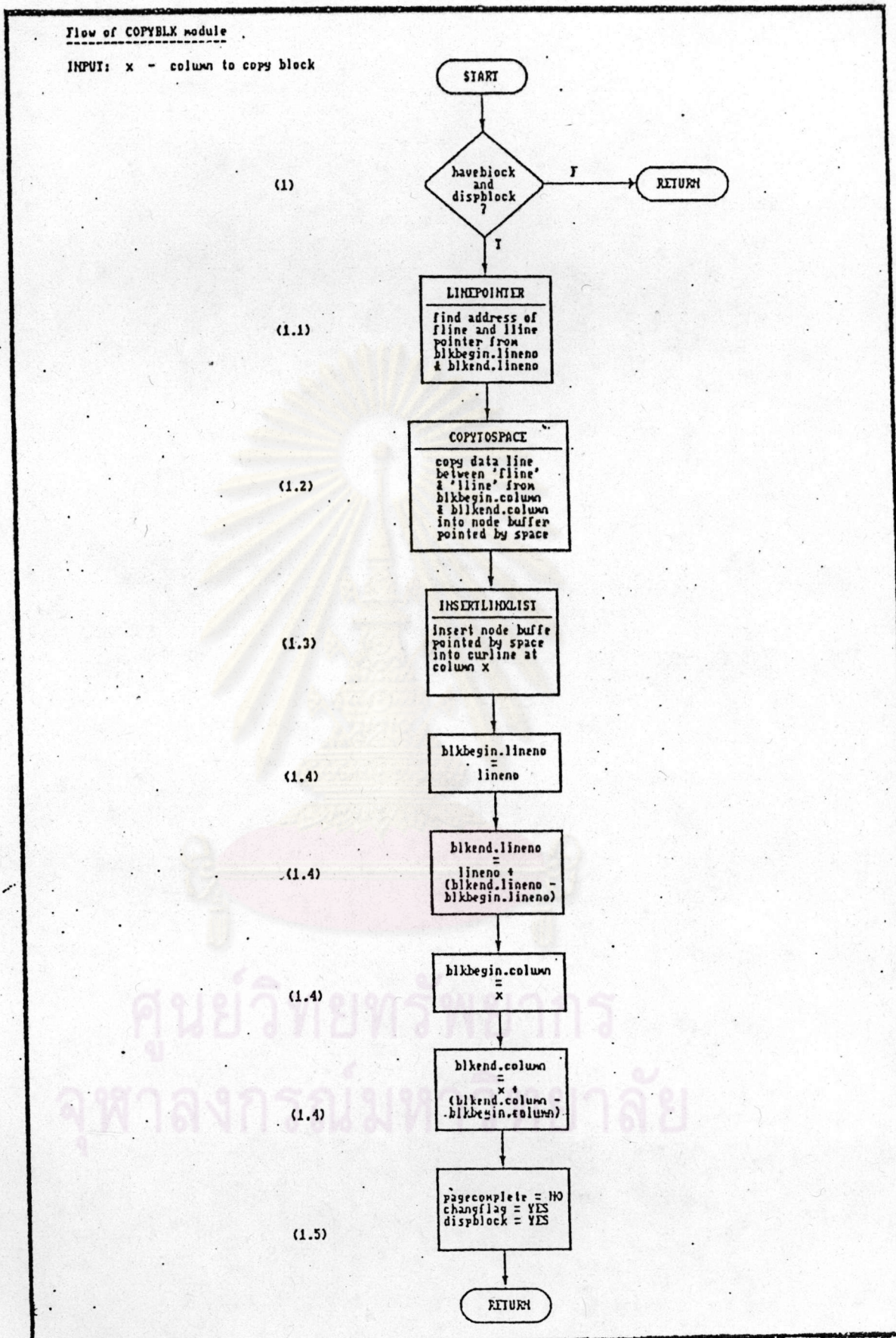
<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. ตรวจสอบว่าขณะนั้นได้มีการกำหนดบล็อกหรือไม่

- ถ้าเป็นจริง จะให้ทำงานตามขั้นตอนดังนี้คือ

1.1 ทำงานในโมดูล linepointer เพื่อหาตำแหน่งของโหนดพอยน์เตอร์ fline และ lline จากค่าตำแหน่งบรรทัด lineno ของ blkbegin และ blkend ที่เป็นตำแหน่งเริ่มต้นและสุดท้ายของบล็อก

1.2 ทำงานในโมดูล copytospace เพื่อคัดลอกข้อมูลในบรรทัดที่อยู่ระหว่าง fline และ lline จากค่าตำแหน่ง column ของ blkbegin และ blkend ไปไว้ในบัฟเฟอร์โหนดว่าง ที่ถูกชี้ด้วยโหนดพอยน์เตอร์ space



รูปภาพที่ 4.75 ผังงานแสดงขั้นตอนการทำงานของโมดูล copyblk

1.3 ทำงานในโมดูล deletelinklist เพื่อจะเป็นการลบชุดของบรรทัดข้อมูลที่อยู่ระหว่างบรรทัด fline และ lline จากตำแหน่งคอลัมน์ column ของ blkbegin และ blkend

1.4 เปลี่ยนค่าตำแหน่งบรรทัด lineno และตำแหน่ง column ของ blkbegin และ blkend ดังนี้คือ

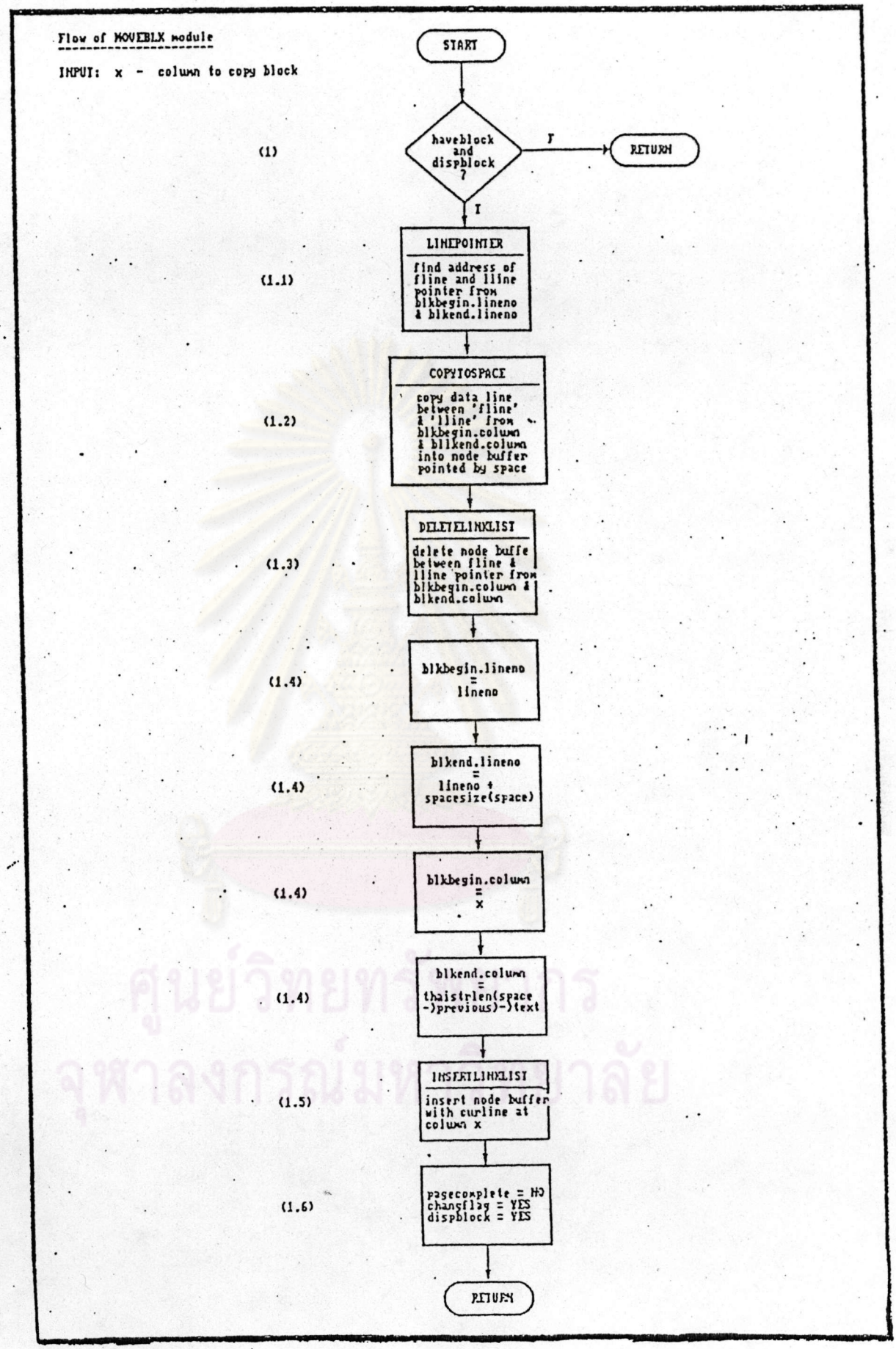
- กำหนดค่า lineno ของ blkbegin ให้เท่ากับค่าหมายเลขบรรทัด lineno ของบรรทัดที่ใช้งานอยู่
- กำหนดค่า lineno ของ blkend ให้เท่ากับผลบวกของค่าหมายเลขบรรทัด lineno กับจำนวนบรรทัดทั้งหมดที่เคลื่อนย้ายมา
- กำหนดค่า column ของ blkbegin ให้เท่ากับคอลัมน์ x ที่กำหนดให้
- กำหนดค่า column ของ blkend ตามเงื่อนไขดังนี้คือ
  - ถ้าข้อมูลในบรรทัดที่เคลื่อนย้ายมา มีเพียง 1 บรรทัด แล้ว ให้มีค่าเท่ากับผลบวกของ column ของ blkbegin กับค่าจำนวนตัวอักษรทั้งหมดในบรรทัดนั้น
  - ถ้าข้อมูลในบรรทัดที่เคลื่อนย้ายมามากกว่า 1 บรรทัด แล้วให้มีค่าเท่ากับค่าจำนวนตัวอักษรทั้งหมดในบรรทัดที่ใช้งานนั้น

1.5 ทำงานในโมดูล insertlinklist เพื่อนำข้อมูลในบัฟเฟอร์โหนดนั้นไปเชื่อมต่อกับข้อมูลในบรรทัด curline คอลัมน์ x ซึ่งเป็นบรรทัดที่ใช้งานอยู่

1.6 เปลี่ยนค่าในตัวแปร pagecomplete ให้เป็น NO และเปลี่ยนค่า changflag และ dispblock ให้เป็น YES

- ถ้าเป็นเท็จ ก็จะเลิกทำงานไป

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.76 แล้ว



รูปภาพที่ 4.76 ผังงานแสดงขั้นตอนการทำงานของโมดูล moveblk

4.3.19.11 การลบบล็อก (Delete Block)

จะเป็นการลบข้อมูลในบล็อกที่กำหนดไว้ให้หายไปจากที่ใช้งานอยู่ ซึ่งจะเป็นการทำงานภายใต้โมดูล deleteblk

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. ตรวจสอบว่าขณะนั้นได้มีการกำหนดบล็อกหรือไม่

- ถ้าเป็นจริง จะให้ทำงานตามขั้นตอนดังนี้คือ

1.1 ทำงานในโมดูล linepointer เพื่อหาตำแหน่งของโหนดพอยน์เตอร์ fline และ lline จากค่าตำแหน่งบรรทัด lineno ของ blkbegin และ blkend ที่เป็นตำแหน่งเริ่มต้นและสุดท้ายของบล็อก

