สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การรับส่งข้อมูลตามเวลาจริงโดยใช้ลีนุกซ์

T058666

ชนินทร์ สวัสดิผล

โครงงานพิเศษเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาฟิสิกส์ประยุกต์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

วพ.
ส154 ก
2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขหมู่ 58666
เลขทะเบียน
วัน,เดือน,ปี 31 ส.ค. 2549

b
i

Real-Time Data Acquisition Using Linux System

Mr.Chanin Sawatdipon

A Special Project Submitted in Partial Fulfillment of the Requirement for the Degree of

Bachelor of Science

Department of Applied Physics

Faculty of Science

King Mongkut's Institute of Technology Ladkrabang

Academic Year 2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษเรื่อง การรับส่งข้อมูลตามเวลาจริงโดยใช้ลีนุกซ์

นักศึกษา นายชนินทร์ สวัสดิผล

ภาควิชา ฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์

สาขาวิชา ฟิสิกส์ประยุกต์

อาจารย์ที่ปรึกษา ดร.ปิติพร ถนอมงาม

อาจารย์ที่ปรึกษาร่วม อ. เบญจพล ตันฮู้

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

อนุมัติให้โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต

คณะกรรมการตรวจสอบ	ลายมือชื่อ
ประธานกรรมการ อ. ภารุจ บัณฑิธาดาวิทย์	
กรรมการ อ. ทุติยาภรณ์ ทิวาวงศ์	
กรรมการ อ. สุรชาติ กมลดิลก	
กรรมการที่ปรึกษา ดร. ปิติพร ถนอมงาม	
กรรมการที่ปรึกษาร่วม อ.เบญจพล ตันฮู้	

.....

(รองศาสตราจารย์วิชาญ เตชิตธีระ)

หัวหน้าภาควิชาฟิสิกส์ประยุกต์

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลากกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษ การรับส่งข้อมูลตามเวลาจริงโดยใช้ลีนุกซ์
นักศึกษา นาย ชนินทร์ สวัสดิผล
ภาควิชา ฟิสิกส์ประยุกต์
สาขาวิชา ฟิสิกส์ประยุกต์
ปีการศึกษา 2547
อาจารย์ที่ปรึกษา ดร.ปิติพร ถนอมงาม
อาจารย์ที่ปรึกษาร่วม อ. เบญจพล ตันฮู้

บทคัดย่อ

โครงงานพิเศษนี้เป็นการศึกษาการรับส่งข้อมูลตามเวลาจริง (Real Time) โดยใช้ลีนุกซ์เป็นระบบปฏิบัติการ โดยระบบประกอบด้วย เรียลไทม์ลีนุกซ์และ ตัวส่งข้อมูล โดยลีนุกซ์จะถูกคอมไพล์ใหม่โดยใช้ RT-Linux 2.0 โดยตัวส่งข้อมูลออกแบบโดยใช้ AT892051 เป็นตัวสร้างสัญญาณพัลส์และ ส่งสัญญาณพัลส์โดย real-time kernel และทำการตรวจสอบพัลส์ทุก 1 วินาที และถูกส่งไปยัง non real-time FIFO โดยจำนวนพัลส์ที่นับได้เป็นหน่วยเฮิรตซ์ โดยผลลัพธ์จะแสดงทางหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Specail Project Title	Real-Time Data Acquisition Using Linux System
Name	Mr.Chanin Sawatdipon
Department	Applied Physics
Program	Applied Physics
Academic Year	2004
Special Project Advisor	Dr.Pitiporn Thanomngam
Special Project co-advisor	Benchapol Tunhoo

Abstract

This special project is the researching about the sending-receiving which is in Real-time Linux operation System. This research consists of RT-Linux, sender, receiver. Linux is compiled by RT-Linux2.0.The sender is controlled by AT892051 that generate many of pulse, RT-kernel always check pulse every a second then send to non real-time FIFO with Hz unit and display via monitor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงได้ด้วยดี เนื่องมาจากได้รับความเมตตาเป็นอย่างยิ่งจาก ดร. ปิติพร ถนอนงาน อาจารย์ประจำภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อาจารย์ที่ปรึกษาปริญญานิพนธ์ซึ่งให้คำปรึกษาและแนะแนวทางในการดำเนินการวิจัย รวมทั้งจัดหาอุปกรณ์และงบประมาณในการดำเนินการวิจัย ข้าพเจ้ารู้สึกซาบซึ้งในความกรุณาของท่าน ขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณอาจารย์เบญจพล ตันฮู้ ที่ได้เอื้อเฟื้อเทคนิคต่างๆ และให้คำแนะนำต่างๆ ในการดำเนินการวิจัย

ขอขอบคุณ รศ.วิชิต ศิริโชติ ที่ให้คำแนะนำและเอื้อเฟื้ออุปกรณ์ในการดำเนินการวิจัย

ขอขอบคุณสันติพงษ์ บุญผลิตกุล นักศึกษาชั้นปีที่ 3 วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2547 ที่ช่วยในการปรับปรุงเอกสารในโครงงานนี้

ขอขอบคุณนายเฉลิมพล สาริบุตร นักศึกษาชั้นปีที่ 4 ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังปีการศึกษา 2547ที่คอยให้กำลังใจในการทำงาน

ขอขอบคุณนางสาวสุพรรณิการ์ โรจน์รัตนวาณิชย์ นักศึกษาชั้นปีที่ 4 ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังปีการศึกษา 2547 ที่ช่วยในการปรับปรุงเอกสารในโครงงานนี้

คุณค่า และประโยชน์อันพึงมีจากโครงงานพิเศษนี้ ผู้วิจัยขอมอบแด่ บิดามารดา น้องสาว และผู้มีพระคุณในการศึกษาทุกท่าน

ชนินทร์ สวัสดิผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ฉ
สารบัญตาราง	ช
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงงานพิเศษ	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตการวิจัย	2
1.4 ขั้นตอนการวิจัยและวิธีการดำเนินการ	2
1.5 ผลที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎี	3
2.1 ลีนุกซ์	3
2.1.1 คุณสมบัติพื้นฐานของลีนุกซ์	3
2.1.2 ลีนุกซ์ค่ายต่างๆ	4
2.1.3 การนำลีนุกซ์มาใช้งาน	5
2.1.4 การเตรียมความพร้อมก่อนใช้ลีนุกซ์	5
2.1.5 สิ่งที่ควรทราบก่อนการติดตั้ง	6
2.2 ระบบปฏิบัติการเวลาจริง (real time OS)	7
2.2.1 เวลาจริงแบบแข็งกับแบบอ่อน (Hard and Soft Real Time)	8
2.2.2 Linux กับการทำงานในเวลาจริง	9
2.2.3 โครงสร้างการทำงานของ RTLinux	9
2.2.4 การติดต่อแลกเปลี่ยนข้อมูลระหว่างกระบวนการของ RTLinux	11
2.2.5 ทางเลือกอื่นนอกจาก RTLinux	12
2.3 พอร์ตขนาน	13
2.3.1 ความรู้เบื้องต้นของพอร์ตขนาน	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.1 พอร์ตดาต้า 15
2.3.1.2 พอร์ตคอนโทรล 16
2.3.1.3 พอร์ตแสดงสถานะหรือพอร์ตสเตตัส 18
บทที่ 3 วิธีดำเนินการวิจัย 19
3.1 ส่วนประมวลผล 19
3.1.1 ทำการติดตั้ง Linux และ RTLinux 19
3.1.2 ศึกษาการเขียนโปรแกรม 20
3.2 ส่วนส่งข้อมูล 20
3.3 ทำการทดลองการทำงานของ RTLinux และ Microcontroller 22
บทที่ 4 ผลการทดลอง 23
4.1 ผลการทดลอง 23
4.1.1 เป็นการเปรียบเทียบระหว่างลีนุกซ์ที่ทำงานโปรแกรมเดียวกับทำงานหลายโปรแกรม 24
4.1.2 เป็นการเปรียบเทียบระหว่างเรียลไทม์ลีนุกซ์ที่ทำงานโปรแกรมเดียวกับทำงานหลายโปรแกรม 24
บทที่ 5 สรุปและข้อเสนอแนะ 26
5.1 สรุปผลการทดลอง 26
5.2 ปัญหาที่พบและข้อเสนอแนะ 26
เอกสารอ้างอิง 27
ภาคผนวก 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 โครงสร้างของ RTLinux	10
รูปที่ 2.2ลักษณะการทำงานของ FIFO	12
รูปที่ 2.3 พอร์ตดาต้า(Data Port)	15
รูปที่ 3.1 แผนผังการจัดแสดงโครงงาน	19
รูปที่ 3.2 ตัวส่งข้อมูล(สัญญาณพัลส์)	21
รูปที่ 3.3 วงจรตัวส่งข้อมูล	22
รูปที่ 4.1 เป็นสัญญาณที่ใช้ในการทดลอง 700 Hz	23
รูปที่ 4.2 ทำงานโปรแกรมเดียวในลีนุกซ์	24
รูปที่4.3เป็นการทำงานสองโปรแกรมในลีนุกซ์	24
รูปที่ 4.4 ทำงานโปรแกรมเดียว ในเรียวทามส์	24
รูปที่4.5ทำงานสองโปรแกรมในเรียวทามส์	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ลีนุกซ์ค่ายต่างๆ	4-5
ตารางที่ 2.2 สถานะของขาคอนเนคเตอร์	16-17
ตารางที่ 3.1 หน้าที่การทำงานของตัวส่งข้อมูล	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงานพิเศษ

ในปัจจุบันลีนุกซ์เป็นระบบปฏิบัติการโอเพ่นซอร์สที่กำลังแพร่หลายในบ้านเราเนื่องด้วยว่ามันเป็นของ "ฟรี" ทั้งในความหมาย "ฟรี" แบบอิสระคือสามารถนำไปก๊อปปี้แจกจ่ายให้ใครใช้ก็ได้ ไม่เสียค่าลิขสิทธิ์หรือจะนำไปดัดแปลงแก้ไขการใช้งานก็ได้ ส่วนอีกความหมายหนึ่ง คือ "ฟรี" ที่ปราศจากค่าใช้จ่ายทำให้ผู้ผลิตอุปกรณ์คอมพิวเตอร์หลากหลายหยิบยก ลีนุกซ์ ขึ้นมาเป็นพระเอกที่ช่วยลดต้นทุนทำให้ผู้บริโภคได้รับผลประโยชน์มากขึ้น

เหตุผลที่เรานำ ลีนุกซ์ มาใช้งานอย่างกว้างขวางเนื่องจากว่าใช้ต้นทุนในการพัฒนาต่ำสามารถพัฒนาได้รวดเร็ว

โดยเราได้นำระบบเรียลไทม์มาใช้ควบคู่กับระบบ ลีนุกซ์ โดยระบบเรียลไทม์นั้นเป็นระบบที่ทำงานตามเงื่อนไขของเวลาโดยเราสามารถนำมาประยุกต์ใช้งานได้ดังนั้นในโครงงานพิเศษนี้จึงนำระบบเรียลไทม์มาใช้

