บทที่ 2

ลักษณะของจอควบคุมระบบแบบกราฟิก

ในบทนี้เป็นการอธิบายเรื่องการแสดงผลทางจอภาพ ทั้งการแสดงผลแบบข้อความ และการแสดงผลกราฟิกบนระบบยูนิกซ์ ระบบที่ 5 รุ่น 4.0 จุดมุ่งหมายของบทนี้คือ ให้ทราบวิธีในการติดต่อและควบคุมแผ่นวงจรแสดงผลทางจอภาพ (video adapter board) บนระบบยูนิกซ์

จอควบคุมบนระบบยูนิกซ์ คือ อุปกรณ์รับข้อมูลเข้า/ส่งข้อมูลออกที่ต่อโดยตรงกับแผ่นวงจรควบคุมการแสดงผลบนตัวคอมพิวเตอร์แม่ข่าย ซึ่งทำให้จอควบคุมมีความสามารถที่จะทำงานในภาวะกราฟิกได้ ขึ้นอยู่กับชนิดของแผ่นวงจรควบคุมการแสดงผลที่ใช้อยู่ โดยลักษณะของการทำงานในการติดต่อกับแผ่นวงจรควบคุมการแสดงผล สามารถแบ่งเป็นส่วนๆได้ดังต่อไปนี้

<u>แผ่นวงจรควบคุมการแสดงผลบนจอภาพ</u>

การแสดงผลทางจอภาพจะถูกควบคุมโดย แผ่นวงจรควบคุมการแสดงผลทางจอภาพโดยชนิดของแผ่นวงจรควบคุมการแสดงผล จะเป็นตัวกำหนดความละเอียดของการแสดงผล และความสามารถในการแสดงสีทั้งภาวะข้อความ (text mode) และภาวะกราฟิก (graphic mode)

สิ่งแรกที่จะต้องคำนึงถึงในการเขียนโปรแกรมควบคุมการแสดงผลทางจอภาพ คือ การกำหนดชนิดของแผ่นวงจรควบคุมการแสดงผลที่จะใช้ เพื่อที่จะสามารถใช้ประโยชน์จากแผ่นวงจรนั้นได้สูงสุด

หน่วยความจำ (memory)

การเข้าถึงฮาร์ดแวร์จะกระทำโดยผ่านเคอร์เนลของระบบปฏิบัติการยูนิกซ์ (UNIX system kernel) และ โปรแกรมควบคุมอุปกรณ์ ในกรณีของจอควบคุมแบบกราฟิก จะถูกควบคุมโดยโปรแกรมควบคุมการแสดงผล "kd" (kd display driver) เพื่อเป็นการป้องกันไม่ให้เขียนโปรแกรมเข้าถึงฮาร์ดแวร์โดยตรงในระดับหนึ่ง การติดต่อกับฮาร์ดแวร์จะต้องทำผ่านคำสั่ง ioctl ซึ่งเป็นคำสั่งในการเชื่อมต่อกับระบบยูนิกซ์ (UNIX system call interface) ตัวอย่างการใช้งานคำสั่งนี้ได้แก่ การเลือกภาวะ (mode) ในการแสดงผลการตั้งสถานะของแผ่นวงจรควบคุม การรับส่งข้อมูลกับแผ่นวงจรควบคุมการแสดงผล การดึงและกำหนดค่าสำหรับการแสดงผลทางจอภาพ ในการที่จะติดต่อกับแผ่นวงจรควบคุมส่วนสำคัญที่จำเป็นต้องมี คือ ความสามารถที่จะจัดการกับหน่วยความจำส่วนการแสดงผล