1.2 ทำงานในโมดูล copytospace เพื่อคัดลอกข้อมูลในบรรทัดที่อยู่ระหว่าง fline และ lline จากค่าตำแหน่ง column ของ blkbegin และ blkend ไปไว้ในบัฟเฟอร์โหนดว่างที่ถูกชี้ด้วยโหนดพอยน์เตอร์ space

1.3 ทำงานในโมดูล deletelinklist เพื่อจะเป็นการลบชุดของบรรทัดข้อมูลที่อยู่ระหว่างบรรทัด fline และ lline จากตำแหน่งคอลัมน์ column ของ blkbegin และ blkend

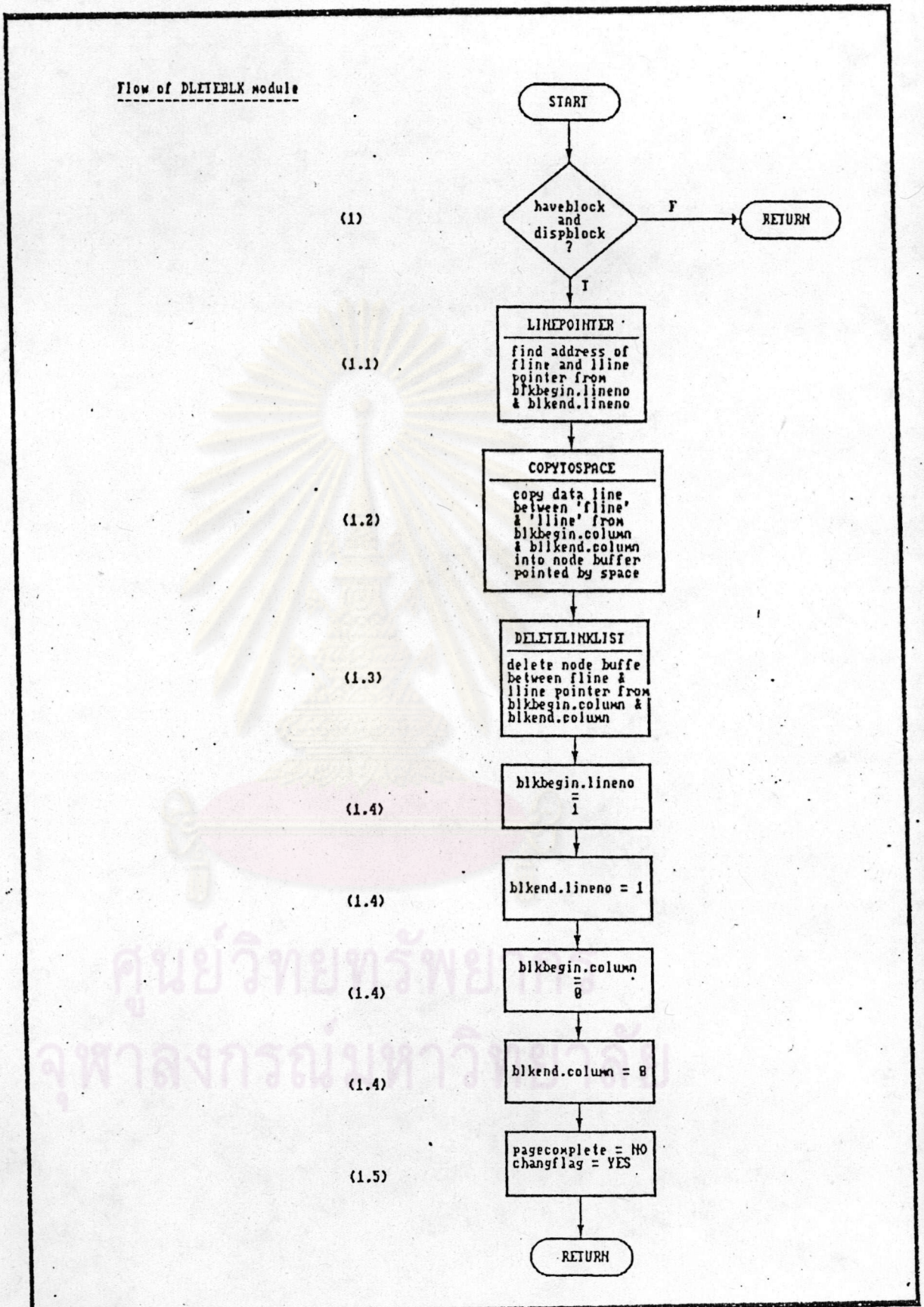
1.4 เปลี่ยนค่าตำแหน่งบรรทัด lineno และตำแหน่ง column ของ blkbegin และ blkend ดังนี้คือ

- กำหนดค่า lineno ของ blkbegin ให้เท่ากับค่า 1
- กำหนดค่า lineno ของ blkend ให้เท่ากับค่า 1
- กำหนดค่า column ของ blkbegin ให้เท่ากับค่า 0
- กำหนดค่า column ของ blkend ให้เท่ากับค่า 0

1.5. เปลี่ยนค่าในตัวแปร pagecomplete ให้เป็น NO และตัวแปร changflag ให้เป็น YES

- ถ้าเป็นเท็จ ก็จะเลิกทำงานไป

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.77 แล้ว



รูปภาพที่ 4.77 ผังงานแสดงขั้นตอนการทำงานของโมดูล deleteblk

4.3.20 การเคลื่อนที่ของเคอร์เซอร์ (Cursor Movement)

เป็นการเคลื่อนย้ายตำแหน่งของเคอร์เซอร์จากที่หนึ่งไปอีกที่หนึ่งเพื่อให้สามารถไปทำงานในบรรทัดต่างๆได้อย่างทั่วถึง เช่น ไปแทรกหรือลบตัวอักษรในบรรทัดหรือคอลัมน์ใดก็ได้ ในตำแหน่งที่เคอร์เซอร์เคลื่อนที่ไปถึง เป็นต้น

สำหรับการทำงานของการเคลื่อนที่ของเคอร์เซอร์ จะมีผลต่อการเปลี่ยนแปลงค่าในตัวแปรทั่วไปที่สำคัญ ดังนี้คือ

1. curline เป็นตัวแปรพอยน์เตอร์ ที่จะชี้ไปยังโหนดพอยน์เตอร์ในบรรทัดต่างๆได้ตั้งแต่บรรทัดแรกไปจนถึงบรรทัดสุดท้ายของข้อมูลทั้งหมด ดังนั้นจึงใช้เป็นตัวชี้บ่งบอกถึงบรรทัดที่จะใช้งาน รวมทั้งใช้คู่กับตัวแปร lineno เพื่อนับบรรทัดของการทำงานอีกด้วย

2. curpage เป็นตัวแปรพอยน์เตอร์ ที่จะชี้ไปยังโหนดพอยน์เตอร์ในบรรทัดต่างๆเช่นเดียวกันกับ curline แต่ตัวแปรนี้จะใช้ชี้เฉพาะโหนดพอยน์เตอร์ ที่จะถูกแสดงบนหน้าจอภาพเท่านั้นคือ ชี้ไปตั้งแต่บรรทัดแรกไปจนถึงบรรทัดสุดท้ายของข้อมูลที่ปรากฏบนหน้าจอภาพเดียวกันเท่านั้น

3. lineno เป็นตัวแปรชนิดตัวเลขจำนวนเต็มที่ใช้เก็บหมายเลขของบรรทัดที่ใช้งานอยู่ขณะนั้น

4. firstcol เป็นตัวแปรชนิดตัวเลขจำนวนเต็มที่ใช้เก็บหมายเลขของคอลัมน์เริ่มต้นที่จะใช้แสดงข้อความบนจอภาพ โดยปกติจะกำหนดค่าเป็นศูนย์

5. pagecomplete เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม ที่ใช้แสดงถึงสภาวะการแสดงข้อความบนจอภาพที่สมบูรณ์แบบ โดยถ้าเป็นค่า YES หรือ 1 แสดงว่ามีการแสดงข้อความบนจอภาพที่สมบูรณ์แบบ โดยไม่มีการขัดจังหวะโดยการกดคีย์และถ้าเป็นค่า NO หรือ 0 แสดงว่า มีการขัดจังหวะด้วยการกดคีย์ จึงเป็นการแสดงข้อความที่ไม่สมบูรณ์แบบ

ลักษณะการทำงานของการเคลื่อนที่ของเคอร์เซอร์ ที่สำคัญมีดังนี้คือ

4.3.20.1 การเคลื่อนที่ไปยังบรรทัดที่ต้องการ

เป็นการเคลื่อนที่ไปยังบรรทัดที่ระบุหมายเลขบรรทัดไว้ โดยจะเป็นการทำงานภายใต้โมดูล "goline" โดยระบุค่า linetogo ซึ่งเป็นค่าของหมายเลขบรรทัดที่จะเคลื่อนที่ไป

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. กำหนดให้ curline ชี้ไปยังบรรทัดแรกสุดของข้อมูล
2. กำหนดให้ lineno เป็นค่า 1
3. ตรวจสอบค่าของ lineno ว่ายังไม่เท่ากับ linetogo หรือไม่ รวมทั้งตรวจสอบอีกว่า curline ยังชี้ไปไม่ถึงบรรทัดสุดท้ายของข้อมูลหรือไม่
   - ถ้าเป็นเท็จไม่ว่ากรณีใด จะไปทำงานในหัวข้อถัดไป
   - ถ้าเป็นจริง จะทำการเปลี่ยนค่า curline ให้ชี้ไปยังบรรทัดถัดไป และเพิ่มค่าของตัวแปร lineno อีก 1
4. กำหนดให้ curpage ชี้ไปยังบรรทัดเดียวกันกับที่ curline ชี้อยู่
5. กำหนดให้ pagecomplete เป็น NO

4.3.20.2 การเคลื่อนที่ไปยังคอลัมน์ที่ต้องการ

เป็นการเคลื่อนที่ไปยังคอลัมน์ที่ระบุหมายเลขคอลัมน์ไว้โดยจะเป็นการทำงานภายใต้โมดูล "gocol" โดยระบุตัวแปร coltogo ซึ่งจะเก็บค่าของหมายเลขคอลัมน์ที่จะเคลื่อนที่ไป และตัวแปร x ที่จะเก็บตำแหน่งของคอลัมน์ที่แท้จริงในบัฟเฟอร์ทำงาน ซึ่งได้จากการทำงานภายในโมดูล

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. ตรวจสอบค่าของ coltogo ว่ามีค่ามากกว่า MAXCOL (ค่าคอลัมน์ที่มากที่สุด ซึ่งมีค่าเป็น 256) หรือไม่ ถ้ามากกว่าจริง จะกำหนดให้ coltogo เท่ากับค่า MAXCOL
2. ตรวจสอบค่าของ coltogo ว่ามีค่ามากกว่า ความยาวของตัวอักษรในบัฟเฟอร์ทำงานระดับกลาง (workline.middle) หรือไม่ ถ้ามากกว่าจริง จะกำหนดให้

coltogo เท่ากับ ค่าความยาวของตัวอักษรในบัฟเฟอร์ทำงานระดับกลาง

3. ตรวจสอบค่าของ coltogo ว่ามีค่าน้อยกว่า firstcol หรือไม่ ถ้าน้อยกว่าจริง จะกำหนดให้ firstcol เท่ากับ coltogo
4. กำหนดค่าให้ตัวแปร x เท่ากับผลต่างของ coltogo กับ firstcol โดยค่าในตัวแปร x จะเก็บค่าคอลัมน์ในบัฟเฟอร์ทำงาน ซึ่งจะนำไปใช้งานต่อไปได้

