บทที่ 4

หน้าที่ของโปรแกรมต่าง ๆ

ชื่อโปรแกรม main()

ตัวแปรที่ส่งผ่าน ไม่มี

หน้าที่ เป็นโปรแกรมหลัก

ขั้นตอนการทำงาน

1.จัดค่าเริ่มต้น ของตัวแปรต่าง ๆ

2.รับคำสั่งจากผู้ใช้งาน โดยเรียกโปรแกรม accept โดยคำสั่งที่ได้มาถูกเก็บอยู่ในบัฟเฟอร์ชื่อ qbuff

3.เรียก stoke เพื่อหาคำสำคัญคำแรกสุดในคำสั่ง ถ้าพบก็เรียก go เพื่อวิ่งไปยังโปรแกรมที่ทำหน้าที่ปฏิบัติตามคำสั่งนั้น ๆ ถ้าไม่พบคำสำคัญก็เรียก csent

4.ถ้า cquit ยังมีค่าเป็น 1 วนกลับไปที่ข้อ 2 ถ้าไม่เป็น 1 เลิกทำงาน

ชื่อโปรแกรม create()

ตัวแปรที่ส่งผ่าน ไม่มี

หน้าที่ สร้างแฟ้มข้อมูล

รายละเอียดการทำงาน

1.ตรวจดูว่ามีแฟ้มที่เปิดค้างไว้หรือไม่ โดยดูจาก fnum ตัวแรกสุดในตารางโครงสร้างแฟ้ม ถ้าเป็น '¥0' แสดงว่าไม่มีแฟ้มเปิดค้างไว้

2.ถ้าผู้ออกคำสั่งไม่ได้ใส่ชื่อแฟ้มมาในคำสั่ง ให้ถามผู้ใช้ว่าแฟ้มชื่ออะไร

3.เติมไฟล์เอ็กเทนชัน (File Extension) ให้โดยอัตโนมัติ

4.ให้ผู้ใช้งานกำหนดโครงสร้างแฟ้มโดยสะดวก โดยรับข้อมูลในแบบเต็มจอ

5.ทำการสร้างแฟ้มตามที่ผู้ออกคำสั่งกำหนด โดยการสร้างแฟ้มนั้นนอกจากสร้างแฟ้มข้อมูลแล้วยังต้องสร้างแฟ้มเก็บโครงสร้างอีกด้วย

ชื่อโปรแกรม use()
ตัวแปรที่ส่งผ่าน ไม่มี
หน้าที่ เปิดแฟ้มข้อมูล
รายละเอียดการทำงาน

1.ตรวจดูว่ามีการเปิดแฟ้มค้างไว้หรือไม่ ถ้าเปิดแฟ้มค้างไว้แสดงข้อความแนะนำแล้วเลิกทำงาน

2.ถ้าผู้ออกคำสั่งไม่ได้บอกชื่อแฟ้มที่ต้องการเปิดให้ถามว่าต้องการเปิดแฟ้มชื่ออะไร

3.ทำการเปิดแฟ้มโครงสร้างแล้วอ่านโครงสร้างเข้ามาเก็บไว้ในตารางโครงสร้างและคำนวณค่าตัวแปรต่างๆ ที่จำเป็นต่อการทำงาน หน้าที่นี้ทำโดยการการเรียกโปรแกรมย่อย read_st()

4.ทำการเปิดแฟ้มข้อมูลเพื่อเตรียมพร้อมสำหรับการทำงาน

ชื่อโปรแกรม spec()
ตัวแปรที่ส่งผ่าน ไม่มี
หน้าที่ แสดงโครงสร้างของแฟ้มข้อมูล
รายละเอียดการทำงาน

1.ตรวจดูว่ามีแฟ้มที่เปิดเอาไว้หรือไม่ ถ้าไม่มี แสดงข้อความแนะนำแล้วเลิกทำงาน

2.นำรายละเอียดของโครงสร้างที่เก็บไว้ในตารางโครงสร้างของแฟ้มข้อมูลแสดงออกทางจอภาพให้อยู่ในรูปแบบที่สะดวกต่อการอ่านของผู้ใช้งาน

ชื่อโปรแกรม close_fil()
ตัวแปรที่ส่งผ่าน ไม่มี
หน้าที่ ปิดแฟ้มข้อมูลและแฟ้มโครงสร้าง

ชื่อโปรแกรม append()
ตัวแปรที่ส่งผ่าน ไม่มี
หน้าที่ เพิ่มข้อมูลเข้าสู่แฟ้มข้อมูล
รายละเอียดการทำงาน

