หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

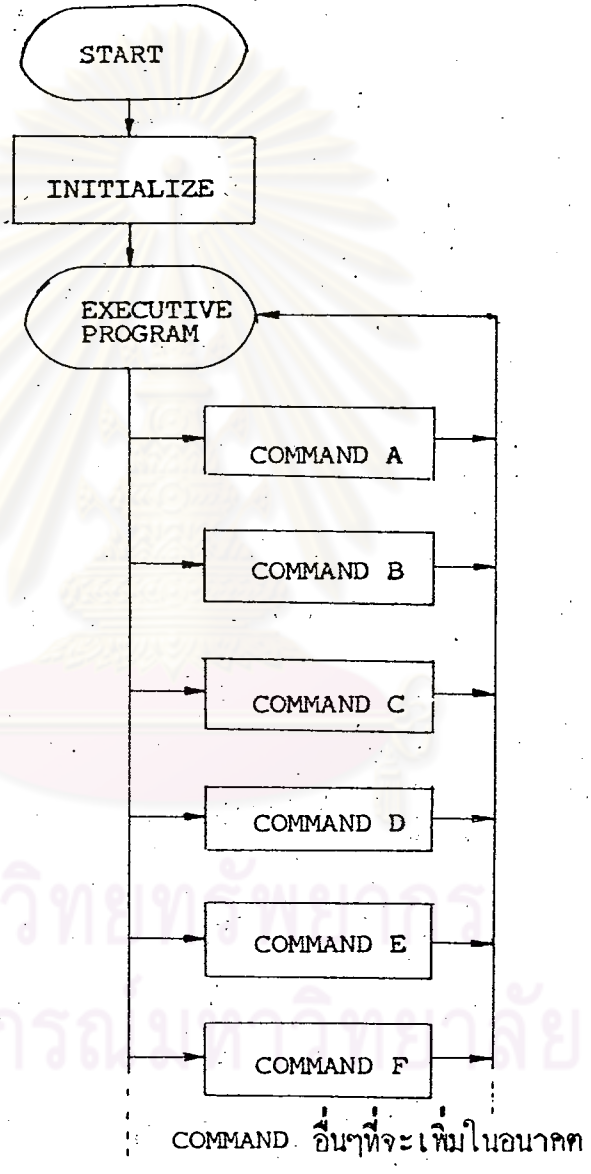
วิธีควบคุมการทำงานโดยใช้โปรแกรม

SOFTWARE ที่เขียนขึ้นนี้มีจุดประสงค์เพื่อให้ง่ายต่อการทำงาน สะดวกต่อการเปลี่ยนแปลงและเพิ่มเติมในภายหลัง โดยจะออกแบบเป็น SOFTWARE MODULE ต่างๆ แต่ละ MODULE อาจจะมีการใช้ COMMAND SUBROUTINE ร่วมกันหรือแยกกันต่างหากซึ่งจะกล่าวถึงต่อไป

SOFTWARE หลัก หรือ MODULE แรก ที่จะกล่าวถึงคือ EXECUTIVE MODULE โดยที่ SOFTWARE ส่วนนี้จะทำหน้าที่คอยตรวจสอบสภาวะของ อินพุท จากส่วนอินเตอร์เฟสทั้งสองคือจาก อินพุทที่มาจากเครื่องเทเล็กซ์และอินพุทที่มาจากชุมสายเทเล็กซ์ การตรวจสอบนี้จะยึดหลักว่า ถ้าสัญญาณเรียกมาจากตัวชุมสายเทเล็กซ์ก็หมายความว่า มีผู้ใช้เทเล็กซ์รายอื่นต้องการติดต่อกับเครื่องเทเล็กซ์ที่ตัวเครื่องนี้ต่ออยู่ ดังนั้นตัวเครื่องซึ่งกำลังทำงานภายใต้ EXECUTIVE PROGRAM นี้ก็จะไปเรียกโปรแกรมอีก MODULE หนึ่งมาใช้ ซึ่งจะเป็นโปรแกรมที่จะทำหน้าที่เชื่อมวงจรระหว่างชุมสายเทเล็กซ์เข้ากับเครื่องเทเล็กซ์ โดยโปรแกรมนี้จะคอยตรวจสอบว่า ถ้ามีการเลิกติดต่อเมื่อใด ตัวโปรแกรมนี้ก็จะคืนการควบคุมการทำงานกลับไปให้ EXECUTIVE PROGRAM ต่อไปใหม่

แต่ถ้าหากตัวเครื่องได้สัญญาณการเรียกมาจากตัวเครื่องเทเล็กซ์ ตัว EXECUTIVE PROGRAM ก็จะตีความหมายได้ว่าผู้ใช้เครื่องเทเล็กซ์ กำลังต้องการส่ง COMMAND มาให้ตัวเครื่องดังกล่าวเพื่อให้ทำงานอย่างใดอย่างหนึ่งตาม COMMAND ที่จะส่งมา ดังนั้นตัว EXECUTIVE PROGRAM ก็จะเตรียมตัวที่จะรับ COMMAND ที่ส่งมาจากเครื่องเทเล็กซ์ โดยที่ตัว EXECUTIVE PROGRAM นี้จะตรวจสอบ FORMAT ของ COMMAND ว่าถูกต้องหรือไม่และถ้า COMMAND ดังกล่าวต้องการ PARAMETER ตัวอื่นประกอบ ตัว COMMAND SUBROUTINE นี้ก็จะส่งคำขอไปให้ผู้ใช้ส่ง PARAMETER เพิ่มเติมมาให้ เมื่อโปรแกรมนี้ได้รับ

COMMAND และ PARAMETER ถูกต้องเรียบร้อยแล้ว ตัว COMMAND SUBROUTINE ก็จะไปทำการเรียก SUBROUTINEอื่นๆ ที่เกี่ยวข้องมาควบคุมการทำงานต่อไป หลังจากโปรแกรมต่างๆสิ้นสุดการทำงานแล้ว ก็จะคืนการควบคุมการทำงานกลับมาให้ EXECUTIVE PROGRAM อีกครั้งหนึ่งเพื่อควบคุมการทำงานต่อไป



รูปที่ 15 SOFTWARE SYSTEM FLOW CHART

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ลักษณะการทำงานตาม ROUTINE นี้ก็เพื่อที่จะทำการรีเซ็ทอุปกรณ์ต่างๆ และเซ็ทค่า PARAMETER ให้กับส่วนต่างๆของ HARDWARE เพื่อที่จะได้ให้ทำงานตามต้องการ ซึ่งการทำงานตาม ROUTINE จะรวมถึงการ CLEAR TABLE ต่างๆ ที่ใช้และ BUFFER AREA จึงจะแยกกล่าวดังนี้

1. เซ็ท 8251 ทั้ง 2 ตัวให้อยู่ในสภาพว่างพร้อมที่จะทำงาน และโปรแกรมให้มีการรับส่งข้อมูลขนาด 5 บิท, $1\frac{1}{2}$ BIT STOP ASYNCHRONOUS ความเร็วในการทำงาน 50 BAUD

จุดประสงค์ของการเซ็ทนี้เพื่อที่ให้ส่วน ไอโอ ที่จะใช้ติดต่อกับเครื่องเทเล็กซ์และชุมสายพร้อมที่จะทำงานในลักษณะของสัญญาณเทเล็กซ์ ที่ใช้ติดต่อกันอยู่

2. เซ็ท 8253 เพื่อให้ทำหน้าที่เป็น REAL TIME CLOCK โดยเซ็ทเคาว์เตอร์ตัวแรกและตัวที่ 2 เพื่อให้หารความถี่ 2 MHz ให้เหลือความถี่ 1 Hz เพื่อใช้เป็นสัญญาณไป อินเตอร์รัพท์ ซีพียู ต่อไป

จุดประสงค์ของการเซ็ทนี้เพื่อใช้ TIMER ให้เป็นเวลาจริง และเป็น SOFTWARE TIMER LOOP ให้กับโปรแกรมต่างๆ

3. CLEAR ABBREVIATE SELECTION TABLE ทำการ CLEAR พื้นที่ในหน่วยความจำซึ่งมีไว้สำหรับเก็บหมายเลขย่อ ให้ว่างพร้อมที่จะรับหมายเลขย่อ

จุดประสงค์ของการทำดังกล่าวเนื่องจากว่า เมื่อเราเปิดเครื่องขึ้นใหม่ ค่าต่างๆในหน่วยความจำเป็นค่าที่เราสามารถกำหนดได้แน่นอน จึงต้องทำการ CLEAR ให้พร้อมที่จะทำงาน

4. CLEAR BUFFER MEMORY การ CLEAR อันนี้ก็จะทำการจัดบัฟเฟอร์แต่ละ BLOCK ให้ LINK เข้ากับ VACANT BUFFER POOL พร้อมที่จะให้ผู้ใช้เก็บข้อความเทเล็กซ์ต่อไป

จุดประสงค์ของการทำดังกล่าวเนื่องจากเหตุผลเช่นเดียวกับข้อ 3

5. CLEAR BUFFER MEMORY TABLE ในการใช้ BUFFER MEMORY เพื่อเก็บข้อความเทเล็กซ์นั้น เราจำเป็นต้องใช้ TABLE ช่วยเพื่อบอกให้รู้ว่า MESSAGE ฉบับหมายเลขเท่าใด อยู่ที่ไหน ในการทำการ CLEAR นี้เพื่อที่จะได้รู้ว่าขณะนี้ไม่มีข้อความเทเล็กซ์ เก็บอยู่เลย

จุดประสงค์ของการทำดังกล่าวก็เช่นเดียวกับข้อ 4

6. CLEAR WAITING LIST TABLE TABLE นี้เราสร้างขึ้นมาเพื่อบ่งบอกว่าขณะนี้มีข้อความเทเล็กซ์ใดบ้างที่กำลังรอส่งอยู่ และการส่งนี้มีเงื่อนไขใดบ้าง เช่น เวลา เป็นต้น

จุดประสงค์ของการ CLEAR นี้เพื่อที่จะบ่งว่าขณะนี้ไม่มีข้อความเทเล็กซ์ใดอยู่ในระบบที่จะรอส่ง ดังแสดงในรูปที่ 16