4.3.20.3 การเคลื่อนที่ไปยังต้นบรรทัด

เป็นการเคลื่อนที่ไปยังต้นบรรทัดหรือคอลัมน์แรกของบรรทัดโดยจะเป็นการทำงานภายใต้โมดูล "home" โดยระบุตัวแปร x ที่จะเก็บตำแหน่งของคอลัมน์ที่แท้จริงในบัฟเฟอร์ทำงาน ซึ่งได้จากการทำงานภายในโมดูล โดยจะกำหนดให้ตัวแปร x และ firstcol เป็นค่าศูนย์ และกำหนดให้ตัวแปร pagecomplete เป็น NO

4.3.20.4 การเคลื่อนที่ไปยังท้ายบรรทัด

เป็นการเคลื่อนที่ไปยังท้ายบรรทัดหรือ คอลัมน์สุดท้ายของบรรทัด โดยจะเป็นการทำงานภายใต้โมดูล "endline" โดยจะไปทำงานในโมดูล "gocol" โดยระบุตำแหน่งคอลัมน์ที่เป็นค่าขนาดของตัวอักษรในบัฟเฟอร์ทำงานขนาดกลาง

4.3.20.5 การเคลื่อนที่ขึ้น 1 บรรทัด

เป็นการเคลื่อนที่ของเคอร์เซอร์ขึ้นไปอีก 1 บรรทัด โดยจะเป็นการทำงานภายใต้โมดูล "cursor_up"

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. ลดค่าของ lineno อีก 1
2. ตรวจสอบว่าบรรทัดก่อนหน้าของบรรทัดที่ curline ชี้อยู่ เป็นบรรทัดแรกของข้อมูลบนหน้าจอภาพ หรือไม่

   - ถ้าเป็นจริง จะทำงานดังนี้คือ

     2.1 กำหนดให้ curpage ชี้ไปยังบรรทัดก่อนหน้าจากบรรทัดที่ใช้งานอยู่

2.2 กำหนดให้ curline ชี้ไปยังบรรทัดก่อนหน้าจากบรรทัดที่ใช้งานอยู่

2.3 กำหนดให้ pagecomplete เป็นค่า NO

- ถ้าไม่จริง จะกำหนดให้ curline ชี้ไปยังบรรทัดก่อนหน้าจากบรรทัดที่ใช้งาน

4.3.20.6 การเคลื่อนที่ลง 1 บรรทัด

เป็นการเคลื่อนที่ของเคอร์เซอร์ลงไปอีก 1 บรรทัด โดยจะเป็นการทำงานภายใต้โมดูล "cursor_down" โดยระบุค่าของ y ที่เป็นตำแหน่งบรรทัดที่ใช้งานอยู่

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. เพิ่มค่าของ lineno อีก 1
2. กำหนดให้ curline เปลี่ยนไปชี้บรรทัดถัดไป จากบรรทัดที่ใช้งานอยู่
3. ตรวจสอบค่าของ y ว่าเป็นบรรทัดล่างสุดของจอภาพ หรือไม่

- ถ้าเป็นจริง จะทำงานดังนี้คือ

3.1 จะตรวจสอบว่าบรรทัดขณะนั้นเป็นบรรทัดที่ครบ 1 หน้าหรือไม่ ถ้าครบหน้าจริงแล้ว จะกำหนดให้ curpage เปลี่ยนไปชี้บรรทัดถัดไป จากบรรทัดที่ใช้งานอยู่

3.2 กำหนดให้ curpage เปลี่ยนไปชี้บรรทัดถัดไป จากบรรทัดที่ใช้งานอยู่

3.3 กำหนดให้ pagecomplete เป็น NO

- ถ้าไม่จริง จะทำงานดังนี้คือ

3.4 จะตรวจสอบว่าบรรทัดขณะนั้นเป็นบรรทัดที่ครบ 1 หน้าหรือไม่ ถ้าครบหน้าจริงแล้วจะกำหนดให้ curpage เปลี่ยนไปชี้บรรทัดถัดไป จากบรรทัดที่ใช้งานอยู่

3.5 กำหนดให้ pagecomplete เป็น NO

4.3.20.7 การเคลื่อนที่ไปทางซ้าย 1 ตัวอักษร

เป็นการเคลื่อนที่ของเคอร์เซอร์ไปทางซ้าย 1 ตัวอักษรโดยจะเป็นการทำงานภายใต้โมดูล "cursor_left" โดยระบุตำแหน่งคอลัมน์ x

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. กำหนดให้ i เป็นค่าผลบวกของ x กับ firstcol
2. ตรวจสอบค่าของ i ว่าเท่ากับศูนย์หรือไม่
   - ถ้าเท่ากับศูนย์ แสดงว่าขณะนั้นเคอร์เซอร์อยู่ที่ตำแหน่งต้นบรรทัด จะทำงานดังนี้คือ
     2.1 ไปทำงานในโมดูล "cursor_up" เพื่อเลื่อนตำแหน่งเคอร์เซอร์ขึ้นไปอีก 1 บรรทัด
     2.2 ไปทำงานในโมดูล "endline" เพื่อเลื่อนตำแหน่งเคอร์เซอร์ไปท้ายสุดของบรรทัด
   - ถ้าไม่เท่ากับศูนย์ แสดงว่าขณะนั้นเคอร์เซอร์อยู่ที่ตำแหน่งอื่นๆที่ไม่ใช่ต้นบรรทัด จะไปทำงานในโมดูล "gocol" โดยกำหนดตำแหน่งคอลัมน์ที่จะเคลื่อนไปทางซ้าย 1 คอลัมน์ (i - 1)

4.3.20.8 การเคลื่อนที่ไปทางขวา 1 ตัวอักษร

เป็นการเคลื่อนที่ของเคอร์เซอร์ไปทางขวา 1 ตัวอักษรโดยจะเป็นการทำงานภายใต้โมดูล "cursor_right" โดยระบุตำแหน่งคอลัมน์ x

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. ตรวจสอบค่าผลบวกของ x กับ firstcol + 1 ว่าเท่ากับขนาดความยาวของตัวอักษรในบัฟเฟอร์ หรือไม่
   - ถ้าเท่ากัน แสดงว่าขณะนั้นเคอร์เซอร์อยู่ที่ตำแหน่งท้ายสุดของบรรทัด จะทำงานดังนี้คือ
     1.1 ไปทำงานในโมดูล "cursor_down" เพื่อเลื่อนตำแหน่งเคอร์เซอร์ลง 1 บรรทัด

1.2 ไปทำงานในโมดูล "home" เพื่อเลื่อนตำแหน่งเคอร์เซอร์ไปต้นสุดของบรรทัด

- ถ้าไม่เท่ากัน แสดงว่าขณะนั้นเคอร์เซอร์อยู่ที่ตำแหน่งอื่นๆที่ไม่ใช่ท้ายบรรทัด ซึ่งจะทำงานดังนี้คือ

1.3 จะตรวจสอบค่าในตัวแปร x ว่ามีค่าเกินกว่าขนาดจอภาพด้านขวาสุดหรือไม่ ซึ่งถ้ามีค่าเกินกว่าจริง จะไปทำงานในโมดูล "shiftscrn" เพื่อขยับจอภาพไปทางขวามือ

1.4. เพิ่มค่าในตัวแปร x อีก 1

4.3.20.9 การเลื่อนจอภาพขึ้น 1 บรรทัด

เป็นการเคลื่อนที่ของบรรทัดบนจอภาพขึ้น 1 บรรทัด โดยตำแหน่งเคอร์เซอร์ไม่เคลื่อนที่ ซึ่งจะเป็นการทำงานภายใต้โมดูล "scroll_up"

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. ตรวจสอบบรรทัดก่อนหน้าของ curpage ว่าขณะนั้นอยู่ที่บรรทัดแรกสุดของข้อมูลหรือไม่ ถ้าอยู่บรรทัดแรกสุดจริงแล้ว จะทำงานดังนี้คือ

1.1 กำหนดให้ curpage เปลี่ยนไปชี้ที่บรรทัดก่อนหน้า

1.2 กำหนดให้ pagecomplete เป็น NO

4.3.20.10 การเลื่อนจอภาพลง 1 บรรทัด

เป็นการเคลื่อนที่ของบรรทัดบนจอภาพลง 1 บรรทัด โดยตำแหน่งเคอร์เซอร์ไม่เคลื่อนที่ ซึ่งจะเป็นการทำงานภายใต้โมดูล "scroll_down"

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. ตรวจสอบบรรทัดก่อนหน้าของ curpage ว่าขณะนั้นอยู่ที่บรรทัดแรกสุดของข้อมูลหรือไม่ ถ้าอยู่บรรทัดแรกสุดจริงแล้ว จะทำงานดังนี้คือ

1.1 กำหนดให้ curpage เปลี่ยนไปชี้ที่บรรทัดก่อนหน้า

1.2 กำหนดให้ pagecomplete เป็น NO

4.3.20.11 การเลื่อนจอภาพขึ้น 1 หน้า

เป็นการเคลื่อนที่ของบรรทัดบนจอภาพขึ้นไป 1 หน้า โดยจะเป็นการทำงานภายใต้โมดูล "page_up"

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. ตรวจสอบบรรทัดก่อนหน้าของ curpage ว่าเป็นบรรทัดแรกสุดของข้อมูล หรือไม่
   - ถ้าเป็นบรรทัดแรกสุดแล้ว จะทำงานดังนี้คือ
     1.1 กำหนดให้ curline ชี้ไปบรรทัดเดียวกันกับ curpage ชี้อยู่
     1.2 กำหนดให้ lineno เป็นค่า 1
   - ถ้าไม่ได้เป็นบรรทัดแรกสุดแล้ว จะมีขั้นตอนทำงานดังนี้คือ
     1.1 กำหนดให้ count เป็นค่าของจำนวนบรรทัดต่อหน้าบนจอภาพ
     1.2 ตรวจสอบว่า count ยังคงมากกว่าศูนย์หรือไม่ และ curpage ชี้ไปจนถึงบรรทัดแรกสุดของข้อมูลหรือไม่ โดยถ้าเงื่อนไขทั้งสองกรณีเป็นจริงแล้ว จะทำงานดังนี้คือ
        1.2.1 กำหนดให้ curpage เปลี่ยนชี้ไปยังบรรทัดก่อนหน้า
        1.2.2 กำหนดให้ curline เปลี่ยนชี้ไปยังบรรทัดก่อนหน้า
        1.2.3 ลดค่าของ lineno และ count ลงไปอีก 1
        1.2.4 กลับไปทำงานในหัวข้อ 1.2 อีกครั้ง
     1.3 กำหนดให้ pagecomplete เป็น NO

4.3.20.12 การเลื่อนจอภาพลง 1 หน้า

เป็นการเคลื่อนที่ของบรรทัดบนจอภาพลงอีก 1 หน้า โดยจะเป็นการทำงานภายใต้โมดูล "page_down"

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. ตรวจสอบบรรทัดถัดไปของ curpage ว่าเป็นบรรทัดสุดท้ายของข้อมูล หรือไม่
   - ถ้าเป็นบรรทัดสุดท้ายแล้ว จะทำงานดังนี้คือ

1.1 กำหนดให้ curpage ชี้ไปบรรทัดเดียวกันกับ curline ชี้อยู่

1.2 กำหนดให้ pagecomplete เป็น NO

- ถ้าไม่ได้เป็นบรรทัดสุดท้ายแล้ว จะทำงานดังนี้คือ

1.1 กำหนดให้ count เป็นค่าของจำนวนบรรทัดต่อหน้าบนจอภาพ

1.2 ตรวจสอบว่า count ยังคงมากกว่าศูนย์หรือไม่ และ curline ชี้ไปจนถึงบรรทัดสุดท้ายของข้อมูลหรือไม่ โดยถ้าเงื่อนไขทั้งสองกรณีเป็นจริงแล้ว จะทำงานดังนี้คือ