ในระบบเรียลไทม์ที่เคยมีการพัฒนาขึ้นมานั้น จะมีขอบเขตและขนาดการใช้งานที่แตกต่างกันตั้งแต่ นาฬิกาข้อมือ เตาอบไมโครเวฟ จนถึงระบบอัตโนมัติในอุตสาหกรรม โรงงานพลังงานนิวเคลียร์รวมถึงระบบทีวีและวีดีโอ เครื่องเล่นเพลง ระบบควบคุมเครื่องบินหรือแม้กระทั้ง ระบบควบคุมจรวด ซึ่งต้องการให้สามารถทำงานตามที่เรากำหนดได้

โดยระบบปฏิบัติการเป็นส่วนประกอบที่สำคัญในระบบทั่วๆไป ซึ่งในระบบเรียลไทม์จำเป็นต้องมีระบบปฏิบัติการแบบเรียลไทม์ เพื่อสามารถรับงานแบบเรียลไทม์ได้ การที่มีระบบปฏิบัติการแบบเรียลไทม์ทำให้เราคาดเดาเวลาที่ใช้ในการทำงานให้เสร็จได้ และยังมีความน่าเชื่อถือรวมทั้งสามารถจัดลำดับของงานเพื่อทำการรันได้ แต่เนื่องจากระบบปฏิบัติการแบบเรียลไทม์ในปัจจุบันส่วนใหญ่เป็นเชิงการค้า ซึ่งราคาค่อนข้างแพงและต้องเสียค่าใช้จ่ายในการบำรุงรักษาเป็นอย่างมาก ทำให้ไม่เหมาะที่จะนำมาใช้ในระบบขนาดเล็ก จึงได้นำระบบปฏิบัติการที่มีอยู่เดิมและมีราคาถูกมาพัฒนาเพิ่มเพื่อให้สามารถทำงานแบบเรียลไทม์ โดยระบบปฏิบัติการที่เรานำมาใช้กับระบบเรียลไทม์ที่นิยมนำมาใช้คือ "ลีนุกซ์" ซึ่งทำการเพิ่มความสามารถในด้านเรียลไทม์ให้กับลีนุกซ์ด้วยเหตุผลคือสามารถเปิดเผยโค้ดได้ สามารถทำการปรับปรุงและแก้ไขง่ายและไม่เสียค่าใช้จ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์

1. เพื่อศึกษาระบบปฏิบัติการลีนุกซ์
2. เพื่อศึกษาระบบเรียลไทม์และการรับ-ส่งข้อมูลแบบเรียลไทม์
3. ศึกษาและค้นคว้าการประยุกต์ใช้งานลีนุกซ์กับระบบเรียลไทม์

1.3 ขอบเขตการวิจัย

ต้องการพัฒนา การรับส่งข้อมูลแบบเรียลไทม์ซึ่งสามารถนำไปต่อยอดเพื่อ งานวิจัยที่จำเป็นต้องมีการการทำการรับส่งข้อมูลแบบเรียลไทม์

1.4 ขั้นตอนการวิจัยและวิธีการดำเนินงาน

แผนการดำเนินการแบ่งเป็นขั้นตอนต่างๆดังนี้

ขั้นตอนที่ 1. ศึกษาระบบปฏิบัติการลีนุกซ์และทำการติดตั้งระบบปฏิบัติการลีนุกซ์

ขั้นตอนที่ 2. ศึกษาระบบ Real-time (RT) และทำการติดตั้งระบบ Real-time (RT)

ขั้นตอนที่ 3. ศึกษาการเขียนโปรแกรม RTLinux และเขียนโปรแกรมเพื่อรับและส่งข้อมูล

ขั้นตอนที่ 4. สร้าง Microcontroller เพื่อรับ-ส่งข้อมูล

ขั้นตอนที่ 5. ทดลอง

ขั้นตอนที่ 6. ปรับปรุงแก้ไขข้อบกพร่องต่างๆที่เกิดขึ้นและทำการแก้ไข

ขั้นตอนที่ 7. สรุปผลและเสนอแนะแนวทางในการพัฒนาต่อไป

1.5 ผลที่คาดว่าจะได้รับ

สามารถนำเรียลไทม์ลีนุกซ์ (RT Linux) มารับและส่งข้อมูลที่ต้อง ใช้เวลาในการปฏิบัติงาน ทั้งการควบคุมเวลาที่เที่ยงตรงและความเร็วในการรับ-ส่งข้อมูลโดยมีความผิดพลาดน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 ลีนุกซ์

ลีนุกซ์นั้นถูกสร้างขึ้นโดยนาย Linux Torvalds ขณะนั้นยังเรียนอยู่ที่มหาลัย Helsinki ประเทศฟินแลนด์ ในขณะนั้น Linux สนใจในระบบปฏิบัติการณ์ Minix จึงเกิดความคิดที่จะพัฒนาระบบปฏิบัติการณ์ที่มีความสามารถขึ้นตามแบบของ Minix และทำงานบนเครื่องตระกูล PC-Intel ได้ด้วย Linus เริ่มต้นพัฒนาในปี 1991 และได้ออก Kernel เวอร์ชั่น 1.0 ได้ จวบจนปัจจุบัน kernel ได้ถูกพัฒนาจนถึงเวอร์ชั่น 2.5.x และยังคงถูกพัฒนาต่อไปอีก

ลีนุกซ์ เป็นระบบปฏิบัติการเช่นเดียวกับ ดอส ไมโครซอฟต์วินโดวส์ หรือยูนิกซ์ โดยลีนุกซ์นั้นจัดว่าเป็นระบบปฏิบัติการยูนิกซ์ประเภทหนึ่ง การที่ลีนุกซ์เป็นที่กล่าวขานกันมากขณะนี้เนื่องจากความสามารถของตัวระบบปฏิบัติการและโปรแกรมประยุกต์ที่ทำงานบนระบบลีนุกซ์ สิ่งที่สำคัญที่สุดก็คือระบบลีนุกซ์เป็นระบบปฏิบัติการประเภทฟรีแวร์ (Freeware) คือไม่เสียค่าใช้จ่ายในการซื้อโปรแกรม

ถึงแม้ว่าในขณะนี้ลีนุกซ์ยังไม่สามารถแทนที่ไมโครซอฟต์ วินโดวส์ บนพีซีหรือแมคโอเอส (Mac OS) ได้ทั้งหมดก็ตาม แต่ผู้ใช้จำนวนไม่น้อยที่หันมาใช้และช่วยพัฒนาโปรแกรมประยุกต์บนลีนุกซ์กัน และเรื่องของการดูแลระบบลีนุกซ์นั้น ภายในระบบลีนุกซ์เองมีเครื่องมือช่วยสำหรับดำเนินการให้สะดวกยิ่งขึ้น

2.1.1 คุณสมบัติพื้นฐานของลีนุกซ์

Multitasking ลีนุกซ์สามารถทำงานได้หลายโปรแกรมพร้อมๆกัน โดยที่โปรแกรมต่างๆทำงานเป็นอิสระต่อกัน

Multiuser ลีนุกซ์ยอมให้ผู้ใช้สามารถเข้ามาทำงานพร้อมๆกันได้ โดยผู้ใช้แต่ละคนทำงานได้เป็นอิสระจากกัน

ทำงานได้ทั้งแบบ Text Mode และ Window Mode

ทำงานได้ทั้งแบบหน้าเครื่อง (console) หรือจากเครื่องอื่น (Remote)

ลีนุกซ์สามารถใช้งานตั้งแต่เครื่องรุ่นเล็กๆ ไปจนถึงเครื่องขนาดใหญ่ เราสามารถใช้งานลีนุกซ์ด้วยเมมโมรีอย่างน้อย 4 MB และใช้เนื้อที่บนฮาร์ดดิสก์ต่ำสุด 15 MB แต่ยิ่งทรัพยากรที่ใช้น้อยเท่าไร ก็ยิ่งเกิดข้อจำกัดในการใช้งานลีนุกซ์มากขึ้นเท่านั้น สำหรับเครื่องที่ลีนุกซ์สนับสนุนในการทำงานมีดังนี้

แบบ PC-Base

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Intel/AND/Cyrix 386SX/DX/SL/DXL/SLC

-Intel/AMD/Cyrix 486SX/DX/SL/SX2/DX2/DX4

-AMD K5,K6,K6-2,K6-3 and K7/Athlon

-Cyrix 6x86,6x86MX

-Intel Pentium, Pentium Pro, Pentium2 (including the Celeron series) and Pentium

-IDT WinChip C6

-Symmetrical Multiprocessing (multiple CPUs)

แบบที่ไม่ใช่ PC-Base

-Digital Alpha

-Sun SPARC

-Macintosh Power PC

-IBM Mainframes(370/390,AS400)

-Hewlett-Packard PA-RISC

-Amiga

2.1.2 **ลีนุกซ์ค่ายต่างๆ**

ในปัจจุบันลีนุกซ์ได้ถูกพัฒนาขึ้นมาจากหลายๆ ค่ายด้วยกัน ซึ่งแต่ละค่ายก็พัฒนาขึ้นมาเป้าหมายที่ต่างกันไป ซึ่งลีนุกซ์บางค่ายก็สนับสนุนเครื่องที่ทำงานได้แตกต่างกันไปบ้าง ดังตัวอย่าง

Distributor	Website
Yellow Dog	www.yellowdoglinux.com
KSI-Linux	www.ksi-linux.com
Corel Linux	www.linux.corel.com
Debian	www.debian.com
Redhat	www.redhat.com
Slackware	www.slackware.com
Elfstone Linux	www.elflinux.com
Best Linux	www.bestlinux.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Libranet	www.libranet.com
Stampede	www.stamplde.org
SuSE	www.suse.com
ASPLinux	www.asp-linux.com
Linux-Mandrake	www.linux-mandrake.com
OpenLinux	www.openlinux.net
Phat Linux	www.phatlinux.net
LuteLinux	www.lutelinux.com
MkLinux	www.mklinux.org

ตารางที่ 2.1 ลีนุกซ์ค่ายต่าง ๆ

2.1.3 การนำลีนุกซ์มาใช้งาน

ปัจจุบันได้มีการนำระบบปฏิบัติการลีนุกซ์ไปประยุกต์เป็นระบบปฏิบัติการสำหรับงานด้านต่างๆเช่นงานด้านการคำนวณทางวิทยาศาสตร์ใช้เป็นสถานีงาน สถานีบริการ อินเทอร์เน็ต อินทราเน็ต หรือใช้ใน การเรียนการสอนและการทำวิจัยทางคอมพิวเตอร์ใช้พัฒนาโปรแกรม เนื่องจาก มีเครื่องมือมากมาย เช่น โปรแกรมภาษาซี (C) ซีพลัสพลัส (C++) ปาสคาล (Pascal) ฟอร์แทรน (Fortran) ลิสป์ (Lisp) โปรล็อก (Prolog) เอดา (ADA) มีภาษาสคริปต์ เช่น เชลล์ (Shell) บาสซ์เชลล์ (Bash Shell) ซีเชลล์ (C Shell) คอร์นเชลล์ (Korn Shell) เพิร์ล (Perl) พายตัน (python) TCL/TK