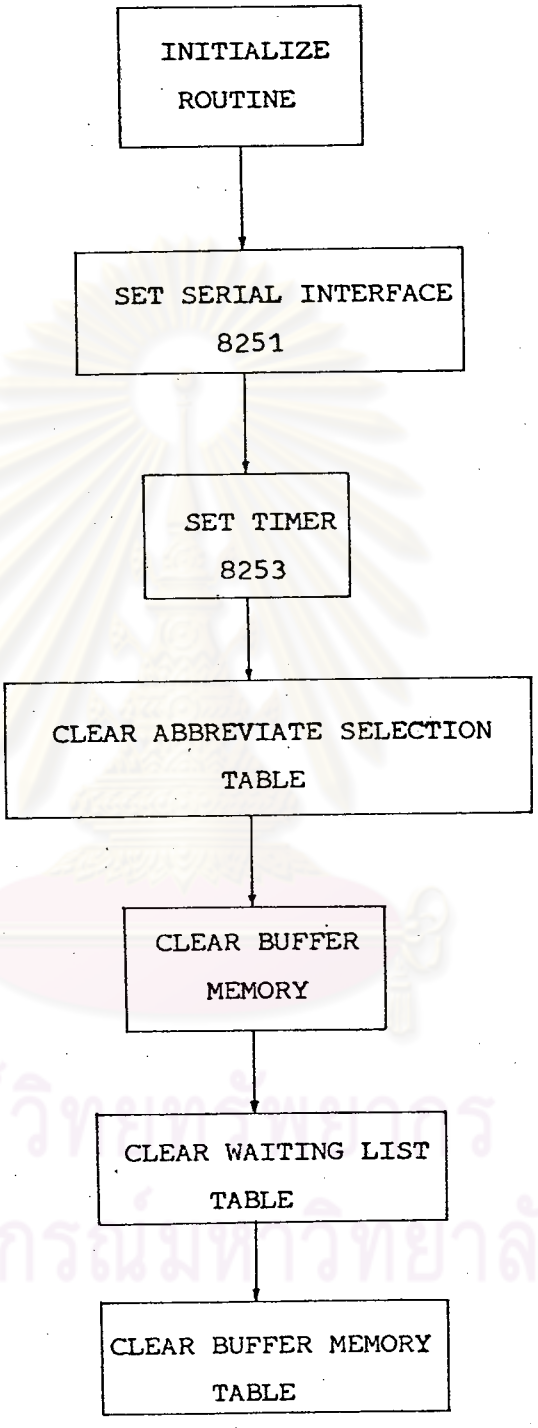
EXECUTIVE PROGRAM

EXECUTIVE PROGRAM นี้เป็น PROGRAM ที่สำคัญมากและถือได้ว่าเป็นหัวใจของการทำงาน เปรียบได้กับ OPERATING SYSTEM ของระบบใหญ่

การทำงานของ PROGRAM นี้จะเป็นการตรวจสภาวะต่างๆของระบบ ซึ่งจะทำการวน ตรวจสอบจนกว่าจะพบการเปลี่ยนแปลง และจะเริ่มทำงานต่อการเปลี่ยนแปลงสภาวะนั้นๆ

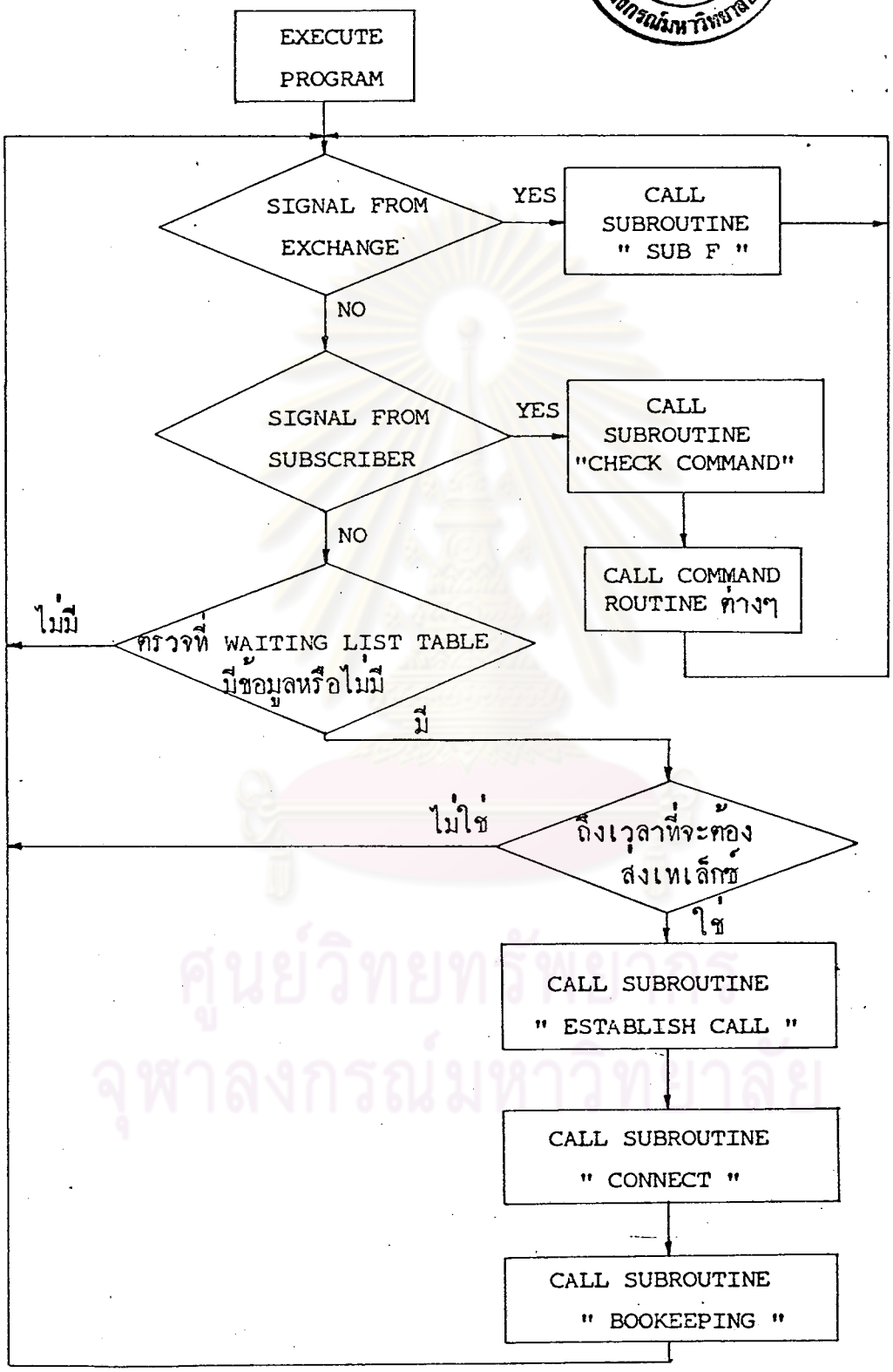
การตรวจสอบมี 3 สภาวะคือ

1. ตรวจสอบการเรียก ที่มาจากชุมสายเทเล็กซ์
2. ตรวจสอบการเรียก ที่มาจากผู้ใช้เครื่องเทเล็กซ์
3. ตรวจสอบเวลาจาก WAITING LIST TABLE



รูปที่ 16 INITIALIZE ROUTINE

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 17 SYSTEM FLOWCHART OF EXECUTIVE PROGRAM

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

I 15812996

การตรวจสอบอันแรกที่จะกล่าวถึงคือ การตรวจสอบสภาวะการเรียกที่มาจากชุมสายเทเล็กซ์ ซึ่งถ้า PROGRAM นี้ตรวจพบก็จะทำการต่อผ่านข้อมูลตรงระหว่างตัวชุมสายเข้ากับตัวเครื่องเทเล็กซ์ ด้วยการทำงานของ SUBROUTINE CONNECT ซึ่งจะกล่าวถึงต่อไป หลังจากการเรียกติดต่อนั้นจบลงแล้ว SUBROUTINE CONNECT ก็จะส่งกลับการทำงานมายัง EXECUTIVE PROGRAM ใหม่

การตรวจสอบต่อไปเป็นการตรวจสอบสภาวะการเรียกจากผู้ใช้เครื่องเทเล็กซ์ ซึ่งเมื่อตรวจสอบพบ ตัว PROGRAM นี้จะทำการเรียก SUBROUTINE CHECK COMMAND ซึ่งจะตรวจดูว่า COMMAND CODE ที่ส่งมาถูกต้องหรือไม่ เมื่อถูกต้องแล้วก็จะทำการเรียก SOFTWARE MODULE ต่างๆตาม COMMAND ที่ส่งมา ภายหลังจากที่ SOFTWARE MODULE ต่างๆทำงานเสร็จสิ้นแล้วก็คืนกลับการทำงานมายัง EXECUTIVE PROGRAM อีกครั้งหนึ่ง

การตรวจสอบอันสุดท้ายคือการตรวจสอบ WAITING LIST TABLE ดูว่ามีข้อความเทเล็กซ์ รอการส่งอยู่หรือไม่ ถ้ามีก็จะดูว่าเงื่อนไขสำหรับการส่งนั้นถูกต้องหรือไม่ ซึ่งเงื่อนไขดังกล่าวก็คือ เวลา ที่จะส่ง ถ้ายังไม่ถึงเวลาก็จะกลับไปยัง EXECUTIVE PROGRAM เพื่อตรวจสอบสภาวะอื่นต่อไป แต่ถ้าถึงเวลาแล้ว ก็จะเรียก SUBROUTINE ESTABLISH CALL ซึ่งจะทำการเรียกติดต่อระหว่างตัวเครื่องกับชุมสาย ถ้าหากไม่สามารถเรียกติดต่อได้ เช่นกรณีสายไม่ว่าง ตัวโปรแกรมก็จะตรวจดูเงื่อนไขและทำการ เซ็ทเงื่อนไขใหม่ เช่น เงื่อนไขเดิมบอกว่าให้เรียกใหม่ทุก 2 นาที ถ้าสายไม่ว่างตัวโปรแกรมก็จะทำการเซ็ทเงื่อนไขใหม่ให้ทำการเรียกในอีก 2 นาทีข้างหน้าเป็นต้น