ในลักษณะการทำงานของนักเขียนโปรแกรมที่ต้องการแสดงผลข้อมูล ออกทางจอภาพจะต้องทำการติดต่อกับแผ่นวงจรควบคุมการแสดงผล โดยการติดต่อกับหน่วยความจำในการแสดงผล การเปลี่ยนค่าภายในหน่วยความจำในการแสดงผลจะทำให้การแสดงผลทางหน้าจอเปลี่ยนไปทันที สิ่งสำคัญในการเขียนโปรแกรมจัดการจอภาพอย่างหนึ่งคือจะต้องทราบความสัมพันธ์ระหว่างการจัดเรียงของหน่วยความจำและการแสดงผลบนจอภาพ การจัดเรียงหน่วยความจำจะมีลักษณะเป็นอย่างไรขึ้นกับแผ่นวงจรแสดงผล และภาวะที่แผ่นวงจรแสดงผลใช้อยู่ซึ่งจะกล่าวถึงต่อไป

รีจิสเตอร์ (register)

การปฏิบัติการจริงๆ ของแผ่นวงจรควบคุมการแสดงผล จะถูกควบคุมโดยรีจิสเตอร์บนแผ่นวงจรควบคุมการแสดงผลนั้น บนระบบปฏิบัติการยูนิกซ์ รุ่นที่ 4.0 โปรแกรมควบคุมการแสดงผล "kd" จะเป็นตัวจัดการกำหนดค่ารีจิสเตอร์ควบคุมการแสดงผล ในขณะที่เลือก "ภาวะ" การแสดงผลเป็นแบบกราฟิก ในโปรแกรมย่อย "kd" นี้จะไม่มีการป้องกัน

การผิดพลาดที่เกิดจากการเขียนโปรแกรม ในบางกรณีนักเขียนโปรแกรมจะได้รับอนุญาตให้อ่านและเขียนลงบนรีจิสเตอร์ควบคุมการแสดงผลได้ แต่จะต้องแน่ใจในผลที่จะเกิดขึ้นมิฉะนั้นจะเกิดข้อผิดพลาดที่เป็นผลเสียต่อระบบได้

ภาวะข้อความและภาวะกราฟิก

การเขียนโปรแกรมสำหรับการแสดงผลทางจอภาพ ในภาวะข้อความและภาวะกราฟิกจะแตกต่างกันโดยสิ้นเชิง แผ่นวงจรควบคุมการแสดงผล จะสามารถอยู่ในภาวะใดภาวะหนึ่งเท่านั้น ดังนั้นนักเขียนโปรแกรมจะต้องเขียนโปรแกรมควบคุมการแสดงผลในภาวะใดภาวะหนึ่งเท่านั้น ไม่สามารถเขียนโปรแกรมให้ควบคุมทั้ง 2 สภาวะได้ในเวลาเดียวกัน

ภาวะข้อความ

ภายใต้ภาวะข้อความนักเขียนโปรแกรมสามารถที่จะกำหนดรูปแบบข้อความ (text format) กลุ่มของอักขระ (character set) ที่จะใช้ สีของตัวอักขระ (foreground color) และสีของพื้นหลังตัวอักขระ (background color) สามารถควบคุมการเคลื่อนที่ของตัวชี้ตำแหน่ง (cursor) และสามารถกำหนดการแปลงส่งระหว่างตัวอักขระกับแป้นพิมพ์ (character mapping) ภายใต้ภาวะข้อความความละเอียดในการแสดงผลทางจอภาพจะวัดในรูปของจำนวนตัวอักขระที่สามารถแสดงได้ภายใน 1 หน้าของจอภาพ เช่น ความละเอียดขนาด 80x25 คือในหนึ่งบรรทัดจะมีได้ 80 ตัวอักขระ และมี 25 บรรทัด ใน 1 หน้า ภาวะข้อความจะเป็นภาวะเริ่มต้นของการแสดงผลโดยทั่วไป

ในภาวะนี้สามารถเลือกลักษณะตัวอักขระที่แตกต่างกันได้ เช่น สามารถเลือกตัวอักขระที่เป็นกราฟิก ได้แก่ ลูกศร เส้นตรง เป็นต้น เนื่องจากภาวะข้อความจะทำงานได้เร็วและง่ายต่อการใช้มากกว่าภาวะกราฟิก จึงควรพิจารณาว่าจะเขียนโปรแกรมประยุกต์ต่างๆ บนภาวะข้อความ หรือ ภาวะกราฟิก