นอกจากนี้ยังมีโปรแกรมประยุกต์ในสาขาต่างๆ อีกมากมาย

2.1.4 การเตรียมความพร้อมก่อนใช้ลีนุกซ์

ก่อนที่จะทำการติดตั้งก็ต้องเตรียมความพร้อมทางด้านอุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ให้เป็นที่เรียบร้อยก่อน ระบบลีนุกซ์ต้องการฮาร์ดแวร์ที่มีคุณสมบัติ ขั้นต่ำสุดดังต่อไปนี้

1. หน่วยประมวลผลกลางของ Intel 80386 ขึ้นไป

2. หน่วยประมวลผลทางคณิตศาสตร์ มีหรือไม่มีก็ได้ เพราะระบบปฏิบัติการ Linux ได้มีการจำลอง หน่วยประมวลผลทางคณิตศาสตร์ไว้ในระดับของเคอร์เนล (Kernel) แล้ว

3. หน่วยความจำอย่างน้อย 8 เมกะไบต์ แต่แนะนำให้มีอย่างน้อย 16 เมกะไบต์จะทำให้ระบบมีประสิทธิภาพที่ดีกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ฮาร์ดดิสก์อย่างน้อย 101 เมกะไบต์ สำหรับการติดตั้งแบบพื้นฐาน 266 เมกะไบต์ สำหรับการติดตั้งแบบทั่วไป และ 716 เมกะไบต์ สำหรับการติดตั้งแบบทั้งหมดตัวเลขที่ระบุทั้งหมดเฉพาะส่วนระบบปฏิบัติการ ถ้าต้องการใช้เป็น File Server หรือ Database Server จะ ต้องเผื่อเนื้อที่ไว้สำหรับใช้งานด้วย

2.1.5 สิ่งที่ควรทราบก่อนการติดตั้ง

1. คุณสมบัติของฮาร์ดดิสก์ที่ต้องการติดตั้ง
   - จำนวนของฮาร์ดดิสก์ที่ต้องการติดตั้ง
   - ขนาดความจุของฮาร์ดดิสก์ที่จะใช้ในการติดตั้ง
   - ประเภทการเชื่อมต่อของฮาร์ดดิสก์ที่จะใช้ในการติดตั้ง IDE, EIDE หรือ SCSI
   - มีการใช้ประเภทการเชื่อมต่อของฮาร์ดดิสก์หลายๆ ประเภทในเครื่องเดียวกันหรือไม่
2. ขนาดของหน่วยความจำหลัก เพื่อที่จะคำนวณหาขนาดของ Linux Swap Partition
3. ประเภทการเชื่อมต่อของเครื่องอ่านซีดีรอม IDE (ATAPI), SCSI
4. รุ่นและยี่ห้อของแผงวงจรเชื่อมต่อ SCSI
5. รุ่นและยี่ห้อของแผงวงจรเชื่อมต่อเครือข่าย
6. จำนวนปุ่มกด และประเภทเชื่อมต่อของเมาส์
7. รุ่นและยี่ห้อของแผงวงจรเชื่อมต่อจอภาพ รุ่นและยี่ห้อของจอภาพ.
8. รายละเอียดการกำหนดโปรโตคอล TCP/IP ของเครื่องที่ต้องการติดตั้ง.
   - IP Address
   - Net Mask
   - GateWay Address
   - Name Server Address
   - Domain Name
   - Host Name

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RTLinuxเป็นระบบปฏิบัติการที่รวมเอาความสามารถของระบบปฏิบัติการทั่วไป (general-purpose OS) ซึ่งก็คือ Linux เข้ากับความสามารถของการทำงานในเวลาจริง (real time) ทำให้เราสามารถใช้ RTLinux ในการทำงานที่ต้องการความ เที่ยงตรงของเวลาตอบสนอง เช่น ในการเก็บตัวอย่างข้อมูลหรือการควบคุมวงปิดโดยใช้คอมพิวเตอร์และยังสามารถใช้ความสามารถของ Linux เช่น ความสามารถในด้านเครือข่าย (networking) หรือ การติดต่อกับผู้ใช้แบบกราฟิก (GUI) ได้อีกด้วย

เช่นเดียวกัน Linux, RTLinux เป็นซอฟแวร์ที่ใช้ลิขสิทธ์แบบ GPL ผู้ใช้สามารถนำต้นรหัส (source code) ไปพัฒนาต่อให้เหมาะสมกับการใช้งานยิ่งขึ้นได้ และผู้ใช้ ไม่ต้องเสียค่าใช้จ่ายให้ผู้ทำ RTLinux แต่อย่างใด แต่หากผู้ใช้ต้องการเก็บต้นรหัสไว้ หรือ ต้องการขายโปรแกรมที่พัฒนาขึ้นมา ผู้ใช้สามารถซื้อลิขสิทธิ์ได้จากผู้ที่ทำ RTLinux สำหรับในรุ่นที่ขาย นอกจากจะมีความสามารถเหมือน RTLinux รุ่นที่เป็น GPL แล้ว ยังได้เพิ่มความสามารถบางอย่างเข้าไปด้วย แต่ สำหรับผู้เขียนแล้วคิดว่าส่วนที่เพิ่มเข้ามานี้ไม่จำเป็นนักในการใช้งานทั่วๆ ไป

2.2 ระบบปฏิบัติการเวลาจริง (real time OS) คืออะไร

ในการนำคอมพิวเตอร์ไปประยุกต์ใช้งานบางประเภท เราต้องการมากกว่าการได้ผลลัพธ์ที่ถูกต้อง เราอาจต้องการให้คอมพิวเตอร์ทำงานได้ภายในระยะเวลาที่เรา กำหนด หรือทำงานที่เวลาที่เรากำหนดโดยความผิดพลาดทางเวลาที่ยอมรับได้อาจจะน้อยมาก อาจจะเป็นมิลลิวินาที หรือ น้อยกว่านั้น งานประเภทที่ว่านี้ก็เช่น ในการควบคุมเครื่องจักรกล เราต้องอ่านข้อมูลสถานะของเครื่องจักรกลเพื่อ ใช้ในการคำนวณสัญญาณควบคุม และต้องส่งสัญญาณควบคุมที่คำนวณได้ไปควบคุมเครื่องจักรกลนั้น ทั้งหมดนี้ต้องทำภายในระยะเวลาที่กำหนดแน่นอน ความผิดพลาดจากเวลาที่กำหนด อาจทำให้ไม่ได้สมรรถนะตามที่ต้องการ หรืออาจทำให้ระบบขาดเสถียรภาพและเกิดความเสียหายได้ เราเรียกการทำงานในลักษณะนี้ว่าการทำงานในเวลาจริง

จากความต้องการดังกล่าวการทำงานในเวลาจริงส่วนใหญ่จะใช้ฮาร์ดแวร์และซอฟท์แวร์ที่ออกแบบมาโดยเฉพาะสำหรับงานนั้นๆ ทำให้การพัฒนาระบบและนำไปใช้งานมีความยุ่งยากและอาจเสียค่าใช้จ่ายสูง นอกจากนั้น การเพิ่มความสามารถให้แก่ระบบที่มีอยู่เดิม เช่น การเพิ่มความสามารถในการเชื่อมต่อกับเครือข่าย การติดต่อกับผู้ใช้แบบกราฟิกหรือการติดต่อกับระบบฐานข้อมูล จะทำได้ยากหรือทำไม่ได้ ทั้งที่ความสามารถเหล่านี้มีอยู่ในระบบปฏิบัติการทั่วไปอยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจึงมีการพัฒนาระบบปฏิบัติการที่สามารถทำงานในเวลาจริงและยังสามารถทำงาน อื่นๆ ที่ระบบปฏิบัติการทั่วไปสามารถทำได้ รวมทั้งสามารถใช้งานบนเครื่อง คอมพิวเตอร์มาตรฐานเช่น IBM PC compatible ซึ่งมีใช้กันอย่างแพร่หลายและราคาถูก การมีระบบปฏิบัติการเวลาจริง ทำให้สามารถพัฒนาระบบและนำไปใช้งานได้อย่างรวดเร็ว มีค่าใช้จ่ายต่ำ รวมทั้งเรายังใช้ความสามารถของระบบปฏิบัติการทั่วไปได้อย่างสะดวก

ระบบปฏิบัติการเวลาจริงมีให้เลือกในท้องตลาดอยู่หลายผลิตพันธ์ เช่น VxWorks, QNX, LynxOS หรือ Windows NT ที่เพิ่มความสามารถด้านการทำงานเวลา จริงด้วย INTime, Hyperkernel หรือ RTX ส่วน RTLinux ก็เป็นอีกทางเลือกหนึ่ง ที่น่าสนใจเนื่องจากใช้ลิขสิทธ์แบบ GPL สามารถนำมาใช้ได้ฟรีไม่ต้องเสียค่าใช้จ่าย และสามารถนำต้นรหัสไปพัฒนาต่อให้ดียิ่งขึ้นได้ โดยมีข้อแม้เพียงว่าต้นรหัสที่พัฒนาต่อนั้นต้องใช้ลิขสิทธ์แบบ GPL ด้วย

2.2.1 เวลาจริงแบบแข็งกับแบบอ่อน (Hard and Soft Realtime)

โดยทั่วไปเราสามารถแบ่งประเภทของงานในเวลาจริงได้เป็น 2 ประเภทคือ เวลาจริงแบบแข็ง (Hard Realtime) กับ เวลาจริงแบบอ่อน (Soft Realtime)

เวลาจริงแบบแข็ง หมายถึง งานที่ต้องการความเที่ยงตรง และแม่นยำสูงในเรื่องของเวลา เมื่อมีการสั่งให้ทำงาน ต้องสามารถทำงานนั้นๆ ได้ทันที งานประเภทนี้ เช่น การควบคุมเครื่องจักรกลหรือการควบคุมอากาศยาน ความผิดพลาดทางเวลาเกินกว่าค่าที่ยอมรับได้อาจทำให้เกิดความเสียหายหรืออุบัติเหตุได้

เวลาจริงแบบอ่อน หมายถึง งานที่ต้องการการทำงานที่รวดเร็วพอ แต่อาจมีความผิดพลาดในเรื่องของเวลาได้ เป็นบางครั้ง งานประเภทนี้ เช่น ในการประมวลผลภาพเคลื่อนไหว เราต้องการให้ความเร็วในการประมวล ผลมากกว่า 30 เฟรมต่อวินาที การประมวลผลที่ช้าเกินไปจะทำให้ภาพเคลื่อนไหวที่ได้ มีคุณภาพลดลง แต่เราอาจยอมรับได้ถ้าเกิดขึ้นไม่บ่อยนัก โดยปกติแล้ว Linux และ MS-Windows ในตระกูล NT จะให้ความสามารถด้านเวลาได้เพียงระดับเวลาจริงแบบอ่อนนี้เท่านั้น