แต่ถ้าหากทำการเรียกติดต่อได้สำเร็จตัวโปรแกรมก็จะไปทำการเรียก SUBROUTINE CONNECT เพื่อใช้ส่งข้อความติดต่อระหว่างกัน และหลังจากได้ส่งข้อความเสร็จสิ้นแล้วก็จะทำการเรียก SUBROUTINE BOOKEEPING ซึ่ง SUB ROUTINE นี้จะเซ็ทเงื่อนไขต่างๆ เช่น ข้อความดังกล่าวได้ส่งเสร็จสิ้นแล้ว ฯลฯ

ลักษณะของ COMMAND CODE

ลักษณะของ COMMAND CODE ที่เครื่องเทเล็กซ์จะส่งให้ตัว EXECUTIVE PROGRAM เพื่อการทำงานนั้นจะเป็นตัวอักษรตัวเดียว โดยที่ตัว EXECUTIVE PROGRAM จะส่งคำถามไปปรากฏบนเครื่อง เทเล็กซ์ว่า CMD ? เพื่อรับรู้การเรียก ซึ่งหมายความว่า ENTER COMMAND HERE ผู้ใช้ก็จะกดระหัส COMMAND ที่ต้องการโดยเป็นตัวอักษรเพียงตัวเดียว ตัว EXECUTIVE PROGRAM จะรับระหัสของคำสั่งนั้นแล้วดูว่าถูกต้องหรือไม่ ถ้าต้องการ PARAMETER เพิ่มก็จะส่งคำสั่งต่างๆ ไปให้ผู้ใช้เพิ่มเติม PARAMETER ต่างๆต่อไป ตามที่จะได้กล่าวถึงต่อไปนี้

COMMAND CODE ต่างๆ ที่ได้รับการออกแบบเพื่อให้เครื่องรับรู้และทำงานให้มีดังนี้

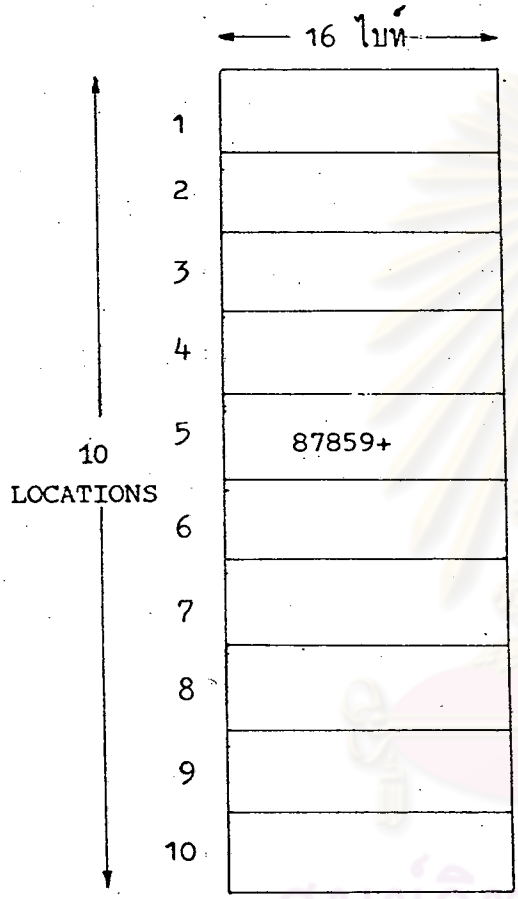
A	หมายถึง	ส่งข้อความเทเล็กซ์ตามหมายเลขย่อ
B	,,	กำหนดหรือเปลี่ยนแปลงหมายเลขย่อ
C	,,	ทำการเรียกติดต่อซ้ำเนื่องจากปลายทางสายไม่ว่าง
D	,,	เตรียมข้อความที่จะส่งออกเก็บไว้ใน บัฟเฟอร์ของตัวเครื่อง
E	,,	ส่งข้อความในบัฟเฟอร์ที่เตรียมไว้ ตามเวลาที่ต้องการ
F	,,	ทำการต่อตรงระหว่างเครื่องเทเล็กซ์กับชุมสายโดยไม่ใช้การทำงานของตัวเครื่องตามคำสั่งต่างๆ

ต่อไปนี้จะกล่าวถึงการทำงานของ SOFTWARE MODULEต่างๆตามที่จะได้รับมา

COMMAND A : ส่งข้อความเทเล็กซ์ตามหมายเลขย่อ

การทำงานตาม COMMAND นี้ตัวโปรแกรมจะมี MEMORY AREA อยู่แห่งหนึ่ง ซึ่งจะเป็น LOOKEEP TABLE โดยที่หมายเลขเต็มจะเก็บอยู่ในหน่วยความจำดังกล่าว

โครงสร้างของหน่วยความจำกังกล่าวจะมีลักษณะดังนี้



รูปที่ 18 ABBREVIATE TABLE

หน่วยความจำจะแบ่งออกเป็น 10 LOCATION LOCATION ละ 16 ไบท์ การใช้ TABLE ชี้จะเป็นตัวชี้ (INDEX) ด้วยหมายเลข ของเลขย่อ โดยที่หมายเลขย่อ " 0 " จะเป็น LOCATION ที่ 1 หมายเลขย่อ " 9 " จะเป็น LOCATION ที่ 10 ในแต่ละ LOCATION ซึ่งจะมีข้อมูลอยู่ 16 ไบท์ แต่ละไบท์ จะใช้เก็บหมายเลขเทเล็กซ์ 1 ตัว

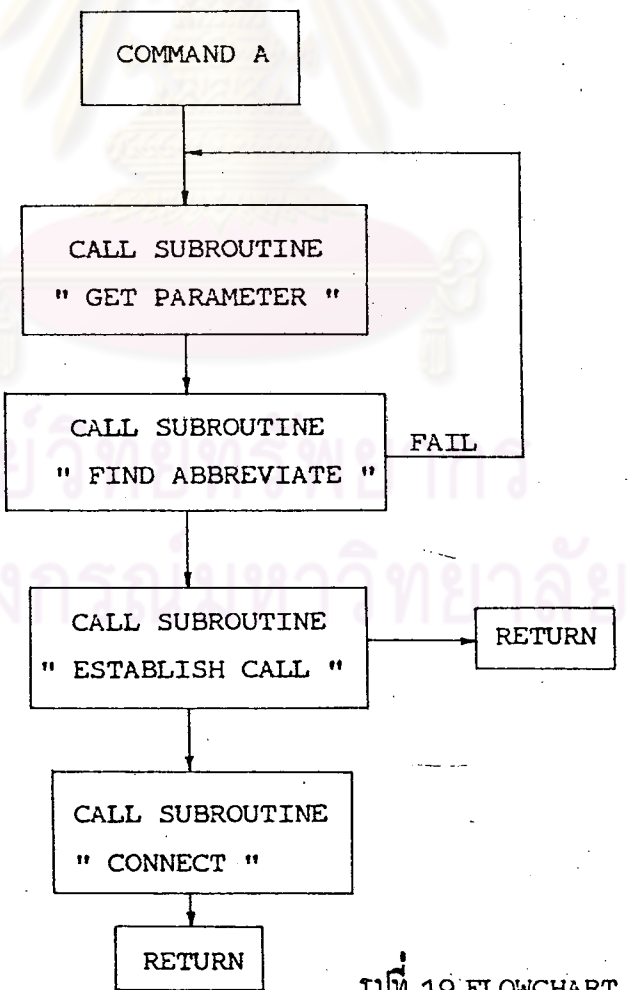
ดังตัวอย่าง LOCATION ที่ 5 จะเก็บหมายเลขเทเล็กซ์ 87859+ ซึ่งจะใช้ AREA ไป 6 ไบท์ การที่เลือกใช้ MEMORY AREA 16 ไบท์ ต่อ 1 LOCATION ก็เนื่องจากว่าหมายเลขเทเล็กซ์ ที่ใช้เรียกติดต่อนั้น ยอมให้มีหมายเลขสูงสุดได้ถึง 15 DIGIT (รวมทั้ง AREA CODE)

SOFTWARE MODULE ของ COMMAND นี้จะส่งคำขอ PARAMETER เพิ่มเติมกลับไปยังเครื่องเทเล็กซ์ให้ผู้ใช้ส่งหมายเลขย่อ ที่จะส่งข้อความเทเล็กซ์ออกไปมาให้ หลังจากผู้ใช้ได้ส่งหมายเลขย่อมาแล้วตัวโปรแกรมก็จะไปดูที่ หน่วยความจำตามตำแหน่งที่หมายเลขย่อนั้นบอกมา เพื่อหาหมายเลขเต็มที่จะใช้ในการส่งข้อความพร้อมกับส่งหมายเลขเต็มกลับมายังเครื่องเทเล็กซ์เพื่อตรวจสอบความถูกต้อง และทำการเรียก SUBROUTINE ESTABLISH CALL และ SUBROUTINE CONNECT

ขั้นตอนการเรียก

CMD ? A	ผู้ใช้เครื่องพิมพ์ตัว	A
? 2+	ผู้ใช้เครื่องพิมพ์เลข	2+
82567+	หมายเลขเต็มที่เครื่องเก็บไว้	

หลังจากที่ตัวเครื่องได้ทำการเรียกติดต่อเสร็จสิ้นแล้ว ก็จะคืนการควบคุมการทำงานกลับไปให้ EXECUTIVE PROGRAM ต่อไป



