



ระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย  
โปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา ด้วย Nagios บน Raspberry Pi  
Network Devices Service Status Monitoring System of Computer  
Department Songkhla Rajabhat University with Nagios on Raspberry Pi



กฤษดา เบ็ญกาเสมอ  
พงษ์นรินทร์ พงศาปาน

โครงการหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์  
โปรแกรมวิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี  
มหาวิทยาลัยราชภัฏสงขลา

2559

ชื่อโครงการ ระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่ายโปรแกรมวิชาคอมพิวเตอร์  
มหาวิทยาลัยราชภัฏสงขลา ด้วย Nagios บน Raspberry Pi

ชื่อผู้จัดทำ นายกฤษดา เป็นกาเสมอ รหัสนักศึกษา 564234044  
นายพงษ์นรินทร์ พงศาปาน รหัสนักศึกษา 564234059

---

**อาจารย์ที่ปรึกษา**

**คณะกรรมการสอบ**

..... อาจารย์ที่ปรึกษาหลัก ..... ประธานกรรมการ  
(อาจารย์ภาณุกร ภูริปัญญาพันธ์) (ผู้ช่วยศาสตราจารย์ ดร.ศศลักษณ์ ทองขาว)

..... อาจารย์ที่ปรึกษาร่วม (อาจารย์โชติธรรม ธารรักษ์)  
(อาจารย์โชติธรรม ธารรักษ์) ..... กรรมการ

..... (อาจารย์ญาณพัฒน์ ชูชื่น)

**อาจารย์ประจำวิชา**

..... กรรมการและเลขานุการ  
(อาจารย์จกสิทธิ์ โอบาริกชาติ)  
.....  
(ผู้ช่วยศาสตราจารย์ ดร.ทวิรัตน์ นวลช่วย)

โปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา อนุมัติโครงการฉบับนี้เป็นส่วนหนึ่ง  
ของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

.....

(อาจารย์โชติธรรม ธารรักษ์)

ประธานโปรแกรมวิชาคอมพิวเตอร์

ชื่อโครงการ	ระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่ายโปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา ด้วย Nagios บน Raspberry Pi		
ชื่อผู้จัดทำ	นายกฤษดา เป็นกาสม	รหัสนักศึกษา	564234044
	นายพงษ์นรินทร์ พงศาปาน	รหัสนักศึกษา	564234059
ระดับการศึกษา	วิทยาศาสตร์บัณฑิต		
ปีการศึกษา	2559		

## บทคัดย่อ

โปรแกรมวิชาคอมพิวเตอร์ เป็นโปรแกรมวิชาหนึ่งที่อยู่ภายใต้สังกัดของ คณะวิทยาศาสตร์และเทคโนโลยี ซึ่งในโปรแกรมวิชาคอมพิวเตอร์ มีการใช้อุปกรณ์เครือข่ายและเครื่องบริการภายในโปรแกรมวิชาเป็นจำนวนมากไม่น้อย ผู้ดูแลระบบของโปรแกรมวิชาจึงต้องทำการตรวจสอบการทำงานของอุปกรณ์และเครื่องบริการสม่ำเสมอ เพื่อให้ทราบว่าอุปกรณ์และเครื่องบริการนั้นมีการทำงานที่เป็นปกติ และเมื่ออุปกรณ์หรือเครื่องบริการเกิดสถานะหยุดให้บริการอาจทำให้ผู้ดูแลระบบทราบถึงสถานะการหยุดการทำงานของอุปกรณ์หรือเครื่องบริการได้ล่าช้า จึงทำให้การแก้ไขให้กลับมาทำงานปกติไม่ทันท่วงที

ระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย เป็นระบบสำหรับการตรวจสอบสถานะการหยุดให้บริการของอุปกรณ์เครือข่าย โดยสามารถตรวจสอบสถานะของระบบได้ผ่านเว็บไซต์พร้อมทั้งสามารถส่งข้อความสั้นไปยังโทรศัพท์เคลื่อนที่ของผู้ดูแลระบบ ทำให้ผู้ดูแลระบบสามารถทราบเมื่ออุปกรณ์เครือข่ายเกิดสถานะการหยุดให้บริการได้รวดเร็วยิ่งขึ้น จึงสามารถแก้ไขปัญหาได้อย่างทันท่วงที อีกทั้งผู้ดูแลระบบสามารถตรวจสอบการทำงานของอุปกรณ์เครือข่ายทั้งในเวลาปัจจุบัน และย้อนหลังได้โดยแสดงผ่านเว็บไซต์ ทั้งในรูปแบบของเอกสาร ตาราง และกราฟ

ดังนั้นระบบสามารถตอบสนองความต้องการในการตรวจสอบ และมีความสามารถในการแจ้งเตือนเมื่อเกิดสถานะการหยุดให้บริการไปยังผู้ดูแลระบบผ่านทางข้อความสั้น (SMS) และแสดงสถานะผ่านทางเว็บไซต์ ทำให้ผู้ดูแลระบบสามารถทราบปัญหา และแก้ไขปัญหาได้อย่างทันท่วงที

## กิตติกรรมประกาศ

ระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย โปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา ด้วย Nagios บน Raspberry Pi (Network Devices Service Status Monitoring System of Computer Department Songkhla Rajabhat University with Nagios on Raspberry Pi) สำเร็จได้ด้วยความช่วยเหลือจากบุคคลหลายฝ่ายดังนี้

อาจารย์ที่ปรึกษาหลักโครงการ อาจารย์ภาณุกร ภูริปัญญานันท์ และอาจารย์ที่ปรึกษาร่วมโครงการ โชติธรรม ธารรักษ์ ที่คอยให้คำแนะนำ คำปรึกษา และสละเวลาส่วนตัวเพื่อให้คำแนะนำ ตั้งแต่เริ่มต้นจนกระทั่งโครงการสำเร็จลุล่วงไปได้ด้วยดี รวมถึงให้ความช่วยเหลือในการแก้ปัญหาต่าง ๆ ที่เกิดขึ้นในระหว่างการดำเนินโครงการ และตรวจสอบความถูกต้องในการจัดทำโครงการ

คณะกรรมการตรวจสอบโครงการวิทยาศาสตร์บัณฑิตทุกท่าน ที่กรุณาตรวจสอบความถูกต้อง และให้คำปรึกษาโครงการครั้งนี้

อาจารย์ประจำโปรแกรมวิชาคอมพิวเตอร์ทุกท่าน ที่คอยประสิทธิ์ประสาทวิชาความรู้ทางวิชาการ ซึ่งทางผู้จัดทำโครงการได้นำความรู้มาใช้เป็นพื้นฐานในการทำโครงการ

โครงการนี้จะไม่สำเร็จลุล่วงไปได้ หากปราศจากแรงสนับสนุนจากบุคคลดังรายนามข้างต้น ทางผู้พัฒนาจึงขอขอบคุณทุก ๆ ท่านเป็นอย่างสูงมา ณ โอกาสนี้

คณะผู้จัดทำ

2 ตุลาคม 2560

## สารบัญ

	หน้า
บทคัดย่อ .....	I
กิตติกรรมประกาศ .....	II
สารบัญ .....	III
สารบัญตาราง .....	VI
สารบัญภาพ .....	XIII
<b>บทที่ 1 บทนำ .....</b>	<b>1</b>
หลักการและเหตุผล .....	1
เป้าหมายของโครงการ .....	2
วัตถุประสงค์ของโครงการ .....	2
ขอบเขตและความสามารถของระบบ .....	2
แผนการดำเนินงาน .....	3
1. ขั้นตอนการดำเนินงาน .....	3
2. ระยะเวลาการดำเนินงาน .....	4
สถานที่ดำเนินโครงการ .....	4
เครื่องมือและอุปกรณ์ที่ใช้ในการพัฒนา .....	5
1. ฮาร์ดแวร์ .....	5
2. ด้านซอฟต์แวร์ .....	5
ประโยชน์ที่คาดว่าจะได้รับ .....	6
อาจารย์ที่ปรึกษาโครงการ .....	6
อาจารย์ประจำวิชา .....	6
<b>บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง .....</b>	<b>7</b>
ความรู้พื้นฐานในการทำระบบ .....	7