1.2.1 กำหนดให้ curline เปลี่ยนชี้ไปยังบรรทัดถัดไป

1.2.2 กำหนดให้ curpage เปลี่ยนชี้ไปยังบรรทัดถัดไป

1.2.3 ลดค่าของ count ลงไปอีก 1 และ เพิ่มค่าของ lineno ขึ้นไปอีก 1

1.2.4 กลับไปทำงานในหัวข้อ 1.2

1.3 กำหนดให้ pagecomplete เป็น NO

4.3.20.13 การเคลื่อนที่ไปต้นไฟล์

เป็นการเคลื่อนที่ของเคอร์เซอร์ ไปตำแหน่งบรรทัดแรกสุดของข้อมูลทั้งหมด โดยจะเป็นการทำงานภายใต้โมดูล "topfile" ซึ่งจะต้องกำหนดตัวแปร x ที่จะเก็บตำแหน่งของคอลัมน์ที่แท้จริงในบัฟเฟอร์ทำงาน

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. ตรวจสอบว่า curpage ชี้อยู่ที่บรรทัดแรกสุดของข้อมูลหรือไม่ โดยถ้าไม่ได้ชี้อยู่ที่บรรทัดแรกสุดแล้ว จะทำงานดังนี้คือ

1.1 กำหนดให้ curpage ชี้ไปยังบรรทัดแรกสุดของข้อมูล

1.2 กำหนดให้ pagecomplete เป็น NO

2. กำหนดให้ curline ชี้ไปบรรทัดเดียวกันกับที่ curpage ชี้อยู่
3. กำหนดให้ x และ firstcol เป็นค่าศูนย์
4. กำหนดให้ lineno เป็นค่า 1

4.3.20.14 การเคลื่อนที่ไปท้ายไฟล์

เป็นการเคลื่อนที่ของเคอร์เซอร์ ไปตำแหน่งบรรทัดสุดท้ายของข้อมูลทั้งหมด โดยจะเป็นการทำงานภายใต้โมดูล "endfile" ซึ่งจะต้องกำหนดตัวแปร x ที่จะเก็บตำแหน่งของคอลัมน์ที่แท้จริงในบัฟเฟอร์ทำงาน

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. กำหนดให้ curline ชี้ไปยังบรรทัดสุดท้ายของข้อมูล
2. กำหนดให้ curpage ชี้ไปบรรทัดเดียวกันกับที่ curline ชี้อยู่
3. ไปทำงานในโมดูล "findlineno" เพื่อหาหมายเลขบรรทัดของ curline แล้วไปเก็บไว้ใน lineno
4. กำหนดให้ count เป็นค่าของจำนวนบรรทัดต่อหน้าบนจอภาพ
5. ตรวจสอบค่า pagebreak ว่าขณะนั้นมีการขึ้นหน้าใหม่หรือไม่ โดยถ้ามีการขึ้นหน้าใหม่จริงแล้ว จะทำงานในโมดูล "findlineno" เพื่อหาหมายเลขบรรทัดของ curpage แล้วไปเก็บไว้ใน linenum
6. ตรวจสอบว่า count ยังไม่เป็นศูนย์หรือไม่และ curpage ชี้ไปจนถึงบรรทัดแรกสุดของข้อมูลหรือไม่ โดยถ้าเงื่อนไขทั้งสองกรณีเป็นจริงแล้ว จะทำงานดังนี้คือ
   6.1 ตรวจสอบค่า pagebreak ว่าขณะนั้นมีการขึ้นหน้าใหม่หรือไม่
   - ถ้ามีการขึ้นหน้าใหม่จริงแล้ว จะทำงานดังนี้คือ
     6.1.1. กำหนดให้ curpage เปลี่ยนชี้ไปยังบรรทัดก่อนหน้า
     6.1.2. ลดค่าของ linenum ลงไปอีก 1
   - ถ้าไม่มีการขึ้นหน้าแล้ว จะกำหนดให้ curpage เปลี่ยนชี้ไปยังบรรทัดก่อนหน้า
   6.2 ลดค่าของ count ลงไปอีก 1
   6.3. กลับไปทำงานในหัวข้อ 6
7. ทำงานในโมดูล "endline" เพื่อจะเลื่อนเคอร์เซอร์ไปตำแหน่งท้ายสุดของบรรทัด
8. กำหนดให้ pagecomplete เป็น NO

4.3.20.15 การเคลื่อนที่ไปทางซ้ายทีละคำ

เป็นการเคลื่อนที่ของเคอร์เซอร์ไปทางซ้ายทีละคำซึ่งในแต่ละคำจะถูกคั้นด้วยตัวอักษรช่องว่าง โดยการทำงานจะอยู่ภายใต้โมดูล "backword" ซึ่งจะต้องกำหนดตัวแปร x ที่จะเก็บตำแหน่งของคอลัมน์ที่แท้จริงในบัฟเฟอร์ทำงาน

การทำงาน จะมีขั้นตอนดังนี้คือ

1. กำหนดให้ i เป็นค่าผลบวกของ x กับ firstcol + 1
2. ตรวจสอบค่า i ว่าเป็นค่า 1 หรือไม่
   - ถ้าค่าของ i เป็น 1 แสดงว่าขณะนั้นอยู่ในตำแหน่งต้นบรรทัด จะทำงานดังนี้คือ
     - 2.1 ทำงานในโมดูล "cursor_up" เพื่อเลื่อนเคอร์เซอร์ขึ้น 1 บรรทัด
     - 2.2 ทำงานในโมดูล "endline" เพื่อเลื่อนเคอร์เซอร์ไปยังท้ายสุดของบรรทัด
   - ถ้าค่าของ i ไม่เป็น 1 แสดงว่า ไม่ได้อยู่ในตำแหน่งต้นบรรทัด จะทำงานดังนี้คือ
     - 2.3 จะตรวจสอบค่าในบัฟเฟอร์ทำงานระดับกลาง ตัวที่ i ไปเรื่อยๆจนกว่าจะพบตัวอักษรช่องว่างโดยขณะที่ตรวจสอบค่าจะลดค่าของ i ลงไปอีก 1 ด้วย
     - 2.4 ไปทำงานในโมดูล "gocol" โดยระบุคอลัมน์ i ซึ่งเป็นคอลัมน์ที่จะเลื่อนไป

4.3.20.16 การเคลื่อนที่ไปทางขวาทีละคำ

เป็นการเคลื่อนที่ของเคอร์เซอร์ไปทางขวาทีละคำ ซึ่งแต่ละคำจะถูกคั้นด้วยตัวอักษรช่องว่างโดยการทำงานจะอยู่ภายใต้โมดูล "nextword" ซึ่งจะต้องกำหนดตัวแปร x ที่จะเก็บตำแหน่งของคอลัมน์ที่แท้จริงในบัฟเฟอร์ทำงาน

การทำงาน จะมีขั้นตอนดังนี้คือ

1. กำหนดให้ i และ j เป็นค่าผลบวกของ x กับ firstcol + 1
2. จะตรวจสอบค่าในบัฟเฟอร์ทำงานระดับกลางตัวที่ i ไปเรื่อยๆจนกว่าจะพบตัวอักษรช่องว่าง โดยขณะที่ตรวจสอบค่าจะเพิ่มค่าของ i อีก 1 ด้วย

3. ตรวจสอบค่า i ว่าเท่ากับค่ามากที่สุดของคอลัมน์ (MAXCOL) หรือไม่

- ถ้าไม่เท่ากันแล้ว จะทำงานดังนี้คือ

    3.1 ลดค่าของ i ลงอีก 1

    3.2 ทำงานในโมดูล "gocol" โดยระบุคอลัมน์ i ให้เป็นคอลัมน์ที่จะเลื่อนไป

- ถ้าเท่ากันแล้ว จะทำงานดังนี้คือ

    3.3 ทำงานในโมดูล "endline" โดยระบุตำแหน่งของ x เพื่อให้เลื่อนเคอร์เซอร์ไปท้ายสุดของบรรทัด

    3.4. ตรวจสอบค่าผลบวกของ x กับ firstcol + 1 ว่าไม่เกินค่า j หรือไม่

    โดยถ้าเป็นจริง จะทำงานดังนี้คือ

        3.4.1 ทำงานในโมดูล "cursor_down" เพื่อที่จะเลื่อนเคอร์เซอร์ลงไป 1 บรรทัด

        3.4.2 ทำงานในโมดูล "home" เพื่อเลื่อนเคอร์เซอร์ไปที่ต้นบรรทัด

4.3.21 <u>การค้นหาและแทนที่ของข้อความ</u>

จะเป็นการค้นหาข้อความที่ต้องการ ตั้งแต่บรรทัดที่ใช้งานอยู่ขณะนั้น โดยถ้าค้นหาข้อความพบก็จะแสดงตำแหน่งของข้อความนั้นๆให้เห็น ซึ่งอาจจะให้มีการแทนที่ด้วยข้อความใหม่ที่กำหนดไว้ก็ได้

การค้นหาและแทนที่ข้อความ จะทำงานได้โดยใช้ตัวแปรทั่วไปที่สำคัญอยู่ 3 ตัวคือ

1. source เป็นตัวแปรตัวอักษรขนาด 80 ตัวอักษร ใช้สำหรับเก็บข้อความที่จะต้องการใช้ค้นหา

2. replace เป็นตัวแปรตัวอักษรขนาด 80 ตัวอักษร ใช้สำหรับเก็บข้อความที่จะต้องการแทนที่ข้อความเดิมที่ค้นหาพบ

3. option เป็นตัวแปรตัวอักษรขนาด 5 ตัวอักษร ใช้สำหรับเก็บรูปแบบวิธีการค้นหาและแทนที่ข้อความ รูปแบบนี้เป็นรหัสตัวอักษร 1 ตัว ซึ่งจะมีค่าต่างๆดังนี้คือ

   G = ค้นหาและแทนที่ข้อความตั้งแต่ต้นไฟล์จนถึงสุดท้ายของไฟล์

   U = ค้นหาข้อความที่เป็นตัวอักษรภาษาอังกฤษโดยไม่คำนึงถึงตัวอักษรตัวเล็กหรือตัวใหญ่