เนื่องจาก RTLinux ได้รับการออกแบบมาเพื่อแก้ปัญหาเวลาจริงแบบแข็ง ดังนั้นถ้าไม่ได้กล่าวไว้เป็นการเฉพาะ คำว่า “เวลาจริง” จะหมายถึง เวลาจริงแบบแข็ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 Linux กับการทำงานในเวลาจริง

Linux เป็นระบบปฏิบัติการที่สามารถใช้ได้โดยไม่ต้องเสียค่าใช้จ่าย สามารถทำงานได้ดีกับเครื่องคอมพิวเตอร์ ทั้งรุ่นใหม่และเก่า และจำนวนผู้ใช้ก็มีมากขึ้นเรื่อยๆ ดังนั้นจึงไม่ใช่เรื่องแปลกที่เราจะลองนำ Linux มาประยุกต์ใช้กับการทำงานในเวลาจริง แต่เราพบว่า Linux (และระบบปฏิบัติการแบบทั่วไปอื่นๆ) อาจไม่สามารถ ให้ความเที่ยงตรงได้อย่างเพียงพอ เช่นถ้าเราต้องการอ่านข้อมูลจากอุปกรณ์ภายนอก คอมพิวเตอร์ทุกๆ 10 มิลลิวินาทีและนำมาเก็บไว้ในฮาร์ดดิสค์ เราจะพบว่า Linux ไม่สามารถให้ความเที่ยงตรงในระดับนี้ได้ทั้งที่ในทฤษฎีแล้วฮาร์ดแวร์ ของเราสามารถตอบสนองที่ความเร็วระดับนี้ได้

สาเหตุที่ทำให้ Linux ไม่สามารถตอบสนองความต้องการของการทำงานในเวลาจริงมี อยู่ด้วยกันหลายประการ เช่น

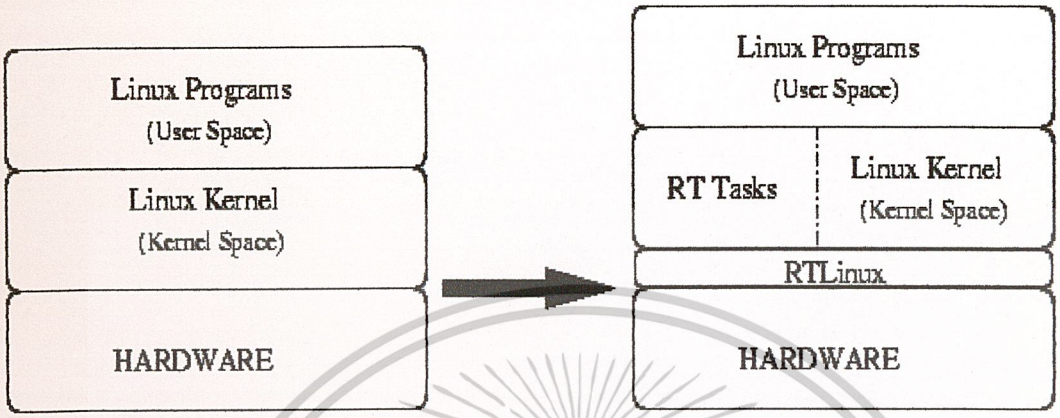
- context switch time เมื่อ Linux ต้องการหยุดการทำงานของโปรแกรมหนึ่งเพื่อให้อีกโปรแกรมหนึ่งได้ทำงาน จะมีช่วงเวลาระยะหนึ่งที่ Linux จะต้องใช้เก็บข้อมูลสถานะของโปรแกรมเดิม ซึ่งอาจทำให้การทำงานของเราไม่ทันตามกำหนด
- interrupt disabled delay บางขณะ Linux จะทำการ disable Interrupt เป็นช่วงระยะเวลาหนึ่งเพื่อแก้ไขข้อมูลในส่วนที่อาจมี Interrupt routine อื่นใช้งานเนื่องจากการทำงานในเวลาจริงต้องการใช้ Interrupt เพื่อกำหนดเวลา (Timer Interrupt) หรือใช้ Interrupt จากฮาร์ดแวร์ภายนอกเพื่อเป็นสัญญาณบอกให้เริ่มทำงาน การที่Linux ทำการ disable Interrupt ไว้จะทำให้เกิดความผิดพลาดทางเวลา
- non-preemptive kernel โดยปกติ Linux สามารถที่จะหยุดการทำงานของโปรแกรมที่กำลังทำงานอยู่เพื่อให้โปรแกรมที่มีความสำคัญกว่าได้ทำงาน แต่ในกรณีที่โปรแกรมที่กำลังทำงานอยู่นั้น อยู่ในระหว่างการทำ system call เช่น fork(), exec() Linux จะไม่สามารถหยุดการทำงานของโปรแกรมนั้นได้ (เนื่องจากเป็นการทำงานในตัว Linux เอง) จะต้องรอให้ออกจาก system call ก่อน ทำให้เกิดความผิดพลาดทางเวลาได้เช่นกัน

2.2.3 โครงสร้างการทำงานของ RTLinux

ดังที่กล่าวในหัวข้อที่ผ่านมาว่า Linux โดยตัวของมันเองไม่เหมาะที่จะนำมาใช้ในการทำงานในเวลาจริง ดังนั้นจึงได้มีการพัฒนาระบบปฏิบัติการ RTLinux ขึ้นโดยกลุ่มนักวิจัย ที่ New

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mexico Institute of Technology ซึ่งมี Dr. Victor Yodaiken เป็นหัวหน้า โครงการ การพัฒนาได้เริ่มมาตั้งแต่ปี คศ. 1996 ในขณะนี้ได้พัฒนาถึงรุ่นที่ 3.1



รูปที่ 2.1 โครงสร้างของ RTLinux

โครงสร้างของ RTLinux ที่จะทำให้ Linux มีความสามารถทำงานในเวลาจริงได้ จะทำโดยการเพิ่มชั้นของซอฟท์แวร์ระหว่างฮาร์ดแวร์ของคอมพิวเตอร์กับ Linux kernel ดังรูปที่ 2.1

RTLinux จะมอง Linux kernel เป็นงานที่มีความสำคัญต่ำ จะทำงานได้ก็ต่อเมื่อไม่มีงาน ในเวลาจริง (RT Tasks) ต้องการทำงานเท่านั้น และเมื่อมีงานในเวลาจริงต้องการ ทำงาน RTLinux จะสามารถ

หยุดการทำงานของ Linux kernel ได้ตลอดเวลา ไม่ว่า Linux จะทำอะไร อยู่ก็ตาม ส่วนตัว Linux เองจะไม่เห็นความแตกต่างไปจากการที่ Linux ทำงานบน ฮาร์ดแวร์โดยตรง จากจุดนี้เองที่ทำให้เราสามารถใช้ความสามารถเดิมของ Linux ได้ทุกอย่าง ไม่ว่าจะ เป็นความสามารถด้านเครือข่าย หรือด้านกราฟิก

ในกรณีที่เกิด Interrupt จากฮาร์ดแวร์ RTLinux จะทำการดักสัญญาณ Interrupt นั้นไว้ ถ้าไม่มีงานในเวลาจริงอื่นๆ ต้องการสัญญาณที่ดักได้นี้จึงจะส่งต่อไปให้ Linux และกรณีที่ Linux ทำการ disable Interrupt เพื่อเหตุผลใดก็ตาม RTLinux จะไม่ทำการ disable Interrupt จริงๆ เพียงแต่จะไม่ส่งสัญญาณ Interrupt นั้นให้กับ Linux kernel เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อความรวดเร็วในการทำงาน RTLinux จึงเป็นชั้นของซอฟท์แวร์ที่มีขนาดเล็ก ใช้ scheduler แบบง่ายๆ โดยจะให้งานที่มีความสำคัญมากกว่าได้ทำงานก่อนเสมอ และงานในเวลาจริงจะไม่มีการถูกสลับลง ฮาร์ดดิสก์ อีกทั้งงานในเวลาจริงทั้งหมดจะใช้ memory space เดียวกันกับ Linux kernel (เราเรียกว่าทำงานใน kernel space) จึงทำให้ไม่เสียเวลามากในการหยุดการ ทำงานของงานหนึ่งและให้อีกงานหนึ่งทำต่อ (fast context switch time) จากเหตุผลดังที่กล่าวมาทำให้ RTLinux สามารถให้ความเที่ยงตรงในระดับไมโครวินาทีบนเครื่องคอมพิวเตอร์ระดับ Pentium Class ได้

ข้อแตกต่างอย่างหนึ่งของงานในเวลาจริงกับโปรแกรมทั่วไปบน Linux คือ งานในเวลาจริง จะไม่ได้รับการป้องกันความผิดพลาดใดๆ จาก Linux ดังนั้นความผิดพลาดของซอฟท์แวร์ เช่นการหารด้วยศูนย์ จะทำให้เครื่องคอมพิวเตอร์หยุดการทำงานทันที ซึ่งต่างจาก โปรแกรมทั่วไปบน Linux (เราเรียกว่าทำงานใน user space) ซึ่ง Linux จะทำการหยุด การทำงานของโปรแกรมนั้นโดยไม่มีผลกระทบต่อโปรแกรมอื่นๆ ที่กำลังทำงานอยู่ คอมพิวเตอร์จะยังคงสามารถ ทำงานต่อไปได้อย่างปกติ จะมีเพียงโปรแกรมที่ทำงานผิดพลาดเท่านั้นที่จะหยุดทำงาน ตรงจุดนี้จึงเป็นข้อพึงระวังอย่างมากในการพัฒนาโปรแกรมที่ทำงานในเวลาจริง

2.2.4 การติดต่อแลกเปลี่ยนข้อมูลระหว่างกระบวนการของ RTLinux (RTLinux Interprocess Communication Mechanism)

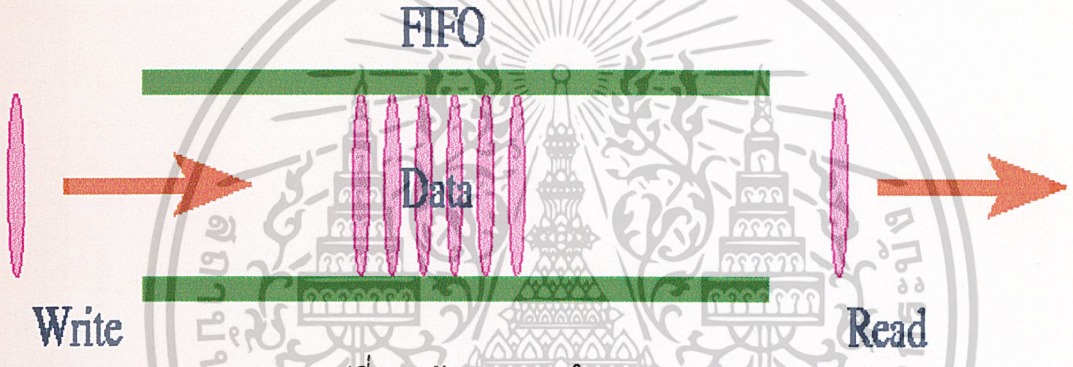
RTLinux ได้รับการออกแบบให้มีขนาดเล็กและไม่ซับซ้อน ดังนั้นงานในเวลาจริง จึงไม่สามารถทำงานที่ต้องการการบริการของ Linux ได้ เช่น การติดต่อกับผู้ใช้งาน การอ่านเขียนข้อมูลในฮาร์ดดิกส์ หรือการติดต่อกับเครือข่าย งานดังกล่าวจะต้องทำโดยโปรแกรมที่ทำงานใน Linux