## สารบัญ (ต่อ)

	หน้า
1. Nagios .....	8
2. SSH .....	9
3. SNMP .....	9
4. SMS Tools 3.....	15
5. Perl Script .....	15
6. PNP .....	17
7. NRPE.....	18
8. OID.....	19
9. RRDTOol.....	22
10.DNS.....	22
11.GSM.....	24
12.งานวิจัยที่เกี่ยวข้อง .....	24
สรุปงานวิจัยที่ได้จากการศึกษา .....	26
<b>บทที่ 3 การวิเคราะห์และออกแบบระบบ .....</b>	<b>27</b>
ความต้องการของระบบ .....	27
นิยามคำศัพท์พร้อมความหมายที่เกี่ยวข้อง.....	27
การวิเคราะห์และออกแบบระบบ .....	28
การออกแบบระบบ .....	42
1. แนวคิดในการออกแบบระบบ .....	42
2. โครงสร้างหน้าจอ.....	43
ส่วนประกอบหลักของระบบ .....	44

## สารบัญ (ต่อ)

	หน้า
1. หลักการ Monitoring ของ Nagios .....	44
2. ความปลอดภัยของ Nagios Server .....	45
<b>บทที่ 4 การพัฒนาระบบและการทดสอบ.....</b>	<b>46</b>
โครงสร้างการดำเนินงาน .....	46
หลักการทำงานของโปรแกรม.....	47
1. การเข้าใช้งานระบบของผู้ดูแล .....	47
2. การติดตั้งโปรแกรมที่ใช้สำหรับตรวจสอบอุปกรณ์บนเครือข่าย ดังนี้ .....	51
การพัฒนาโปรแกรม.....	51
การทดสอบระบบ .....	166
<b>บทที่ 5 บทสรุปและข้อเสนอแนะ.....</b>	<b>176</b>
สรุปผลการดำเนินงาน.....	176
ปัญหาและอุปสรรคในการดำเนินงาน.....	176
ข้อจำกัดของระบบ .....	176
ข้อเสนอแนะ .....	177
<b>บรรณานุกรม.....</b>	<b>178</b>
<b>ภาคผนวก ก</b>	
<b>สภาพแวดล้อมของระบบและการติดตั้ง .....</b>	<b>180</b>
สภาพแวดล้อมของระบบ .....	181
1. การติดตั้งระบบปฏิบัติการ Raspbian (Linux Debian).....	181
2. การติดตั้งโปรแกรม SSH Secure Shell Client.....	195
<b>ประวัติผู้จัดทำโครงการ .....</b>	<b>200</b>

## สารบัญตาราง

ตารางที่	หน้า
1.1 ระยะเวลาในการดำเนินงาน.....	4
4.1 ไฟล์ Config เพื่อตรวจสอบการให้บริการของอุปกรณ์บนเครือข่าย.....	52
4.2 รายละเอียดของการ Config ไฟล์ contacts.cfg ที่นำมาใช้งานในระบบ.....	54
4.3 รายละเอียดการกำหนด Host ไฟล์ localhost.cfg.....	62
4.4 รายชื่ออุปกรณ์ที่ใช้ระบบปฏิบัติการ Linux ที่ระบบจะทำการตรวจสอบสถานะ.....	63
4.5 รายละเอียดการกำหนดค่าของ Host Group ในระบบ.....	63
4.6 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการ Ping ในไฟล์ localhost.cfg.....	65
4.7 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบพื้นที่ว่างของฮาร์ดดิสก์ในไฟล์ localhost.cfg.....	67
4.8 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบจำนวนผู้ใช้งานของอุปกรณ์เครือข่าย ในไฟล์ localhost.cfg.....	69
4.9 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการประมวลผลทั้งหมดของอุปกรณ์ เครือข่าย ในไฟล์ localhost.cfg.....	71
4.10 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการไหลของอุปกรณ์ ไฟล์ localhost.cfg.....	73
4.11 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการแลกเปลี่ยนการทำงาน ของอุปกรณ์ ไฟล์ localhost.cfg.....	75
4.12 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะ SSH ของอุปกรณ์บนเครือข่าย ไฟล์ localhost.cfg.....	78
4.13 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะ HTTP ของอุปกรณ์บนเครือข่าย ไฟล์ localhost.cfg.....	80
4.14 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบขนาดของการแลกเปลี่ยนการทำงาน ของอุปกรณ์บนเครือข่าย โดยใช้โปรโตคอล SNMP ไฟล์ localhost.cfg.....	81
4.15 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบพื้นที่ว่างของการแลกเปลี่ยนการทำงาน ของอุปกรณ์บนเครือข่าย โดยใช้โปรโตคอล SNMP ไฟล์ localhost.cfg.....	82



## สารบัญตาราง (ต่อ)

ตารางที่	หน้า
4.16 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบขนาดของหน่วยความจำหลัก ของอุปกรณ์บนเครือข่าย โดยใช้โปรโตคอล SNMP ไฟล์ localhost.cfg .....	83
4.17 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบพื้นที่ว่างของหน่วยความจำหลัก ของอุปกรณ์บนเครือข่าย โดยใช้โปรโตคอล SNMP ไฟล์ localhost.cfg .....	84
4.18 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบระยะเวลาทำงานของอุปกรณ์ บนเครือข่าย โดยใช้โปรโตคอล SNMP ไฟล์ localhost.cfg .....	85
4.19 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบพื้นที่ว่างฮาร์ดดิสก์ของอุปกรณ์ บนเครือข่าย ไฟล์ localhost.cfg .....	86
4.20 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบผู้ใช้งานอุปกรณ์บนเครือข่าย โดยใช้ NRPE ไฟล์ localhost.cfg .....	87
4.21 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบการประมวลผลทั้งหมดของอุปกรณ์บนเครือข่าย โดยใช้ NRPE ไฟล์ localhost.cfg .....	87
4.22 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบการไหลของอุปกรณ์บนเครือข่าย โดยใช้ NRPE ไฟล์ localhost.cfg .....	88
4.23 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบการแลกเปลี่ยนการทำงานของ อุปกรณ์บนเครือข่าย โดยใช้ NRPE ไฟล์ localhost.cfg .....	89
4.24 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบการประมวลผลที่สิ้นสุดแล้ว แต่ยังคงค้างอยู่ในระบบของอุปกรณ์บนเครือข่าย โดยใช้ NRPE ไฟล์ localhost.cfg .....	90
4.25 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบพื้นที่ว่างในฮาร์ดดิสก์ของอุปกรณ์บน เครือข่าย โดยใช้ NRPE ไฟล์ localhost.cfg .....	91
4.26 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการไหลของหน่วยประมวลผลกลาง ของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ไฟล์ localhost.cfg .....	92

## สารบัญตาราง (ต่อ)

ตารางที่	หน้า
4.27 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการทำงานของ eth0 ของอุปกรณ์บน เครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ไฟล์ localhost.cfg .....	93
4.28 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการทำงานของหน่วยความจำหลัก ของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ไฟล์ localhost.cfg.....	93
4.29 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการทำงานของทำงานของการ ประมวลผลบริการของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ไฟล์ localhost.cfg.....	94
4.30 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบการแลกเปลี่ยนพื้นที่การทำงานของ อุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ไฟล์ localhost.cfg.....	95
4.31 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบพื้นที่ว่างภายใต้ / ของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ไฟล์ localhost.cfg.....	96
4.32 คำศัพท์ในไฟล์ localhost.cfg ที่ใช้ในการ Config.....	96
4.33 รายละเอียดการกำหนด Host ของเครื่องพิมพ์เอกสาร ในไฟล์ printer.cfg .....	99
4.34 รายชื่อเครื่องพิมพ์เอกสารที่ระบบจะทำการตรวจสอบสถานะ .....	99
4.35 รายละเอียดการกำหนดค่าของ Host Group ของเครื่องพิมพ์เอกสารในระบบ .....	100
4.36 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะของเครื่องพิมพ์เอกสาร โดยการใช้โปรโตคอล SNMP ไฟล์ printer.cfg.....	101
4.37 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการตอบสนองการ Ping ไปยัง เครื่องพิมพ์เอกสาร ไฟล์ printer.cfg.....	102
4.38 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะของพอร์ตที่ 23 ของเครื่องพิมพ์เอกสาร ไฟล์ printer.cfg.....	103
4.39 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบข้อมูล และรายละเอียด ของเครื่องพิมพ์เอกสาร โดยการใช้โปรโตคอล SNMP ไฟล์ printer.cfg.....	104