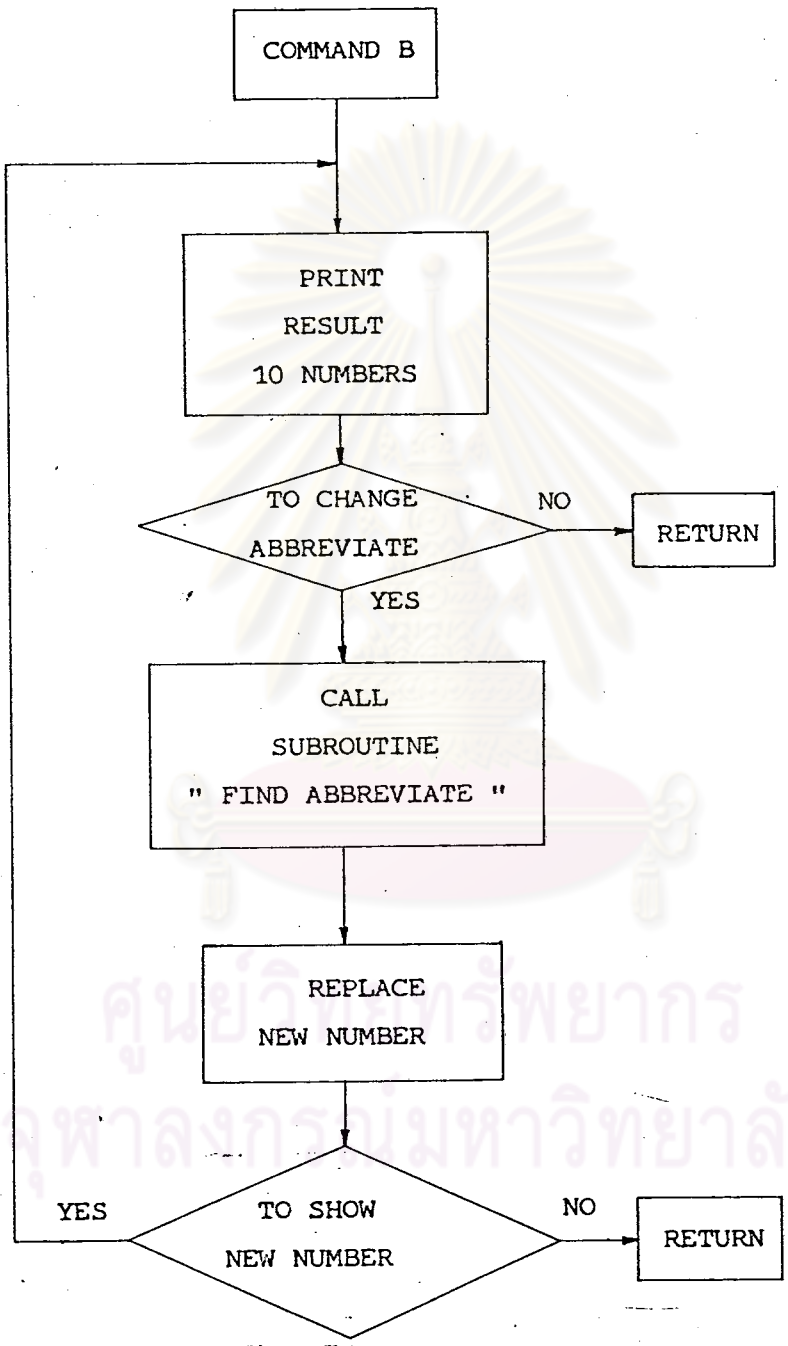
รูปที่ 19 FLOWCHART OF COMMAND A

COMMAND B : กำหนดหรือเปลี่ยนแปลงหมายเลขย่อรวมทั้งแสดงหมายเลขย่อที่เก็บไว้ด้วย

การทำงานตามคำสั่งนี้ หรือการเปลี่ยนแปลงหรือกำหนดหมายเลขย่อชิ้นใหม่ใน TABLE ที่จัดเตรียมไว้ตามที่ได้อธิบายในตอนที่กล่าวถึง COMMAND A SOFTWARE MODULE ของ COMMAND นี้จะทำงานคล้ายกันกับ COMMAND A แต่จะต่างกันที่ COMMAND A จะทำการเรียกติดต่อกับชุมสายด้วย ส่วน COMMAND B นี้จะเพียงแสดงหมายเลขย่อที่เก็บไว้หรือเปลี่ยนแปลงเท่านั้น

ลักษณะการทำงานเมื่อผู้ใช้กดแป้นอักษรตัว B เพื่อเรียกใช้โปรแกรมสำหรับ COMMAND ตัวเครื่องจะพิมพ์เลขย่อออกมาทั้ง 10 หมายเลข เรียงกัน ถ้าไม่ต้องการเปลี่ยนแปลงเลขย่อก็ให้กด (CR) เมื่อเครื่องส่งเครื่องหมายขอ PARAMETER เพิ่มเติมแล้วตัวโปรแกรมจะส่งคืนการทำงานให้ EXECUTIVE PROGRAM แต่ถ้าต้องการเปลี่ยนเลขย่อใหม่ ก็พิมพ์หมายเลขย่อที่ต้องการเปลี่ยนเมื่อเครื่องส่งเครื่องหมายขอ PARAMETER เพิ่มได้เลย และเมื่อเครื่องส่งเครื่องหมายขอ PARAMETER (?) มาอีกครั้ง ก็พิมพ์เลขที่ต้องการเปลี่ยนใหม่ รวมทั้งเครื่องหมายบวก (+) ได้ทันที

เครื่องจะส่งเครื่องหมาย ? อีกครั้ง ถ้าผู้ใช้ต้องการดูว่าเครื่องเปลี่ยนแปลงเลขย่อให้ถูกต้องหรือไม่ ก็ให้พิมพ์ ตัว `B´ แต่ถ้าไม่ต้องการดูก็ให้กด `CR´ ตัวโปรแกรมจะส่งคืนการทำงาน ให้แก่ EXECUTIVE PROGRAM ต่อไป



รูปที่ 20 FLOWCHART OF COMMAND B

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
CMD ? B

    0 = . . . . . .

    1 = . . . . . .

    2 = . . . . . .

    3 = . . . . . .