RTLinux ได้จัดเตรียมช่องทางที่จะให้งานในเวลาจริงสามารถติดต่อกับโปรแกรมที่ทำงานใน Linux ไว้ 2 วิธี คือ การติดต่อผ่านทาง FIFO ซึ่งมีลักษณะคล้ายกับการติดต่ออ่าน เขียนแฟ้มข้อมูล และอีกช่องทาง คือติดต่อผ่านทาง Shared Memory

การสื่อสารระหว่างงานในเวลาจริงด้วยกันสามารถทำได้ง่ายเนื่องจากงานในเวลาจริงทั้งหมดใช้ Memory Space เดียวกัน ทำให้สามารถเห็นข้อมูลของกันและกันได้หมด งานในเวลาจริงสามารถจัดสรรเวลาการเข้าถึงข้อมูลได้โดยใช้ semaphores และ mutex ซึ่งเป็นวิธีมาตรฐานของระบบปฏิบัติการแบบ UNIX (และ POSIX) โดยในโครงงานพิเศษนี้ได้ใช้การติดต่อผ่านทาง FIFO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIFO(First In First Out) หรือ เข้าก่อนออกก่อนคือวิธีการรับส่งข้อมูลระหว่างกระบวนการวิธีหนึ่ง เป็นชื่อที่ได้มาจากลักษณะการถ่ายเทข้อมูล โดยข้อมูลที่ส่งเข้าไปใน FIFO ก่อนหากมีการเรียกจะถูกเรียกออกมาก่อน ดังรูปที่ 2.1 ใน RTLinux ได้จัดเตรียมวิธีการสร้าง FIFO ให้ การสร้างจะต้องสร้างด้วยโมดูล การอ่านและเขียนลงไปใน FIFO จึงจะทำได้ทั้งจากโมดูลและโปรแกรมประยุกต์ โดยปกติเราจะสามารถสร้าง FIFO ได้ทั้งหมด 64 ชุด(เราสามารถเพิ่มจำนวนชุดได้จากการปรับแก้ตัวเลือกของ RTLinux ในตัวเลือกที่เกี่ยวกับ Driver) ขนาดหน่วยความจำที่จะใช้สร้าง FIFO แต่ละชุดรวมกันแล้วจะถูกจำกัดโดยขนาดหน่วยความจำของเครื่องคอมพิวเตอร์



รูปที่ 2.2 ลักษณะการทำงานของ FIFO

การอ่านหรือเขียนข้อมูลของ FIFO ในโมดูลจะใช้ฟังก์ชันเฉพาะแต่ในโปรแกรมประยุกต์จะมีการติดต่อกับ FIFO เหมือนกับการติดต่ออ่านหรือเขียนแฟ้มข้อมูลทั่วไปผ่านแฟ้มข้อมูลชื่อ ``/dev/rtf{fifo}" (เมื่อ fifo มีค่าได้ตั้งแต่ 0 ถึง 63) ที่ถูกสร้างขึ้นจากขั้นตอนการติดตั้ง RTLinux หากคอมพิวเตอร์ของท่านใดยังไม่มีให้ไปที่ไดเร็กตอรี่ของ RTLinux ที่ใช้ในการติดตั้งแล้วสั่ง ``make install"

นอกจาก FIFO จะเป็นทางผ่านของข้อมูลแล้ว RTLinux ยังได้เพิ่มความสามารถในการเรียกใช้งานฟังก์ชันที่กำหนดขึ้นหากมีการอ่านหรือเขียน FIFO ด้วยโปรแกรมประยุกต์หรือโมดูล ความสามารถนี้เราจะนำไปใช้ในการสั่งงานโมดูลเวลาจริงให้เริ่มหรือหยุดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สอดคล้องกับมาตรฐานPOSIX 1003.13 PES51 และมีความเป็นธุรกิจมากขึ้น ในขณะที่ RTAI ยังใช้ชุดคำสั่งเดิม และยังคงความเป็น GPL ต่อไป นอกจากนั้น RTAI ยังได้เพิ่มความสามารถของการทำงานในเวลาจริงให้กับโปรแกรมที่ทำงานอยู่บน Linux หรือทำงานอยู่ ใน user space memory ซึ่ง RTLinux ก็ได้เพิ่มส่วนนี้ด้วยเช่นกันแต่มีเฉพาะในรุ่นที่ขายเท่านั้น

นอกจากแนวทางที่ RTLinux ใช้แล้วยังมีอีกแนวทางหนึ่งที่พยายามทำให้ Linux สามารถทำงานในเวลาจริงได้ โดยแนว ทางนี้ได้พยายามพัฒนาให้ Linux kernel จากที่เป็น non-preemptive kernel ไปเป็น preemptive kernel สำหรับในแนวทางนี้ ผู้สนใจสามารถหาข้อมูลได้เพิ่มเติมจากเอกสารอ้างอิง

2.3 พอร์ตขนาน (Parallel Port)

2.3.1ความรู้เบื้องต้นของพอร์ตขนาน

พอร์ตขนาน สาเหตุที่มีชื่อนี้ เนื่องจากการถ่ายทอดข้อมูลของพอร์ตนี้เป็นแบบขนาน การประมวลผลส่วนใหญ่จะขนาน 8 บิต

เพื่อให้เข้าถึงการนำเอาพอร์ตขนานไปใช้งาน ก่อนอื่นต้องทำความเข้าใจก่อนว่า ปกตินั้นการสั่งพิมพ์รายงานจากคอมพิวเตอร์ไปยังพอร์ตขนาน กับเครื่องพิมพ์ เริ่มจากสัญญาณพอร์ตดาต้า (data) ถูกส่งออกไปยังเครื่องพิมพ์ พร้อมทั้งส่งสัญญาณสโตรบ (strobe) ออกไปด้วยเพื่อให้เครื่องรับรู้ว่ามีการส่งข้อมูลใหม่มาที่ขาดาต้า (data) แล้วจากนั้นคอมพิวเตอร์จะต้องรอการตอบกลับจากเครื่องพิมพ์นั่นคือเครื่องพิมพ์จะสร้างสัญญาณ busy หรือเพื่อบอกว่าเครื่องพิมพ์ยังไม่พร้อมที่จะรับข้อมูลใหม่ จนกระทั่งเมื่อเครื่องพิมพ์พร้อม เครื่องพิมพ์จะส่งสัญญาณ Ack ส่งไปยังคอมพิวเตอร์เพื่อแจ้งว่า พร้อมที่จะรับข้อมูล

นอกจากสัญญาณทั้งสามแล้วส่วนใหญ่การติดต่อกับเครื่องพิมพ์ยังต้องมีสัญญาณ อื่นๆ ร่วมด้วย เนื่องจากเครื่องพิมพ์ต้องทำหน้าที่ถึง 3 อย่างด้วยกันคือ รับข้อมูลจากคอมพิวเตอร์, พิมพ์ข้อมูลที่รับเข้ามา และตอบสนองต่อการใช้งานของผู้ใช้

พอร์ตขนานของคอมพิวเตอร์ยังแยกย่อยออกเป็นอีก 3 พอร์ตได้แก่ พอร์ตเอาต์พุตที่ทำหน้าที่ส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ พอร์ตเอาต์พุตสำหรับสัญญาณสโตรบ (Strobe) และรีเซต (Reset) พอร์ตอินพุตสำหรับการอ่านค่าสัญญาณ Acknowledge, busy และสัญญาณ Error จากเครื่องพิมพ์

โดยปกติพอร์ตขนานออกแบบมาให้มีสายสัญญาณอยู่ทั้งหมด 17 เส้น สายสัญญาณเหล่านี้จะมีรีจิสเตอร์ 3 ตัวควบคุมการทำงานดังนี้

1.พอร์ตเอาต์พุตสำหรับข้อมูล 8 เส้นมีรีจิสเตอร์ Data ควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากสัญญาณทั้งสามแล้วส่วนใหญ่การติดต่อกับเครื่องพิมพ์ยังต้องมีสัญญาณ อื่นๆ ร่วมด้วย เนื่องจากเครื่องพิมพ์ต้องทำหน้าที่ถึง 3 อย่างด้วยกันคือ รับข้อมูลจากคอมพิวเตอร์, พิมพ์ข้อมูลที่รับเข้ามา และตอบสนองต่อการใช้งานของผู้ใช้

พอร์ตขนานของคอมพิวเตอร์ยังแยกย่อยออกเป็นอีก 3 พอร์ตได้แก่ พอร์ตเอาต์พุตที่ทำหน้าที่ส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ พอร์ตเอาต์พุตสำหรับสัญญาณสโตรบ (Strobe) และรีเซต (Reset) พอร์ตอินพุตสำหรับการอ่านค่าสัญญาณ Acknowledge, busy และสัญญาณ Error จากเครื่องพิมพ์

โดยปกติพอร์ตขนานออกแบบมาให้มีสายสัญญาณอยู่ทั้งหมด 17 เส้น สายสัญญาณเหล่านี้จะมีรีจิสเตอร์ 3 ตัวควบคุมการทำงานดังนี้

1.พอร์ตเอาต์พุตสำหรับข้อมูล 8 เส้นมีรีจิสเตอร์ Data ควบคุม

2.พอร์ตอินพุตสำหรับการอ่านค่าสถานะต่างๆ จากภายนอกมีอยู่ด้วยกัน 5 เส้นใช้รีจิสเตอร์ STATUS ในการควบคุม

3.พอร์ตเอาต์พุตสำหรับการส่งสัญญาณควบคุมไปยังอุปกรณ์ภายนอก มีอยู่ด้วยกัน 4 เส้นใช้รีจิสเตอร์ Control ในการควบคุม

การติดต่อกับพอร์ตขนานจะต้องมีการอ้างแอดเดรส ตำแหน่งที่ใช้ในการอ้างแอดเดรสถึงจะเป็นตำแหน่ง A0-A9 และใช้ขา IOR และ IOW สำหรับเป็นตัวเลือกว่าต้องการอ่านหรือเขียนรีจิสเตอร์ตัวใด จากการดีโค้ดแอดเดรส A0-A9 นี้เองทำให้ได้สัญญาณออกมาไปควบคุมหรืออีนาเบิลวงจรบัฟเฟอร์ต่างๆ ดังนี้

Data write สัญญาณอีนาเบิลสำหรับนำข้อมูลที่อยู่ในบัส Data ออกไปที่ขาดาต้า (Data) ของพอร์ตขนาน

Data read สัญญาณอีนาเบิลสำหรับอ่านข้อมูลจากขาดาต้า (Data) ของพอร์ตขนานมาเก็บไว้ในบัสของ Data