## สารบัญตาราง (ต่อ)

ตารางที่	หน้า
4.40 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการทำงานของบริการ FTP ของ เครื่องพิมพ์เอกสาร ไฟล์ printer.cfg.....	105
4.41 คำศัพท์ในไฟล์ printer.cfg ที่ใช้ในการ Config.....	106
4.42 คำศัพท์จากไฟล์ commands.cfg ใช้ในการ Config เพื่อให้ระบบเรียกคำสั่ง สำหรับการตรวจสอบการทำงานต่าง ๆ.....	114
4.43 รายละเอียดการกำหนด Host ของเครื่องพิมพ์เอกสาร ในไฟล์ windows.cfg.....	116
4.44 รายชื่อเครื่องพิมพ์เอกสารที่ระบบจะทำการตรวจสอบสถานะ.....	117
4.45 รายละเอียดการกำหนดค่าของ Host Group ในระบบ.....	117
4.46 รายละเอียดการกำหนดคำสั่งให้ระบบตรวจสอบสถานะบริการ และรุ่นของ NSClient++.....	118
4.47 รายละเอียดการกำหนดคำสั่งให้ระบบตรวจสอบเวลาในการทำงานของ Windows Server.....	118
4.48 รายละเอียดการกำหนดคำสั่งให้ระบบตรวจสอบสถานะการทำงานของ หน่วยประมวลผลกลาง.....	119
4.49 รายละเอียดการกำหนดคำสั่งให้ระบบตรวจสอบสถานะสถานะการใช้หน่วยความจำหลัก ...	120
4.50 รายละเอียดการกำหนดคำสั่งให้ระบบตรวจสอบพื้นที่ว่างใน C:\.....	120
4.51 คำศัพท์ในไฟล์ windows.cfg ที่ใช้ในการ Config.....	121
4.52 รายละเอียดการกำหนด Host ไฟล์ switch.cfg.....	122
4.53 รายชื่ออุปกรณ์ที่ระบบจะทำการตรวจสอบสถานะ.....	123
4.54 รายละเอียดการกำหนดค่าของ Host Group ในระบบ.....	123
4.55 รายละเอียดการกำหนดคำสั่งให้ระบบตรวจสอบสถานะการตอบสนองของอุปกรณ์ โดยการ Ping ไปยังอุปกรณ์.....	124
4.56 คำศัพท์ในไฟล์ switch.cfg ที่ใช้ในการ Config.....	124
4.57 ไฟล์ที่สำคัญในการ Config อุปกรณ์ Aircard.....	125

## สารบัญตาราง (ต่อ)

ตารางที่	หน้า
4.58 คำศัพท์ที่แสดงใช้ในการตั้งค่าอุปกรณ์ Aircard เข้ากับเครื่อง Nagios Server .....	126
4.59 คำศัพท์ที่แสดงใช้ในการตั้งค่าคำสั่งในการส่งข้อความไปยังผู้ดูแลระบบ .....	129
4.60 การเรียกคำสั่งในระบบเพื่อการส่งข้อความสั้นไปยังผู้ดูแลระบบ .....	130
4.61 ส่วนที่ผู้ดูแลระบบต้อง Config ไฟล์ nrpe.cfg .....	132
4.62 ส่วนที่ผู้ดูแลระบบต้อง Config ไฟล์ nrpe.....	133
4.63 ไฟล์ และรายละเอียดของไฟล์โปรแกรมภาษา Perl ที่ได้เพิ่มเข้าไปในระบบ.....	134
4.64 ไฟล์ check_snmp_int.pl สถานะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการ ที่แสดงผ่านเว็บไซต์.....	136
4.65 ไฟล์ check_snmp_int.pl สำหรับการตรวจสอบ SNMP Datas.....	136
4.66 ไฟล์ check_snmp_int.pl สถานะของอุปกรณ์ที่ตรวจสอบ .....	139
4.67 ไฟล์ check_snmp_load.pl สถานะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการ ที่แสดงผ่านเว็บไซต์.....	142
4.68 ไฟล์ check_snmp_load.pl สำหรับการตรวจสอบสถานะทั่วไปด้วย MIB ที่ฝังอยู่ใน Host.....	142
4.69 ไฟล์ check_snmp_load.pl สำหรับการตรวจสอบสถานะการทำงานของ หน่วยประมวลผลกลางของ Linux.....	143
4.70 ไฟล์ check_snmp_load.pl สำหรับการตรวจสอบสถานะการทำงานของ หน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Cisco .....	144
4.71 ไฟล์ check_snmp_load.pl สำหรับการตรวจสอบสถานะการทำงานของ หน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Cisco Catalyst .....	145
4.72 ไฟล์ check_snmp_load.pl สำหรับการตรวจสอบสถานะการทำงานของ หน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Netscreen .....	146
4.73 ไฟล์ check_snmp_load.pl สำหรับการตรวจสอบสถานะการทำงานของ หน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ AS/400.....	146

## สารบัญตาราง (ต่อ)

ตารางที่	หน้า
4.74 ไฟล์ check_snmp_load.pl สำหรับการตรวจสอบสถานะการทำงานของ หน่วยประมวลผลกลางที่ถูกใช้โดย Net-SNMP .....	147
4.75 ไฟล์ check_snmp_load.pl สำหรับการตรวจสอบสถานะการทำงานของ หน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Procurve.....	148
4.76 ไฟล์ check_snmp_load.pl สำหรับการตรวจสอบสถานะการทำงานของ หน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Nokia.....	148
4.77 ไฟล์ check_snmp_load.pl สำหรับการตรวจสอบสถานะการทำงานของ หน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Bluecoat .....	149
4.78 ไฟล์ check_snmp_load.pl สำหรับการตรวจสอบสถานะการทำงานของ หน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Fortigate.....	149
4.79 ไฟล์ check_snmp_load.pl สำหรับการตรวจสอบสถานะการทำงานของ หน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Linkproof .....	150
4.80 ไฟล์ check_snmp_load.pl สำหรับการตรวจสอบสถานะการทำงานของ หน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ HP-UX.....	150
4.81 ไฟล์ check_snmp_mem.pl สถานะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการที่แสดงผ่านเว็บไซต์.....	152
4.82 ไฟล์ check_snmp_mem.pl สำหรับการตรวจสอบหน่วยความจำหลัก Net-SNMP Memory .....	153
4.83 ไฟล์ check_snmp_mem.pl สำหรับการตรวจสอบหน่วยความจำหลัก Cisco .....	154
4.84 ไฟล์ check_snmp_mem.pl สำหรับการตรวจสอบหน่วยความจำหลัก HP Procurve.....	155
4.85 ไฟล์ check_snmp_process.pl สถานะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการที่แสดงผ่านเว็บไซต์.....	157
4.86 ไฟล์ check_snmp_process.pl สถานะการทำงานของกระบวนการบนเครื่องบริการ.....	157

## สารบัญตาราง (ต่อ)

ตารางที่	หน้า
4.87 ไฟล์ check_snmp_storage.pl สถานะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการที่แสดงผ่านเว็บไซต์.....	160
4.88 ไฟล์ check_snmp_process.pl สถานะการทำงานเกี่ยวกับหน่วยจัดเก็บข้อมูลบนเครื่อง บริการ ในการทำงานส่วน SNMP Datas.....	161
4.89 ไฟล์ check_snmp_process.pl สำหรับตรวจสอบ Storage Types.....	162



## สารบัญภาพ

ภาพที่	หน้า
2.1 โครงสร้างของระบบบริหารจัดการเครือข่าย .....	11
2.2 การอ้างอิง TCP กับ SNMP .....	14
2.3 การรับส่ง Message ระหว่าง Manager และ Agent .....	14
2.4 คอมไพเลอร์ภาษา C .....	17
2.5 อินเตอร์พรีเตอร์ (Interpreter) .....	17
2.6 การทำงานของ NRPE .....	18
2.7 ผลใบรับรองอิเล็กทรอนิกส์ .....	20
2.8 โครงสร้างของ OID แต่ละโหนด .....	21
3.1 ขั้นตอนการทำงานของระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่ายและเครื่องบริการ .....	30
3.2 ขั้นตอนการทำงานของ Agent ในการตรวจสอบสถานะการให้บริการ อุปกรณ์เครือข่าย และเครื่องบริการ .....	31
3.3 โครงสร้างกระบวนการทำงานของระบบตรวจสอบสถานะการให้บริการของ อุปกรณ์เครือข่าย และเครื่องบริการ .....	32
3.4 การ Remote เพื่อติดตั้ง Package บนเครื่อง Nagios Server .....	33
3.5 การตรวจสอบสถานะบน Localhost ซึ่งเป็น Nagios Server โดยใช้การ Monitoring .....	34
3.6 การตรวจสอบสถานะบน printer.1 ซึ่งเป็นอุปกรณ์เครือข่าย โดยใช้ โพรโตคอล SNMP .....	35
3.7 การตรวจสอบสถานะบน printer.4 ซึ่งเป็นอุปกรณ์เครือข่าย โดยใช้ โพรโตคอล SNMP .....	35
3.8 การตรวจสอบสถานะบน AP-8-401[.2] โดยใช้การ Monitoring ไปเก็บข้อมูล .....	36
3.9 การตรวจสอบสถานะบน AP-8-406[.3] โดยใช้การ Monitoring ไปเก็บข้อมูล .....	36
3.10 การตรวจสอบสถานะบน server8407 โดยใช้การ Monitoring ไปเก็บข้อมูล .....	37
3.11 การตรวจสอบสถานะบน gateway-8402 โดยใช้การ Monitoring ไปเก็บข้อมูล .....	37
3.12 การตรวจสอบสถานะบน VMServer.15 โดยใช้การ Monitoring ไปเก็บข้อมูล .....	38
3.13 การตรวจสอบสถานะบน NASserver.16 โดยใช้การ Monitoring ไปเก็บข้อมูล .....	38
3.14 การตรวจสอบสถานะบน VMServer.18 โดยใช้การ Monitoring ไปเก็บข้อมูล .....	39