   W = ค้นหาเป็นคำเฉพาะที่ต้องการเท่านั้น

   N = ไม่ต้องมีการถามตอบ ให้มีการแทนที่ข้อความได้ทันที

หลักการทำงานที่สำคัญต่างๆ มีดังนี้คือ

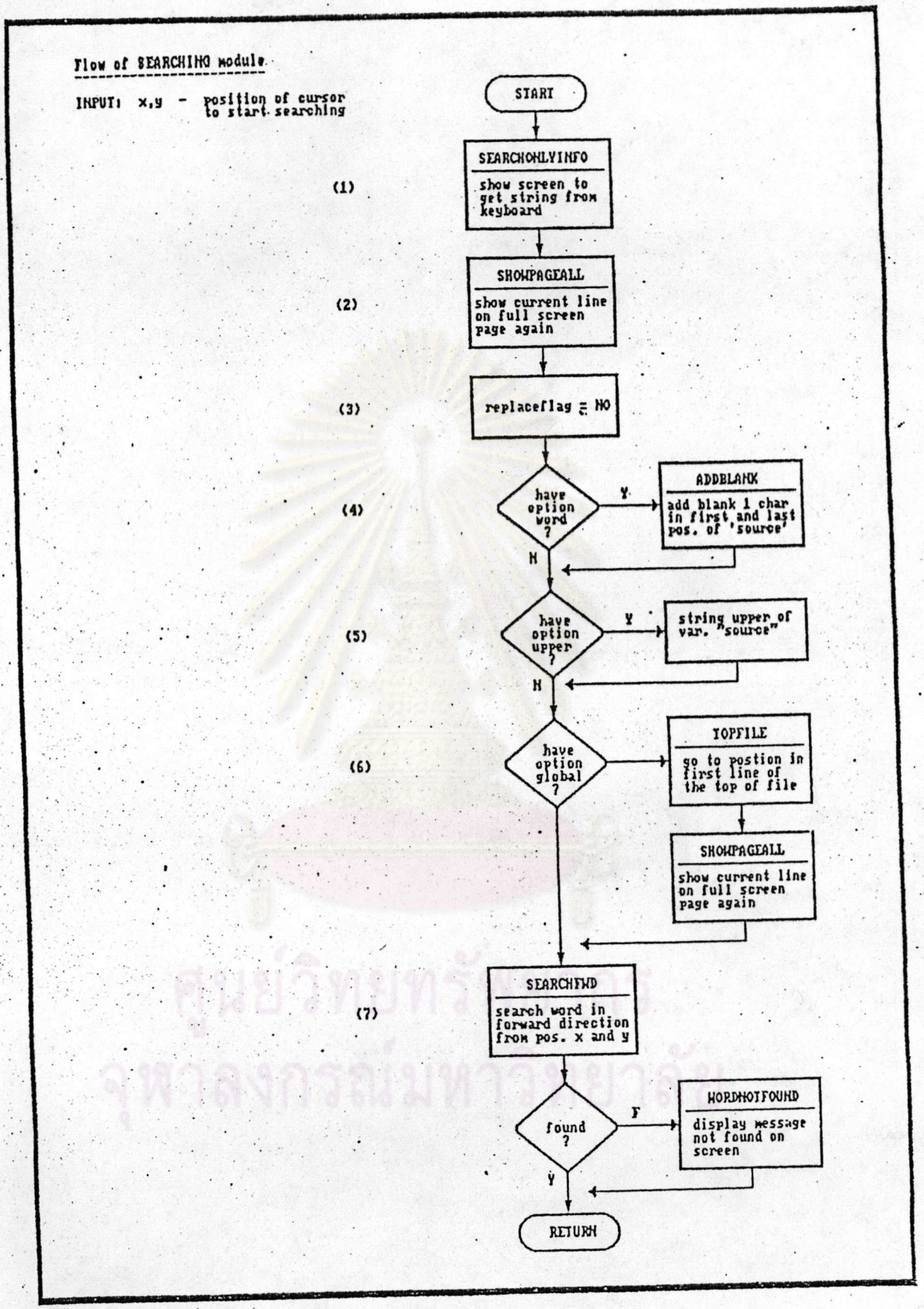
4.3.21.1 การค้นหาข้อความที่ต้องการ

จะเป็นการค้นหาข้อความที่ต้องการตั้งแต่บรรทัดที่ใช้งานอยู่ขณะนั้น โดยถ้าค้นหาข้อความพบในบรรทัดใด ก็จะแสดงให้เห็นข้อความในบรรทัดนั้นบนจอภาพ การทำงานนี้จะอยู่ภายใต้โมดูล searching โดยจะกำหนดตำแหน่ง x และ y ให้

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. ทำงานในโมดูล "searchonlyinfo" เพื่อจะรับข้อความและวิธีที่จะต้องการค้นหาจากแป้นพิมพ์บนจอภาพ มาเก็บค่าไว้ในตัวแปร source และ option
2. ทำงานในโมดูล "showpageall" เพื่อจะแสดงบรรทัดบนจอภาพอีกครั้ง
3. กำหนดให้ค่าตัวแปร replaceflag เป็น NO
4. ทำงานในโมดูล "optionword" เพื่อตรวจสอบว่าค่าในตัวแปร option มีตัวอักษร W อยู่หรือไม่ ถ้ามีอยู่จริง จะไปทำงานในโมดูล "addblank" เพื่อเพิ่มช่องว่าง 1 ตัวอักษรไว้ในตำแหน่งแรก และสุดท้ายของข้อความในตัวแปร source
5. ทำงานในโมดูล "optionupper" เพื่อตรวจสอบว่าค่าในตัวแปร option มีตัวอักษร U อยู่หรือไม่ ถ้ามีอยู่จริง จะไปแปลงข้อความในตัวแปร source ให้เป็นตัวอักษรตัวใหญ่ให้หมด
6. ทำงานในโมดูล "optionglobal" เพื่อตรวจสอบว่าค่าในตัวแปร option มีตัวอักษร G อยู่จริงหรือไม่ ถ้ามีอยู่จริง จะไปทำงานในโมดูล "topfile" เพื่อจะเปลี่ยนตำแหน่งค้นหา ให้ไปอยู่ในตอนต้นของไฟล์ และทำงานในโมดูล "showpageall" เพื่อจะแสดงบรรทัดบนจอภาพอีกครั้ง
7. ทำงานในโมดูล "searchfwd" เพื่อทำการค้นหาข้อความในแต่ละบรรทัด โดยนับตั้งแต่ตำแหน่งของ x และ y ซึ่งจะเป็นการเปรียบเทียบข้อความในบรรทัดกับข้อความในตัวแปร source ว่าตรงกันหรือไม่
   - ถ้าตรงกันจริง แสดงว่าค้นหาข้อความนั้นพบ ก็จะเลิกทำงานไป
   - ถ้าไม่ตรงกัน แสดงว่าค้นหาข้อความนั้นไม่พบ ก็จะไปทำงานในโมดูล "wordnotfound" เพื่อแสดงข้อความไม่พบข้อความบนจอภาพ แล้วก็เลิกทำงานไป

   สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.78 แล้ว



รูปภาพที่ 4.78 ผังงานแสดงขั้นตอนการทำงานของโมดูล searching

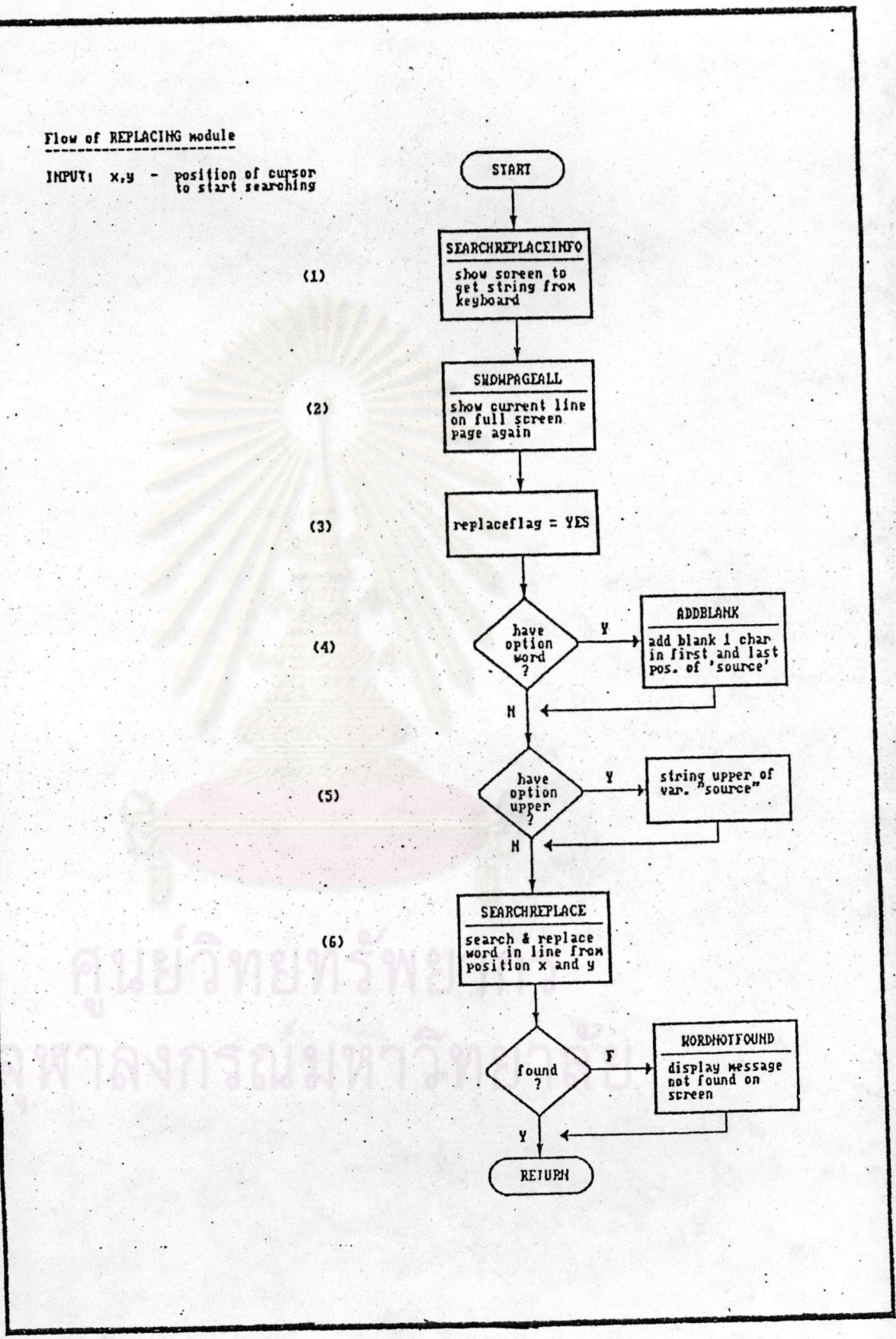
4.3.21.2 การค้นหาและแทนที่ข้อความ

จะเป็นการค้นหาข้อความที่ต้องการตั้งแต่บรรทัดที่ใช้งานอยู่ขณะนั้น โดยถ้าค้นหาข้อความพบอยู่ในบรรทัดใด ก็จะมีการแทนที่ด้วยข้อความใหม่ที่ต้องการนั้น การทำงานนี้จะอยู่ภายใต้โมดูล replacing โดยจะกำหนดตำแหน่ง x และ y ให้

<u>การทำงาน</u> จะมีขั้นตอนดังนี้คือ

1. ทำงานในโมดูล "searchreplaceinfo" เพื่อจะรับข้อความและวิธีที่จะต้องการค้นหาจากแป้นพิมพ์บนจอภาพ มาเก็บค่าไว้ในตัวแปร source และ option
2. ทำงานในโมดูล "showpageall" เพื่อจะแสดงบรัรทัดบนจอภาพอีกครั้ง
3. กำหนดให้ค่าตัวแปร replaceflag เป็น YES
4. ทำงานในโมดูล "optionword" เพื่อตรวจสอบว่าค่าในตัวแปร option มีตัวอักษร W อยู่หรือไม่ ถ้ามีอยู่จริง จะไปทำงานในโมดูล "addblank" เพื่อเพิ่มช่องว่าง 1 ตัวอักษรไว้ในตำแหน่งแรก และสุดท้ายของข้อความในตัวแปร source
5. ทำงานในโมดูล "optionupper" เพื่อตรวจสอบว่าค่าในตัวแปร option มีตัวอักษร U อยู่หรือไม่ ถ้ามีอยู่จริง จะไปแปลงข้อความในตัวแปร source ให้เป็นตัวอักษรตัวใหญ่ให้หมด
6. ทำงานในโมดูล "searchreplace" เพื่อทำการค้นหาข้อความในแต่ละบรรทัดโดยนับตั้งแต่ตำแหน่งของ x และ y ซึ่งจะเป็นการเปรียบเทียบข้อความในบรรทัดกับข้อความในตัวแปร source ว่า ตรงกันหรือไม่
   - ถ้าตรงกันจริง แสดงว่าค้นหาข้อความนั้นพบ ก็จะเลิกทำงานไป
   - ถ้าไม่ตรงกัน แสดงว่าค้นหาข้อความนั้นไม่พบ ก็จะไปทำงานในโมดูล "wordnotfound" เพื่อแสดงข้อความไม่พบข้อความบนจอภาพ แล้วจะเลิกทำงาน