    9 = . . . . . .
```

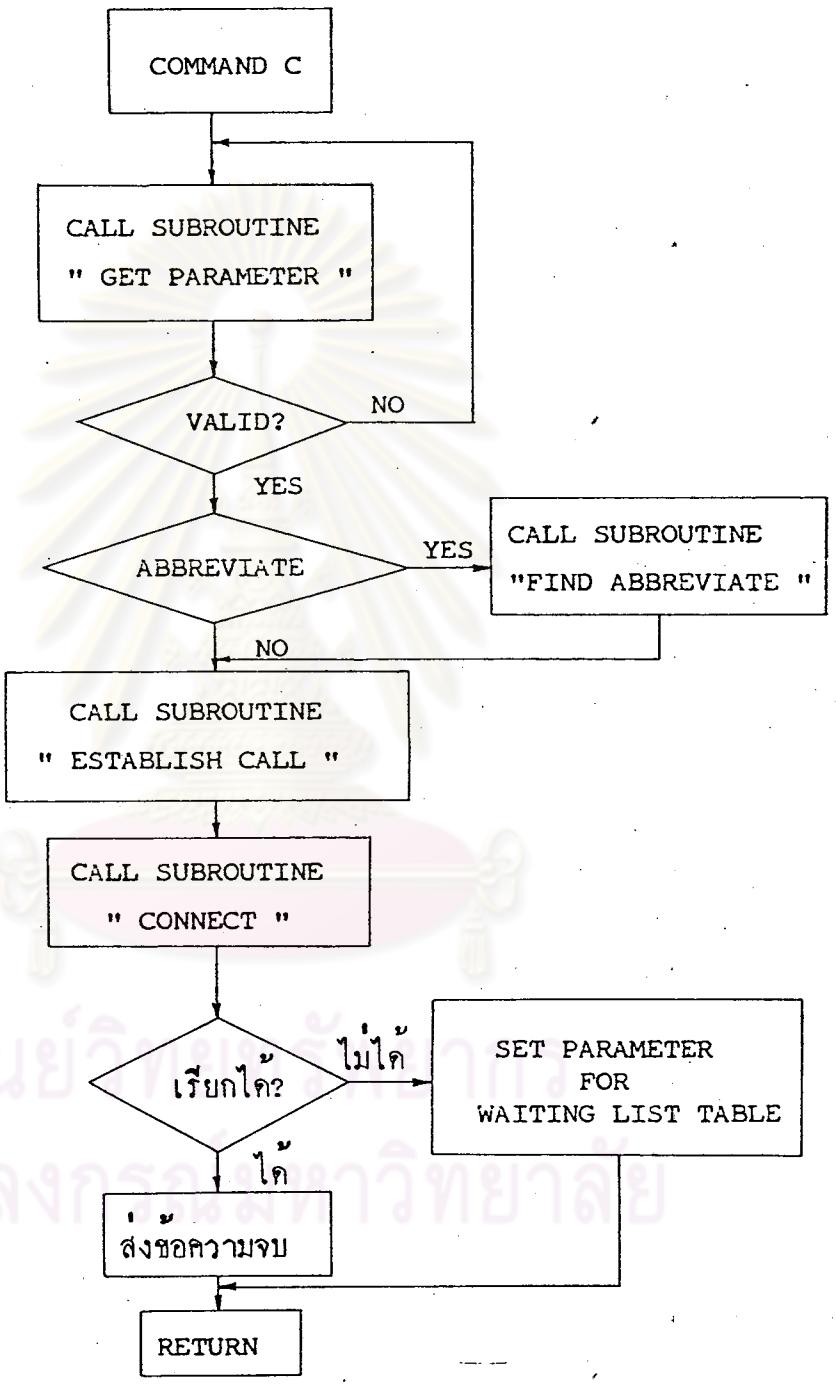
? (กด CR) โปรแกรมจะคืนการทำงานให้ EXECUTIVE PROGRAM

? 2 เลขย่อหมายเลข 2 ต้องการจะเปลี่ยนเลขเต็มใหม่

? 74632+ เลขที่เปลี่ยนใหม่ รวมทั้งเครื่องหมายบวก (+)

? (กดตัว B) เมื่อต้องการให้พิมพ์ผลการเปลี่ยนแปลงใหม่ทั้งหมด
หรือ
? (กด CR) โปรแกรมจะคืนการทำงานให้ EXECUTIVE PROGRAM
หรือ ถ้าต้องการเปลี่ยนเลขย่อใหม่ ก็พิมพ์เลขย่อที่ต้องการเปลี่ยนได้เลย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 21 FLOWCHART OF COMMAND C

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

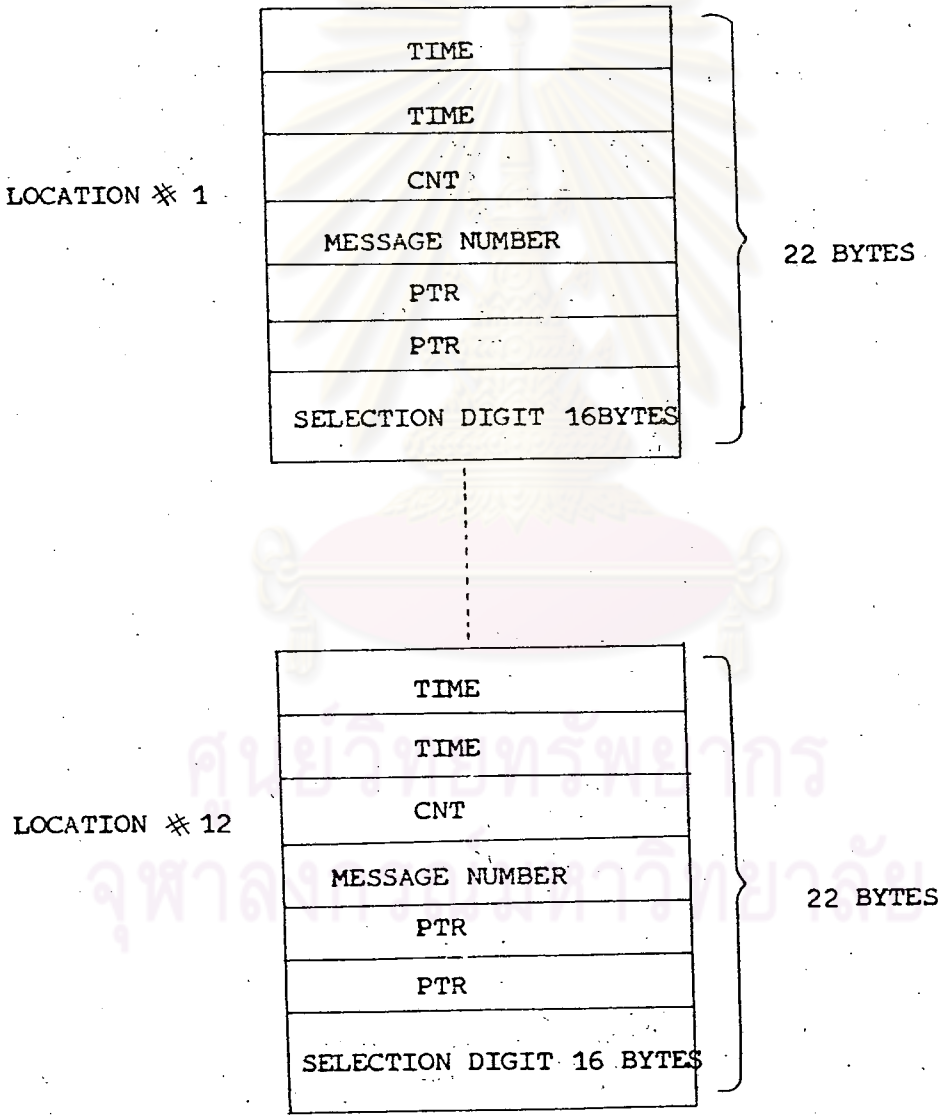
COMMAND C : ทำการเรียกติดต่อซ้ำเนื่องจากปลายทางสายไม่ว่าง

การใช้งานตามโปรแกรมนี้ ตัวโปรแกรมจะใช้ส่วนหนึ่งของหน่วยความจำมาใช้เป็น WAITING TABLE การจัดหน่วยความจำมาใช้เป็น WAITING TABLE แสดงได้ตามรูปที่ 22 การจัดหน่วยความจำนี้จะแบ่งออกเป็น 12 LOCATION แต่ละ LOCATION จะมี 22 ไบท์ 2 ไบท์แรกจะใช้เก็บเวลาที่จะส่งข้อความเทเล็กซ์
ไบท์ที่ 3 ใช้เพื่อทำหน้าที่ เป็น COUNT ใช้แสดงจำนวนครั้งที่ยังจะต้องเรียก ซึ่งใช้ในกรณี COMMAND C นี้ซึ่งจะบอกค่าให้ลองเรียกกี่ครั้ง ถ้ายังเรียกไม่สำเร็จ ในกรณีนี้ COUNT (CNT) จะเซ็ทด้วยจำนวนครั้งที่ต้องการและจะลดลงทีละหนึ่งทุกครั้งที่ลองเรียก
ไบท์ที่ 4 ใช้เพื่อทำหน้าที่ เป็น MESSAGE NO. ซึ่งจะใช้เป็น POINTER ที่จะชี้ไปยัง MESSAGE TABLE ว่าหมายเลขที่กำลังเรียกอยู่นี้เมื่อเรียกได้แล้วจะส่งข้อความเทเล็กซ์ฉบับใดในหน่วยความจำออกไป แต่ในกรณี COMMAND C นี้ผู้ใช้จะเป็นผู้ส่งข้อความเองดังนั้นในกรณีนี้จะใส่ค่า NULL ไว้ ซึ่งจะทำให้ตัวโปรแกรมหลังจากเรียกติดต่อได้จะส่งสัญญาณไปยังเครื่องเทเล็กซ์เพื่อให้ผู้ใช้เรียกติดต่อกันต่อไป
ไบท์ที่ 5,6 เป็น POINTER ชี้ไปยัง LOCATION อื่นๆ ในกรณีที่มีหมายเลขหลายหมายเลขรอเวลาการเรียกส่งอยู่ โดยจะให้ LINK ถึงกันในลักษณะเรียงตามเวลา เพื่อที่ว่าถ้าตรวจ LOCATION แรกของ LIST แล้วยังไม่ถึงเวลาที่จะเรียกส่งจะได้ไม่ต้องเสียเวลาไปตรวจ LOCATION อื่นต่อไป

อีก 16 ไบท์ที่เหลือนั้นจะใช้เก็บหมายเลขเทเล็กซ์ ที่ต้องการจะให้เรียกติดต่อ

ลักษณะการทำงาน SOFTWARE MODULE ของ COMMAND นี้จะขอ PARAMETER เพิ่มเติมคือหมายเลขที่ต้องการเรียกติดต่อ และจำนวนครั้งที่ต้องการให้พยายามเรียกในกรณีที่สายไม่ว่าง ในกรณีที่ผู้ใช้พิมพ์หมายเลขย่อมาก็จะทำการหาหมายเลขเต็มและจะเก็บข้อมูลเหล่านี้ไว้เพื่อดำเนินการเรียกติดต่อ ถ้าการลองเรียกแล้ว

สายไม่ว่าง ตัวโปรแกรมก็จะเอาข้อมูลไปเก็บไว้ใน WAITING TABLE พร้อมกับเซ็ทเวลาที่จะเรียกในครั้งต่อไป พร้อมกับตรวจสอบว่า การเรียกนี้จะอยู่ในลำดับใดใน TABLE พร้อมกับทำการจัดอันดับใหม่ด้วยการเปลี่ยน LINK LIST



รูปที่ 22 การจัดหน่วยความจำเพื่อใช้เป็น WAITING TABLE

การเรียกติดต่อจะใช้เวลาติดต่อห่างกันครั้งละ 2 นาที ถ้าสามารถเรียกติดต่อกันได้ โปรแกรมก็จะคอยตรวจสอบการติดต่อนั้น และจะคืนการควบคุมการทำงานกลับไปให้ EXECUTIVE PROGRAM ต่อไป แต่ถ้าหากเรียกครบจำนวนครั้งที่บังไว้แล้วและยังไม่สามารถเรียกติดต่อได้ โปรแกรมก็จะส่งข้อความไปบอกผู้ใช้ว่าไม่สามารถเรียกติดต่อได้

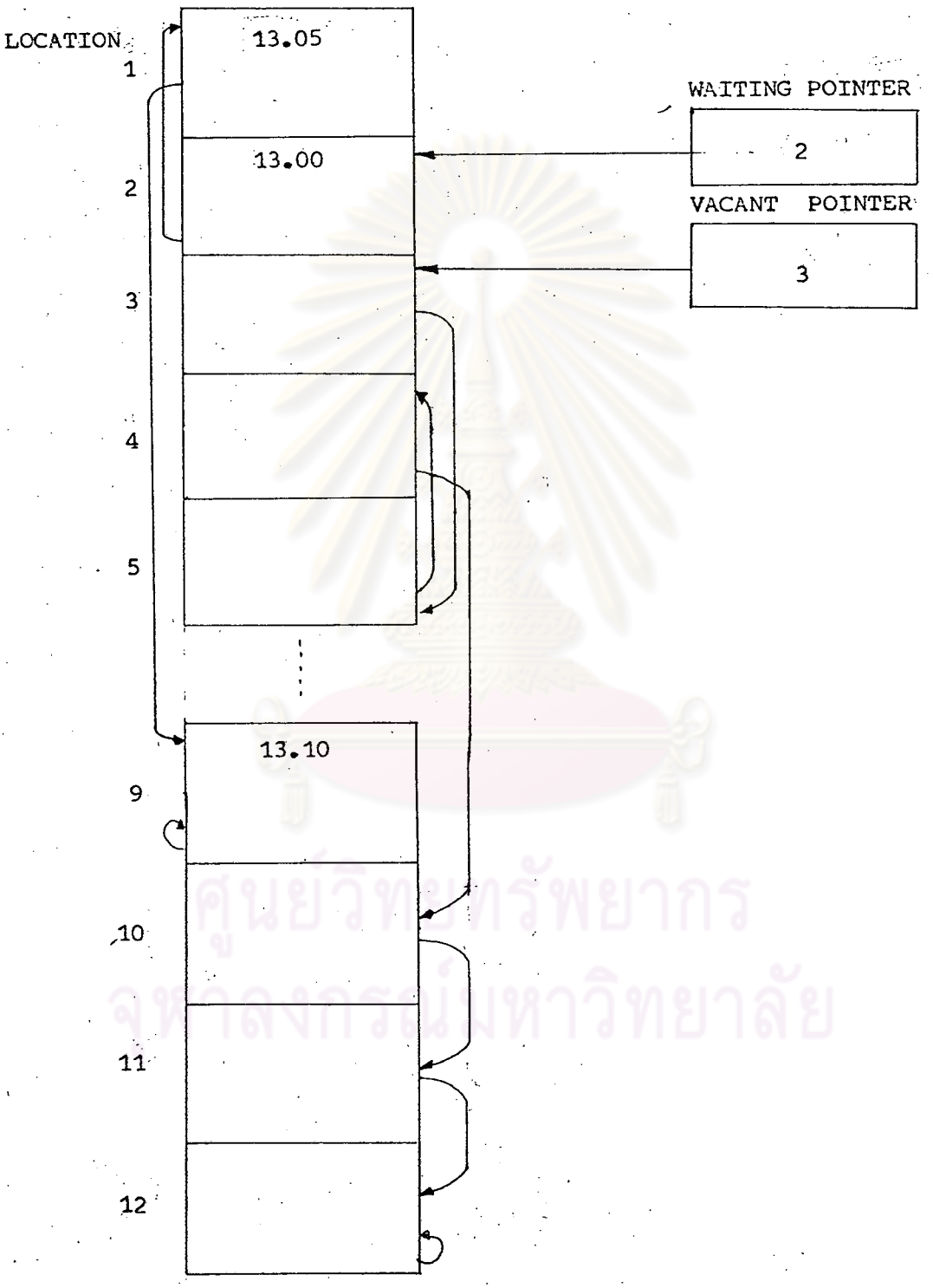
อนึ่งในระหว่างช่วงเวลาที่ว่างอยู่ระหว่างการเรียกติดต่อแต่ละครั้งนั้น ตัวโปรแกรมจะคืนการควบคุมการทำงานกลับไปให้ EXECUTIVE PROGRAM ดังนั้นผู้ใช้ก็จะสามารถสั่งให้ตัวเครื่องทำ COMMANDอื่นๆ ได้ด้วย

