บทที่ 6
คำสั่งใน PSU*ProC

การพัฒนาระบบงานนอกจากจะใช้คำสั่งใน SQL แล้ว จำเป็นอย่างยิ่งที่จะต้องเขียนเป็นโปรแกรมด้วยภาษาต่างๆ เช่น COBOL, Pascal หรือ C เพื่อให้ทำงานได้นอกเหนือกว่าคำสั่งพื้นฐานที่มีอยู่ ดังนั้นในระบบ PSUbase จึงได้ออกแบบไว้ให้มีฟังก์ชันต่างๆ เพื่อให้โปรแกรมเมอร์ได้เขียนโปรแกรมติดต่อกับไฟล์ PSUbase ในระดับของ macro code แทนการใช้ฟังก์ชัน I/O พื้นฐานในภาษา C เช่นใช้ฟังก์ชัน use(fn,"r") แทนฟังก์ชัน fgets(fp) เป็นต้น และจุดเด่นอีกประการหนึ่งก็คือ การอางอิงข้อมูลในไฟล์จะกระทำในระดับฟิลด์ แทนการอางอิงทีละหนึ่งตัวอักษร จึงมีความยืดหยุ่นกว่า เช่น ถ้ามีการเพิ่ม/ลด ขนาดของฟิลด์ โปรแกรมก็ยังอางอิงข้อมูลได้ถูกต้อง ไม่ต้องแก้ไขตัวโปรแกรมมาก ใน PSUbase จะมี library ชื่อ libinx และ libdbf ซึ่งทำงานทางด้านอินเด็กซ์ไฟล์ และดาต้าเบส ตามลำดับ

6.1 หลักการเขียนโปรแกรมใน PSU*ProC

วิธีการเขียนโปรแกรมเพื่อเรียกใช้ไฟล์ชนิด PSUbase หรือภายใต้ PSU*ProC นั้น จะเขียนคล้ายคลึงกับโปรแกรมใน dBASE เช่น ใช้ฟังก์ชัน SELECT, USE, SKIP, EOF() และ SEEK เป็นต้น แต่เนื่องจาก PSU*ProC จะต้องใช้ตัวแปร (variable) ภายใต้ภาษา C มาตรฐาน (ANSI C) เพราะว่าเมื่อเขียนเสร็จแล้ว ต้องนำมาแปร (compile) ด้วย ANSI C ซึ่งต่างกับ dBASE ที่เป็น Interpreter ไม่อิงกับคอมไพเลอร์ และไม่ต้องแปรก่อนใช้งาน

เนื่องจากได้กำหนดให้มีชนิดของฟิลด์ในไฟล์ PSUbase ได้ 2 ชนิด คือ ตัวอักษร(char) กับ ตัวเลข (Num) ดังนั้นใน PSU*ProC จึงกำหนดให้มีตัวแปรได้ 2 ชนิด คือ char กับ double ในการอ้างอิงกับข้อมูลในแต่ละฟิลด์ ตัวอย่าง เช่น

```
char   *empno;
double salary;
```

ตัวแปร *empno ถูกกำหนดเป็นชนิด pointer of charater เพื่อให้เกิดความยืดหยุ่นในกรณีที่ความกว้าง ของฟิลด์ EmpNo ถูกเปลี่ยนไป ก็ยังทำให้โปรแกรมทำงานได้ถูกต้อง

เมื่อกำหนดตัวแปรได้แล้ว จะต้องกำหนดค่าที่อยู่ (address) หรือทำ Mapping เพื่อให้ตัวแปรนั้นชี้ไปยังข้อมูล แต่ละฟิลด์ในไฟล์ PSUbase โดยใช้ฟังก์ชัน mapchar() และ mapreal() สำหรับตัวแปร char และ double ตามลำดับ

เมื่อได้ตัวแปรในภาษา C ที่พร้อมจะใช้งานแล้ว ต่อไปให้ใช้ฟังก์ชัน select() เพื่อกำหนดพื้นที่ในการอ่านข้อมูลจากไฟล์ PSUbase

ขั้นตอนถัดไปให้ใช้ฟังก์ชัน use() เพื่อเปิดไฟล์ แล้วตามด้วยฟังก์ชัน gotop() เพื่ออ่านข้อมูลเร็คคอร์ดแรกลงพื้นที่ที่กำหนดไว้ และข้อมูลจะเข้าไปอยู่ในตัวแปร เช่น empno เลย สามารถนำไปใช้งานได้ทันที

ให้ใช้ฟังก์ชัน skip() ในการอ่านข้อมูลเร็คคอร์ดถัดๆไป

สรุปขั้นตอนเขียนโปรแกรมใน PSU*ProC

1. ตั้งชื่อตัวแปร เช่น char *empno; double salary;
2. ทำ Mapping ตัวแปร ในข้อที่ 1. เข้ากับที่อยู่ของข้อมูลในไฟล์ PSUbase เช่น

   mapchar("EmpNo", &empno);

   mapreal("Salary", &salary);
3. เลือกพื้นที่ เช่น select(1); หมายถึงจะใช้พื้นที่ที่ 1 ในการเก็บข้อมูลที่อ่านได้จากไฟล์
4. เปิดไฟล์ เช่น use("emp","r"); หมายถึงเปิดไฟล์ชื่อ "emp" เพื่ออ่านอย่างเดียว
5. ใช้ฟังก์ชัน gotop() เพื่ออ่านข้อมูลเร็คคอร์ดแรก
6. จะได้ข้อมูลเร็คคอร์ดแรกอยู่ในตัวแปร empno และ salary พร้อมที่จะใช้งานต่อไป
7. ใช้ฟังก์ชัน skip(1) เพื่ออ่านเร็คคอร์ดถัดไป
8. ใช้ฟังก์ชัน eof() เช็คว่าอ่านจบไฟล์แล้วหรือไม่ เช่น while (!eof()) skip(1);

6.2 ฟังก์ชันต่างๆใน PSU*ProC

ฟังก์ชันต่างๆ ที่ใช้จัดการกับไฟล์ PSUbase ภายใต้การเขียนโปรแกรม PSU*ProC ซึ่งใช้หลักการทำงานคล้ายๆ กับฟังก์ชันต่างๆ ใน dBASE ตัวอย่างเช่น

1. ฟังก์ชัน mapchar();
2. ฟังก์ชัน mapreal();
3. ฟังก์ชัน select();
3. ฟังก์ชัน use();
4. ฟังก์ชัน gotop();

5. ฟังก์ชัน skip();
6. ฟังก์ชัน eof();
7. ฟังก์ชัน seek();
8. ฟังก์ชัน found();
9. ฟังก์ชัน delete();
10. ฟังก์ชัน sayget();
11. ฟังก์ชัน setcolor();

6.3 ตัวอย่างโปรแกรมใน PSU*ProC

ตัวอย่างโปรแกรมภาษา C ภายใต้ PSU*ProC ได้แก่ sam1.c และ sam2.c ซึ่งมีรายละเอียดของแต่ละโปรแกรมดังต่อไปนี้

```
/****
* sam1.c: This is a example program #1 for calling the database routine
*       Written by Mongkol kuanhavet, 18 Jan 93
*       Computer center, Prince of Songkla Univ.(PSU), Hatyai, Thailand.
*       Under the VAX 11/785, ULTRIX Operating system v3.11
*
* compile: cc sam1.c /staff/mongkol/bin/libdbf /staff/mongkol/bin/libinx
*
* run:    a.out /staff/mongkol/bin/emp /staff/mongkol/bin/dept
*/
#include <stdio.h>
```

```
/****
* Define the constant name for refer to the field name in the database file.
*/
#define  EMPNO      "empno"
#define  NAME       "name"
#define  SNAME      "sname"
#define  SALARY     "salary"
#define  DEPTNO     "deptno"
#define  DEPTNAME   "deptname"

/****
/* Define the variable name for store the data from a field in database file.
*  The first character (A_ and B_) are show the useing work area 1 and 2.
*/
char  *A_empno;
char  *A_deptno;
char  *A_name;
char  *A_sname;
double A_salary;

char  *B_deptno;
char  *B_deptname;
```

```
/****
*  Map_A() is the function for mapping the field name with variable name
*          in the work area 1.
*/
map_A()
{
  mapchar(EMPNO,    &A_empno);
  mapchar(NAME,     &A_name);
  mapchar(SNAME,    &A_sname);
  mapchar(DEPTNO,   &A_deptno);
  mapreal(SALARY,   &A_salary);
}

/****
*  Map_B() is the function for mapping the field name with variable name
*          in the work area 2.
*/
map_B()
{
  mapchar(DEPTNO,   &B_deptno);
  mapchar(DEPTNAME,&B_deptname);
}
```

```
/****
* This is a main program sam1.c
*/
main(ac,av)
int ac;
char **av;
{
  char pdeptno[10];

  if (ac!=3)
    error("usage: sam1 emp dept"," ");

  select(1);                    /* select the work area #1 before use*/
  use(av[1],"r");               /* open the database file 'emp' */
  map_A();
  gotop();                      /* store data from 1st record to var. */

  select(2);                    /* select the work area #2 before use*/
  use(av[2],"r");               /* open the database file 'dept' */
  map_B();
  gotop();
```

```
  select(1);
  while (!eof()) {                          /* get data from database file */
    if (strcmp(pdeptno,A_deptno)!=0){       /* the deptno is change ? */
      select(2);
      seekdbf(A_deptno);                    /* seek the dept file with deptno*/
      if (found())                          /* if found print the dept name */
        printf("\nDept:%s %s\n",B_deptno,B_deptname);
      else
        printf("\nDept:%s Not found in dept\n",A_deptno);
      strcpy(pdeptno,A_deptno);
    }                                       /* print the detail from file emp */
    printf("    %s %s %s %7.0f\n",A_empno,A_name,A_sname,A_salary);
    select(1);                              /* select work area #1 */
    skip(1);                                /* skip one record of file emp */
  }
  closedata();                              /* close all the database file */
}
```