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.15 การตรวจสอบสถานะบน WinServer.19 โดยใช้การ Monitoring ไปเก็บข้อมูล .....	39
3.16 การตรวจสอบสถานะบน webcomsci โดยใช้การ Monitoring ไปเก็บ .....	40
3.17 การตรวจสอบสถานะของ cacti โดยใช้โปรโตคอล SNMP .....	40
3.18 การแก้ไขเครื่องบริการที่มีปัญหา .....	41
3.19 การเก็บข้อมูลเป็น Text File .....	41
3.20 โครงสร้างหน้าจอของระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย.....	43
4.1 โครงสร้างการดำเนินงานของผู้ดูแลระบบ.....	46
4.2 โครงสร้างการเข้าใช้งานระบบของผู้ดูแลระบบ .....	48
4.3 การเข้าใช้ระบบของผู้ดูแลระบบ .....	49
4.4 ป้อนรหัสผ่านเพื่อเข้าใช้งานระบบของผู้ดูแลระบบ .....	50
4.5 หน้าการทำงานของโปรแกรม Secure Shell.....	50
4.6 หน้าจอสำหรับการ Login เพื่อตรวจสอบสถานะผ่านเว็บไซต์.....	167
4.7 หน้าเว็บหลักเมื่อกรอกชื่อผู้ใช้ และรหัสผ่านถูกต้อง .....	167
4.8 เมนู Current Status ในเมนูย่อย Service แสดงการตรวจสอบสถานะของอุปกรณ์.....	168
4.9 เมนู Current Status ในเมนูย่อย Tactical Overview .....	168
4.10 เมนู Current Status ในเมนูย่อย Map .....	169
4.11 เมนู Current Status ในเมนูย่อย Hosts .....	169
4.12 เมนู Current Status ในเมนูย่อย Host Groups.....	170
4.13 เมนู Current Status ในเมนูย่อย Problems.....	170
4.14 หน้าต่างสำหรับเลือกประเภทของข้อมูลที่ต้องการตรวจสอบย้อนหลัง.....	171
4.15 หน้าต่างสำหรับเลือกอุปกรณ์ที่ต้องการตรวจสอบย้อนหลัง .....	171
4.16 หน้าต่างสำหรับเลือกข้อมูลที่ต้องการให้แสดงการตรวจสอบย้อนหลัง .....	172
4.17 หน้าต่างการแสดงผลข้อมูลการตรวจสอบย้อนหลัง .....	172
4.18 หน้าต่างการแสดงผลข้อมูลการตรวจสอบย้อนหลัง (ต่อ).....	173



## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.19 กราฟสถานะการให้บริการของ gateway-8402 โดยการ Ping การทำงานของอุปกรณ์.....	173
4.20 กราฟสถานะการให้บริการของ gateway-8402 เมื่อหยุดให้บริการ .....	174
4.21 กราฟสถานะการให้บริการของ webcomsci โดยการตรวจสอบ HTTP .....	174
4.22 กราฟขนาดของสถานะการให้บริการของ webcomsci โดยการตรวจสอบ HTTP .....	174
4.23 ข้อความสั้นๆที่ได้รับการแจ้งเตือนจาก Nagios Server .....	175
ก.1 เขียนไฟล์ Image ของระบบปฏิบัติการลงใน SD Card ด้วย Win32DiskImage.....	181
ก.2 หน้าจอแสดงการ Boot ระบบปฏิบัติการ .....	182
ก.3 หน้าจอแสดงการตั้งค่าเริ่มต้นของอุปกรณ์ Raspberry Pi .....	182
ก.4 Expand Filesystem การสั่งให้ระบบทำการจัดการ Partition บน SD Card.....	183
ก.5 เมื่อระบบทำการ Expand Filesystem เสร็จ .....	183
ก.6 หัวข้อ Change User Password.....	184
ก.7 ยืนยันความต้องการในการเปลี่ยน Password.....	184
ก.8 ยืนยัน Password ใหม่อีกครั้ง.....	185
ก.9 เมื่อตั้ง Password เรียบร้อยแล้ว .....	185
ก.10 หัวข้อ Enable Boot to Desktop/Scratch .....	186
ก.11 ตัวเลือกรูปแบบการ Boot การทำงาน .....	186
ก.12 หัวข้อ Internationalisation Options .....	187
ก.13 Change Locale .....	187
ก.14 เลือกรูปแบบภาษาเป็น en_US.UTF-8.....	188
ก.15 ในส่วน Geographic area เลือก Asia.....	188
ก.16 ในส่วน Time zone เลือก Bangkok.....	189
ก.17 ในส่วน Keyboard model เลือก Generic 105-key (Intl) PC.....	189
ก.18 ในส่วน Country of origin for the keyboard เลือก English (US).....	190
ก.19 ในส่วน Keyboard layout เลือกเป็น English (US).....	190

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ก.20 หัวข้อ Overclock.....	191
ก.21 ตัวเลือกในการปรับแต่งการใช้งานทรัพยากรของระบบ .....	191
ก.22 ตัวเลือกในหัวข้อ Advanced Options.....	192
ก.23 การตั้งชื่อให้กับเครื่อง Raspberry Pi.....	192
ก.24 การแบ่งปันหน่วยความจำหลักไปให้กับ GPU.....	193
ก.25 เปิดการทำงานของ SSH .....	193
ก.26 เปิดการทำงานของ Device Tree.....	194
ก.27 ตั้งค่าเสร็จแล้วให้เลือก Finish.....	194
ก.28 Reboot หลังจากตั้งค่าระบบเสร็จแล้ว .....	195
ก.29 ไฟล์ติดตั้งของโปรแกรม SSH Secure Shell Client .....	195
ก.30 หน้าต่างต้อนรับของโปรแกรม SSH Secure Shell Client.....	196
ก.31 หน้าต่างเงื่อนไขการใช้งานโปรแกรม (License).....	196
ก.32 หน้าต่างเลือก Folder สำหรับการติดตั้งโปรแกรม .....	197
ก.33 หน้าต่าง Select Program Folder .....	197
ก.34 หน้าต่าง Select Components เป็นการเลือกส่วนของโปรแกรมที่ต้องการติดตั้ง .....	198
ก.35 หน้าต่าง Check Setup Information.....	198
ก.36 หน้าต่าง Setup Status.....	199
ก.37 หน้าต่าง Setup Complete.....	199

# บทที่ 1

## บทนำ

### หลักการและเหตุผล

ในปัจจุบันเป็นยุคที่เครือข่ายอินเทอร์เน็ตเข้ามามีบทบาท และมีความสำคัญต่อการใช้ชีวิตเป็นอย่างมาก หลายอย่างในชีวิตประจำวันล้วนแล้วต้องอาศัยเครือข่ายอินเทอร์เน็ตในการเข้าถึง ดังนั้นเครือข่ายบริการจึงต้องมีความพร้อมอยู่เสมอที่จะรองรับความต้องการในการเข้าถึงของผู้ใช้ เพื่อให้ระบบเครือข่ายมีประสิทธิภาพ และมีเสถียรภาพในการบริการ ผู้ที่ทำหน้าที่ดูแลระบบจะต้องบริหารจัดการ ดูแล ควบคุม และตรวจสอบการทำงานของอุปกรณ์เครือข่ายให้อยู่ในสถานะที่ปลอดภัย และพร้อมใช้งานตลอดเวลา เพื่อให้สอดคล้องกับแนวคิดด้านการรักษาความปลอดภัย และความมั่นคงของสารสนเทศจากภัยคุกคาม โดยประกอบไปด้วย การบริหารจัดการรักษาข้อมูลไว้เป็นความลับ (Confidentiality) การคงไว้ซึ่งความถูกต้องสมบูรณ์และครบถ้วนของข้อมูล (Integrity) และการอยู่ในสถานะพร้อมใช้เมื่อต้องการ (Availability) ซึ่งหากเครือข่ายหยุดให้บริการ โดยเกิดจากปัญหาการขัดข้อง หรือโดยการถูกโจมตี ทำให้อุปกรณ์เครือข่ายไม่สามารถใช้งานได้ ผู้ดูแลระบบอาจไม่สามารถได้รับการแจ้งเตือน และทราบถึงสถานะการหยุดการทำงานของอุปกรณ์เครือข่ายได้ทันที จึงอาจทำให้ระบบเครือข่ายเกิดความเสียหายมากยิ่งขึ้น