ขั้นตอนการตอบโต้ระหว่างตัวเครื่องและผู้ใช้เทเล็กซ์

CMD ? C — ตัว C ผู้ใช้เครื่องเทเล็กซ์เป็นผู้พิมพ์

? 82341+5 — เป็นหมายเลขที่ผู้ใช้เครื่องต้องการเรียกตามด้วยเลขจำนวนครั้งที่ให้ลองเรียกใหม่เมื่อสายไม่ว่าง

รูปที่ 26 จะแสดงการทำงานของโปรแกรมร่วมกับหน่วยความจำ จะเห็นว่า WAITING POINTER จะชี้ไปที่ WAITING TABLE LOCATION ที่ 2 ซึ่งบ่งว่ามีข้อความเทเล็กซ์รอส่งออกเวลา 13.00 น จาก LOCATIONที่ 2 จะมี POINTER ชี้ไปที่ LOCATION ที่ 1 ซึ่งจะบอกว่ามีข้อความอีกฉบับรอส่งออกเวลา 13.05 น ที่ LOCATION ที่ 1 จะมี PTR ชี้ไปที่ LOCATION ที่ 9 บอกว่ามีข้อความรอส่งออกเวลา 13.10 น ที่ LOCATIONที่ 9 และ PTR จะเป็น NULL ซึ่งบ่งว่าใน WAITING LIST TABLE มี MESSAGE รอส่งออกเพียง 3 ฉบับ ส่วน VACANT POINTER จะคอยบอกว่ามี LOCATION ใดว่างที่จะให้ใช้ และจะชี้ LOCATION แรกที่ว่างเสมอ



รูปที่ 23 การทำงานของโปรแกรมร่วมกับหน่วยความจำของ WAITING TABLE

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

COMMAND D : เตรียมข้อความที่จะส่งออกเก็บไว้ในบัฟเฟอร์ของตัวเครื่อง

ก่อนที่จะกล่าวถึงการทำงานของ COMMAND ดังกล่าวนี้จะขอกล่าวถึงวิธีการจัดหน่วยความจำเสียก่อนเพราะว่า ลักษณะของโปรแกรมนี้คือการเก็บข้อความเทเล็กซ์ลงใน BUFFER MEMORY ดังนั้นปัญหาสำคัญก็คือ MEMORY MANAGEMENT

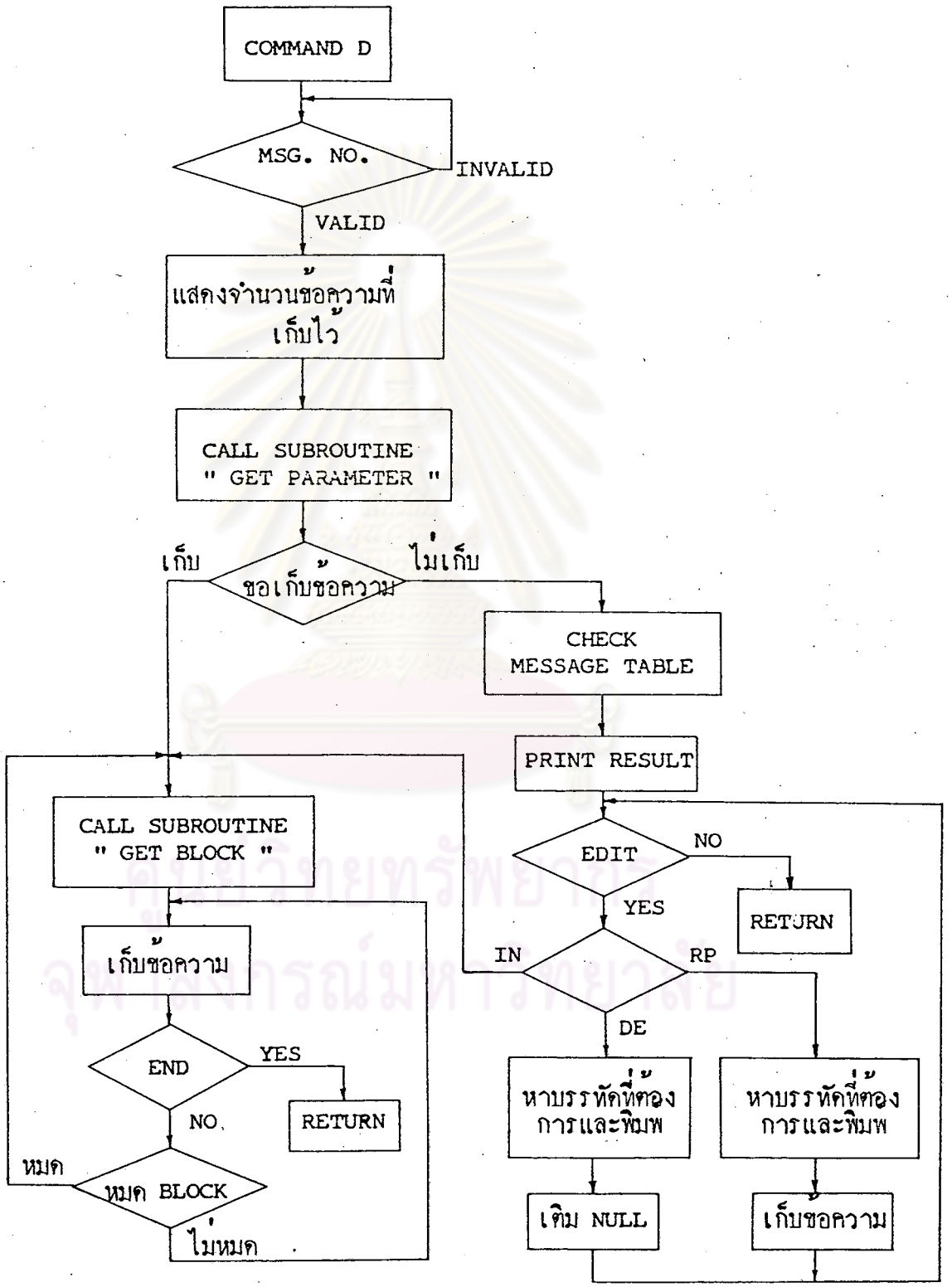
ลักษณะของการจัดหน่วยความจำนี้จะเป็นแบบ DYNAMIC โดยทำเป็นแบบ LINK LIST เนื่องจากส่วนที่เป็น RAM ของตัวเครื่องมีทั้งหมด 4 K ในที่นี้ 3 K จะใช้เป็นหน่วยความจำเพื่อเก็บข้อมูลเทเล็กซ์ ส่วนอีก 1 K จะใช้เป็น SCRATCH PAD, STACK และ WORKING AREA สำหรับโปรแกรมอื่น

หน่วยความจำทั้ง 4 K นี้จะเริ่มต้นเก็บไว้ที่ LOCATION 1000H จนถึง LOCATION 1FFFH ส่วนพื้นที่ที่จะใช้เป็นบัฟเฟอร์ดังกล่าวจะใช้ที่แอดเดรส 1400H ถึง 1FFFH (ดูตามรูปที่ 25) หน่วยความจำทั้ง 3K นี้จะแบ่งออกเป็น SEGMENT ย่อยๆ ซึ่งจะเรียกว่า BLOCK และมี BLOCK ละ 256 ไบท์ ลักษณะของการเก็บข้อมูลจะเก็บข้อมูลใน 255 ไบท์แรก ส่วนไบท์สุดท้ายจะเป็น POINTER เพื่อที่จะชี้ไปที่ BLOCK อื่นใน LIST เดียวกัน

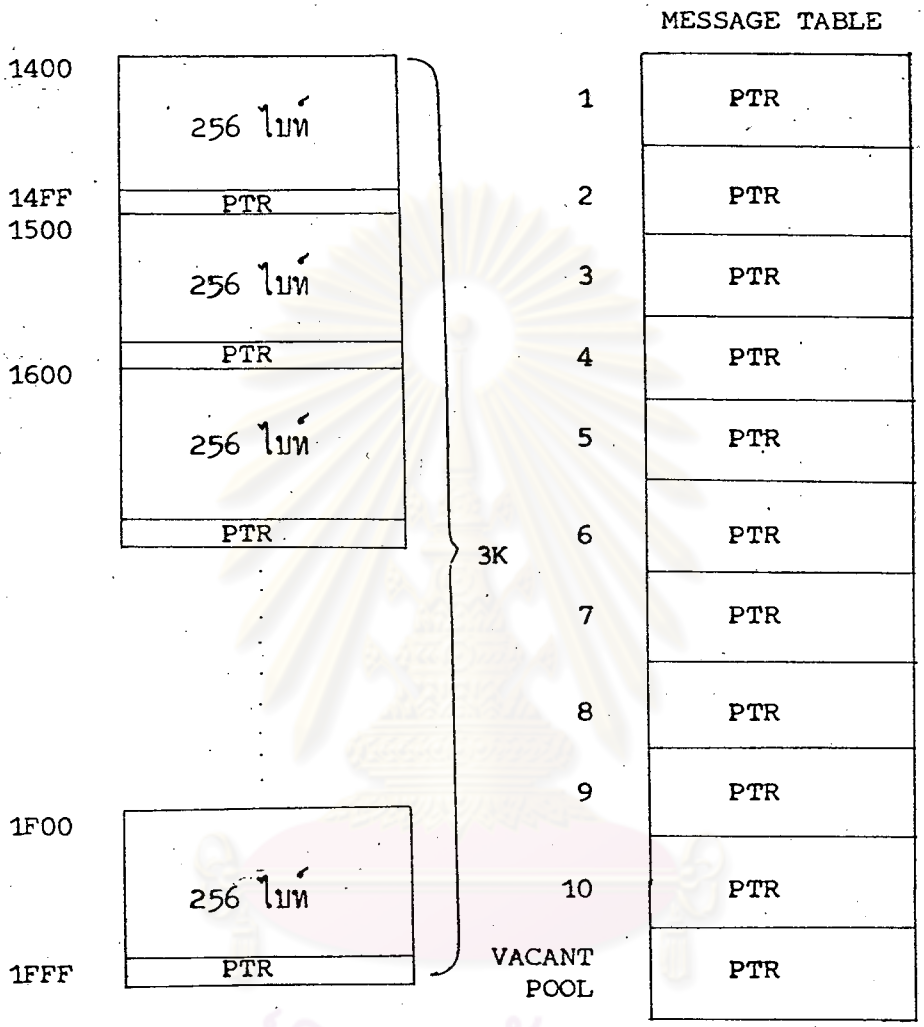
ขณะเดียวกันก็จะมีพื้นที่อีกส่วนหนึ่ง (1 K ที่เหลือ) ซึ่งใช้หน่วยความจำ 11 ไบท์ โดยจะใช้เป็น MESSAGE TABLE ซึ่ง MESSAGE TABLE นี้จะทำหน้าที่เก็บ POINTER ของ MESSAGE NUMBER ต่างๆไว้ โดยจะมีจุดประสงค์ที่ว่า จะได้ทราบว่า MESSAGE ฉบับหมายเลขใดจะเก็บอยู่ในส่วนไหนของหน่วยความจำ

การทำงานเริ่มแรกในตอนส่วน INITIALIZE ROUTINE จะทำการ CLEAR ให้ MESSAGE TABLE เป็นค่าว่างเปล่า (ใส่ค่า FF) เพื่อบ่งให้ทราบว่าขณะนี้ไม่มีข้อความใดๆเก็บอยู่ ส่วน LOCATION ที่ 11 ซึ่งเป็น POINTER ของ VACANT POOL มีหน้าที่ชี้บอก BLOCK ที่ว่าง จะชี้ไปที่ LOCATION 1400H ซึ่งเป็น BLOCK แรกของบัฟเฟอร์

หมายเหตุ POINTER ดังกล่าวนี้ใช้เพียง ไบท์เดียวโดยจะเก็บแอดเดรส



รูปที่ 24FLOWCHART OF COMMAND D



1 BLOCK = 256 BYTE

4 BLOCK = 1K BYTE

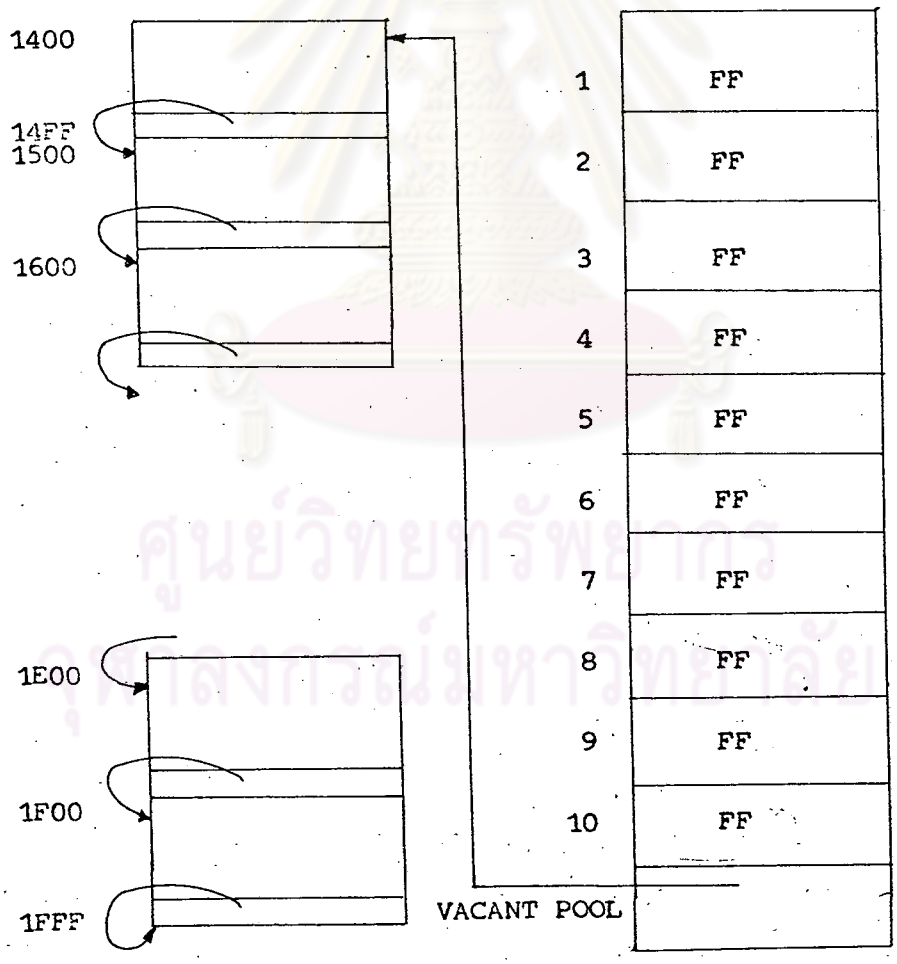
แอดเดรสตำแหน่งที่ 1000 - 13FF ใช้สำหรับทำงานหน้าที่อื่นๆ

รูปที่ 25 การจัดหน่วยความจำ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บิทที่ 8 ถึง บิทที่ 15 เท่านั้นก็พอ เนื่องจากทุก BLOCK LOCATION แรกของ BLOCK ADDRESS 8 บิทหลังจะเป็น " 0 " หมด ส่วนแอดเดรส 4 บิทแรกสุดจะเป็น " 1 " เสมอ