การแสดงผลในภาวะข้อความแบ่งออกได้เป็น 3 ชนิด ได้แก่

1. การแสดงผลตัวอักขระมาตรฐาน (standard character) การแสดงผลแบบนี้จะแสดงผลตัวอักขระที่สามารถพิมพ์ออกมาได้ เช่น พยัญชนะ สระ

2. การแสดงผลตัวอักขระที่ไม่มาตรฐาน (non-standardcharacter) จะแสดงผลตัวอักขระที่ไม่สามารถพิมพ์ออกมาได้ เช่น ตัวอักขระแทนคำสั่งในการป้อนกระดาษ (formfeed character) ผลลัพธ์ที่ได้จะเป็นการลบข้อความบนหน้าจอ เป็นต้น

3. การแสดงผลตัวอักขระที่เป็นลำดับของรหัสหลีก (escape code sequences) การแสดงผลแบบนี้จะใช้ในการควบคุมการเคลื่อนที่ของตัวชี้ตำแหน่ง (cursor) การกำหนดค่าสำหรับแป้นกำหนดหน้าที่ (function keys) และกำหนดลักษณะประจำ (attributes) สำหรับแต่ละตัวอักขระ

ภาวะกราฟิก

ภายใต้ภาวะกราฟิก การแสดงผลจะอยู่ในรูปตำแหน่งของจุด (addressable points) เรียกว่า จุดภาพ (pixels) ชนิดของแผ่นวงจรควบคุม และ ภาวะในการแสดงผลจะเป็นตัวกำหนดความละเอียดของจอภาพ และความสามารถในการแสดงสี นักเขียนโปรแกรมสามารถกำหนดการแสดงผลในแต่ละจุด และกำหนดสีในการแสดงผลได้ นักเขียนโปรแกรมสามารถที่จะอ่านและกำหนดค่าของรีจีสเตอร์ ในแผ่นวงจรควบคุมการแสดงผลได้ทั้งในภาวะข้อความและภาวะกราฟิก

นอกจากนี้ยังมีคำสั่งเพิ่มเติมในการควบคุมแผ่นวงจรควบคุมการแสดงผล ซึ่งอนุญาตให้อ่านและกำหนดสถานะของแป้นพิมพ์ การสร้างเสียงสูงต่ำ และฟังก์ชันอื่นๆ

ขั้นตอนในการเขียนโปรแกรมควบคุมการแสดงผลบนจอภาพ

ก่อนที่จะเริ่มเขียนโปรแกรมจะต้องคุ้นเคยกับคำสั่ง ioctl และแฟ้มหัวเรื่อง (header file) คือ /usr/include/sys/kd.h ซึ่งแฟ้มหัวเรื่องนี้จะเป็นสิ่งที่กำหนดค่าที่จะนำมาใช้ในคำสั่ง ioctl เพื่อควบคุมการแสดงผล นอกจากนี้ยังมีการกำหนดโครงสร้างซึ่งจะอนุญาตให้นักเขียนโปรแกรมอ่านและเขียน รีจีสเตอร์ควบคุม (control register) บนแผ่นวงจรการแสดงผล

แฟ้มหัวเรื่องอื่นๆที่จำเป็นต้องใช้ ได้แก่ /usr/include/sys/types.h และ /usr/include/sys/at_ansi.h

นอกจากนี้สิ่งที่นักเขียนโปรแกรมต้องทราบคือ ภาวะที่ต้องการใช้งานและใช้แผ่นวงจรควบคุมการแสดงผลชนิดใด ซึ่งในการทำวิทยานิพนธ์นี้จะใช้ภาวะกราฟิกในการทำงาน และใช้แผ่นวงจรควบคุมการแสดงผลแบบวีจีเอ