1.ถ้าไม่มีแฟ้มถูกเปิดเอาไว้ แสดงข้อความแนะนำ แล้วเลิกทำงาน

2.อ่านระเบียนสุดท้ายของแฟ้ม(Bottom record) เพื่อนำเลขที่ระเบียนมาเพิ่มอีกหนึ่งให้เป็นเลขที่ระเบียนปัจจุบัน

3.เรียกโปรแกรมบรรฒาธิกรเพื่อรับข้อมูลแบบเต็มจอ ตรวจสอบคำสั่งควบคุมที่ผู้ทำงานพิมพ์เข้ามา ย้ายข้อมูลจากบัฟเฟอร์สำหรับแก้ไข (Edit_buff) ไปยังบัฟเฟอร์ของแฟ้มข้อมูล แล้วเขียนข้อมูลที่ได้รับมาลงแฟ้ม

ชื่อโปรแกรม list()
ตัวแปรที่ส่งผ่าน ไม่มี
หน้าที่ แสดงข้อมูลในแฟ้มออกทางจอภาพ
รายละเอียดการทำงาน

1.ถ้าไม่มีการเปิดแฟ้มข้อมูลเอาไว้ แสดงข้อความแนะนำแล้วเลิกทำงาน

2.ทำการพาสคำสั่งส่วนที่เหลือโดยสแกนหาคำสำคัญ สองคำคือ 'สำหรับ' และ' ถัด' ถ้าพบคำสำคัญ 'สำหรับ' ก็จะเรียกพาสเซอร์ของนิพจน์คือ exp_par() เพื่อทำหน้าที่พาสนิพจน์ และให้ expf มีค่า 1 ถ้าพบ'ถัด' ก็เปลี่ยนข้อความตัวเลขให้เป็นค่าแล้วเก็บไว้ที่ตัวแปรชื่อ nxt และให้ nxtf มีค่า

3.ถ้า nxtf มีค่าไม่เป็น 1 ทำการย้ายระเบียนปัจจุบันไปที่ระเบียนแรก

4.ทำการแสดงข้อมูลไปเรื่อย ๆ จนกว่า nxt มีค่าเท่ากับ 0 หรือสิ้นสุดแฟ้ม และในวงรอบ(Loop)ของการแสดงข้อมูลนั้น ถ้า expf มีค่า 1 ก็จะเรียก expeval()เพื่อคำนวณค่านิพจน์ว่าได้ค่าจริงหรือ เท็จ ถ้าเท็จก็จะไม่แสดงระเบียนนั้น ขยับไปยังระเบียนถัดไป และถ้า nxtf มีค่า 1 ให้ลดค่า nxt ลง 1

ชื่อโปรแกรม go_to()
ตัวแปรที่ส่งผ่าน ไม่มี
หน้าที่ ย้ายระเบียนปัจจุบันไปยังระเบียนที่ต้องการ
รายละเอียดการทำงาน

1.ถ้าไม่มีแฟ้มเปิดอยู่ แสดงข้อความแนะนำแล้วเลิกการทำงาน

2.ตรวจสอบตัวเลขที่ใส่ต่อจากคำสำคัญ 'ไป' ถ้าเกิน 5 หลัก แสดงข้อความผิดพลาดแล้วเลิกทำงาน

3.ใช้เลขที่ได้ในการค้นเลขที่ระเบียนที่ต้องการ ถ้าไม่พบ แสดงข้อความผิดพลาดและให้ ef_flg มีค่า 'F'

ชื่อโปรแกรม edit()
ตัวแปรที่ส่งผ่าน ไม่มี
หน้าที่ แก้ไขระเบียน
รายละเอียดการทำงาน

1.ถ้าไม่มีแฟ้มปัจจุบัน แสดงข้อความแนะนำแล้วเลิกทำงาน

2.ย้ายข้อมูลจากบัฟเฟอร์ของแฟ้มข้อมูล (data_buf) ไปยังบัฟเฟอร์สำหรับการแก้ไขข้อมูล เรียกโปรแกรมบรรณาธิกรเพื่อแก้ไขข้อมูล

3.ตรวจสอบปุ่มควบคุมที่ผู้ใช้สั่งมาว่าต้องการให้ทำอะไร เช่นขยับระเบียนไปยังระเบียนก่อนหน้า หรือระเบียนถัดไป หรือ เขียนข้อมูลปัจจุบันลงแฟ้ม

ชื่อโปรแกรม fval_con()