ในอีกส่วนหนึ่งของ INITIALIZE จะทำการเซ็ท POINTER ของแต่ละ BLOCK ให้ชี้ที่ BLOCK ถัดไปดังแสดงตามรูปที่ 26 และที่ BLOCK สุดท้ายจะเซ็ทค่า NULL (FFH) เพื่อบ่งให้รู้ว่าเป็น BLOCK สุดท้าย



รูปที่ 26 ลักษณะการ INITIALIZE หน่วยความจำ

หน่วยความจำที่ใช้เก็บข้อความเป็น BLOCK นี้สามารถเพิ่มเติมได้ทีละ BLOCK ในกรณีที่ข้อความแต่ละฉบับเกิน 256 ไบท์ โดยจะต่อเชื่อมโยงกันได้ด้วย LINK ดังนั้นจึงทำให้การเก็บข้อความไว้ในหน่วยความจำนี้เป็นแบบ DYNAMIC

การทำงานตามคำสั่งอันนี้ เมื่อผู้ใช้ส่งคำสั่งเป็น COMMAND D มา ตัวเครื่องจะให้สภาวะของข้อความที่เก็บไว้ในหน่วยความจำ โดยจะบอกถึงว่ามี MESSAGE เก็บอยู่แล้วกี่ฉบับ และยังมี MEMORY AREA เหลืออีกเท่าไร ในกรณีนี้ผู้ใช้ก็จะพิมพ์ตอบไปด้วย MESSAGE NUMBER ที่ต้องการจะให้เก็บใหม่ เครื่องจะตอบรับด้วยการให้เครื่องหมายบ่งให้ผู้ใช้เริ่มพิมพ์ข้อความได้ ในกรณีนี้ถ้าผู้ใช้ให้หมายเลขซ้ำกันกับ MESSAGE ที่มีเก็บไว้ตัวโปรแกรมจะส่งคำถามมาให้ผู้ใช้ยืนยันว่าต้องการใช้หมายเลขนี้แน่หรือไม่ ถ้าใช่ตัวโปรแกรมก็จะยกเลิกข้อความเดิมที่เก็บไว้และเตรียมรับข้อความใหม่แทน

เมื่อข้อความเตรียมเสร็จแล้วผู้ใช้จะพิมพ์ MMMEM เพื่อใช้สำหรับเป็นระหัสในการเลิกติดต่อกับชุมสายเทเล็กซ์ ในกรณีนี้ตัวโปรแกรมก็จะรับรู้เช่นเดียวกันว่าเป็นการจบข้อความที่เตรียมไว้ และจะคืนการควบคุมการทำงานกลับไปให้ EXECUTIVE PROGRAM ต่อไป

นอกจากนี้แล้วตามคำสั่งนี้ยังมีความสามารถในการแก้ไขข้อความที่จัดเตรียมไว้ในหน่วยความจำก่อนที่จะส่งออกได้ด้วย ซึ่งจะใช้ร่วมกับ COMMAND พิเศษเพิ่มเติมจากกรณีดังกล่าวข้างต้น กล่าวคือเมื่อโปรแกรมตอบกลับมาให้ผู้ใช้พิมพ์หมายเลขประจำข้อความ ถ้าผู้ใช้พิมพ์เฉพาะหมายเลข ตัวโปรแกรมก็จะรับรู้ว่า ผู้ใช้ต้องการที่จะเตรียมข้อความใหม่ แต่ถ้าผู้ใช้พิมพ์ตัวอักษร " P " นำหน้าหมายเลข ตัวโปรแกรมก็จะนำข้อความตามหมายเลขนั้นซึ่งเก็บไว้ในหน่วยความจำ กลับมาพิมพ์บนเครื่องของผู้ใช้ให้ผู้ใช้ตรวจสอบว่าข้อความที่เตรียมไว้นั้นถูกต้องหรือไม่ ถ้าผู้ใช้ต้องการจะแก้ไขก็สามารถทำได้โดย พิมพ์ EDIT COMMAND ซึ่งจะประกอบด้วย

- DELETE LINE COMMAND ซึ่งผู้ใช้พิมพ์ตัวอักษร "D" และตามด้วย

หมายเลขของบรรทัดที่ให้ข้อความถูกกำจัดออกไป ตัวโปรแกรมก็จะไปหาข้อความที่บรรทัดนั้นและอัดทับด้วย "NULL " character

- INSERT LINE COMMAND เมื่อผู้ใช้พิมพ์อักษร " I " ในกรณีนี้ตัวโปรแกรมจะไปนำเอา MEMORY BLOCK ที่ว่างมาจาก BUFFER POOL แล้วเริ่มเก็บข้อความ ถ้าเก็บข้อความไม่พอก็จะไปเอา BLOCK ใหม่มาอีก เมื่อเก็บข้อความจบก็จะคืนการทำงานไปให้ EXECUTIVE PROGRAM

- REPLACE LINE COMMAND ในกรณีที่ผู้ใช้ต้องการที่จะเปลี่ยนแปลงข้อความในบรรทัดใดก็จะทำได้ด้วยการใช้ COMMAND อันนี้ด้วยการพิมพ์ตัวอักษร" R " นำหน้าหมายเลขบรรทัด

COMMAND E : ส่งข้อความในบัฟเฟอร์ที่เตรียมไว้ ตามเวลาที่ต้องการ

SOFTWARE MODULE ของ COMMAND นี้จะขอ PARAMETER เพิ่มเติม คือหมายเลขที่จะส่งซึ่งจะเป็นหมายเลขเต็มหรือ หมายเลขย่อก็ได้ หลังจากนั้นก็เป็นหมายเลขข้อความที่จะส่ง พร้อมกับนั้นถ้ามีกำหนดเวลา ก็จะไปเซ็ทค่าใน WAITING TABLE เพื่อจัดเรียงลำดับ และรอเวลาที่จะส่งออก เมื่อถึงเวลาดังกล่าวจึงค่อยเรียกส่ง แต่ถ้าไม่กำหนดเวลาตัวโปรแกรมก็จะส่งข้อความไปทันที

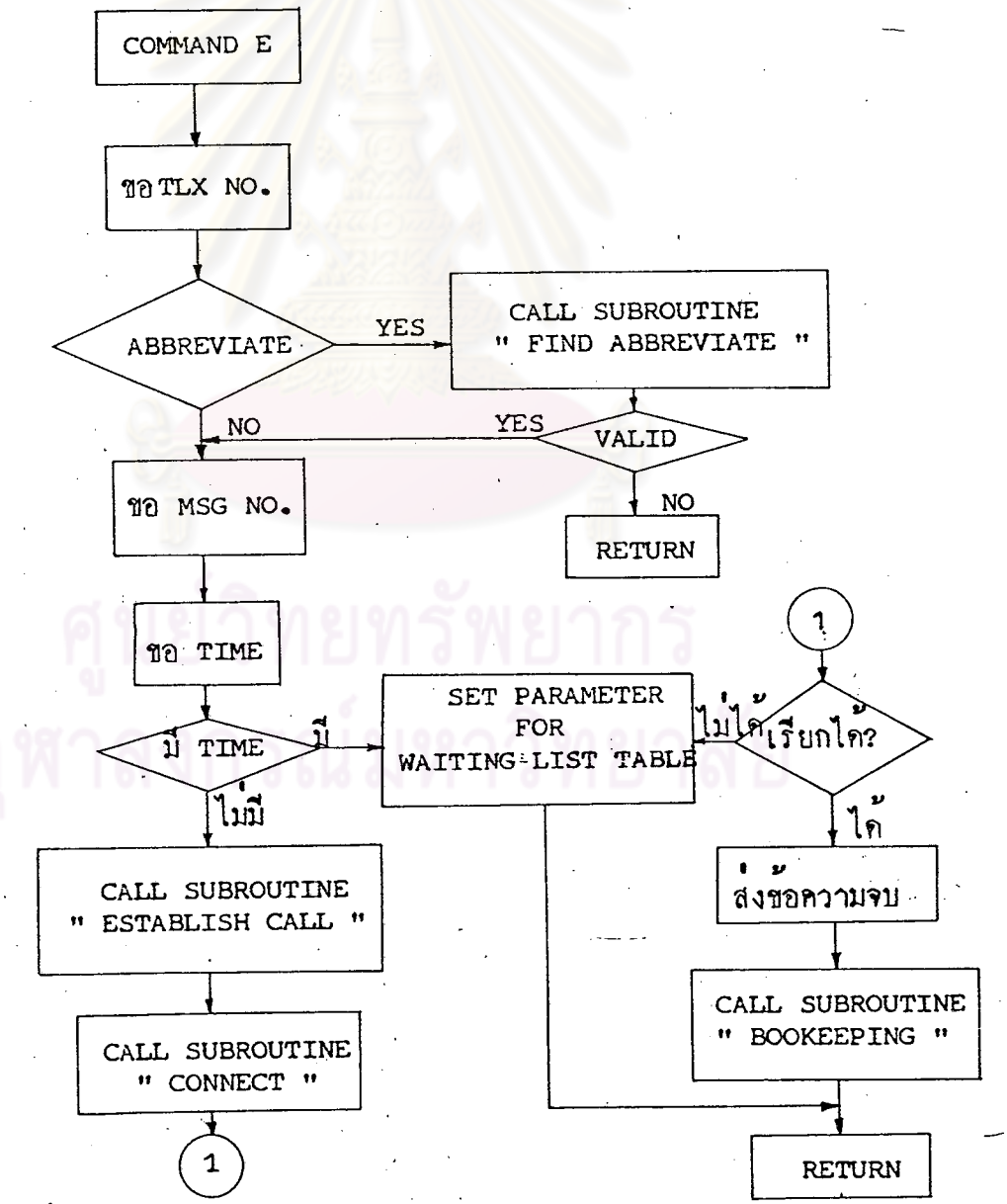
ลักษณะการโต้ตอบของคำสั่งจะเป็นดังนี้