การเขียนโปรแกรมเมื่ออยู่ในภาวะข้อความ

ขั้นตอนการเขียนโปรแกรมเมื่ออยู่ในภาวะข้อความ มีดังนี้

1. กำหนดค่าลำดับของรหัสหลีกที่จำเป็นต้องใช้
2. เปิดโปรแกรมควบคุมอุปกรณ์ "kd"
3. กำหนดว่าใช้แผ่นวงจรควบคุมการแสดงผลชนิดใด
4. เข้าสู่ภาวะข้อความ
5. ลบข้อมูลบนจอภาพ
6. แสดงผลแบบข้อความ

การแสดงผลโดยทั่วไปมี 2 วิธีได้แก่

1. การแสดงผลโดยใช้คำสั่งพิมพ์ เช่น คำสั่ง printf ในภาวะนี้สามารถใช้ลำดับของรหัสหลีก ในการควบคุมสีของตัวอักขระ , สีพื้นหลังตัวอักขระ และ ลักษณะประจำของตัวอักขระ เช่น การกระพริบ (blink),ขีดเส้นใต้ (underline)

2. เขียนโดยตรงลงไปที่หน่วยความจำในการแสดงผล (video memory) เนื้อที่หน่วยความจำที่จะใช้ได้เป็นจำนวนเท่าของ 16 กิโลไบท์ ขึ้นอยู่กับใช้แผ่นวงจรควบคุมการแสดงผลชนิดใด

<u>การเขียนโปรแกรมเมื่ออยู่ในภาวะกราฟิก</u>

เพื่อที่จะใช้งานในภาวะกราฟิกได้อย่างมีประสิทธิภาพ นักเขียนโปรแกรมควรสร้างเป็นคลังโปรแกรม (library program) ของโปรแกรมทางด้านกราฟิกที่เป็นฟังก์ชันของการเขียนข้อมูลลงในหน่วยความจำ และ เชื่อมโยงหน่วยความจำของผู้ใช้เข้ากับหน่วยความจำของแผ่นวงจรควบคุมการแสดงผลเพื่อแสดงผลบนจอภาพ

ขั้นตอนในการเขียนโปรแกรมเมื่ออยู่ในภาวะกราฟิก มีดังนี้

1. เปิดโปรแกรมควบคุมอุปกรณ์ "kd"
2. กำหนดว่าจะใช้แผ่นวงจรควบคุมการแสดงผลแบบใด
3. กำหนดภาวะเป็นภาวะกราฟิก (กำหนดสีที่จะใช้ และ ความละเอียดของการแสดงผลบนจอภาพ)
4. ลบจอภาพ โดยการเขียนสีที่เหมาะสมลงในหน่วยความจำในส่วนของการแสดงผล
5. แสดงผลข้อมูลแบบกราฟิก

เทคนิคทั่วไปในการเขียนโปรแกรมแสดงผลทางจอภาพ

ส่วนควบคุมการแสดงผลทางจอภาพแบ่งออกเป็นหลายชนิดดังนี้

1. ตัวปรับภาพแบบสีเดียว (MDA หรือ monochrome display adapters) สามารถแสดงสีได้ 2 สี คือ สีดำ และ สีอื่นอีกหนึ่งสี ความละเอียดของแผ่นวงจรควบคุมการแสดงผลแบบนี้ประกอบด้วย 720 จุดภาพ ตามแนวนอน และ 350 จุดภาพ ตามแนวตั้ง หรือ 720x350 จุดภาพ

2. ตัวปรับภาพสี (CGA หรือ color graphics adapter) แบ่งออกได้เป็น 2 แบบคือ

- แสดงสีได้ 4 สี มีความละเอียด 320x200 จุดภาพ
- แสดงสีได้ 2 สี มีความละเอียด 640x200 จุดภาพ

3. ตัวปรับภาพเพิ่มความคม (EGA หรือ enhanced graphics adapter) สามารถแสดงสีได้ 16 สี มีความละเอียด 640x350 จุดภาพ