ตัวแปรที่ส่งผ่าน ไม่มี

หน้าที่ แปลงข้อมูลของเขตข้อมูลชนิดตัวเลขซึ่งเป็นข้อความให้เป็นค่า

รายละเอียดการทำงาน

เนื่องจากการเก็บข้อมูลเอาไว้ในแฟ้มของ บีทรีพ เก็บในลักษณะตัวอักขระเพียงอย่างเดียว ดังนั้นถึงแม้ผู้ใช้งานจะกำหนดให้เขตข้อมูลนั้นเป็นชนิดตัวเลขก็ตามในเวลาเก็บไว้ในแฟ้มจึงเก็บไว้แบบตัวอักขระ เมื่อต้องการนำค่าไปคำนวณจึงแปลงกลับไปเป็นค่าทางตัวเลข

ชื่อโปรแกรม open_st()

ตัวแปรที่ส่งผ่าน ไม่มี

หน้าที่ เปิดแฟ้มที่ใช้เก็บโครงสร้าง

รายละเอียดการทำงาน

เรียก บีทรีพ เพื่อเปิดแฟ้มโครงสร้าง

ชื่อโปรแกรม close_st()
ตัวแปรที่ส่งผ่าน ไม่มี
หน้าที่ ปิดแฟ้มที่ใช้เก็บโครงสร้าง

ชื่อโปรแกรม open_dat()
ตัวแปรที่ส่งผ่าน ไม่มี
หน้าที่ เปิดแฟ้มข้อมูล

ชื่อโปรแกรม close_dat()
ตัวแปรที่ส่งผ่าน ไม่มี
หน้าที่ ปิดแฟ้มข้อมูล

ชื่อโปรแกรม read_st()
ตัวแปรที่ส่งผ่าน ไม่มี
หน้าที่ อ่านข้อมูลในแฟ้มโครงสร้างมาเก็บไว้ในตาราง
โครงสร้าง

รายละเอียดการทำงาน

อ่านข้อมูลในแฟ้มโครงสร้าง ซึ่งเป็นรายละเอียดของโครงสร้างแฟ้มข้อมูล เข้ามาเก็บไว้ในตารางโครงสร้างแฟ้มข้อมูล โดยหนึ่งระเบียน (record) ของแฟ้มโครงสร้างประกอบไปด้วยข้อมูลเหล่านี้คือ

-เลขที่เขตข้อมูล (fnum)
-ชื่อเขตข้อมูล (fname)
-ชนิดของเขตข้อมูล (ftyp)
-ความยาว (flen)
-จุดทศนิยม (fdec)

ชื่อโปรแกรม stoke(str,end,toke)

ตัวแปรที่ส่งผ่าน

ตัวแปรรับเข้า

```
char *str; /* ข้อความที่ต้องการนำมาค้นหาคำสำคัญ*/
```

ตัวแปรส่งออก

```
int *end; /*จุดสิ้นสุดคำสำคัญ*/
char *toke; /* โทคเคน ของคำสำคัญนั้น */
```

ค่าส่งกลับจากโปรแกรม(ฟังก์ชันนี้)

ถ้าเป็น 0 แสดงว่าค้นพบคำสำคัญ

หน้าที่ ค้นหาคำสำคัญ (สแกนเนอร์)

ขั้นตอนการทำงาน

1.นำตัวอักขระตัวแรกไปเปรียบเทียบกับตัวอักขระในตารางดัชนี (comktab) จนกว่าจะพบหรือสิ้นสุดตารางดัชนี ถ้าสุดตารางดัชนีแสดงว่าไม่พบคำสำคัญ เลิกทำงาน

2.นำค่าแถว (row) ที่ได้จากตารางดัชนี ไปดึงคำที่ขึ้นต้นด้วยตัวอักขระตัวนั้นในตารางคำสำคัญ (tok_tab)

3.นำอักขระในข้อความ str ไปเปรียบเทียบกับตัวอักขระของคำสำคัญในตารางทีละตัว ถ้าตรงกันหมดไปจนสุดคำสำคัญ แสดงว่าค้นพบคำสำคัญส่งผ่านค่าตัวแปรแล้วเลิกทำงาน

4.ถ้าไม่ตรงกันให้ขยับไปยังคำสำคัญคำต่อไปโดยเพิ่มค่า row ถ้ายังไม่สิ้นสุดกลุ่มของคำสำคัญที่ขึ้นต้นด้วยตัวอักขระเดียวกัน ให้วนกลับไปที่ข้อ 3 ถ้าสิ้นสุด เลิกทำงาน