จากปัญหาดังกล่าว ผู้พัฒนาจึงได้สร้างระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย (Network Devices Status Monitoring System) ระบบนี้จะแจ้งเตือนไปยังผู้ดูแลระบบเมื่ออุปกรณ์เครือข่ายหยุดให้บริการผ่านโทรศัพท์เคลื่อนที่ โดยจะทำการแจ้งเตือนในรูปแบบของการส่งข้อความสั้น (Short Message Service : SMS) ไปยังผู้ดูแลระบบ ผู้ดูแลระบบสามารถดูข้อมูลสถานะพร้อมรายละเอียดของอุปกรณ์ที่ขัดข้องผ่านเว็บไซต์ และสามารถควบคุมการทำงานของอุปกรณ์เครือข่ายผ่านระบบไร้สายเดียวกันโดยที่ผู้ดูแลไม่จำเป็นต้องอยู่ในห้องเครื่องบริการ ผู้ดูแลระบบสามารถตรวจสอบการหยุดให้บริการย้อนหลัง และสามารถดูรายงานข้อมูลสถานะการตรวจสอบการให้บริการในรูปแบบเอกสาร ตาราง และกราฟได้ เพื่อเป็นประโยชน์ในการเปรียบเทียบ และปรับปรุงระบบเครือข่ายให้ดียิ่งขึ้นต่อไป

## เป้าหมายของโครงการ

พัฒนาต้นแบบในการตรวจสอบสถานะการหยุดให้บริการของอุปกรณ์และเครือข่าย โดยการส่งข้อความสั้นพร้อมทั้งสามารถตรวจสอบสถานะของระบบผ่านเว็บไซต์เพื่อแจ้งเตือนการหยุดทำงานไปยังผู้ดูแลระบบ

## วัตถุประสงค์ของโครงการ

เพื่อพัฒนาระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย โปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา ด้วย Nagios บน Raspberry Pi

## ขอบเขตและความสามารถของระบบ

### 1. ผู้ดูแลระบบ

- 1.1 สามารถตรวจสอบสถานะการทำงานของอุปกรณ์เครือข่ายที่เกิดสถานะการหยุดให้บริการผ่านเครือข่ายอินเทอร์เน็ต
- 1.2 สามารถดูสถานะการให้บริการจากระยะไกลได้
- 1.3 สามารถปรับแต่ง กำหนดค่าให้เหมาะสมกับระบบเครือข่ายนั้นๆ ได้
  - 1.3.1 สามารถปรับแต่งผ่านหน้าเว็บ
  - 1.3.2 เวลาในการ Pooling
  - 1.3.3 ข้อความที่จะทำการส่งไปยังผู้ดูแลระบบ
- 1.4 ระบบสามารถทำการแจ้งเตือนไปยังผู้ดูแลระบบ เมื่ออุปกรณ์บนเครือข่ายหยุดให้บริการ โดยผ่านโทรศัพท์เคลื่อนที่ทุกเครือข่ายที่รองรับระบบ GSM
  - 1.4.1 ระบบสามารถส่งข้อความสั้นไปยังผู้ดูแลระบบ โดยรายละเอียดของข้อความประกอบด้วย
    - 1.4.1.1 ชื่ออุปกรณ์
    - 1.4.1.2 ชื่อบริการ ( Services )
    - 1.4.1.3 เวลาและวันที่ที่หยุดให้บริการ
    - 1.4.1.4 สถานะการหยุดให้บริการ และสถานะเริ่มกลับให้บริการของระบบ

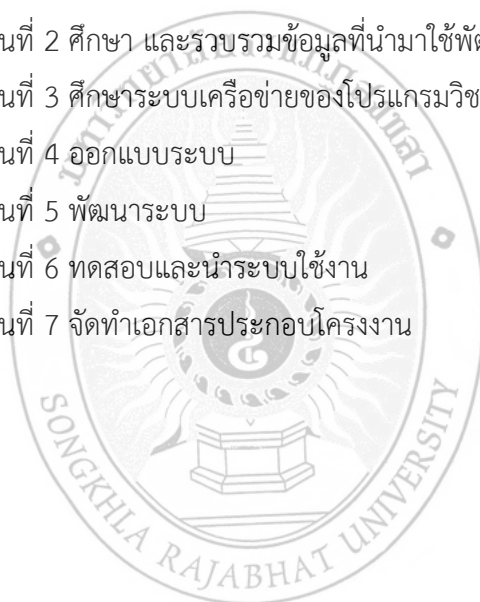
1.5 แสดงข้อมูลการใช้ทรัพยากรต่าง ๆ ของเครื่องบริการ Processor Load , Disk Usage , Memory ในรูปแบบของเอกสาร ตาราง และกราฟ ทั้งปัจจุบัน และย้อนหลัง ผ่านเว็บไซต์

1.6 สามารถตรวจสอบสถานะการให้บริการของอุปกรณ์เครือข่ายได้จากระบบผ่าน Web Page

## แผนการดำเนินงาน

### 1. ขั้นตอนการดำเนินงาน

- ขั้นตอนที่ 1 เสนอเค้าโครงโครงการ
- ขั้นตอนที่ 2 ศึกษา และรวบรวมข้อมูลเพื่อนำมาใช้พัฒนา
- ขั้นตอนที่ 3 ศึกษาและออกแบบของโปรแกรมวิชาคอมพิวเตอร์
- ขั้นตอนที่ 4 ออกแบบระบบ
- ขั้นตอนที่ 5 พัฒนาระบบ
- ขั้นตอนที่ 6 ทดสอบและนำระบบใช้งาน
- ขั้นตอนที่ 7 จัดทำเอกสารประกอบโครงการ



## 2. ระยะเวลาการดำเนินงาน ดังแสดงในตาราง

ตารางที่ 1.1 ระยะเวลาในการดำเนินงาน

แผนการดำเนินงาน	สิงหาคม 2559 - ธันวาคม 2559																			
	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน				ธันวาคม			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. เสนอเค้าโครง โครงการ																				
2. ศึกษา และรวบรวม ข้อมูลที่น่าสนใจ																				
3. ศึกษาระบบ เครือข่ายของโปรแกรม วิชาคอมพิวเตอร์																				
4. ออกแบบระบบ																				
5. พัฒนาระบบ																				
6. ทดสอบและนำ ระบบใช้งาน																				
7. จัดทำเอกสาร ประกอบโครงการ																				

### สถานที่ดำเนินโครงการ

ห้องปฏิบัติการคอมพิวเตอร์ 8-408 โปรแกรมวิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสงขลา

## เครื่องมือและอุปกรณ์ที่ใช้ในการพัฒนา

### 1. ฮาร์ดแวร์

1.1 Raspberry Pi 2 Model B เป็นบอร์ดคอมพิวเตอร์ขนาดเล็กใช้เป็นเครื่องบริการ โดยติดตั้งซอฟต์แวร์หลัก ๆ คือ Raspbian V. Wheezy, Nagios, SNMP Protocol, SMSTools3 โดยบอร์ดคอมพิวเตอร์มีรายละเอียดของอุปกรณ์ดังนี้

- หน่วยประมวลผลกลาง (CPU): Quad Cortex A7@900MHz SoC BCM2836
- หน่วยประมวลผลภาพ (GPU): 250MHz VideoCore IV
- หน่วยความจำหลัก (RAM): SDRAM 1 GB
- หน่วยความจำสำรอง: Micro SD 16 GB Class 10

1.2 อุปกรณ์สื่อสารที่รองรับ GSM ใช้สำหรับเป็นสื่อกลางในการส่งข้อความสั้น SMS จากเครื่องบริการไปยังโทรศัพท์เคลื่อนที่ของผู้ดูแลระบบ