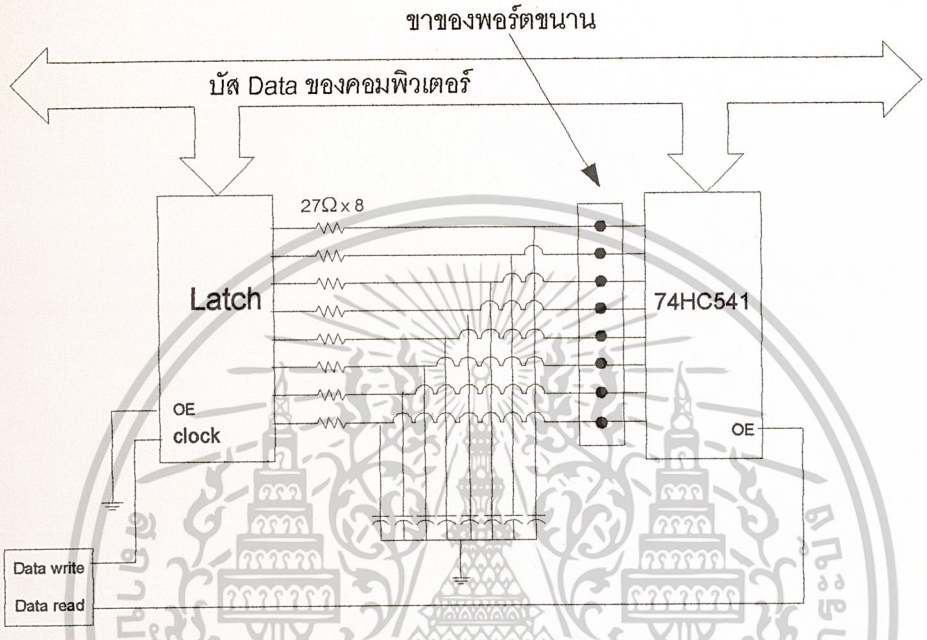
Control write สัญญาณอีนาเบิลสำหรับนำข้อมูลที่อยู่ในบัส Data ออกไปอยู่ที่ขาคอนโทรล (Control) ของพอร์ตขนาน สำหรับพอร์ตนี้นอกจากจะนำข้อมูลออกไปยังพอร์ตขนานแล้ว ยังทำหน้าทิอินเตอร์รัปของการเปลี่ยนแปลงสัญญาณที่พอร์ต Status อีกด้วย

Control read สัญญาณอีนาเบิลสำหรับอ่านข้อมูลจากขาคอนโทรล (Control) มาเก็บไว้ในบัส Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Status read สัญญาณอีนาเบิลสำหรับอ่านข้อมูลจากขาพอร์ตสเตตัส (Status) มาเก็บไว้ในบัส Data

2.3.1.1 พอร์ตดาต้า (Data Port)



รูปที่ 2.3 พอร์ตดาต้า (Data Port)

จากรูปที่ 2.4 แสดงให้เห็นว่าพอร์ตดาต้า (Data Port) ประกอบไปด้วยบัฟเฟอร์ 1 ตัว และไอซีแลตช์อีก 1 ตัว เมื่อคอมพิวเตอร์อ้างการส่งข้อมูล คอมพิวเตอร์จะเขียนข้อมูลไปยังไอซีแลตช์ 1 ตัว ทั้ง 8 บิต เอาต์พุตของไอซีแลตช์ 1 คือ D0-D7 ซึ่งเอาต์พุตนี้จะไปปรากฏอยู่ที่พอร์ตขนานในตำแหน่งขา 2 ถึง 9

สำหรับบัฟเฟอร์สำหรับการอ่านข้อมูลกลับได้แก่เบอร์ 74HC541 ซึ่งเมื่อต้องการอ่านค่าคอมพิวเตอร์จะส่งสัญญาณ Data read ออกมาเพื่ออีนาเบิลไอซี 74HC541 สำหรับพอร์ตขนานแบบมาตรฐาน พอร์ตดาต้า (Data) จะต้องใช้เพื่อการส่งค่าออกเอาต์พุตเท่านั้น แต่สำหรับพอร์ตขนานที่มีการสื่อสารสองทิศทาง สามารถอ่านค่าจากพอร์ตดาต้า (Data) ได้ด้วย แต่ก่อนที่จะอ่านค่าต้องจำไว้เสมอว่าจะต้องป้อนค่าเอาต์พุตให้มีค่าลอจิก “1” ทั้งหมดก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.2 พอร์ตคอนโทรล (Control Port)

DB-25	รีจิสเตอร์	ทิศทาง	ตำแหน่งบิต	ชื่อขาสัญญาณ	หน้าที่การทำงาน
1	Control	Out	C0	STROBE	แอกทีฟ 0 ส่งค่าออกไปเพื่อบอกว่าที่ขา Data มีข้อมูลอยู่แล้ว
2-9	Data	Out	D1-D8	DATA1-DATA2	สำหรับพอร์ตขนานมาตรฐานเดิมขานี้ทำหน้าที่ส่งข้อมูลเท่านั้นสำหรับปัจจุบันขานี้รับข้อมูลอินพุตได้ด้วย
10	Status	In	S6	nACK	เป็นพัลส์ลอจิก 0 ที่ส่งมาจากเครื่องพิมพ์เพื่อบอกว่าได้รับข้อมูลที่ส่งไปแล้ว
11	Status	In	S7	BUSY	เป็นสัญญาณแจ้งมาจากเครื่องพิมพ์ว่ายังไม่พร้อมรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DB-25	รีจิสเตอร์	ทิศทาง	ตำแหน่งบิต	ชื่อขาสัญญาณ	หน้าที่การทำงาน
12	Status	In	S5	PE	แจ้งกระดาษหมด
13	Status	In	S4	SELECT	แจ้งว่าเครื่องพิมพ์ต่ออยู่
14	Control	Out	C1	AUTO FEET	ส่งให้เครื่องพิมพ์เลื่อนบรรทัด
15	Status	In	S3	ERROR	สัญญาณจากเครื่องพิมพ์มายังเครื่องคอมพิวเตอร์เพื่อแสดงข้อผิดพลาดจากการพิมพ์
16	Control	Out	C2	INIT	รีเซตเครื่องพิมพ์โดยให้ลอจิก 0
17	Control	In	C3	SELECT-IN	ส่งสัญญาณไปยังเครื่องพิมพ์เพื่อแจ้งว่าต้องการเลือกเครื่องพิมพ์เครื่องนี้
18-25				GND	กราวด์

ตารางที่ 2.2 สถานะของขาคอนเนคเตอร์

พอร์ตคอนโทรล (Control) ใช้สำหรับคอมพิวเตอร์ควบคุมเครื่องพิมพ์ จากตารางที่ 2.2 จะเห็นว่าพอร์ตควบคุม ประกอบไปด้วยบิตเอาต์พุต 4 บิตที่ต่อออกไปยังเครื่องพิมพ์ ส่วนบิตอีนาเบิลอินเตอรัปต์ไม่ได้ถูกต่อออกไป เอาต์พุตของพอร์ตควบคุมมีอินเวอร์เตอร์แบบคอลเลคเตอร์เปิดต่อร่วมอยู่ด้วย โดยเอาต์พุตเหล่านี้จะถูกพูลอัปไว้ด้วยตัวต้านทานค่า 4.7 กิโลโอห์ม สำหรับบิต C2 จะผ่านอินเวอร์เตอร์ถึงสองตัวทำให้เอาต์พุตของบิต C2 ไม่มีการกลับสถานะลอจิก

สถานะของพอร์ตควบคุมสามารถอ่านค่ากลับได้โดยการใช้บัฟเฟอร์ 74HC541 ซึ่งเอาต์พุตของ 74HC541 มีอินเวอร์เตอร์อยู่ภายใน ทำให้ค่าที่อ่านได้ตรงกับค่าที่ส่งออกไป การควบคุมการอ่าน และเขียนข้อมูลจากพอร์ตควบคุม คอมพิวเตอร์จะส่งข้อมูลที่ขา Control write และขา Control read เนื่องจากเอาต์พุตของพอร์ตคอนโทรล (Control) เป็นแบบคอลเลคเตอร์แบบเปิด ดังนั้นผู้ใช้สามารถอ่านค่าอินพุตจากสัญญาณจากภายนอกได้ โดยก่อนที่จะอ่านค่าจะต้องทำให้ขาพอร์ตที่ต้องการอ่านมีลอจิก “1” เสียก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
58666

2.3.1.3 พอร์ตแสดงสถานะหรือพอร์ตสเตตัส (Status)

พอร์ตสเตตัส (Status) เป็นพอร์ตที่คอมพิวเตอร์ใช้สำหรับการอ่านค่าสถานะจากเครื่องพิมพ์ จะสังเกตได้ว่ามีขาสัญญาณอยู่ทั้งหมด 5 สัญญาณด้วยกัน และจะเรียกชื่อเป็นS3, S4, S5, S6 และ S7 ซึ่งตัวเลขนั้นหมายถึงตำแหน่งบิตของขาเหล่านี้ภายในรีจิสเตอร์ Status นั่นเองสำหรับบิต S7 จะมีค่าต่างจากบิตอื่นๆ ที่เมื่อสัญญาณจากภายนอกเข้ามาแล้วจะไม่ผ่านอินเวอร์เตอร์ ในขณะที่ขาอื่นๆ ผ่านอินเวอร์เตอร์ทั้งหมด ดังนั้นเมื่อข้อมูลผ่านจากขาอินพุตไปยัง 74HC541 ซึ่งเอาต์พุตมีการกลับสถานะทำให้บิต S7 เป็นบิตเดียวที่กลับสถานะ นอกจากนี้ในการใช้งานถ้าต้องการใช้งานในการสร้างสัญญาณอินเตอร์รัปจากของขอบขาขึ้นของขา S6 สามารถกำหนดค่าได้จากพอร์ตคอนโทรล (Control) บิต4

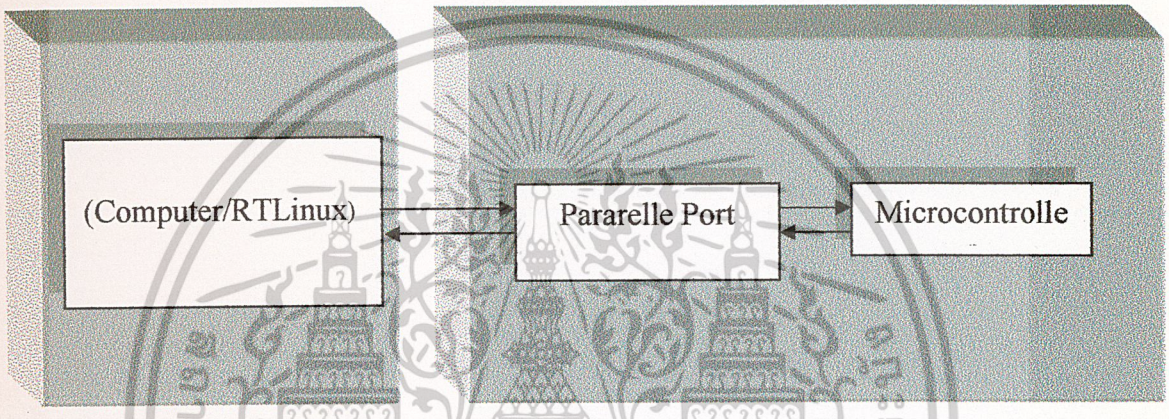
2.3.2 การนำพอร์ตขนานไปใช้งาน

สำหรับพอร์ตขนานแบบมาตรฐาน ผู้ใช้สามารถนำพอร์ตอินพุต 5 บิตพอร์ต เอาต์พุต 4 บิต และพอร์ตเอาต์พุตอีก 8 บิต ไปใช้งานได้โดยตรง โดยที่ 4 บิตของพอร์ตเอาต์พุตหรือพอร์ตคอนโทรล (Control) นั้นสามารถดัดแปลงให้เป็นพอร์ตอินพุตขนาด 4 บิตได้ด้วย ดังนั้นผู้ใช้งานจึงสามารถนำสัญญาณจากพอร์ตขนานที่มากถึง 17 เส้น ไปใช้งานในการควบคุมโดยใช้ระดับสัญญาณ TT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีดำเนินการวิจัย

ในโครงงานนี้มีผังการดำเนินงานการทดลองตามรูปที่ 3.1 โดยมีส่วนประกอบอยู่ 2 ส่วน คือ **หมายเลข 1** เป็นส่วนประมวลผล และ **หมายเลข 2** เป็นส่วนรับ-ส่งข้อมูล



รูปที่ 3.1 แผนผังการจัดแสดงโครงงาน

3.1 ส่วนประมวลผล

ส่วนประมวลผลประกอบไปด้วย Hardware (Computer), Software ที่คอยประมวลผลเมื่อรับ Data เข้ามา โดยมีขั้นตอนการวิจัย ดังนี้

3.1.1 ทำการติดตั้ง Linux และ RTLinux

โดยทำการติดตั้งลีนุกซ์โดยได้ใช้ลีนุกซ์เรดแฮดในการติดตั้งโดยสามารถดาวโหลดได้ที่ http://www.redhat.com แล้วทำการติดตั้ง เรียลไทม์ลีนุกซ์ โดยเราจะติดตั้ง เรียลไทม์เคอร์เนล การติดตั้ง โมดูล ที่จะใช้สำหรับการทำงานในเวลาจริง โดยในการเตรียมการติดตั้งจะต้อง Download rtlinux2.0 โดยในส่วนการติดตั้งเรียลไทม์ลีนุกซ์ต้องทำการขยายไฟล์ของrtlinux-2.0 และ เคอร์เนล 2.2.13 เมื่อถึงขั้นนี้จะได้ต้นรหัสของ kernel ที่ทำการแก้ไขบางส่วนให้ความสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทำงานตามเวลาจริง พร้อมที่จะทำการปรับค่าตัวเลือกและคอมไพล์ต่อไป การปรับค่าตัวเลือกของ kernel จะใช้คำสั่ง *make config*, *make menuconfig* หรือ *make xconfig* ก็ได้ สำหรับคำสั่ง *make config* และ *make menuconfig* เป็นการใช้งานใน Text Mode (Terminal Console) การใช้งานค่อนข้างจะยุ่งยากมากกว่าการใช้คำสั่ง *make xconfig* ที่ทำงานใน Graphic Mode (X-Windows) เพื่อให้ง่ายขอแนะนำให้ใช้คำสั่ง ``make xconfig'' ในการปรับค่าตัวเลือกของ kernel การติดตั้ง kernel สำหรับ Linux ของค่าย RedHat และ Mandrake รุ่นใหม่ๆ คอมไพเลอร์ gcc ในส่วนติดตั้ง modules ของ RTLinux หลังจากที่สามารถบูทเครื่องได้โดย RT-Kernel แล้ว อันดับต่อไปจะเป็นการติดตั้ง modules ของ RTLinux เพื่อใช้ในการเพิ่มคุณสมบัติให้ kernel สามารถทำงานในเวลาจริงได้

3.1.2 ศึกษาการเขียนโปรแกรมและเขียนโปรแกรมเพื่อรับส่งข้อมูลตาเวลาจริงลงใน RTLinux

โดยใช้ภาษา C ในการเขียนโปรแกรมและใช้ภาษาแอสแซมบลีในการควบคุม AT89C2051 เพื่อสร้างสัญญาณพัลส์

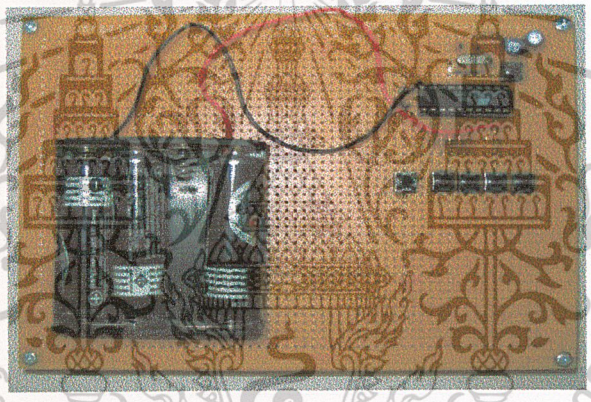
3.2 ส่วนส่งข้อมูล(สัญญาณพัลส์)

ส่วนรับส่งข้อมูลประกอบไปด้วยวงจรรับ-ส่งสัญญาณโดยในส่วนของวงจรรับ-ส่งสัญญาณ Pararellel Port, Microcontroller โดยในส่วนนี้ได้ทำการใช้ AT89C2051 ในการสร้างสัญญาณพัลส์โดยในส่วนนี้จะมีการสร้างสัญญาณพัลส์ได้ต่ำสุดที่ 10 เฮิรตส์และสูงสุดที่ 100 กิโลเฮิรตส์ โดยมีสวิตช์ในการควบคุมโดยแบ่งหน้าที่ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

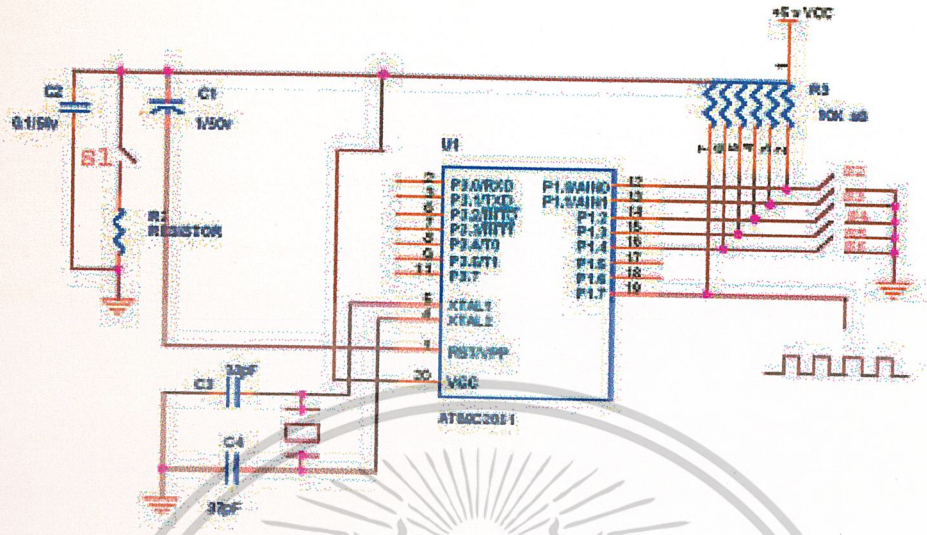
	การทำงาน
สวิตส์ที่1(S1)	ปิด-เปิดการทำงาน
สวิตส์ที่2(S2)	เป็นตัวResetการทำงาน
สวิตส์ที่3(S3)	เป็นตัวปรับสัญญาณพัลส์ 10-100 Hz
สวิตส์ที่4(S4)	เป็นตัวปรับสัญญาณพัลส์ 100-1000 Hz
สวิตส์ที่5(S5)	เป็นตัวปรับสัญญาณพัลส์ 1-10 kHz
สวิตส์ที่6(S6)	เป็นตัวปรับสัญญาณพัลส์ 10-100 kHz

ตารางที่ 3.1 หน้าที่การทำงานของตัวส่งข้อมูล



รูปที่ 3.2 ตัวส่งข้อมูล(สัญญาณพัลส์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจรตัวส่งข้อมูล

3.3 ทำการทดลองการทำงานของ Computer และ Microcontroller

เมื่อทราบการทำงานของ ลีนุกซ์และเรียลไทม์ลีนุกซ์ แล้วจึงนำมาเชื่อมต่อกับส่วนของวงจรรับ-ส่งสัญญาณแล้วทำการทดสอบการรับ-ส่งข้อมูลตามเวลาจริงโดยได้ทำการเปรียบเทียบระหว่างเรียลไทม์ลีนุกซ์และลีนุกซ์ทั่วไป โดยทำการเขียนโปรแกรมขึ้นมาทดสอบการทำงานโดยแบ่งการทดสอบออกเป็นโปรแกรมที่ใช้ในการรับข้อมูลและโปรแกรมที่ใช้ในการเพิ่มการทำงานของซีพียู โดยโปรแกรมนี้เป็นโปรแกรมที่เพิ่มการทำงานของซีพียูโดยการให้โปรแกรมทำการคำนวณทางคณิตศาสตร์และทำงานตลอดเวลา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลองและอภิปรายผล

โครงงานนี้ประกอบไปด้วย 2 ส่วน คือ การติดตั้ง ลีนุกซ์และเรียลไทม์ลีนุกซ์และ การทดสอบการรับส่งข้อมูลโดยจะแสดงออกมาทางหน้าจอ

4.1 ผลการทดลอง

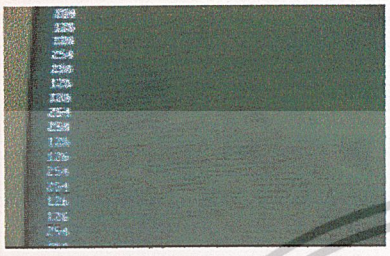
การส่งข้อมูลจะเป็นการส่งมาทางพอร์ทขนานเมื่อสัญญาณพัลซ์ส่งมาจาก Microcontroller จะทำให้ RTLinux สามารถทำงานได้ทันทีโดยเปรียบเทียบกับลีนุกซ์ โดยเราจะจับสัญญาณทุกๆ 1 นาทีโดยได้ทำการเขียนโปรแกรมออกมาสองโปรแกรมเพื่อทำการเปรียบเทียบและดูความแตกต่างระหว่างลีนุกซ์และเรียลไทม์ลีนุกซ์โดยโปรแกรมที่ 1 เป็นโปรแกรมที่ทำการรับส่งข้อมูลในเรียลไทม์ในที่นี้จะเขียนในเคอร์เนลเราเรียกว่าโมดูลส่วนโปรแกรมที่2จะเป็นโปรแกรมการคำนวณทางคณิตศาสตร์เพื่อเป็นการเปรียบเทียบและเพิ่มการทำงานของซีพียู