ผลลัพธ์ของโปรแกรม sam1.c คือ

```
Dept:200 RESEARCH
   3510001 JONES      MILLER        15000
   3510002 BLAKE      PITTY          5000
   3510003 CLARK      ARUN          25000
Dept:300 SALES
   3520001 JAMES      SMITH          8000
   3520002 ADAMES     SCOTT          7500
```

```
/****
* sam2.c: This is a example program #2 for calling the database library PSUbase
*        Written by Mongkol kuanhavet, 18 Dec 93
*        Computer center, Prince of Songkla Univ.(PSU), Hatyai, Thailand.
*        Under the SUN Sparc10, Solaris 2.1 Operating system
* compile: gcc -o sam7 sam7.c libinx libdb
*
* run:     sam7
*/
#include <stdio.h>
#include <ctype.h>
#include <signal.h>
#include <math.h>
#include "cst_db.h"

/* define the variable for refers to field name in PSUbase
*/
char   *empno;
char   *name;
char   *sname;
char   *status;
double dept;
double salary;
double minsal;
double maxsal;
```

```
/* Mapping the address of variable to the field name
*/
define_var()
{
  defchar("empno" ,&empno);
  defchar("name"  ,&name);
  defchar("sname" ,&sname);
  defchar("status",&status);
  defreal("dept"  ,&dept);
  defreal("salary",&salary);
  defreal("minsal",&minsal);
  defreal("maxsal",&maxsal);
}

/* This is a main program
*/
main(ac,av)
int ac;
char **av;
{
  char tmp[30];
  int  ret;

  define_var();
  clear();                    /* Clear the screen */
  initvar();                  /* Calling the Initvar() */
```

```
  initscr();                          /* Initial screen or change to RAW mode*/
  setcolor("w+");                     /* Set color to white and reverse mode */
  while (1) {
    ret=sayget(" \                    /* Calling sayget() for get the data */
      @ 01,02 say 'EMPNO     :' get empno  pict 'ccccc' \
      @ 03,02 say 'NAME      :' get name   pict '!!!!!!!!!!!!!!!!!!!!!!!!' \
      @ 05,02 say 'SNAME     :' get sname  pict 'ccccccccccccc' \
      @ 07,02 say 'Dept      :' get dept   pict '999' \
      @ 09,02 say 'Stat(Y,N) :' get status pict '!' valid 'YN' \
      @ 11,02 say 'SALARY    :' get salary pict '99999.99' range 1,10 \
    ");
    if (ret==CTRL_K)                  /* If control K is out off loop */
      break;
  }
  goyx(23,0);
  setcolor(" ");
  endscr();                           /* Reset the screen mode */
}
```

```
initvar()                    /* Initial the data in variable */
{
  empno="A1234";
  name="aaaaaaaaaaaaaaaaaaaaaaa";
  sname="bbbbbbbbbbbbbbb";
  dept=123;
  Status="Y"
  salary=567.56;
  minsal=0;
  maxsal=1000;
}
```

ผลลัพธ์ของโปรแกรม sam2.c คือ ข้อความที่แสดงบนหน้าจอภาพ และรอรับข้อมูลดังนี้

```
EMPNO     : A1234
NAME      : aaaaaaaaaaaaaaaaaaaaaaa
SNAME     : bbbbbbbbbbbbbbb
Dept      : 123
Stat(Y,N) : Y
SALARY    : 567.56
```