สำหรับขั้นตอนการทำงานได้แสดงเป็นผังงานไว้ในรูปภาพที่ 4.79 แล้ว



รูปภาพที่ 4.79 ผังงานแสดงขั้นตอนการทำงานของโมดูล replacing

4.3.22 การทำงานด้วยพลูดาวน์เมนู (Pulldown Menu)

คำสั่งของโปรแกรมซียูไรท์เตอร์สามารถเลือกใช้ผ่านทางเมนูได้โดย เมนูที่ใช้งานนั้นจะเป็นลักษณะของพลูดาวน์เมนู (Pulldown Menu) คือ จะมีคำสั่งสำคัญอยู่ในเมนูหลัก และในแต่ละเมนูหลักนั้น จะมีเมนูย่อยให้เลือกคำสั่งเพิ่มเติมได้โดยวิธีเลือกใช้คำสั่งจะทำได้ด้วยการกดแป้นพิมพ์ลูกศรซ้าย (<--) หรือลูกศรขวา (-->) เพื่อเลื่อนแถบทึบ (High Light) ไปทางซ้ายหรือขวามือของคำสั่งที่ต้องการในเมนูหลักและใช้แป้นพิมพ์ลูกศรขึ้น (↑) หรือลูกศรลง (↓) เพื่อเลื่อนแถบทึบไปยังคำสั่งที่ต้องการในเมนูย่อย แล้วกดคีย์รีเทิร์น (return) เพื่อไปทำงานในคำสั่งนั้น

นอกจากนั้นการเลือกใช้คำสั่งจากเมนู อาจจะได้จากการกดคีย์พิเศษต่างๆ ตามที่กำหนดไว้เช่น กดคีย์ ^KS (กดแป้นพิมพ์คอนโทรลตามด้วยตัวอักษร K และ D เข้าด้วยกัน) แทนการเลื่อนแถบทึบด้วยลูกศร ไปที่เมนูจัดเก็บแฟ้มข้อมูล หรือกดคีย์ ALT-X (กดแป้นพิมพ์ Alt กับ X ด้วยกัน) แทนการเลื่อนแถบทึบด้วยลูกศรไปที่เมนูเลิกการทำงาน เป็นต้น

การแสดงของเมนู จะมีเป็นระดับต่างๆกันคือ ระดับแรกจะเป็นการแสดงเมนูหลัก โดยในแต่ละเมนูหลักจะมีการแสดงเป็นเมนูย่อยๆ ซึ่งในที่นี้จะเรียกว่า "เมนูย่อยระดับที่หนึ่ง" และในแต่ละเมนูย่อยระดับที่หนึ่ง อาจจะมีการแสดงเป็นเมนูย่อยๆ ได้อีก ซึ่งในที่นี้จะเรียกว่า "เมนูย่อยระดับที่สอง" เช่น เมื่อเราเลือกเมนูหลักไปที่ "เคลื่อนที่" ก็จะปรากฏเมนูย่อยระดับที่หนึ่งให้เลือกอีก 7 เมนู คือ เลื่อนตำแหน่งเลื่อนจอภาพ ไปยังตำแหน่งเริ่มต้น ไปยังตำแหน่งสุดท้าย ไปยังตำแหน่ง TAB ไปหน้าที่ และไปบรรทัดที่ ซึ่งถ้าสมมุติว่าเลือกเมนูเลื่อนตำแหน่งของเมนูย่อยระดับที่หนึ่ง ก็จะปรากฏเมนูย่อยระดับที่สองให้เลือกอีก 6 เมนู คือ ไปทางซ้าย ไปทางขวา เลื่อนขึ้นบน เลื่อนลงล่าง ไปคำถัดไป และ ไปคำที่แล้ว เป็นต้น

การทำงานของพลูดาวน์เมนู จะใช้ข้อมูลในตัวแปรที่สำคัญ ดังนี้คือ

1. curmenu

เก็บค่าของเมนูที่เลือกได้ขณะนั้น เพื่อจะได้นำไปใช้งานต่อไป โดยค่าของตัวแปรนี้จะใช้เป็นตัวเลข Hex (ฐานสิบหก) ขนาด 2 ไบต์ ใช้ตัวเลขทั้งหมด 4 หลัก โดยแต่ละหลักจะมีความหมายดังนี้คือ

โดย เลขหลัก 1 แทนค่าเมนูที่เลือกได้ในเมนูหลัก
เลขหลัก 2 แทนค่าเมนูที่เลือกได้ในเมนูย่อยระดับที่หนึ่ง
เลขหลัก 3 แทนค่าเมนูที่เลือกได้ในเมนูย่อยระดับที่สอง
เลขหลัก 4 จะเป็นค่าศูนย์เสมอ

เช่น ค่าของ curmenu เป็น 0x2430 แสดงว่า เป็นการเลือกเมนูหลักที่ลำดับ 2 (ในที่นี้จะตรงกับเมนูเคลื่อนที่) และเลือกเมนูย่อยระดับที่หนึ่ง ในลำดับ 4 (ในที่นี้จะตรงกับเมนูไปยังตำแหน่งสุดท้าย) รวมทั้งยังมีการเลือกเมนูย่อยระดับที่สอง ในลำดับ 3 (ในที่นี้จะตรงกับเมนูไปท้ายแฟ้มข้อมูล)

2. command_key

เป็นตารางที่ใช้เก็บข้อมูล การแปลงค่าคีย์พิเศษให้กลายเป็นค่าของเมนู เพราะการใช้คำสั่งจากเมนู อาจจะทำได้จากการกดคีย์พิเศษที่กำหนดไว้ก็ได้ จึงต้องมีการนำค่าของคีย์ไปค้นหาค่าของเมนูในตาราง โดยข้อมูลจะแบ่งออกเป็นชุดๆ แต่ละชุดจะประกอบด้วยข้อมูลสำคัญ 4 อย่างดังนี้คือ

2.1 ค่าของคีย์แรก เป็นค่าของคีย์พิเศษตัวแรก ซึ่งได้ถูกกำหนดไว้ในตารางภาคผนวกแล้ว สามารถเลือกมาใช้งานได้ เช่น ค่า ALTX มีค่าเป็น 0x02D0 หรือ ค่า CNTRL_K มีค่าเป็น 0x0250B เป็นต้น ซึ่งค่าของคีย์แรกนี้อาจจะมีการใช้คีย์ที่สองร่วมด้วยก็ได้ และถ้าในกรณีไม่มีการใช้คีย์พิเศษใดๆ ให้ใส่ค่าของคีย์เป็นศูนย์

2.2 ค่าของคีย์ตัวที่สอง เป็นค่าของคีย์ที่จะใช้ร่วมกับคีย์พิเศษตัวแรก ซึ่งอาจจะเป็นตัวอักษรหรือเป็นคีย์พิเศษอีกก็ได้ เช่น ค่าคีย์พิเศษตัวแรกเป็น CNTRL_K แล้ว คีย์ตัวที่สองอาจจะเป็นตัวอักษร S ก็แสดงว่ากดคีย์ ^KS เป็นต้น และถ้าในกรณีไม่มีการใช้คีย์ใดๆ ให้ใส่ค่าของคีย์เป็นศูนย์

2.3 ค่าของเมนู เป็นค่าของเมนูที่จะเลือกได้ ซึ่งจะเป็นข้อมูลเดียวกันกับ curmenu

2.4 ค่าสภาวะบอกเมนูย่อย เป็นค่าที่แสดงถึงการมีเมนูย่อยได้อีกหรือไม่ โดยถ้ามีค่า 0 แสดงว่า ไม่มีเมนูย่อยอีกแล้วแต่ถ้ามีค่า 1 แสดงว่า ยังมีเมนูย่อยต่อไปอีก เช่น ถ้าชุดของข้อมูลเป็น ALTX, 0, 0x1800, 0 แสดงว่า เมื่อเรากดคีย์ ALTX ลงไปคีย์เดียวแล้ว จะมีผลทำให้ได้ค่าเมนูเป็น 0x1800 และไม่มีเมนูย่อย หรือ ชุดของข้อมูลเป็น CNTRL_K , 0x0053, 0x1300, 0 แสดงว่าเราจะต้องกดคีย์พิเศษ Ctrl + K ร่วมกับคีย์อักษรตัวที่สองคือ S (มีค่าเท่ากับ 0x0053) แล้วจะมีผลทำให้ได้ค่าเมนูเป็น 0x1300 หรือ ชุดของข้อมูลเป็น 0, 0, 0x2100, 1 แสดงว่า ไม่ได้กำหนดคีย์พิเศษไว้ใช้แทนได้ แต่ถ้าเลือกเมนูนั้นแล้วได้ค่าเป็น 0x2100 แล้ว จะมีเมนูย่อย ให้เลือกเมนูต่อไปได้อีก

3. mainmenu xlev0 ylev0 nolev0

เป็นตัวแปรที่จะใช้เกี่ยวข้องกับการแสดงเมนูหลักโดยตัวแปร mainmenu จะใช้เก็บข้อความที่ใช้แสดงเมนูหลัก ตัวแปร xlev0 และ ylev0 จะใช้เก็บตำแหน่งที่ใช้แสดงเมนูหลัก ตัวแปร nolev0 ใช้เก็บจำนวนเมนูของเมนูหลัก

4. msglev1 xlev1 ylev1 nolev1

เป็นเป็นตัวแปรที่จะใช้เกี่ยวข้องกับการแสดงเมนูย่อยระดับ 1 โดยตัวแปร msglev1 จะใช้เก็บข้อความที่ใช้แสดงเมนูย่อยระดับ 1 ตัวแปร xlev1 และตัวแปร ylev1 จะใช้เก็บตำแหน่งที่ใช้แสดงเมนูย่อยระดับ 1 ตัวแปร nolev1 ใช้เก็บจำนวนเมนูของเมนูย่อยระดับ 1

5. msglev2 xlev2 ylev2 nolev2

เป็นเป็นตัวแปรที่จะใช้เกี่ยวข้องกับการแสดงเมนูย่อยระดับ 2 โดยตัวแปร msglev2 จะใช้เก็บข้อความที่ใช้แสดงเมนูย่อยระดับ 2 ตัวแปร xlev2 และตัวแปร ylev2 จะใช้เก็บตำแหน่งที่ใช้แสดงเมนูย่อยระดับ 1 ตัวแปร nolev2 ใช้เก็บจำนวนเมนูของเมนูย่อยระดับ 2

สำหรับการทำงานของพลูดาวน์เมนูจะอยู่ภายใต้โมดูล "pulled_down_menu" โดยจะต้องกำหนดตัวแปร curmenu สำหรับใช้เก็บค่าของเมนูที่เลือกไว้

การทำงาน จะมีขั้นตอนดังนี้คือ

1. ไปทำงานในโมดูล "findcurlevel" เพื่อที่จะแยกค่าของเมนู curmenu ออกไปเป็นค่าลำดับที่ เลือกได้จากเมนู แยกออกไปเก็บไว้ในตัวแปรของแต่ละระดับ คือ ตัวแปร lev1, lev2, lev3 และ lev4 เช่น ถ้าค่าของ curmenu เป็น 0x2130 จะแยกออกเป็นค่าลำดับที่ของเมนู เก็บไว้ในตัวแปร lev1 เป็นค่า 2 ตัวแปร lev2 เก็บเป็นค่า 1 ตัวแปร lev3 เก็บเป็นค่า 3 และ ตัวแปร lev4 เป็นค่า 0

2. แสดงข้อมูลในตัวแปร mainmenu ที่เป็นข้อความของเมนูหลักทั้งหมดบนจอภาพ ตามจำนวนค่าที่เก็บไว้ในตัวแปร nolev0 โดยแสดงในตำแหน่งที่กำหนดไว้ในค่าของ xlev0 และ ylev0 รวมทั้งจะแสดงหัวข้อของเมนูหลักตัวแรก ด้วยลักษณะของ Reverse (พื้นขาวตัวอักษรดำ) เพื่อให้เกิดเป็นลักษณะของแถบทึบขึ้นมา