```
CMD ? E
NO      3+
MNO     2
T?      18.00
```

ลักษณะการโต้ตอบตามตัวอย่างนี้ก็คือ ผู้ใช้ต้องการจะส่งข้อความในหน่วยความจำฉบับที่ 2 ไปยังปลายทางตามหมายเลขย่อ 3 ในเวลา 18.00 น

หมายเหตุ ถ้าเรียกแล้วสายไม่ว่างตัวโปรแกรมจะเรียกใหม่ทุก 2 นาทีอีก 10 ครั้ง

การส่งข้อความตาม WAITING LIST TABLE อาจมีกรณีพิเศษเกิดขึ้นคือ ขณะที่กำลังส่งข้อความเทเล็กซ์อยู่นั้น ข้อความยาวจนเลยช่วงเวลาของข้อความฉบับต่อไปซึ่งรอส่งอยู่ กรณีเช่นนี้ ตัวโปรแกรมที่เขียนขึ้น จะมีข้อกำหนดไว้ว่า ถ้าเวลาที่รอเพื่อส่งข้อความฉบับต่อไป ไม่เกิน 30 นาที จากเวลาที่เซ็ทค่าไว้ใน WAITING LIST TABLE ก็ให้ส่งข้อความฉบับที่รอส่งออกต่อไปได้เลย แม้จะเลยเวลาไปแล้วก็ตาม และช่วงเวลา 30 นาทีนี้สามารถเปลี่ยนแปลงได้ตามความเหมาะสม



รูปที่ 27 FLOWCHART OF COMMAND E

COMMAND F : ทำการต่อตรงระหว่างเครื่องเทเล็กซ์กับชุมสายโดยไม่ใช้การทำงานของตัวเครื่องตามคำสั่งต่างๆ

การทำงานตามคำสั่งนี้ตัวโปรแกรมจะทำการต่อสัญญาณโดยตรงระหว่างเครื่องเทเล็กซ์กับชุมสายเทเล็กซ์ โดยตัวโปรแกรมจะคอยตรวจสอบการเรียกติดต่อเมื่อผู้ใช้เลิกใช้ ตัวโปรแกรมก็จะคืนการทำงานกลับไปให้ EXECUTIVE PROGRAM ต่อไป

การที่จะทำงานให้ได้ตามดังกล่าวข้างต้นนี้ตัว HARDWARE จะต้องมีส่วนประกอบของวงจร INTERRUPT และวงจร TIMER เนื่องจากการทำงานตาม COMMAND บางอย่างนั้นขึ้นอยู่กับเวลา ดังนั้นเมื่อถึงเวลาที่กำหนดตัว TIMER จะต้องส่งสัญญาณ INTERRUPT ไปบอก EXECUTIVE PROGRAM ให้ส่งผ่านการทำงานไปยัง SOFTWARE MODULE ต่างๆตามที่เวลาที่ตั้งกำหนดไว้ แต่ถ้าเวลาดังกล่าวตัวโปรแกรมกำลังทำงานตาม COMMAND หนึ่ง COMMANDใดอยู่ ตัว INTERRUPT จะยังไม่มีผลจนกว่า SOFTWARE MODULE นั้นๆจะคืนการทำงานกลับไปให้ EXECUTIVE PROGRAM เสียก่อน

และอีกกรณีหนึ่งก็คือหากมี INTERRUPT ด้วยช่วงเวลาอันเกิดจาก SOFTWARE ที่ทำตาม COMMAND ต่างๆ เราก็ต้องจัด PRIORITY ให้เพื่อที่ว่าจะได้กำหนดได้ว่า COMMAND ใดสมควรได้ทำก่อนหลัง ฯลฯ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย