บทที่ 2

ภาษาไทยกับการแสดงผลบนจอภาพ

ถ้าต้องการแสดงผลภาษาไทยอย่างถูกต้อง ต้องคำนึงถึงลักษณะการวางตัวอักขระซึ่งต่างจากภาษาอังกฤษ และต้องคำนึงถึงโครงสร้างข้อมูลที่สามารถรองรับการแสดงผล ดังจะกล่าวต่อไปในบทนี้

2.1 ลักษณะของภาษาไทย

ตัวอักขระไทยประกอบด้วย พยัญชนะ สระ และวรรณยุกต์ ตำแหน่งในการวางตัวอักขระของวรรณยุกต์และสระบางตัวไม่ได้อยู่บนบรรทัด ถ้าจัดระดับของอักขระไทยตามตำแหน่งการวางตัวอักขระจะแบ่งได้ 3 ระดับคือ

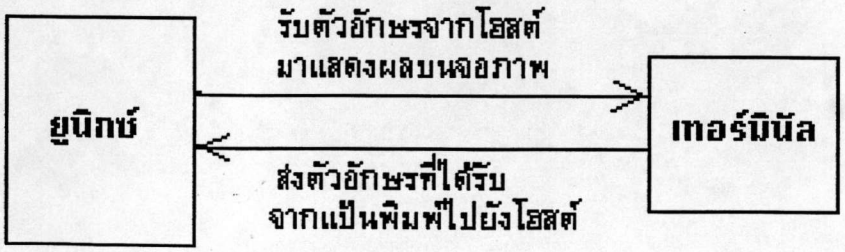
ระดับที่ 3	◌๊	◌์
ระดับที่ 2		◌ิ
ระดับที่ 1	ผ	ท
ระดับที่ 2	◌ุ	

ระดับที่ 1 เป็นตัวอักขระที่อยู่ในบรรทัด ประกอบด้วยพยัญชนะ และ สระบางตัวที่อยู่ในบรรทัด
ระดับที่ 2 เป็นตัวอักขระที่อยู่บนบรรทัด หรือล่างบรรทัด ประกอบด้วยสระบน และ สระล่าง
ระดับที่ 3 เป็นตัวอักขระที่อยู่บนบรรทัด เหนือระดับที่ 2 ประกอบด้วยวรรณยุกต์

รูปที่ 2.1 แสดงการจัดระดับของอักขระไทย

2.2 การแสดงผลภาษาไทย

การแสดงผลข้อมูลบนเทอร์มินัลมีลักษณะการทำงานดังนี้



รูปที่ 2.2 การทำงานของเทอร์มินัล

เทอร์มินัลมีหน้าที่ในการส่งข้อมูลที่ได้รับจากทางคีย์บอร์ดไปยัง โฮสต์ และรับข้อมูลจาก โฮสต์ มาแสดงผลบนจอภาพ

เมื่อต้องการแสดงผลภาษาไทยบนเทอร์มินัล ต้องทำให้เทอร์มินัลรู้จักภาษาไทย เพื่อให้แสดงผลตัวอักขระภาษาไทยได้ในตำแหน่งที่ถูกต้อง การทำให้เทอร์มินัลรู้จักภาษาไทย ทำได้โดย

ก.) ปรับปรุงฮาร์ดแวร์

ข.) โดยการใช้โปรแกรมเลียนแบบเทอร์มินัลร่วมกับไทยไดรเวอร์

ค.) โดยการใช้โปรแกรมเลียนแบบเทอร์มินัลที่จัดการภาษาไทยได้ในตัว

แต่วิธีการเหล่านี้ก็มีข้อจำกัดคือไม่สามารถ แสดงผล 80 คอลัมน์ได้ ทั้งการชดเชยสระยังทำให้เกิดข้อจำกัดในการแสดงผลดังที่ได้กล่าวในบทที่ 1

ถ้าสามารถแสดงผลภาษาไทยได้ถูกต้องตามลักษณะของภาษาไทย คือ แสดงผลตัวอักขระในระดับต่างๆ ได้ถูกต้อง และแสดงได้ 80 คอลัมน์จริง จะไม่มีความจำเป็นที่ต้องใช้การชดเชยสระ การแสดงผลก็จะเป็นไปอย่างเหมาะสม

2.3 การปรับปรุงเพื่อแสดงผลภาษาไทยให้เหมาะสม

เพื่อให้การแสดงผลภาษาไทยเป็นไปอย่างเหมาะสม การแก้ไขที่โปรแกรมเลียนแบบการทำงานของเทอร์มินัล หรือ แก้ไขที่โปรแกรมจัดการแสดงผลภาษาไทย เพียงอย่างเดียวไม่เพียงพอ เนื่องจากเทอร์มินัลรับข้อมูลมากจาก โฮสต์ ถ้า โฮสต์ ส่งข้อมูลในลักษณะที่ไม่รู้จักภาษาไทยมา โปรแกรมเลียนแบบเทอร์มินัลจะไม่สามารถจัดการให้มีการแสดงผล 80 คอลัมน์ได้

บนยูนิกซ์มีไลบรารีสำหรับงานแสดงผลชื่อเคิร์ส ถูกพัฒนาขึ้นที่มหาวิทยาลัยแคลิฟอร์เนียที่เบิกเลย์ (University of California at Berkeley) ไลบรารีนี้ถูกพัฒนาเพิ่มเติม และเป็นส่วนหนึ่งของ AT&T System V

เคิร์ส ประกอบไปด้วยชุดคำสั่งสำหรับการจัดการจอภาพและแป้นพิมพ์ โดยหลักของเคิร์สคือโครงสร้างของชุดคำสั่งจะอยู่ในระดับสูง ไม่เข้าไปยุ่งเกี่ยวกับคุณสมบัติทางกายภาพของเทอร์มินัล จึงทำให้โปรแกรมที่เขียนโดยใช้เคิร์ส สามารถเคลื่อนย้ายไปใช้กับเทอร์มินัลต่างๆ ได้สะดวก ส่วนข้อมูลเกี่ยวกับเทอร์มินัลซึ่งมีหลายชนิดบนระบบปฏิบัติการยูนิกซ์ เคิร์สจะใช้ผ่านทางเทอมอินโฟ(terminfo) อีกที่หนึ่ง โดยเทอมอินโฟ(terminfo) มีฐานข้อมูลที่เก็บรายละเอียดของเทอร์มินัลชนิดต่างๆ

การทำให้ โฮสต์ เสมือนรู้จักภาษาไทย ทำได้โดยให้โปรแกรมที่ทำงานบน โฮสต์ รู้จักภาษาไทย โดยให้โปรแกรมเรียกใช้ฟังก์ชันของเคิร์สที่รู้จักภาษาไทย ในการจัดการแสดงผล ต้องปรับปรุงโปรแกรมเลียนแบบเทอร์มินัล เพื่อให้สอดคล้องกับการทำงานของไลบรารีด้วย

2.4 การพัฒนาโปรแกรมโดยใช้เคิร์สไลบรารี

เคิร์ส เป็นไลบรารีที่ช่วยในการพัฒนาโปรแกรมแสดงผลทางจอภาพ (screen base) ที่มีการโต้ตอบกับผู้ใช้ (interactive) เป็นไลบรารีสำหรับภาษาซี ภายใต้ระบบปฏิบัติการยูนิกซ์

ในหัวข้อนี้จะกล่าวถึงการใช้เคิร์สในการพัฒนาโปรแกรม และการทำงานของเคิร์สอย่างย่อๆ

โปรแกรมที่ใช้เคิร์สในการพัฒนา ต้องมีโครงสร้างดังต่อไปนี้

1. เรียกใช้ไฟล์ curses.h

2. เรียกใช้ฟังก์ชัน initscr() เพื่อเข้าสู่เคิร์ส

3. กำหนดสถานะของอินพุท/เอาพุท (I/O mode) เช่นการกำหนดให้เทอร์มินัลเป็น raw mode หรือการกำหนดให้มีการ echo อักขระที่ได้รับทางจอภาพ

4. เรียกใช้ฟังก์ชันของเคิร์สที่ต้องการ

5. เรียกฟังก์ชัน endwin() เพื่อกลับคืนสู่ภาวะปกติก่อนเข้าสู่เคิร์ส

จากนั้นนำโปรแกรมมาทำการคอมไพล์ โดยต้องอ้างถึงเคิร์สไลบรารีดังต่อไปนี้

```
cc MyProgram.c -lcurses -o MyProgram
```

ก่อนนำโปรแกรมที่ได้จากการคอมไพล์ไปใช้งาน ต้องกำหนดประเภทของเทอร์มินัลที่ใช้ให้ถูกต้อง โดยกำหนดด้วยตัวแปรระบบ (environment variable) TERM ดังตัวอย่างต่อไปนี้

```
export TERM=vt100
```

เป็นการกำหนดว่าประเภทของเทอร์มินัลที่ใช้คือ vt100 กรณีที่ทำงานบน korn shell

2.5 การทำงานของเคิร์ส

เคิร์สใช้วินโดว์เป็น data structure ในการเก็บข้อมูลที่ต้องการให้แสดงผลบนเทอร์มินัล โดยขนาดของวินโดว์กำหนดได้โดยผู้ใช้ และสามารถมีได้หลายวินโดว์ในขณะเดียวกัน

เคิร์สจะปรับปรุงการแสดงผลบนเทอร์มินัลตามข้อมูลที่กำหนดไว้ใน data staructure WINDOW เมื่อมีการเรียกใช้ฟังก์ชัน refresh() โดยเคิร์สใช้ 2 WINDOW คือ curscr และ stdscr

curscr จะเก็บข้อมูลที่กำลังแสดงบนเทอร์มินับในขณะนั้น

stdscr เป็น default วินโดว์ของการทำงานที่ไม่ระบุ WINDOW

นอกจากนี้เราสามารถสร้างวินโดว์อื่นๆ ได้ตามต้องการ เมื่อต้องการปรับปรุงให้จอภาพของเทอร์มินัลแสดงผลตามข้อมูลในวินโดว์ สามารถทำโดยเรียกใช้ฟังก์ชัน refresh() กรณีที่วินโดว์เป็น stdscr และเรียกใช้ฟังก์ชัน wrefresh() กรณีต้องการระบุวินโดว์

เนื่องจากเคิร์สมีหลายเวอร์ชัน โครงสร้างของวินโดว์ในแต่ละเวอร์ชันจะแตกต่างกันออกไป โดยโครงสร้างของวินโดว์จะกำหนดไว้ใน curses.h ดังตัวอย่างต่อไปนี้

```
struct _win_st
{
        short   _cury, _curx;
        short   _maxy, maxx;
        short   _begy, begx;
        short   _flags;
        chtype  _attrs;
        bool    _clear;
        bool    _leave;
        bool    _scroll;
        bool    _use_idl;
        bool    _use_keypad;
        bool    _use_meta;
```

```
    bool    _nodelay;
    chtype  **_y;
    short   *_firstch;
    short   *_lastch;
    short   _tmarg, _bmarg;
};
typedef struct _win_st WINDOW;
extern WINDOW *stdscr, *curscr;
```

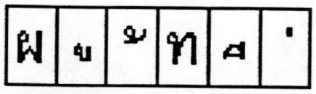
โครงสร้างของข้อมูลของ WINDOW จะเก็บข้อมูลที่จำเป็นสำหรับเคิร์สในการจัดการกับจอภาพของเทอร์มินัล เคิร์สจะจัดการกับข้อมูลต่างๆ ในโครงสร้างนี้ ผู้ใช้ไม่ควรเข้าไปจัดการโดยตรงกับข้อมูลในโครงสร้างนี้ การปรับปรุงค่าเหล่านี้ควรทำผ่านฟังก์ชันของเคิร์ส ต่อไปเป็นตัวอย่างของข้อมูลที่เก็บในโครงสร้างข้อมูล WINDOW

- _cury และ _curx เก็บตำแหน่งปัจจุบัน ตัวอักขระที่จะถูกเพิ่มเข้าไปในวินโดว์ จะถูกเพิ่มเข้าไปในตำแหน่งปัจจุบันที่กำหนดด้วย _cury และ _curx
- _maxx และ _maxy บอกขนาดของวินโดว์ที่ต้องการ ซึ่งอาจเล็ก หรือ ใหญ่กว่า จอภาพจริงได้
- _begx และ _begy บอกตำแหน่งเริ่มต้น ของวินโดว์ โดยอ้างอิงจาก stdscr
- _attrs attribute ของวินโดว์ เมื่อมีการเพิ่มอักขระเข้าในวินโดว์ อักขระนั้นจะมี attribute ตาม _attrs
- _flags เคิร์สใช้ในการทำ optimization
- _scroll บอกว่าวินโดว์สามารถ scroll ได้หรือไม่
- _nodelay wgetch() ฟังก์ชันใช้ในการตัดสินใจว่าการรับ input จะใช้ input queue หรือไม่
- -firstch และ _lastch เป็น arrays ที่ curses ใช้ช่วยในการทำ optimization โดยจะเป็น variable ที่บอกตำแหน่งเริ่มต้น และสิ้นสุดของการเปลี่ยนแปลงในแต่ละบันทัด

เนื่องจากเทอร์มินัลของยูนิกซ์มีหลายชนิด เพื่อให้โปรแกรมที่พัฒนาโดยเคิร์สสามารถทำงานได้บนเทอร์มินัลทุกประเภท เคิร์สจึงใช้ฟังก์ชันในไลบรารี terminfo ในการสอบถามรหัสควบคุมที่ใช้กับเทอร์มินัลที่กำหนดในตัวแปรระบบ TERM

2.6 ภาษาไทยกับโครงสร้างข้อมูลที่เหมาะสม

ภาษาไทยเป็นภาษาที่มีตัวอักขระอยู่ในระดับต่างๆกันในบรรทัด คือมีการแสดงผลใน 2 มิติ ในขณะที่ลักษณะโครงสร้างข้อมูลที่ใช้ในการแสดงผลทั่วไปเป็นแบบ 1 มิติ คือจะเก็บข้อมูลเป็นสายอักขระ (string) ในแนวนอน ดังรูป 2.3



รูปที่ 2.3 ลักษณะการเก็บข้อมูลเป็นสายอักขระ

ลักษณะการเก็บข้อมูลเช่นนี้ ไม่สัมพันธ์กับการแสดงผลภาษาไทย 3 ระดับ ความยาวของข้อมูลและการนับคอลัมน์ของข้อมูลในโครงสร้างข้อมูลไม่ตรงกับการแสดงผลบนจอภาพ ปัญหาเหล่านี้แก้ไขได้โดยใช้โครงสร้างข้อมูลแบบ 2 มิติ ซึ่งเก็บข้อมูลในลักษณะเดียวกับที่แสดงผล ดังรูป 2.4

รูปที่ 2.4 ลักษณะการเก็บข้อมูลแบบ 2 มิติ

ลักษณะการเก็บข้อมูลเช่นนี้จะสัมพันธ์กับการแสดงผลภาษาไทย 3 ระดับ สามารถกำหนดคอลัมน์ที่ตรงกันได้ และจำนวนคอลัมน์บนจอภาพและในโครงสร้างข้อมูลตรงกัน

2.7 การปรับปรุงการแสดงผลภาษาไทยบนเทอร์มินัล

การปรับปรุงการแสดงผลภาษาไทยบนเทอร์มินัล อาจปรับปรุงได้ 2 วิธีดังต่อไปนี้

ก) ปรับปรุงให้เคิร์ส และ โปรแกรมเลียนแบบเทอร์มินัล เป็นแบบ 2 มิติ

ข) ปรับปรุงให้เคิร์สเป็นแบบ 2 มิติ ส่วนโปรแกรมเลียนแบบเทอร์มินัลเป็นแบบ 1 มิติเช่นเดิม แต่ปรับปรุงให้รองรับจำนวนคอลัมน์ให้มากพอ

ในวิทยานิพนธ์นี้เลือกข้อ ข) เนื่องจากเป็นวิธีที่มีการเปลี่ยนแปลงน้อยที่สุด ซึ่งต้องมีการปรับปรุง 2 ส่วนคือส่วนที่เป็นเคิร์สและส่วนที่เป็นโปรแกรมเลียนแบบเทอร์มินัล

ในการปรับปรุงเคิร์ส ให้เป็น 2 มิติ ทำได้ 2 วิธีคือ

ก) แก้ไขโปรแกรมเดิม

ข) เขียนเคิร์สขึ้นใหม่ ให้ครอบคลุมฟังก์ชั่นเคิร์สเดิมที่จำเป็น

ในวิทยานิพนธ์นี้เลือกทำข้อ ข) เนื่องจาก

- การแก้ไขเป็นการปรับปรุงโครงสร้างข้อมูลสำคัญ คือโครงสร้างข้อมูลที่ใช้เก็บข้อมูลในแต่ละคอลัมน์ ซึ่งจะกระทบหลายฟังก์ชัน เคิร์สเป็นโปรแกรมที่ซับซ้อน การเข้าไปแก้ไขในหลายฟังก์ชั่นจำเป็นต้องรู้โครงสร้างทั้งหมดเป็นอย่างดี ซึ่งทำได้ยาก และเมื่อเกิดความผิดพลาด

จะยากต่อการตรวจหา

- โปรแกรม (source code) ที่หาได้เป็นของเวอร์ชั่นที่ไม่ตรงกับคำอธิบายในหนังสือที่ใช้อ้างอิง ตัวอย่างที่พบ คือ ไม่มีฟังก์ชัน wnoutrefresh() เป็นต้น

- เมื่อลองใช้เคิร์สเดิม แสดงผลภาษาไทยจะพบว่ามีปัญหาในฟังก์ชั่ง บางฟังก์ชัน เช่น addch(), insch() จะแสดงผลไม่ถูกต้อง

ในบทต่อไปจะกล่าวถึงฟังก์ชั่นของเคิร์สที่พัฒนาขึ้นและการปรับปรุงโปรแกรมเลียนแบบเทอร์มินัล