ต่อไปนี้เป็นโปรแกรมซึ่งทำหน้าที่เกี่ยวกับการรับและแสดงข้อมูลทางจอภาพ ตัวแปรซึ่งโปรแกรมเหล่านี้ใช้ร่วมกันมีดังนี้คือ

lm,rm,tm,bm = left , right , top and bottom margin
= กำหนดขอบเขตของจอภาพ

bc,cl = background color, color
= สีพื้นของจอ,สีตัวอักขระที่จะแสดง หรือรูปแบบตัวอักขระ

ccol,crow = current column, current row
= ตำแหน่งปัจจุบันของเคอเซอร์

scol,srow = start column,row.
= ตำแหน่งต้นต้นบนจอภาพของข้อความที่จะรับเข้า

chr,echr = character, extended char.
= ตัวอักขระที่เพิ่งรับมาจากแป้นพิมพ์

curtyp = current type
= ชนิดของอักขระที่รับเข้ามา

ins = insert flag
= แฟลกสำหรับบอกว่า เป็นการรับข้อมูลแบบแทรกหรือพิมพ์ทับ

*ptr = พอยเตอร์ซึ่งชี้ไปยังตัวอักขระปัจจุบันในบัฟเฟอร์ซึ่งตัวอักขระนี้ถูกชี้โดยเคอเซอร์บนจอภาพ

wbuff = บัฟเฟอร์สำหรับเก็บข้อความที่กำลังแก้ไข อยู่บนจอภาพ

mlen = maximum length
= ขนาดความกว้างสูงสุดของข้อความที่กำลังแก้ไขบนจอ

rlen = real length (rlen <= mlen)
= ความยาว(ในบัฟเฟอร์)จริงๆของข้อความที่กำลังแก้ไขบนจอภาพ

slen = screen length
= จำนวน column ของข้อความที่กำลังแก้ไข

cpos = ตำแหน่ง column ของตำแหน่งปัจจุบัน โดยนับจาก column ของจุดตั้งต้นข้อความ(ฆ.จุดตั้งต้นข้อความ cpos = 1)

epage = หมายเลขจอ ของข้อความแรกของจอภาพที่กำลังแก้ไข

ชื่อโปรแกรม say(str,row,col,att)

ตัวแปรที่ส่งผ่าน

ตัวแปรรับเข้า

```
char *str; /* ข้อความที่ต้องการนำมาแสดง*/
int row,col; /*ตำแหน่งที่ต้องการแสดงข้อความ */
char  att;   /*สี หรือ รูปแบบ (Attribute) ของตัวอักขระ */
```

หน้าที่ แสดงข้อความบนจอภาพในตำแหน่งที่ต้องการ

ชื่อโปรแกรม pgsay(str,row,col,att,page)

ตัวแปรที่ส่งผ่าน

ตัวแปรรับเข้า

```
char *str;   /* ข้อความที่ต้องการนำมาแสดง*/
int row,col; /*ตำแหน่งที่ต้องการแสดงข้อความ */
```

```
char att;  /*สี หรือ รูปแบบ (Attribute) ของตัว
                อักขระ*/
char page; /*หมายเลขจอ ของข้อความนี้*/
```

หน้าที่ แสดงข้อความบนจอภาพในตำแหน่งที่ต้องการ

รายละเอียดในการทำงาน

โปรแกรมย่อยนี้ จะใช้คู่กับโปรแกรม get โดยโปรแกรม pgsay ทำหน้าที่ในการแสดงข้อความในจอที่จะทำการรับข้อมูล ตัวอย่างข้อความที่จะแสดงเช่น คำแนะนำการทำงาน ชื่อเขตข้อมูลที่จะรับข้อมูล การที่ต้องกำหนด page ก็เพราะในการรับข้อมูลแบบเต็มจอ ในบางครั้งจะกินเนื้อที่มากกว่า 1 จอ ดังนั้นเวลาขึ้นจอใหม่ ข้อความที่เป็นของจอนั้นจะต้องถูกแสดง ข้อความทุกข้อความที่ส่งมาแสดงจะนำไปเก็บไว้ในตารางชื่อ saytab

ชื่อโปรแกรม get(row,col,pg,str,dec,typ,clr,bch
,l,r,ln)

ตัวแปรที่ส่งผ่าน

ตัวแปรรับเข้า

```
char row,col; /*ตำแหน่งบนจอภาพ ที่จะรับข้อมูลนี้*/
char pg;      /*หมายเลขจอ*/
char *str;    /*ที่อยู่ของตัวแปร*/
char dec;     /*จำนวนทศนิยม */
char typ;     /*ชนิดของข้อมูล*/
char  clr;    /*สีตัวอักขระ  หรือรูปแบบตัวอักขระ
                (attribute)*/
char  bch;    /*อักขระกั้นขอบเขต*/
char  l;      /*ขอบซ้ายมือ(left margin)*/
```

```
char r;        /*ขอบขวามือ (right margin)*/
char ln;       /*ความยาว*/
```

หน้าที่ เตรียมที่สำหรับการรับข้อมูลแบบเต็มจอ

รายละเอียดการทำงาน

รายละเอียดทุกอย่างจะถูกบันทึกอยู่ในตารางชื่อ grectab ข้อความจะถูกแสดง ออกทางจอภาพ โปรแกรมย่อยนี้ใช้คู่กับ โปรแกรม pgsay

ชื่อโปรแกรม vread(itc,c,exc)

ตัวแปรที่ส่งผ่าน

ตัวแปรรับเข้า

```
char itc;     /* ลำดับที่ของตัวแปรที่จะรับข้อมูลเข้า*/
```

ตัวแปรส่งออก

```
char *c,*exc; /* ตัวอักขระที่ส่งกลับ c = char.
                 exc = extend */
```

หน้าที่ รับข้อมูลเข้าสู่ตัวแปร

รายละเอียดการทำงาน

โปรแกรมนี้ใช้คู่กับ get ในการเรียก get นั้น ยังไม่ได้รับข้อมูลเข้าสู่ตัวแปรเพียงแต่เป็นการเตรียมที่บนจอภาพ ไว้สำหรับตัวแปรแต่ละตัว ดังนั้น itc จะนับตามลำดับของตัวแปรที่เรา get เมื่อผู้ใช้งานกดตัวอักขระที่ไม่ใช่คำสั่งของโปรแกรมบรรจุธิกร vread จะส่งตัวอักขระนั้นกลับมาใน c และ exc โดย c คืออักขระธรรมดา exc คืออักขระพิเศษ(Extended character) เช่น ฟังก์ชันคีย์

ชื่อโปรแกรม clrget()

ตัวแปรที่ส่งผ่าน ไม่มี

หน้าที่ เริ่มต้นการ get ใหม่

รายละเอียดการทำงาน

ในการ get นั้น โปรแกรม get จะเก็บรายละเอียดเอาไว้ในตาราง ดังนั้น การเรียก clrget ก็เพื่อที่จะลบรายละเอียดในตารางทิ้งไปและจัดค่าตัวแปรบางตัวเป็นค่าเริ่มต้นใหม่

ชื่อโปรแกรม getline(s,lim)

ตัวแปรที่ส่งผ่าน

ตัวแปรรับเข้า

```
char s;      /*ที่อยู่ของตัวแปรที่จะรับข้อมูล */
int lim;     /*ขนาดของ s*/
```

ตัวแปรส่งออก

```
char  *s;
```

หน้าที่ รับข้อมูลเข้าสู่ตัวแปรทีละบรรทัด

ชื่อโปรแกรม accept(str,lim,ip)

ตัวแปรที่ส่งผ่าน

ตัวแปรรับเข้า

```
char str;    /*ข้อความที่จะแสดง */
int lim;     /*ขนาดของ ip*/
char *ip;    /*ที่อยู่ของตัวแปรที่จะรับข้อมูล*/
```

ตัวแปรส่งออก

```
char  *ip;
```

หน้าที่ แสดงข้อความและรับข้อมูลเข้าสู่ตัวแปรทีละบรรทัด

รายละเอียดการทำงาน

ทำงานเหมือน getline เพียงแต่สามารถแสดงข้อความพร้อมกับการรับข้อมูลด้วย

ชื่อโปรแกรม tprint(s)

ตัวแปรที่ส่งผ่าน

ตัวแปรรับเข้า

```
char *s;    /*ข้อความที่จะแสดง */
```

หน้าที่ แสดงข้อความออกทางจอภาพ

รายละเอียดการทำงาน

แสดงข้อความออกทางจอภาพทีละบรรทัด (Line Display) โดยที่เราไม่สามารถกำหนดตำแหน่งได้ ตำแหน่งที่จะแสดงขึ้นอยู่กับว่าเคอเซอร์อยู่ ณ.ตำแหน่งไหน และจะเลื่อนเคอเซอร์ ไปทางขวามือให้เอง