รูปที่ 4.1 สัญญาณที่ใช้ในการทดลองที่ความถี่ 700 Hz

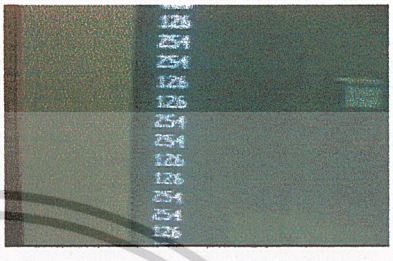
โดยผลการทดลองจะแบ่งออกเป็น 2 ส่วนดังนี้(ในที่นี้เราจะดูการทำงานของโปรแกรมรับส่งข้อมูลเมื่อมีการทำงานเพียงโปรแกรมเดียวและมีการทำงานของโปรแกรมอีกโปรแกรมหนึ่งพร้อมกัน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 เป็นการเปรียบเทียบระหว่างลีนุกซ์ที่ทำงานโปรแกรมเดียวกับทำงานหลายโปรแกรม



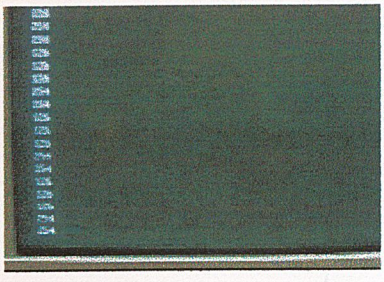
รูปที่ 4.2 การทำงานโปรแกรมเดียวในลีนุกซ์



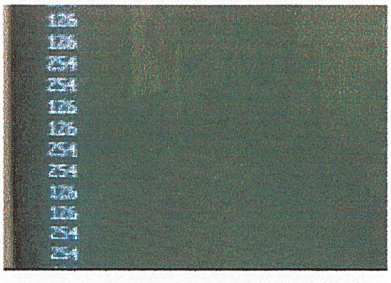
รูปที่ 4.3 การทำงานของสองโปรแกรมในลีนุกซ์

โดยในรูปที่ 4.2 จะเป็นการทำงานของโปรแกรมรับส่งข้อมูลเพียงอย่างเดียว ส่วนในรูปที่ 4.3 จะเป็นการทำงานของโปรแกรมรับส่งข้อมูลและโปรแกรมคณิตศาสตร์พร้อมกัน โดยรูปที่ 4.2 และ 4.3 จะแสดงให้เห็นถึงผลของการทำงานทั้งสองแบบทางหน้าจอคอมพิวเตอร์ จากการที่ได้สังเกตการณ์ทำงานทั้งสองแบบทั้งการงานแบบโปรแกรมเดียวและการทำงานแบบสองโปรแกรมในลีนุกซ์ การทำงานของทั้งสองแบบทำให้เห็นได้ว่ามีการกระตุกของการทำงานโดยในรูปที่ 4.2 มีการทำงานของโปรแกรมที่สม่ำเสมอส่วนรูปที่ 4.3 จะมีการกระตุกในขณะที่มีการสั่งพิมพ์ค่าตัวเลขออกมาโดยเกิดจากการที่ซีพียูทำการประมวลผลทั้งสองโปรแกรมพร้อมกัน

4.1.2 การเปรียบเทียบระหว่างเรียลไทม์ลีนุกซ์ที่ทำงานโปรแกรมเดียวกับทำงานหลายโปรแกรม



รูปที่ 4.4 ทำงานโปรแกรมเดียว ในเรียลไทม์



รูปที่4.5 ทำงานสองโปรแกรมในเรียลไทม์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในรูปที่ 4.4 จะเป็นการทำงานของโปรแกรมรับส่งข้อมูลเพียงอย่างเดียว ส่วนในรูปที่ 4.5 จะเป็นการทำงานของโปรแกรมรับส่งข้อมูลและโปรแกรมคณิตศาสตร์พร้อมกัน โดยรูปที่ 4.4 และ 4.5 จะแสดงให้เห็นถึงผลของการทำงานทั้งสองแบบทางหน้าจอคอมพิวเตอร์ จากการที่ได้สังเกตการณ์ทำงานทั้งสองแบบทั้งการงานแบบโปรแกรมเดียวและการทำงานแบบสองโปรแกรมในเรียลไทม์ การที่ได้สังเกตการทำงานของทั้งสองแบบทำให้เห็นได้ว่าไม่มีการกระตุกของการทำงานโดยในรูปที่ 4.4 มีการทำงานของโปรแกรมที่สม่ำเสมอ ส่วนรูปที่ 4.5 มีการทำงานของโปรแกรมที่สม่ำเสมอออกมาโดยเกิดจากการเรียลไทม์ลีนุกซ์หยุดการทำงานของโปรแกรมคณิตศาสตร์จึงทำให้ซีพียูประมวลผลของโปรแกรมประยุกต์เพียงอย่างเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการทดลองโครงงานการรับส่งข้อมูลตามเวลาจริง โดยโครงงานนี้ประกอบไปด้วย 3 ส่วน คือ

1. การติดตั้ง Linux, RTlinux
2. สร้างตัวส่งข้อมูลและการเขียนโปรแกรมเพื่อรับส่งข้อมูล
3. การทดสอบการทำงานตามเวลาจริง

โดยในโครงงานนี้จะใช้ระบบปฏิบัติการลีนุกซ์ซึ่งทำการคอมไพล์เคอร์เนลใหม่โดยใช้ RTLinux 2.0 เป็นส่วนขยายเพิ่มเติม ให้ลีนุกซ์สามารถทำงานได้ตามเวลาจริง ซึ่งสามารถทำให้รับข้อมูลที่ถูกส่งมาได้ครบถ้วนสมบูรณ์ ในส่วนของการทำงานของซีพียูนั้นสามารถเพิ่มประสิทธิภาพในการทำงานและการประมวลผลเพื่อให้ได้ผลออกมาตามเวลาจริง โดยสามารถดูได้จากผลการทดลองที่ 4.1.2 (ในขณะที่ซีพียูกำลังทำการประมวลผลของโปรแกรมคณิตศาสตร์และในขณะเดียวกันก็ทำการรันโปรแกรมรับส่งข้อมูลแบบเรียลไทม์ซีพียูก็จะทำการหยุดการทำงานของโปรแกรมคณิตศาสตร์แล้วหันมาให้ความสำคัญของโปรแกรมรับส่งข้อมูลแทน) ซึ่งจะต่างจากลีนุกซ์ทั่วไปที่จะสลับไปทำงานอื่นก่อนทำให้รับข้อมูลที่ถูกส่งมาได้ไม่ครบถ้วนและจากผลการทดสอบนี้เรียลไทม์ลีนุกซ์สามารถทำงานในเวลาจริงได้ดีกว่าลีนุกซ์

5.2 ปัญหาที่พบและข้อเสนอแนะ

1. ในการทำการติดตั้ง RTLinux จะมีปัญหาในเรื่องฮาร์ดแวร์ของ Computer ไม่สามารถรองรับกับเพกเกจของ RTLinux แนวทางแก้ไขคือควรมีการศึกษา RTLinux ที่เรานำมาใช้สามารถรองรับกับฮาร์ดแวร์ของ Computer
2. ในส่วนของการนำไปประยุกต์สามารถนำไปประยุกต์ใช้กับเครื่องจักรในโรงงานอุตสาหกรรม, อุปกรณ์ต่างๆที่ต้องการความเที่ยงตรงของเวลาในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

David Pitts and Bill Ball(1998).Red Hat Linux THIRD EDITION.Indianapolis USA: Printed in the United States of America.

Bill W.Crum.An Application of Data Acquisition using Real-Time Linux.New Maxico, 1997.

พิเชษฐ์ บุญหนุน.Linux สำหรับงานจริง.http://linux.thai.net/plone/Members/rut/RTLinux.

RTLinux,

"http://www.fsmlabs.com/"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการการติดตั้งRTLinux

Copy the tar-archive to /usr/src/rtlinux-2.0-prepatched.tgz

Untar the archive: **tar -xzvf /usr/src/rtlinux-2.0-prepatched.tgz**

The directory /usr/src/rtlinux-2.0 will be created.

In this directory there is a directory called linux.

Create a symbolic link: /usr/src/linux -> /usr/src/rtlinux-2.0/linux

rm /usr/src/linux

ln -s /usr/src/rtlinux-2.0/linux /usr/src/linux

2. Go into the /usr/src/linux directory and create your kernel-configuration

make config

or

make menuconfig

or

make xconfig

Whatever you do, be sure to select the "hard real time" option in the basic configuration (under "Processor type and features").

Also, please disable APM support (under "general setup").

APM BIOS calls may have unpredictable effect on real-time performance.

(run also the command **"chkconfig --del apmd"** to make sure that apmd will not be started in any runlevel)

3. Compile the kernel

make dep

make bzImage

4. Create and install the modules

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

make modules

make modules_install

6. Create the rtl-system-map (/boot/System.map-<ver>-rtl2.0)

(this will also copy the kernel to /boot/vmlinuz-2.2.13-rtl2.0)

make install

7. Edit your lilo-configuration to be able to boot the real-time kernel

for example:

boot=/dev/hda

map=/boot/map

install=/boot/boot.b

promt

timeout=50

default=rtlinux

image=/boot/vmlinuz-<normalversion>

label=linux

initrd=/boot/initrd-<normalversion>

read-only

root=/dev/<yourroot>

image=/boot/vmlinuz-2.2.13-rtl2.0

label=rtlinux

read-only

root=/dev/<yourroot>

run lilo:

/sbin/lilo

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. Boot the real-time kernel

shutdown -r now

You can verify whether the rt-linux symbols are in the kernel

/sbin/ksyms -a | grep rt_

9. Now you can cd to the /usr/src/rtlinux-2.0/rtl/ directory and type

make

make install

this will install real-time modules (rtl_fifo.o, rtl_sched.o, ...),

the rtlinux man pages,

as well as some header-files (/usr/include/rtlinux/rtl_*.h)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source code

โปรแกรมที่ใช้เขียนบน real time linux

```
#include<rtl.h>
#include<time.h>
#include<pthread.h>
#include<asm/io.h>
#include<rtl_sync.h>
#include<linux/kd.h>
#include<rtl_fifo.h>

#defind BUFSIZE 65535
unsigned int buf[BUFSIZE];

pthread_t thread;

void *start_routine(void *arg)
{
        struct sched_param p;
        int o,n,d;
        unsigned int count;
        count=9990;
        o=0;

        outb(0xff, 0x378);
        p . sched_priority=1;
        pthread_setschedparam(pthread_self(), SCHED_FIFO, &p);

        pthread_make_periiodic_np(pthread_self(), gethrtime(), 100000000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        while (1)
        {
                n=inb(0x379);
                d=n-o;
                o=n;
                if(d<0)
                {
                        count++;
                        if(count>10000)
                                count=1;
                rtf_put(0,buf,count);
                }
                pthread_wait_np();

        }
        return 0;
}

int init_module(void);
        rtf_create(0,1024*1024);
        return pthread_create(&thread,NULL,start_routine,0);
}

void cleanup_module(void) {
        rtf_destroy(0);
        pthread_delete_np(thread);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้