3. เริ่มต้นทำงานซ้ำๆกัน ดังนี้คือ

3.1 ทำงานในโมดูล "savescrn" เพื่อทำการเก็บข้อความบางส่วนบนจอภาพในบริเวณที่จะใช้แสดงเมนูย่อยระดับที่ 1

3.2 ทำงานในโมดูล "selectmenu" เพื่อให้เกิดการแสดงเมนูย่อยระดับที่ 1 และเลือกคำสั่งจากเมนูย่อยนั้นได้ โดยจะให้ผลเป็นค่าของเมนูที่เลือกได้เก็บไว้ใน curmenu และค่าของคีย์ที่ใช้ในเมนู เก็บไว้ใน key

3.3 ตรวจสอบค่าของคีย์ที่ได้ แล้วแยกกรณีการทำงานดังนี้คือ

3.3.1 คีย์ ESCKEY จะเลิกทำงานจากโปรแกรม

3.3.2 คีย์ลูกศรซ้าย (CNTRL_S หรือ LEKEY) จะทำงานดังนี้คือ

- ทำงานในโมดูล "resscrn" เพื่อนำข้อความที่เก็บไว้ มาแสดงกลับคืนบนจอภาพอีกครั้ง เป็นการลบการแสดงเมนูย่อยระดับที่ 1
- ตรวจสอบค่า lev1 ถ้า lev1 ไม่เป็นค่า 1 ให้ลดค่าของ lev1 อีก 1 แต่ถ้าค่าของ lev1 เป็น 1 พอดี จะกำหนดให้ค่าของ lev1 เท่ากับค่าจำนวนเมนูหลักทั้งหมด ที่เก็บไว้ใน nolev0
- แสดงหัวข้อของเมนูหลักตัวก่อนหน้าด้วยลักษณะของ Reverse (พื้นขาวตัวอักษรดำ) ให้เห็นเป็นแถบทึบ

3.3.3 คีย์ลูกศรขวา (CNTRL_D หรือ RIKEY) จะทำงานดังนี้คือ

- ทำงานในโมดูล "resscrn" เพื่อนำข้อความที่เก็บไว้ มาแสดงกลับคืนบนจอภาพอีกครั้ง เป็นการลบการแสดงเมนูย่อยระดับที่ 1

- ตรวจสอบค่า levl ถ้า levl ไม่เป็นค่าของจำนวนเมนูหลักทั้งหมดให้เพิ่มค่าของ levl อีก 1 แต่ถ้าค่าของ levl เป็นค่าของจำนวนเมนูหลักทั้งหมดพอดี ก็จะกำหนดให้ค่าของ levl เท่ากับค่า 1
- แสดงหัวข้อของเมนูหลัก ตัวถัดไปด้วยลักษณะของ Reverse (พื้นขาวตัวอักษรดำ) ให้เห็นเป็นแถบทึบ

3.3.4 คีย์อื่นๆ ซึ่งได้แก่ คีย์ RETKEY หรือคีย์ CNTRL_M และคีย์พิเศษต่างๆ ก็จะทำงานดังนี้คือ

- ไปทำงานในโมดูล "menu_to_sub" ของค่า curmenu เพื่อจะตรวจสอบดูว่าค่าของ curmenu นี้จะมีเมนูย่อยระดับที่ 2 อีกหรือไม่ โดยจะเข้าไปค้นหาข้อมูลในตาราง command_key โดยถ้าไม่มีเมนูย่อยระดับ 2 แล้วจะเลิกทำงานไป แต่ถ้ามีเมนูย่อยระดับ 2 แล้ว จะทำงานตามขั้นตอนดังนี้คือ

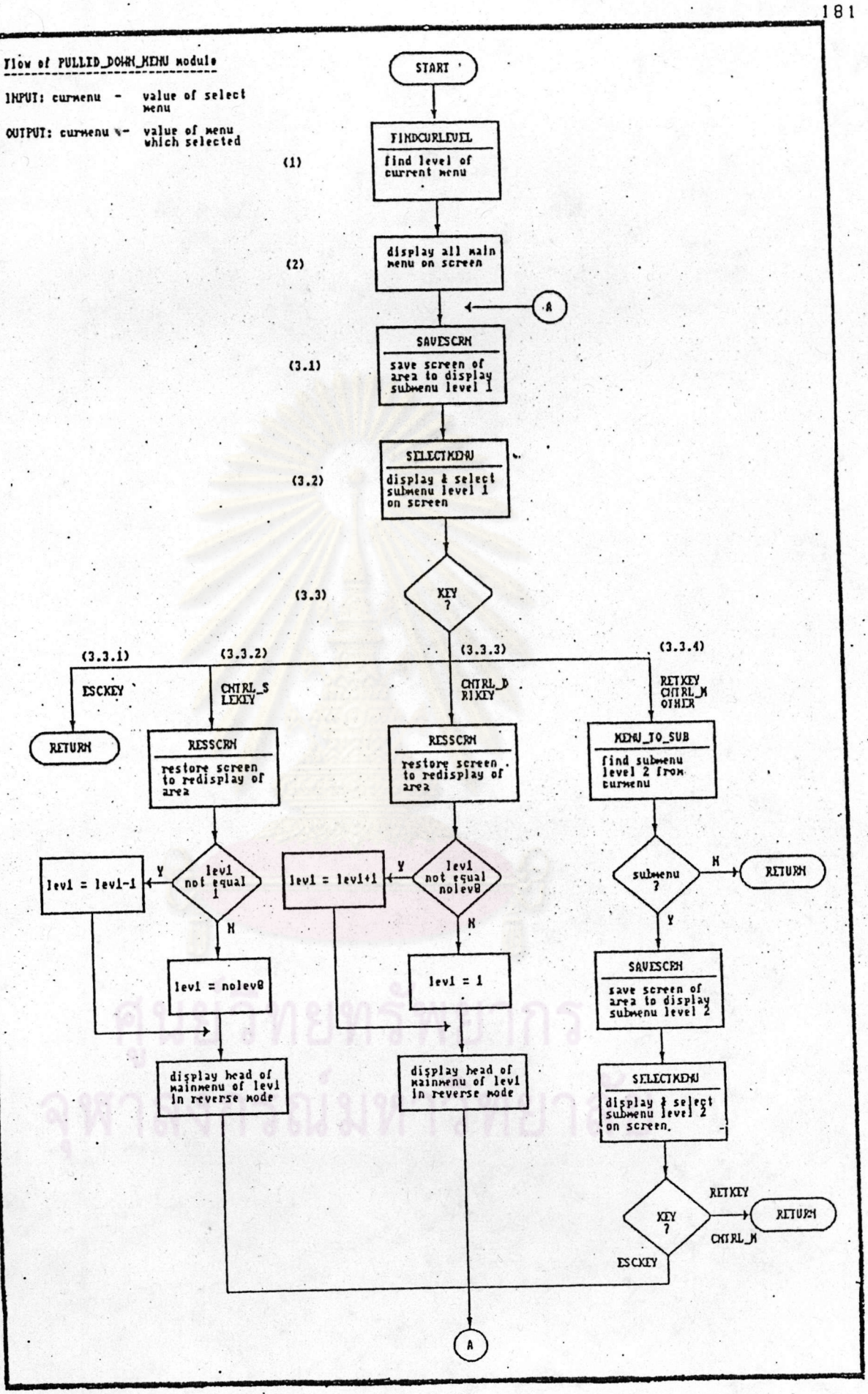
  - ทำงานในโมดูล "savescrn" เพื่อทำการเก็บข้อความบางส่วนบนจอภาพในบริเวณที่จะแสดงเมนูย่อยระดับที่ 2

  - ทำงานในโมดูล "selectmenu" เพื่อให้เกิดการแสดงเมนูย่อยระดับที่ 2 และเลือกคำสั่งจากเมนูย่อยนั้นได้ โดยจะให้ผลเป็นค่าของเมนูที่เลือกได้เก็บไว้ใน curmenu และค่าของคีย์ที่ใช้ในเมนูเก็บไว้ใน key

  - ตรวจสอบค่าของคีย์ที่ใช้ถ้าเป็น ESCKEY จะกลับไปทำงานในหัวข้อ 3.4 ต่อไป แต่ถ้าเป็น RETKEY หรือ CNTRL_M แล้ว ก็จะเลิกทำงาน โดยจะให้ค่าเมนูที่เลือกได้เก็บไว้ใน curmenu

3.4 ทำงานในโมดูล "resscrn" เพื่อนำเพื่อนำข้อความที่เก็บไว้ มาแสดงกลับคืนบนจอภาพอีกครั้ง เป็นการลบการแสดงเมนูย่อยระดับที่ 1 และเมนูย่อยระดับที่ 2 ด้วย

สหรับขั้นตอนการทำงานได้แสดงเป็นผังงาน ไว้ในรูปภาพที่ 4.80



รูปภาพที่ 4.80 ผังงานแสดงขั้นตอนการทำงานของโมดูล pulled_down_menu

4.3.23 สรุปโมดูลย่อยในระบบการประมวลผลคำของโปรแกรมซียูไรท์เตอร์

โมดูลย่อยที่สำคัญต่างๆ มีได้ดังนี้คือ

setupnode	- จะใช้สำหรับจัดเตรียมโหนดปัฟเฟอร์ และปัฟเฟอร์ทำงาน เป็นครั้งแรกของการทำงาน
insert_line	- จะใช้สำหรับแทรกบรรทัดข้อมูลใหม่ในระหว่างบรรทัดทั้งสอง
insert_char	- จะใช้สำหรับแทรกตัวอักษร 1 ตัว ในปัฟเฟอร์ทำงานของบรรทัดที่ใช้งานอยู่ขณะนั้น
ovrwrite_char	- จะใช้สำหรับแทนที่ตัวอักษร 1 ตัว ในปัฟเฟอร์ทำงานของบรรทัดที่ใช้งานอยู่ขณะนั้น
shiftscrn	- จะใช้สำหรับขยับจอภาพไปทางขวามือ เมื่อพิมพ์ข้อความไปจนเกินตำแหน่งขวาสุดของจอภาพ
insert_ret	- จะใช้สำหรับตัดข้อความ ในตำแหน่งที่เคอร์เซอร์อยู่ให้เลื่อนลงไปขึ้นบรรทัดใหม่อีกบรรทัดหนึ่ง
returnkey	- ทำให้ตัวเคอร์เซอร์เลื่อนลงไปขึ้นบรรทัดใหม่อีกบรรทัดหนึ่ง โดยอยู่ในคอลัมน์แรกของบรรทัด
inscntrl	- เป็นการแทรกรหัสควบคุมตัวอักษร ลงไปในปัฟเฟอร์ทำงานของบรรทัดที่ใช้งานอยู่ขณะนั้น
editmacro	- ใช้กำหนดข้อความที่ต้องการไปเก็บไว้ในตัวแปร macro
dispmacro	- ใช้แสดงข้อความคำอธิบายของคีย์ Ctrl ต่างๆ
insertmacro	- ใช้เรียกข้อความที่ได้เก็บไว้ในตัวแปร macro มาแทรกไว้ในระหว่างตัวอักษรที่ตำแหน่งขณะนั้น
deleteline	- เป็นการลบข้อความในโหนดปัฟเฟอร์ของบรรทัดที่ระบุไว้ให้หายไป

del_return - ใช้ลบรหัสรีเทริ์นของบรรทัด เพื่อจะนำข้อความในบรรทัดถัดไปมาเชื่อมต่อกับบรรทัดขณะนั้น