4. ตัวปรับภาพแบบวีจีเอ (VGA หรือ video graphics adapter) แบ่งออกเป็น 2 แบบคือ

- แสดงสีได้ 16 สี ความละเอียด 640x480 จุดภาพ
- แสดงสีได้ 256 สี ความละเอียด 320x200 จุดภาพ

นอกจากนี้ยังมี ตัวปรับภาพสีเดียวแบบเฮอร์คิวลิส (hercules monochrome graphics adapter) และ ตัวปรับภาพแบบอื่นๆอีก โดยแผ่นวงจรควบคุมการแสดงผลแต่ละแบบจะมีความสามารถในการแสดงผล ,รีจีสเตอร์,และ ลักษณะการแปลงส่งหน่วยความจำ (memory mapping) ที่แตกต่างกัน

ขั้นตอนในการควบคุมแผ่นวงจรควบคุมการแสดงผลมี 2 ขั้นตอนคือ

1. เปิดโปรแกรมควบคุมอุปกรณ์
2. ดึงข้อมูลของแผ่นวงจรควบคุมการแสดงผล

คำสั่งที่ใช้ในการเปิดอุปกรณ์ คือ

```
open ("/dev/video",O_RDWR);
```

การเปิดแฟ้ม /dev/video เป็นการควบคุมการแสดงผลทางจอภาพในรูปของเครื่องปลายทางเสมือน (virtual terminal) การเปิดแฟ้มนี้จะขัดข้องถ้าเป็นการต่อโดยใช้สายต่อเครื่องปลายทางแบบอนุกรม (serial terminal port) ถ้าการเปิดอุปกรณ์ขัดข้อง จะมีการส่งค่ากลับที่เป็น "ลบ" และจะใช้คำสั่ง ioctl ใช้เพื่อการดึงข้อมูลการกำหนดภาวะการใช้งาน เป็นต้น

เนื่องจากมีการพัฒนาการใช้งานมาเรื่อยๆ ทำให้คำสั่งนี้มีหลายรุ่น (Version) ดังนี้

```
ioctl(fd,KIOCINFO,0)
```

คำสั่งนี้ใช้ในกรณีที่อุปกรณ์การแสดงผลทางจอภาพถูกควบคุมโดย โปรแกรมควบคุมอุปกรณ์ "kd" ถ้าสามารถทำคำสั่งนี้ได้สำเร็จ จะมีการส่งค่ากลับเป็น ('k'<<8¦'d')

หรือ `ioctl(disp,KDVDCTYPE,&disp_info);`

&disp_info คือตำแหน่งโครงสร้างข้อมูล ซึ่งมีรูปแบบดังนี้

```
struct kd_vdctype disp_info;
```

ค่าที่ส่งกลับจากคำสั่ง ioctl ที่เก็บอยู่ในโครงสร้างข้อมูลนี้จะถูกกำหนดไว้ในแฟ้มข้อมูล /usr/include/sys/kd.h ซึ่งแสดงได้ดังตารางข้างล่างนี้

ค่าที่ส่งกลับ	ชนิดของแผ่นวงจรควบคุมการแสดงผล
KD_MONO	IBM monochrome display adapter
KD_HERCULES	Hercules monochrome adapter
KD_CGA	IBM colorgraphics adapter
KD_EGA	IBM enhanced graphics adapter
KD_VGA	IBM video graphics adapter
KD_VDC 400	AT&T VDC 400 adapter
KD_VDC 750	AT&T VDC 750 adapter
KD_VDC 600	AT&T VDC 600 adapter

ตาราง 2.1 แสดงค่าส่งกลับที่บอกถึงชนิดของแผ่นวงจรควบคุมการแสดงผลทางจอภาพ

ค่าที่ส่งกลับ	ชนิดของจอภาพ
KD_UNKNOWN	กรณีที่ไม่รู้ชนิดของจอภาพ
KD_STAND_M	จอภาพแบบ monochrome มาตรฐาน
KD_STAND_C	จอภาพแบบมีสี
KD_MULTI_M	จอภาพแบบ monochrome ซึ่งมีได้หลายภาวะ
KD_MULTI_C	จอภาพแบบสีซึ่งมีได้หลายภาวะ

ตาราง 2.2 แสดงค่าส่งกลับที่บอกถึงชนิดของจอภาพ

ถ้าค่าที่ส่งกลับมีค่าเป็น -1 แสดงว่าเกิดความผิดพลาดในการส่งข้อมูลขึ้น

การดึงและการกำหนดภาวะการใช้งาน

การกำหนดภาวะการใช้งานจะกำหนดโดยใช้รหัสซึ่งมีรูปแบบ SW_type โดยที่ส่วน "type" จะเป็นตัวกำหนดชนิดของตัวปรับภาพ และภาวะการใช้งาน ตัวอย่าง เช่น SW_C80x25 ใช้ในการกำหนดภาวะเป็นภาวะข้อความแบบซีจีเอ (CGA text mode) มีความละเอียดเท่ากับ 80x25 ตัวอักขระ

รูปแบบการเรียกใช้จะเรียกผ่านคำสั่ง ioctl ดังนี้

```
ioctl(disp,SW_type,0);
```

ถ้ามีข้อผิดพลาดเกิดขึ้นจะมีการส่งค่าที่เป็นลบกลับมา

การหาว่าตอนนี้อยู่ในภาวะใด ทำโดยใช้คำสั่ง CONS_GET ค่าที่ได้จากการใช้ชุดคำสั่งนี้คือ DM_type ซึ่งจะถูกเก็บไว้เพื่อเรียกใช้เมื่อต้องการกำหนดภาวะการทำงานให้กลับสู่ภาวะเดิม การใช้ CONS_GET จะเรียกผ่าน ioctl ดังนี้

```
ioctl(disp,CONS_GET,0);
```

เมื่อได้ค่า DM_type มาแล้ว เมื่อต้องการกำหนดภาวะให้เป็นตามเดิมคือ DM_type จะต้องนำไป "หรือ" กับค่าคงที่ MODESWITCH ดังเช่น

```
SW_type = DM_type ¦ MODESWITCH;
```

แล้วจึงนำค่า SW_type ที่ได้เป็นค่าที่นำไปใช้ในการกำหนดภาวะการทำงานตามคำสั่ง ioctl ดังตัวอย่างข้างต้น

การเขียนโปรแกรมควบคุมหน่วยความจำในการแสดงผลทางจอภาพ

การเข้าถึงหน่วยความจำในส่วนการแสดงผลทางจอภาพ สามารถทำได้ทั้งภาวะข้อความและภาวะกราฟิก โดยขั้นตอนในการเข้าถึงมี 3 ขั้นตอนดังนี้

1. เปิดโปรแกรมควบคุมอุปกรณ์ "kd" โดยอ้างถึงแฟ้ม /dev/video
2. ดึงค่าเลขที่อยู่ (address) ของหน่วยความจำส่วนการแสดงผลทางจอภาพ
3. แปลงส่งหน่วยความจำส่วนการแสดงผลเข้ากับหน่วยความจำที่ผู้ใช้เข้าถึงได้

วิธีการดึงค่าเลขที่อยู่ขึ้นมา

ทำได้โดยใช้ค่า KDDISPTYPE ในคำสั่ง ioctl เพื่อดึงเลขที่อยู่เริ่มต้นของหน่วยความจำส่วนการแสดงผล ในการใช้คำสั่ง ioctl จะเป็นการกำหนดโครงสร้างซึ่งโปรแกรมควบคุมอุปกรณ์ จะเป็นผู้ใส่ค่าลงในโครงสร้างข้อมูล ในการใช้ KDDISPTYPE อาร์กิวเมนต์ (argument) ที่ใช้คือตัวชี้ (pointer) ที่ชี้ไปยังโครงสร้างข้อมูล

โครงสร้างข้อมูลที่ถูกใช้โดย KDDISPTYPE มีโครงสร้างดังนี้

```
struct kd_disparam {
        long    type;           /* display type */
        char    *physaddr;      /* display memory address */
        unshort  ioaddr[MKDIOADDR] /* valid I/O address */
}
```

โดย type เป็นตัวแปรสำหรับเก็บชนิดของการแสดงผล
(CGA,VGA,etc...)