### 2. ด้านซอฟต์แวร์

1.3 Operating System : Raspbian Ver.Wheezy ใช้เป็นระบบปฏิบัติการสำหรับ Raspberry Pi

1.4 Nagios เป็นโปรแกรมแบบโอเพนซอร์สที่ใช้ในการเฝ้าระวังระบบเครือข่ายในองค์กร (Network Monitoring)

1.5 Secure Shell ซอฟต์แวร์สำหรับควบคุมระยะไกลไปยังเครื่องคอมพิวเตอร์ที่เป็นเครื่องบริการ

1.6 SNMP ( Simple Network Management Protocol ) เป็นโพรโตคอลที่ใช้ควบคุมตรวจสอบสถานะของเครือข่าย มีระบบแจ้งเตือนเมื่อมีส่วนหนึ่งส่วนใดของเครือข่ายทำงานผิดพลาด

1.7 SMSTools3 เป็นซอฟต์แวร์แบบโอเพนซอร์สที่สำหรับการส่ง SMS

## ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ระบบที่สามารถแจ้งเตือนไปยังผู้ดูแลระบบ เมื่ออุปกรณ์เครือข่ายโปรแกรมวิชา คอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา หยุดให้บริการ
2. ผู้ดูแลระบบสามารถบริหารจัดการ ควบคุม และแก้ไขปัญหาได้อย่างทันท่วงที
3. สามารถนำระบบมาประยุกต์ใช้กับอุปกรณ์เครือข่าย หรือระบบอื่น
4. ผู้ดูแลระบบสามารถรวบรวมข้อมูลมาปรับปรุงระบบเครือข่ายให้ดียิ่งขึ้นต่อไป

## อาจารย์ที่ปรึกษาโครงการ

อาจารย์ภาณุกร ภูริปัญญานันท์

อาจารย์โชติธรรม ธารรักษา

โปรแกรมวิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสงขลา

## อาจารย์ประจำวิชา

ผู้ช่วยศาสตราจารย์ ดร.ทวิรัตน์ นวลช่วย

โปรแกรมวิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสงขลา



## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การจัดทำโครงการในครั้งนี้ มีวัตถุประสงค์เพื่อพัฒนาระบบตรวจสอบสถานะการหยุดให้บริการของอุปกรณ์และเครือข่าย โดยการส่งข้อความสั้นพร้อมทั้งสามารถตรวจสอบสถานะของระบบผ่านเว็บไซต์เพื่อแจ้งเตือนการหยุดทำงานไปยังผู้ดูแลระบบ ดังนั้นเพื่อให้เกิดองค์ความรู้ในการทำระบบดังกล่าว คณะผู้จัดทำจึงได้ศึกษาทฤษฎี และงานวิจัยที่เกี่ยวข้องโดยครอบคลุมหัวข้อต่าง ๆ ดังนี้

#### ความรู้พื้นฐานในการทำระบบ

ในการบริหารจัดการระบบเครือข่ายทุกกระบวนการต้องอยู่ภายใต้หลักที่เรียกว่า FCAPS คือ หลักสำคัญในการเลือกใช้งานระบบ NMS (Network Management System) ที่เหมาะสมกับองค์กร FCAPS (เอฟแคปส์) เป็นชื่อย่อของหลักการบริหารและจัดการเครือข่าย 5 ตัวคือ Fault (ความผิดพลาด), Configuration (การตั้งค่าอุปกรณ์), Accounting (การจดบันทึกการใช้งาน), Performance (ประสิทธิภาพการทำงาน) และ Security (ความปลอดภัย). ซึ่งรายละเอียดของ FCAPS สามารถอธิบายพอสังเขปได้ ดังนี้

##### Fault Management

- การตรวจสอบสถานะของระบบคอมพิวเตอร์ และ เครือข่าย
- การตรวจสอบปัญหาและการแจ้งปัญหา
- การแก้ไขปัญหา
- การบันทึกปัญหา และวิธีการแก้ไข

##### Performance Management

- การดำเนินการเพื่อให้ระบบเครือข่ายสามารถใช้อย่างเต็มประสิทธิภาพ
- Response Time

### Configuration Management

- สามารถกำหนดค่า (Config) ต่าง ๆ ให้สามารถตรวจจับการทำงานของระบบ Server หรือระบบ Network รวมไปถึงควบคุมอุปกรณ์ต่าง ๆ ที่มีพื้นฐานการทำงานบนโปรโตคอล TCP/IP
- สามารถแสดงลักษณะการเชื่อมต่อ และความสัมพันธ์ต่าง ๆ ของระบบ
- สามารถทราบถึงข้อมูลต่าง ๆ ของอุปกรณ์ได้

### Accounting Management

- สามารถตรวจสอบกิจกรรมต่าง ๆ ได้ ว่าใครใช้บริการอะไร มากน้อยแค่ไหน เพื่อนำมาวิเคราะห์กำหนดแผนการต่าง ๆ ได้

### Security Management

- มีการเข้ารหัสข้อมูลในการตรวจสอบ (Monitoring)
- มีการกำหนดสิทธิในการเข้าใช้งาน

## 1. Nagios

Nagios ได้รับการออกแบบโดย Rock Solid Framework เพื่อใช้ในการ Monitor , scheduling และ alerting ในระบบเครือข่าย และมีความสามารถที่จะเพิ่มศักยภาพในการทำงานอีกได้ตามที่ผู้ใช้ต้องการ ระบบนี้สามารถใช้งานที่จะนำข้อมูลเหล่านี้ไปทำการตั้งค่า (Config) ระบบต่อไป โปรแกรมนี้เหมาะสำหรับผู้ดูแลระบบ (Administrator) ทั่วไปที่ต้องการใช้ระบบ Monitoring Network System ในส่วนของระบบ (System) และ บริการ (Service) ต่าง ๆ ที่ผู้ดูแลระบบต้องการ และที่สำคัญโปรแกรมนี้เป็น free-ware และยังสามารถทำการพัฒนาระบบให้เหมาะสมกับองค์กรได้

Nagios นั้นถูกออกแบบมาให้ทำงานภายใต้ระบบปฏิบัติการ Linux ในส่วนของการพัฒนานั้นสามารถทำการสร้างให้ระบบแสดงผลตามที่ต้องการ หรือการพัฒนา Plug-in ต่างๆโดยใช้ ภาษา C, Perl หรือ shell scripts ได้

Nagios คือ ซอฟต์แวร์ประยุกต์ (Application) ที่ใช้ในการตรวจสอบระบบผ่าน Web - Application เพื่อใช้ในการดูทำงานของ Host และ Service ที่เราต้องการ เช่น Disk space , Ram , CPU และ Application เมื่อเกิดปัญหานั้นจะมีการส่ง (Alert) มายังผู้ดูแลระบบ (Administrator) เพื่อทำการตรวจสอบ เพื่อใช้ในการบริหารในส่วนของ

### ความสามารถของ Nagios

1. ตรวจสอบสถานะการทำงานของ Service ต่าง ๆ ได้ เช่น SMTP, POP3, HTTP, NNTP, Ping และอื่น ๆ
2. สามารถตรวจสอบการใช้ทรัพยากรของ Host ได้ เช่น สถานะการทำงานของ Processor, Disk, Memory Usage และอื่น ๆ
3. สามารถออกแบบ Plug in เพื่อนำมาใช้ในการตรวจจับข้อมูลของ Service ที่ต้องการ
4. สามารถแสดงฝั่งของ Server หรือ ระบบเครือข่ายได้ เพื่อสามารถวิเคราะห์ระบบได้ โดยจะมีการตรวจสอบแบบ Parent
5. สามารถกำหนดเหตุการณ์ควบคุมเมื่อเกิดปัญหาได้
6. มีการเก็บข้อมูลเพื่อนำไปวิเคราะห์ได้
7. สามารถบูรณาการ (Integrate) ใช้งานกับโปรแกรมอื่น ๆ ได้ เช่น Snort, Syslog-NG และอื่น ๆ

## 2. SSH

SSH (Secure Shell) คือโพรโทคอลสำหรับเครือข่ายคอมพิวเตอร์ (RFC 4254) ที่ออกแบบมาเพื่อให้เข้าถึงไปยังคอมพิวเตอร์เครื่องอื่น ๆ และทำงานต่าง ๆ บนเครื่องนั้น SSH ถูกออกแบบมาเพื่อใช้ทดแทน Rlogin, TELNET และ RSH เนื่องจากเหตุผลด้านความปลอดภัย เพราะ SSH ใช้วิธีการเข้ารหัสข้อมูลที่ส่งผ่านระหว่างเครื่อง พอร์ตมาตรฐานในการเชื่อมต่อของ SSH คือ พอร์ต 22 นอกจากนี้ SSH ยังเป็นพื้นฐานของการสื่อสารอื่นๆ เช่น SFTP หรือ SCP

