

## บทที่ 5

### คำสั่งใน PSU\*MENU

คำสั่งใน PSU\*MENU มีอยู่ 5 คำสั่งย่อยคือ say, menu\_bar, menu\_to sayget และ getvar ซึ่งรายละเอียดของแต่ละคำสั่งมีดังต่อไปนี้

#### 5.1 คำสั่ง say

##### หน้าที่

เป็นโปรแกรมสำหรับแสดงข้อความบนจอภาพ

##### รูปแบบ

```
say {row col text}
```

##### ความหมาย

row เป็นค่าของ row มีค่าตั้งแต่ 01-24

col เป็นค่าของ column มีค่าตั้งแต่ 01-79

##### ตัวอย่าง

```
say 01 01 'Hello'
```

##### หมายเหตุ

เหมาะสำหรับทำเมนูบาร์ ซึ่งจะกล่าวต่อไป

คำสั่งที่ต่อเนื่องกันคือ menu\_bar, menu\_to, sayget, getvar

## 5.2 คำสั่ง menu\_bar

หน้าที่

เป็นโปรแกรมสำหรับทำเมนูบาร์

รูปแบบ

```
menu_bar [color1 color2 color3] {text}
```

ความหมาย

color1 ใช้กำหนดแถบสีของเมนูบาร์(menu bar)

color2 ใช้กำหนดสีพื้น(background color)

color3 ใช้กำหนดสีของข้อความ "โปรดรอสักครู่"

(สีจะใช้ Escape sequence (ANSI.SYS) กำหนด เช่น ^[[31m=สีแดง

และเราจะกำหนดตัวแปร global ด้วยคำสั่ง setenv ในไฟล์ .login

เช่น setenv b\_red ^[[31m เป็นต้น ซึ่งสามารถเรียกใช้ได้ง่าย

ดูไฟล์ .login ใน /rdbms/aptool ประกอบ)

text เป็นข้อความของเมนูแต่ละรายการ

ประกอบด้วย "[@ row,col say 'x. Message'] [...]"

ตัวอย่าง

```
menu_bar "$b_blue" "$a_off$a_reverse" "$b_red" \  
    @ 06,40 say '1. Enter Data' \  
    @ 08,40 say '2. Compute Data' \  
    @ 10,40 say '3. Print Data' \  
    .
```

จากตัวอย่าง จะปรากฏเมนูบาร์ดังนี้

1. Enter Data (row 06, col 40)
2. Compute Data
3. Print Data

### หมายเหตุ

การขึ้นบรรทัดใหม่ใน text จะใช้ back slash(\) เป็นตัวต่อบรรทัด คำสั่งนี้ควรใช้ใน C shell และ ต้องให้ควบคู่กับคำสั่ง menu\_to ผู้ใช้จะเลื่อนลูกศร ขึ้น-ลง แล้วกด ENTER เพื่อเลือกรายการ หรือ กดตัวอักษรตัวแรกของเมนูก็ได้ เช่น กด 1 ซึ่งตัวอักษรตัวแรก จะมีค่าระหว่าง 1-9 และ A-Z เท่านั้น ส่วนค่า 0 จะสำรองไว้ใช้เมื่อกด F12 เพื่อเลิกงาน ค่า default ของ color1-3 จะเป็นสี ขาว-ดำ ดูตัวอย่าง Shell script ชื่อ ~/PSU/mis ประกอบ

## 5.3 คำสั่ง menu\_to

### หน้าที่

เป็นโปรแกรมสำหรับอ่านค่าตัวเลือกที่ผู้ใช้กดเลือกเมนู ใน menu\_bar

### รูปแบบ

```
menu_bar
....
....
set num=`menu_to`
```

### ความหมาย

คำสั่ง menu\_to จะให้ค่า 0-99 ออกมาเก็บไว้ในตัวแปร num

### หมายเหตุ

ค่าของ num จะมีค่าระหว่าง 0-99

โดยถ้าเลือกเมนูบรรทัดแรก ค่า num จะเป็น 1

บรรทัดที่ 2 ค่า num จะเป็น 2

บรรทัดที่ n ค่า num จะเป็น n

และกด F12 ค่า num จะเป็น 0 (ใช้สำหรับเลิกงาน)

ดูตัวอย่าง Shell script ชื่อ /rdbms/aptool/bin/mis ประกอบ

## 5.4 คำสั่ง sayget

หน้าที่

เป็นโปรแกรมสำหรับแสดงข้อความ และรับข้อมูลจากแป้นพิมพ์

รูปแบบ

```
sayget [color1 color2 color3] {text1 text2 ...}
```

ความหมาย

color1 ใช้กำหนดสีของข้อความ

color2 ใช้กำหนดสีบริเวณที่รองรับข้อมูล

color3 ใช้กำหนดสีข้อความ "โปรดรอสักครู่"

(สีจะใช้ Escape sequence ทำนองเดียวกับ menu\_bar)

text1 เป็นข้อความที่จะ say-get แต่ละรายการ

(ในกรณี text ยาวมาก สามารถแบ่งเป็น text1, text2,,)

ส่วนประกอบของ text คล้ายๆกับ menu\_bar แต่เพิ่มในส่วนของ

get field\_name รองรับข้อมูล

pict 'XXX' กำหนดขนาด และชนิดของฟิลด์

value 'YYY' ค่าเริ่มต้นในฟิลด์

(pict 'XXX' มีรูปแบบดังนี้

U ตัวอักษรใหญ่(Upper case)

L ตัวอักษรเล็ก (Lower case)

C ตัวอักษร (Character)

9 ตัวเลข (Numeric) เป็นทั้ง real และ integer)

range const1,const2 เป็นการกำหนดช่วงข้อมูล

เช่น range 1000,2000

valid 'ABC' เป็นการตรวจสอบการป้อนข้อมูลตัวอักษร

เช่น valid 'YN' ตัวอักษรที่ป้อนได้คือ Y หรือ N เท่านั้น

echo 'n' เป็นการกำหนดให้ echo หรือไม่

เช่น echo 'n' ไม่ echo และค่า default จะ echo

## ตัวอย่าง

```
sayget "$b_blue" "$a_off$a_reverse" "$b_red" \  
@ 04,40 say 'รหัสผ่าน??' : ' get paswd pict 'cccccc' \  
echo 'n' \  
@ 06,40 say 'จำนวนใหม่(y/n)' : ' get yn pict 'L' \  
value 'n' \  
valid 'yn' \  
@ 08,40 say 'รหัสคณะ(10-99)' : ' get fac pict '99' \  
value '10' \  
range 1,99 \  
.
```

จากตัวอย่าง จะปรากฏเมนูบาร์ดังนี้

```
รหัสผ่าน??      :  
จำนวนใหม่(Y/N) :  
รหัสคณะ(1-99) :
```

## หมายเหตุ

คำสั่งนี้ควรใช้ใน C shell และ ต้องใช้ควบคู่กับคำสั่ง getvar  
ส่วนค่า 0 จะสำรองไว้ใช้เมื่อกด ESC เพื่อเลิกงาน  
ค่า default ของ color1-3 จะเป็นสี ขาว-ดำ  
ค่า return จะเก็บไว้ในตัวแปรชื่อ return ซึ่งเป็นค่าเฉพาะ  
ใช้ใช้กรณีที่ยกเลิก คือ กด F12 ค่า return="F12"  
ดูตัวอย่าง Shell script ชื่อ /rdbms/aptool/bin/mis ประกอบ

## 5.5 คำสั่ง `getvar`

หน้าที่

เป็นโปรแกรมสำหรับอ่านค่าตัวแปรที่ได้จากคำสั่ง `sayget`

รูปแบบ

```
set var_name = `getvar var_name`
```

ความหมาย

จะใช้ร่วมกับคำสั่ง `set` และใช้หลังคำสั่ง `sayget`

ตัวอย่าง

หลังจากใช้คำสั่ง `sayget` ในหัวข้อที่ 5.4 แล้วสามารถใช้คำสั่ง `getvar` ได้ดังนี้

```
set return = `getvar return`
```

```
set yn = `getvar yn`
```

```
set fac = `getvar fac`
```

หมายเหตุ

จากตัวอย่างจะได้ค่าในตัวแปร `return`, `yn` และ `fac` เพื่อนำไปใช้งานต่อไป