*physaddr เก็บเลขที่อยู่ของหน่วยความจำทางกายภาพของแผ่นวงจรแสดงผล

ioaddr[ ] เป็นแถวลำดับ (array) ของเลขที่อยู่ของอุปกรณ์นำเข้าและอุปกรณ์แสดงผลที่โปรแกรมควบคุมอุปกรณ์ อนุญาติให้ใช้ได้ เลขที่อยู่นี้เป็นเลขที่อยู่ของทางเข้า/ออก (input/output port) บนแผ่นวงจรแสดงผล

การแปลงส่งเพื่อเข้าถึงหน่วยความจำในส่วนการแสดงผล

การเข้าถึงหน่วยความจำในการแสดงผลจะต้องแปลงส่งหน่วยความจำของแผ่นวงจรแสดงผล เข้ากับเลขที่อยู่ที่ผู้ใช้สามารถเข้าถึงได้ ซึ่งสามารถทำได้โดยใช้คำสั่ง ioctl KDDISPTYPE ในการค้นข้อมูลเลขที่อยู่เริ่มต้นของหน่วยความจำในการแสดงผลขึ้นมา และ KDMAPDISP ioctl จะเป็นคำสั่งในการแปลงส่งหน่วยความจำ อาร์กิวเมนต์ของคำสั่ง KDMAPDISP จะเป็นตัวชี้ไปที่โครงสร้างข้อมูล kd_memloc ซึ่งมีสมาชิกดังนี้

```
struct kd_memloc{
        char  *vaddr;     /* virtual address to map to */
        char  *physaddr;  /* physical address to map from */
        long  length;     /* size in bytes to map */
        long  ioflag;     /* enable i/o addresses if set */
}
```

โดย vaddr เป็นตัวชี้ไปยังพื้นที่ของหน่วยความจำ จากบัฟเฟอร์ (buffer) ที่อยู่ในพื้นที่ใช้งานของผู้ใช้ โดยจะต้องใช้คำสั่ง malloc เพื่อสร้างพื้นที่ขึ้นมา

physaddr เป็นเลขที่อยู่ที่จะเป็นข้อมูลของ KDMAPDISP ซึ่งเลขที่อยู่นี้ได้มาจากการเรียกใช้ KDDISPTYPE

length จำนวนไบต์ของหน่วยความจำที่จะทำการแปลงส่ง

ioflag ตัวบ่งชี้การเปิดทางของเลขที่อยู่ทางเข้า/ออก

การใช้ MAPCONS ในคำสั่ง ioctl

การเรียกใช้ MAPCONS เป็นการสร้างตัวชี้ให้ชี้ไปยังหน่วยความจำส่วนการแสดงผลการเรียกใช้แบบนี้จะไม่ต้องจัดสรรหน่วยความจำในโปรแกรมของผู้ใช้ การใช้ MAPCONS จะทำการหยุดการทำงานเครื่องปลายทางเสมือน ทำให้ผู้ใช้ไม่สามารถเปลี่ยนไปใช้เครื่องปลายทางเสมือนได้จนกว่าจะออกจากโปรแกรมนี้เสียก่อน

ดังนั้นการใช้ KDMAPDISP จะทำให้การใช้งานมีประสิทธิภาพดีกว่าในกรณีที่มีการเรียกใช้เครื่องปลายทางเสมือน