โปรแกรม SSH (Secure Shell) มีการเข้ารหัสข้อมูลก่อนส่ง ทำให้ผู้ที่ลักลอบไม่สามารถมองเห็นข้อมูลที่แท้จริง ปัจจุบันระบบปฏิบัติการ Unix หรือ Linux มีบริการ SSH มาให้ผู้ใช้เสมอ แต่เครื่องของผู้ใช้ที่ต้องการติดต่อเครื่องบริการจำเป็นต้องมีโปรแกรม SSH Client ติดตั้งไว้ก่อน

## 3. SNMP

SNMP ย่อมาจาก Simple Network Management Protocol ซึ่งเป็นโพรโตคอลที่อยู่ระดับบนในชั้นการประยุกต์ และเป็นส่วนหนึ่งของชุดโพรโตคอล TCP/IP ในการบริการและจัดการเครือข่ายต้องใช้อุปกรณ์ต่าง ๆ มีส่วนของการทำงานร่วมกับระบบจัดการเครือข่าย ซึ่งเราเรียกว่า เอเจนต์ (Agent) เอเจนต์เป็นส่วนของซอฟต์แวร์ที่อยู่ในอุปกรณ์ต่าง ๆ ที่เชื่อมต่ออยู่ในเครือข่ายโดยมีคอมพิวเตอร์หลักในระบบหนึ่งเครื่องเป็นตัวจัดการและบริหาร เครือข่ายหรือเรียกว่า NMS-Network Management System

โพรโตคอล SNMP มีการพัฒนามาอย่างต่อเนื่องตั้งแต่ SNMPv1 จนถึงปัจจุบัน คือ SNMPv3 โดยในเวอร์ชัน 1 และ 2 นั้นมีลักษณะของสถาปัตยกรรมและการทำงานที่คล้ายคลึงกัน ซึ่งในเวอร์ชัน 2 ได้พัฒนาเพื่อยกระดับความสามารถ และประสิทธิภาพของการทำงานจากเวอร์ชัน 1 เช่น เพิ่มคำสั่งสำหรับใช้ในการจัดการเครือข่าย เพิ่มกลุ่มของอ็อบเจกต์ภายในฐานข้อมูล MIB เป็นต้น แต่วัตถุประสงค์หลักหนึ่งในการพัฒนา SNMPv2 ที่ยังไม่ประสบความสำเร็จ คือ การยกระดับในด้านความปลอดภัย ต่อมาจึงได้มีการพัฒนาเป็น SNMPv3 ที่ได้มีการแก้ไขปัญหาด้านความปลอดภัยของ SNMP ทั้งสองเวอร์ชันก่อนหน้านี้ โดยสรุป SNMP ทั้งสามเวอร์ชันได้ดังนี้

1) SNMP Version 1 (SNMPv1) เป็นมาตรฐานปัจจุบันและเป็นที่ยอมรับเพราะความง่ายของโพรโตคอล SNMP ซึ่งถูกระบุใน RFC1157 ส่วน MIB-1 ประกาศใน RFC 1156 และ MIB-2 ถูกประกาศใน RFC 1213 และได้รับอนุมัติให้เป็นมาตรฐานที่สมบูรณ์ ระดับความปลอดภัย SNMPv1 จะขึ้นอยู่กับคอมมูนิตีส์ตริง (Community String) ที่ทำหน้าที่เหมือนรหัสผ่าน หรือพาสเวิร์ด (Password) โดยที่จริงแล้วเป็นเพียงข้อความแบบธรรมดา (Plain Text) ที่บ่งบอกถึงสิทธิการเข้าไปจัดการอุปกรณ์เครือข่าย โดยปกติคอมมูนิตีส์จะมีสามประเภทนั้นคือ อ่านอย่างเดียว (Read-only), อ่านเขียน (Read-write) และแตรป (Trap)

2) SNMP Version 2 (SNMPv2) คือ เวอร์ชันที่ยกระดับความสามารถ และประสิทธิภาพการทำงานจากเวอร์ชัน 1 โดยเพิ่มคำสั่งสำหรับใช้ในการจัดการเครือข่าย เพิ่มกลุ่มของ Object ภายในฐานข้อมูล MIB ซึ่ง SNMPv2 นี้ถูกประกาศใน RFC 1902

3) SNMP Version 3 (SNMPv3) เป็นเวอร์ชันที่ถูกพัฒนาต่อมาจากเวอร์ชัน 2 โดยมีการแก้ปัญหาเรื่องความปลอดภัยของ SNMP (RFC 3411 – RFC 3418) ซึ่งจะมีคุณสมบัติการทำงานที่สำคัญ ดังนี้

3.1) Message Integrity เพื่อให้แน่ใจว่า Packet ที่ส่งนั้นจะไม่ถูกเปลี่ยนแปลง หรือทำลาย

3.2) Authentication เป็นการตรวจสอบว่าข้อความนั้นมาจากแหล่งที่ต้องการ

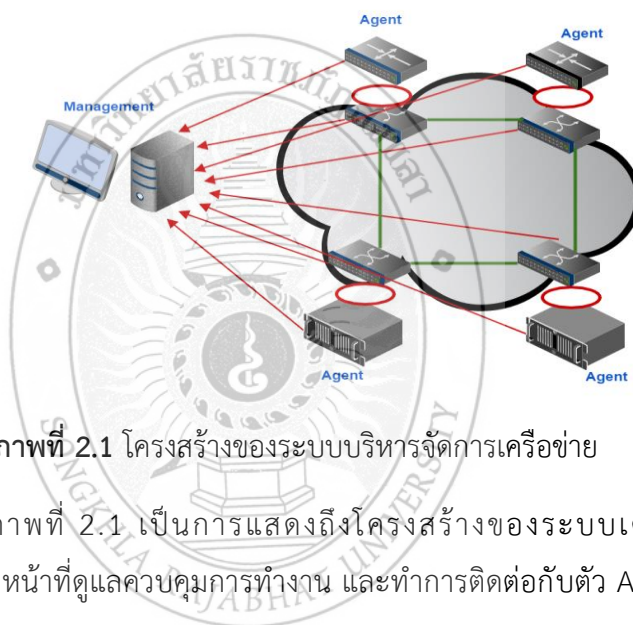
3.3) Encryption ทำการเข้ารหัสของ Packet เพื่อป้องกันการถูกสอดแนมโดยผู้ที่ไม่ได้รับอนุญาต

### 1.1 ระบบดูแลและบริหารเครือข่าย (Network Management System : NMS)

เมื่อเริ่มใช้พีซี ผู้ใช้ต้องรู้จักดูแลรักษาพีซีให้ใช้งานได้ดีตลอดไป การดูแลพีซีเกี่ยวข้องกับทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ ในบางครั้งเราไม่สามารถจัดการแก้ไขปัญหาที่เกิดขึ้นกับพีซีได้ จึงต้องมีการตามช่างหรือบริษัทผู้ขายมาช่วยบริการให้ใช้งานได้ เช่นกันการนำพีซีมาต่อรวมเป็นระบบ LAN มีการจัดตั้งเครื่องบริการเซิร์ฟเวอร์ การดูแลและบริหารเครือข่ายแลนก็เริ่มมีความยุ่งยากขึ้น เพราะมีอุปกรณ์ประกอบหลายชิ้น อุปกรณ์เหล่านี้ต้องทำงานร่วมกันเป็นระบบหากชิ้นหนึ่งชิ้นใด

มีปัญหา การทำงานทั้งระบบก็มีปัญหา การใช้งานในองค์กรมีความซับซ้อนยิ่งขึ้น มีการเชื่อมต่อแลนหลายๆ เครือข่ายให้เป็นเครือข่ายใหญ่ขึ้น หรือที่เรียกว่า อินทราเน็ต มีอุปกรณ์พิเศษจำพวกสวิตชิง (Switching) และ เราเตอร์ (Router) มีการเชื่อมโยงผ่านเครือข่ายบริการต่าง ๆ ในรูปแบบ WAN เพื่อเชื่อมต่อหรือสื่อสารกับหน่วยงานอื่นที่อยู่ห่างไกลความซับซ้อนของระบบจึงมีมากขึ้น