delete_char - เป็นการลบตัวอักษร 1 ตัว ในบัฟเฟอร์ทำงานของบรรทัดที่ใช้งานอยู่ขณะนั้น โดยจะเป็นการลบตัวอักษรในตำแหน่งที่เคอร์เซอร์อยู่

backspace - เป็นการลบตัวอักษร 1 ตัว ในบัฟเฟอร์ทำงานของบรรทัดที่ใช้งานอยู่ขณะนั้น โดยจะเป็นการลบตัวอักษรในตำแหน่งที่อยู่ทางซ้ายของเคอร์เซอร์อยู่

delete_word - ใช้ในการลบข้อความเป็นคำ ในบัฟเฟอร์ทำงานของบรรทัดที่ใช้งานอยู่ขณะนั้น โดยจะเป็นการลบข้อความในตำแหน่งที่เคอร์เซอร์อยู่

deltoendline - ใช้ในการลบข้อความจากตำแหน่ง ที่เคอร์เซอร์อยู่ไปจนหมดข้อความของบรรทัดที่ใช้งานอยู่

createdir - สร้างไดเรคทอรีลิสต์เพื่อเก็บรายชื่อไฟล์ต่างๆไว้ใช้งาน

showpagedir - เป็นการนำชื่อไฟล์ในไดเรคทอรีลิสต์มาแสดงบนจอภาพ

dirpgup dirpgdn
dirup dirdown
dirright dirleft - เป็นการเคลื่อนที่ผ่านไปตามไดเรคทอรีลิสต์ โดยกดคีย์ลูกศรไปตามทิศทางต่างๆกันได้ เพื่อเลือกชื่อไฟล์ในไดเรคทอรีลิสต์

freedir - ทำให้รายชื่อไฟล์ต่างๆทั้งหมด ที่อยู่ในไดเรคทอรีลิสต์หายไป โดยการปลดปล่อยไดเรคทอริโหนดให้เป็นอิสระ

havewild - ใช้ตรวจสอบว่าชื่อไฟล์ที่ระบุไว้ มีการเครื่องหมาย wild card หรือไม่ เช่น เครื่องหมาย * หรือ ? เป็นต้น

selectfile - สำหรับให้มีการเลือกใช้ชื่อไฟล์แบบฟลูสกรีนไดเรคทอรี

findlineno	- ใช้สำหรับนับบรรทัดของโหนดพอยน์เตอร์ จากโหนดพอยน์เตอร์ตัวเริ่มต้น จนถึงโหนดพอยน์เตอร์ที่กำหนดไว้ เพื่อให้ได้ผลเป็นหมายเลขบรรทัดที่ต้องการ
line_node	- ใช้สำหรับค้นหาโหนดพอยน์เตอร์ที่ต้องการ จากการนับบรรทัดเป็นจำนวนตามที่กำหนดไว้
loadtoline	- เป็นการนำข้อมูลจากโหนดบัฟเฟอร์ ที่ถูกชี้ด้วยโหนดพอยน์เตอร์ขณะนั้น มาแยกเก็บไว้ในบัฟเฟอร์ทำงานทั้ง 4 ระดับ
storeline	- เป็นการนำข้อมูลจากบัฟเฟอร์ทั้ง 4 ระดับ เก็บรวมไว้เป็นบรรทัดเดียวกันในโหนดบัฟเฟอร์ที่ใช้งานอยู่
displine	- ใช้สำหรับแสดงข้อความ ในโหนดบัฟเฟอร์ที่ใช้งานอยู่เป็นจำนวน 1 บรรทัด
refreshline	- ใช้สำหรับแสดงข้อความในบัฟเฟอร์ทำงานที่ใช้งานอยู่เป็นจำนวน 1 บรรทัด
disppagebreak	- ใช้แสดงเส้นแบ่งหน้าบนจอภาพ เมื่อแสดงข้อความครบหนึ่งหน้า
showpage	- ใช้ในการแสดงข้อความ ในโหนดบัฟเฟอร์บนจอภาพเป็นจำนวน 1 หน้า
markbegin	- ใช้บอกตำแหน่งเริ่มต้นของบล็อก
markend	- ใช้บอกตำแหน่งสุดท้ายของบล็อก
haveblock	- ตรวจสอบว่า ขณะนี้ได้มีการกำหนดบล็อกขึ้นมาหรือไม่
inblock	- ตรวจสอบว่า ตำแหน่งบรรทัดและคอลัมน์ที่กำหนดไว้นั้นอยู่ภายในบล็อกหรือไม่
toggleblk	- ใช้สำหรับการให้มีหรือยกเลิกความเป็นบล็อก
copytospace	- เป็นการคัดลอกข้อมูลจากโหนดบัฟเฟอร์ของบรรทัดและคอลมน์เริ่มต้นไปจนถึงบรรทัดและคอลัมน์สุดท้ายที่กำหนดไว้

insertlinklist	- เป็นการนำชุดของบรรทัดข้อมูลที่กำหนดไว้มาเชื่อมต่อกันกับบรรทัดข้อมูลผลลัพธ์ที่ใช้งานอยู่ขณะนั้น
deletelinklist	- เป็นการลบชุดของบรรทัดข้อมูลที่ต้องการ จากชุดของบรรทัดข้อมูลทั้งหมด ที่ใช้งานอยู่ขณะนั้น
rdfiletospace	- ใช้อ่านข้อมูลจากไฟล์ที่ได้ระบุชื่อไว้ มาเก็บไว้ในโหนดบัฟเฟอร์ที่เตรียมไว้
abandonfile	- ใช้ในการยกเลิกข้อมูลที่เก็บไว้ในโหนดบัฟเฟอร์ทั้งหมด โดยการปลดปล่อยให้เป็นอิสระทั้งหมด
writeblock	- จะเป็นการนำข้อมูลจากโหนดบัฟเฟอร์มาเขียนเก็บไว้อย่างถาวรในไฟล์ข้อมูล
copyblk	- คัดลอกข้อมูลในบล็อกที่กำหนดไว้ ไปยังตำแหน่งอื่นๆ ของบรรทัดนั้นๆ
moveblk	- เคลื่อนย้ายข้อมูลในบล็อกที่กำหนดไว้ไปยังตำแหน่งอื่นๆ ของบรรทัดขณะนั้นได้
deleteblk	- ลบข้อมูลในบล็อกที่กำหนดไว้ให้หายไป จากที่ใช้งานอยู่
spacesize	- คำนวณหาขนาดเนื้อที่หน่วยความจำที่เหลืออยู่
getfilesize	- คำนวณหาขนาดเนื้อที่ของไฟล์ที่กำหนดไว้
diskfree	- คำนวณหาขนาดเนื้อที่ว่างในแผ่นจานบันทึก
goline	- เคลื่อนที่ไปยังบรรทัดที่ระบุหมายเลขบรรทัดไว้
gocol	- เคลื่อนที่ไปยังคอลัมน์ที่ระบุหมายเลขคอลัมน์ไว้
home	- เคลื่อนที่ไปยังต้นบรรทัดหรือคอลัมน์แรกของบรรทัด
endline	- เคลื่อนที่ไปยังท้ายบรรทัดหรือคอลัมน์สุดท้ายของบรรทัด
cursor_up	- เคลื่อนที่ของเคอร์เซอร์ขึ้นไปอีก 1 บรรทัด
cursor_down	- เคลื่อนที่ของเคอร์เซอร์ลงไปอีก 1 บรรทัด
cursor_left	- เคลื่อนที่ของเคอร์เซอร์ไปทางซ้าย 1 ตัวอักษร
cursor_right	- เคลื่อนที่ของเคอร์เซอร์ไปทางขวา 1 ตัวอักษร

page_up - เคลื่อนที่ของบรรทัดบนจอภาพขึ้นไป 1 หน้า

page_down - เคลื่อนที่ของบรรทัดบนจอภาพลงมา 1 หน้า

scroll_up - เคลื่อนที่ของบรรทัดบนจอภาพขึ้นไป 1 บรรทัด

scroll_down - เคลื่อนที่ของบรรทัดบนจอภาพลงมา 1 บรรทัด

topfile - เคลื่อนที่ของเคอร์เซอร์ไปตำแหน่งบรรทัดแรกสุดของข้อมูลทั้งหมด

endfile - เคลื่อนที่ของเคอร์เซอร์ไปตำแหน่งบรรทัดสุดท้ายของข้อมูลทั้งหมด

backword - เคลื่อนที่ของเคอร์เซอร์ไปทางซ้ายทีละคำ

nextword - เคลื่อนที่ของเคอร์เซอร์ไปทางขวาทีละคำ

gotopage - เคลื่อนที่ไปหน้าที่ต้องการ

gobeginblk - เคลื่อนที่ไปที่ต้นบล็อก

goendblk - เคลื่อนที่ไปที่ท้ายบล็อก

searchonlyinfo - แสดงข้อความบนจอภาพเพื่อจะใช้รับข้อความจากแป้นพิมพ์ ที่จะใช้ค้นหามาเก็บไว้ในตัวแปร source และวิธีที่จะต้องการค้นหามาเก็บไว้ในตัวแปร option

searchreplaceinfo - แสดงข้อความบนจอภาพ เพื่อจะใช้รับข้อความจากแป้นพิมพ์ ที่จะใช้ค้นหามาเก็บไว้ในตัวแปร source และข้อความที่จะใช้แทนที่ มาเก็บไว้ในตัวแปร replace รวมทั้งวิธีที่จะต้องการค้นหามาเก็บไว้ในตัวแปร option

optionglobal optionupper

optionword optionnoask

- ตรวจลอบค่าในตัวแปร option ว่ามีการระบุวิธีการค้นหาเหล่านั้นบ้างหรือไม่

searchline - ค้นหาข้อความที่ต้องการในบรรทัดที่ใช้งานอยู่ 1 บรรทัด

wordnotfound	- แสดงข้อความค้นหาไม่พบบนจอภาพ
searchfwd	- ค้นหาข้อความในแต่ละบรรทัด โดยนับตั้งแต่ตำแหน่งบรรทัดที่ใช้อยู่
addblank	- ใช้เพิ่มช่องว่าง 1 ตัวอักษร ไว้ในตำแหน่งแรกและสุดท้ายของข้อความที่จะค้นหาในตัวแปร source
searching	- ใช้ค้นหาข้อความที่ต้องการ ตั้งแต่บรรทัดที่ใช้งานอยู่
replaceword	- ใช้ในการแทนที่ข้อความใหม่ ลงในข้อความเดิมที่ต้องการ
searchreplace	- ค้นหาข้อความและแทนที่ด้วยข้อความใหม่ในแต่ละบรรทัด โดยนับตั้งแต่ตำแหน่งบรรทัดที่ใช้อยู่
replacing	- เป็นการค้นหาข้อความที่ต้องการ แล้วให้แทนที่ด้วยข้อความใหม่ที่ได้กำหนดไว้
findcurmenu	- ใช้หาค่าของเมนู curmenu จากค่าลำดับของแต่ละตัวแปรระดับเมนู
findcurlevel	- ใช้เพื่อที่จะแยกค่าของเมนู curmenu ออกไปเป็นค่าลำดับที่เลือกได้จากเมนู แยกออกไปเก็บไว้ในตัวแปรของแต่ละระดับเมนู
headmenu	- แสดงหัวข้อของเมนูหลัก ในลักษณะ revese คือพื้นขาวตัวอักษรดำ
selectmenu	- แสดงและเลือกเมนูย่อยบนจอภาพได้
key_to_menu	- ใช้เปลี่ยนค่าคีย์พิเศษให้กลายเป็นค่าของเมนู
menu_to_sub	- ใช้ตรวจสอบค่าของเมนู curmenu ว่าจะมีเมนูย่อยระดับ 2 หรือไม่
pulled_down_menu	- ใช้แสดงพลูดาวน์เมนู และให้เลือกคำสั่งจากเมนูได้