การเข้าถึงรีจิสเตอร์ควบคุมการแสดงผล

ขั้นตอนในการใช้รีจิสเตอร์ควบคุมการแสดงผลของจอภาพทำได้ดังนี้ คือจะต้องเขียนค่าที่รีจิสเตอร์ควบคุม กำหนดไว้ลงที่เลขที่อยู่ในการรับข้อมูลเข้า และ การแสดงผล ตัวอย่าง เช่น ค่า 0x3D4 เป็นค่าหนึ่งที่เป็น ตำแหน่งการรับข้อมูลเข้า และการแสดงผลในแผ่นวงจรควบคุมการแสดงผลบางประเภท อาจจะมีเลขที่อยู่การรับข้อมูลเข้าและการแสดงผลมากกว่า 1 ตำแหน่ง ซึ่งจะมีค่าแตกต่างกันระหว่างแผ่นวงจรแสดงผลแบบสีเดียว (monochrome display adapter) และ แผ่นวงจรแสดงผลแบบสี (Color display adapter) ดังนั้นในการที่จะเข้าถึงรีจีสเตอร์ ควรจะเรียกผ่านฟังก์ชัน "outb"ขั้นตอนในการเขียนข้อมูลลงในเลขที่อยู่นี้ จะเป็นตัวบอกฮาร์ดแวร์ที่ควบคุมว่าต้องการใช้งานรีจีสเตอร์ใด

ตารางข้างล่างนี้เป็นตารางแสดงความสัมพันธ์ระหว่างรีจีสเตอร์และค่าที่จะใช้ในการดึงรีจีสเตอร์ขึ้นมา

รีจิสเตอร์	ชื่อรีจิสเตอร์
0x00	Horizontal Total
0x01	Horizontal displayed End
0x02	Start Horizontal Blank
0x03	End Horizontal Blank
0x04	Start Horizontal Retrace
0x05	End Horizontal Retrace
0x06	Vertical Total
0x07	Overflow
0x08	Preset Row Scan
0x09	Maximum Scan Line Address
0x0A	Cursor Start
0x0B	Cursor End
0x0C	Start High Address
0x0D	Start Low Address
0x0E	Cursor Location High
0x0F	Cursor Location Low
0x10	Vertical Retrace Start
0x11	Vertical Retrace Low
0x12	Vertical display End
0x13	Offset
0x14	Underline Location
0x15	Start Vertical Blanking
0x16	End Vertical Blanking
0x17	Mode Control
0x18	Line Compare

ตาราง 2.3 แสดงความสัมพันธ์ระหว่างรีจีสเตอร์และค่าที่จะใช้ในการอ้างถึงรีจิสเตอร์

การใช้รีจีสเตอร์อย่างมีประสิทธิภาพ

การที่จะเข้าถึงรีจีสเตอร์อย่างมีประสิทธิภาพ จะต้องมีการกำหนดตัวบ่งชี้สำหรับตัวบ่งชี้(flag) การนำข้อมูลเข้าและการแสดงผล ทางหนึ่งในการกำหนดค่าตัวแปร ioflg ของโครงสร้าง kd_memloc ให้มีค่าเท่ากับ 1 ก่อนการใช้ KDMAPDISP หรืออาจกำหนดตัวบ่งชี้สำหรับการนำข้อมูลเข้าและการแสดงผล ผ่าน KDENABIO ioctl เมื่อมีการกำหนดค่าตัวบ่งชี้แล้ว ข้อมูลจะถูกนำเข้าและออกจากรีจีสเตอร์ทีละ 1 ไบท์ โดยใช้ชุดคำสั่งย่อย inb และ outb ซึ่งมีการกำหนดไว้ใน แฟ้มหัวเรื่อง /usr/include/sys/inline.h อาร์กิวเมนต์ของชุดคำสั่งย่อยนี้ ค่าแรกคือ เลขที่อยู่ของรีจีสเตอร์ ค่าที่ 2 คือ ข้อมูลที่จะนำเข้าหรือออกจากรีจีสเตอร์