ดังนั้นจึงมีความจำเป็นที่จะต้องหาวิธีการดูแลและบริหารเครือข่ายให้มีประสิทธิภาพ เชื่อถือได้ สามารถใช้งานได้ทนทาน เทคโนโลยีเครือข่ายจึงต้องมีระบบการจัดการเครือข่าย มีการสร้างโปรโตคอลสำหรับรองรับการทำงานในส่วนเหล่านี้



ภาพที่ 2.1 โครงสร้างของระบบบริหารจัดการเครือข่าย

จากภาพที่ 2.1 เป็นการแสดงถึงโครงสร้างของระบบเครือข่ายโดยมีตัว Management คอยทำหน้าที่ดูแลควบคุมการทำงาน และทำการติดต่อกับตัว Agent ต่าง ๆ ที่อยู่ในระบบเครือข่าย

## 1.2 ความสำคัญของ NMS

NMS ทำหน้าที่เป็นศูนย์กลางการควบคุม และเฝ้ามองเครือข่ายมีระบบเตือนเมื่อมีส่วนหนึ่งส่วนใดของเครือข่ายทำงานผิดพลาด หรือเกิดข้อขัดข้อง ทำให้ผู้ดูแลระบบทราบได้ทันที และเข้าไปทำการแก้ไขได้รวดเร็ว หน้าที่หลักของ NMS คือการตรวจสอบเครือข่ายตลอดเวลา ทำรายงานสถิติการใช้เครือข่าย เช่น สถิติของปริมาณข้อมูล ปริมาณผู้ใช้ สามารถเขียนเป็นกราฟเพื่อให้ผู้ดูแลระบบนำไปวิเคราะห์และวางแผนขยายเครือข่าย ผู้ดูแลระบบยังสามารถตรวจสอบและแก้ไขระบบจากจุดศูนย์กลาง รวมถึงการติดตั้งซอฟต์แวร์ การตั้งค่าระบบให้กับอุปกรณ์เครือข่ายที่อยู่ห่างไกล NMS จึงเป็นอุปกรณ์ที่ระบบเครือข่ายขนาดใหญ่ หรือผู้ให้บริการเครือข่ายแบบสาธารณะที่มีผู้ใช้บริการจำนวนมากจำเป็นต้องมี เพราะเป็นอุปกรณ์ที่ทำให้การเฝ้ามองระบบเป็นไปอย่างมี

ประสิทธิภาพ ปัจจุบันแม้แต่เครื่องอินทราเน็ตมีอุปกรณ์ต่าง ๆ ประกอบรวมกันมีความซับซ้อนมากขึ้น NMS จึงมีส่วนสำคัญในการบริหารและจัดการเครือข่ายอินทราเน็ต

การที่ระบบบริหารและจัดการเครือข่ายจะประสบผลสำเร็จ จึงขึ้นกับระบบซอฟต์แวร์ที่ต้องมีอยู่ในตัวอุปกรณ์เครือข่าย (เอเจนต์) ส่วนของเอเจนต์ยังมีการเก็บข้อมูลไว้ภายในข้อมูลที่เก็บไว้เรียกว่า MIB – Management Information Base การทำงานของอุปกรณ์ต่าง ๆ บนเครือข่ายจะมีข้อมูลของตัวเองเก็บไว้ที่ MIB ดังนั้น NMS จึงส่งคำถามไปยังเอเจนต์ การส่งคำถามและเอเจนต์ส่งข้อมูลคำตอบนี้ย่อมเป็นไปตามมาตรฐานโพรโตคอลที่กำหนด เช่น ลักษณะคำถามคำตอบของ SNMP ที่สอบถามกันเป็นระบบ และเป็นมาตรฐานสากล ข้อมูลในฐานข้อมูลที่เก็บในเอเจนต์ของแต่ละอุปกรณ์ประกอบด้วย ข้อมูลชื่ออุปกรณ์ รหัสอุปกรณ์ หมายเลขแอดเดรสบนเครือข่าย ตารางกำหนดเส้นทางปริมาณข้อมูลที่รับส่ง ข้อผิดพลาดที่ปรากฏ ฯลฯ ดังนั้นระบบ NMS จึงได้ข้อมูลของทุกอุปกรณ์ที่มีเอเจนต์อยู่ และนำข้อมูลเหล่านั้นมาแสดงผลในเชิงวิเคราะห์ต่าง ๆ ไดอะแกรมรูปภาพของเครือข่ายทางฟิสิกส์ การนำข้อมูลมาแสดงผลนี้ NMS ส่งคำถามไปเป็นระยะและรับคำตอบมาปรับปรุงข้อมูล หากส่งคำถามไปยังตัวอุปกรณ์ที่มีในระบบและไม่ได้รับคำตอบก็จะมีวิธีการตรวจสอบอย่างอื่นประกอบ เช่น อุปกรณ์นั้นมีปัญหาอย่างไรหรือไม่หากพบปัญหาก็คงจะแสดงปัญหาเพื่อให้ผู้ดูแลระบบทราบ ระบบบริหารและจัดการเครือข่ายจึงเป็นซอฟต์แวร์ที่นำข้อมูลจากเอเจนต์ต่าง ๆ มาแสดงผล และติดต่อกับผู้ดูแลระบบ ดังนั้นจึงมีผู้พัฒนาระบบ NMS ในรูปแบบต่าง ๆ กันมาก ผู้ดูแลและบริหารเครือข่ายสามารถเลือกใช้ซอฟต์แวร์ NMS ได้ โดยมีผู้ผลิตซอฟต์แวร์ หรือผู้ผลิตอุปกรณ์สื่อสารที่อยู่บนอินเทอร์เน็ตให้ดาวน์โหลดมาลองใช้ดูก่อน พัฒนาการของ NMS ได้ก้าวหน้าขึ้นเป็นลำดับ การสร้างความเชื่อมั่นให้กับผู้ใช้บนเครือข่ายเป็นสิ่งสำคัญ NMS จึงเป็นสิ่งที่มีความจำเป็นและเข้ามามีบทบาทสำคัญในเทคโนโลยีเครือข่ายคอมพิวเตอร์

### 1.3 SNMP Message

ในระบบจัดการเครือข่ายอุปกรณ์ทุกตัวที่ต้องการแลกเปลี่ยนข้อมูลข่าวสารกันนั้นจะใช้ SNMP Message ที่ภายในบรรจุข้อมูลสำหรับจัดการเพื่อส่งไปยังปลายทางที่ต้องการ โดยมีรูปแบบการติดต่อระหว่างตัวจัดการ (Management) กับ ตัวแทน (Agent) ดังนี้

- GetRequest เป็น Message ที่ตัว Management Station (MS) ส่งไปยัง Management Agent (MA) เพื่อบอกว่า MS ต้องการทราบข้อมูลอะไรจาก MA ซึ่งกำหนดโดย Object Identifier (OID) ที่ส่งไปพร้อมกับ Message เช่น MS ระบุ OID เป็น 1.3.6.1.2.1.1.0 ซึ่งเป็นการระบุว่าต้องการทราบข้อมูล sysDescr หรือส่วนของรายละเอียดของอุปกรณ์ที่ตัว MA ทำงานอยู่ ซึ่งทาง MA ก็จะตอบข้อมูลรายละเอียดของอุปกรณ์ตัวที่มันทำงานอยู่กลับมา

- getNextRequest โดย Message ชนิดนี้ต่างจาก GetRequest ตรงที่ข้อมูลที่ส่งกลับมาจาก MA จะไม่ใช่ข้อมูลของ Object Identifier ที่ MS ส่งไปให้แต่จะเป็นข้อมูลของ object Identifier ของตัวถัดไปในโครงสร้าง SMI ซึ่งจะใช้ในกรณีที่ ตัว MS ไม่สามารถที่จะระบุ Object Identifier ได้ โดยจะใช้ Message getNextRequest นี้ไปในลักษณะของการท่อง เข้าไปใน Tree ตัวอย่างเช่น MS ส่ง Message getNextRequest ที่ให้ Object Identifier เป็น 1.3.6.1.2.1.1 ซึ่งเป็นการเข้าถึงกลุ่ม System ใน MIB โดยที่ไม่ได้ระบุว่า ต้องการ ทราบข้อมูลอะไรในกลุ่ม System ดังนั้นเมื่อเวลาที่ MA มี Message GetResponse กลับมาให้มันก็จะส่งค่าของ Object Identifier เป็น 1.3.6.1.2.1.1.0 ซึ่งก็คือค่าของ SysDescr ที่อยู่ในกลุ่ม System ซึ่งเป็นค่าของ Object Identifier ตัวถัดไปใน Tree

- SetRequest เป็น Message ที่ MS ใช้บอกให้ MA เปลี่ยนแปลงค่า Configuration ต่าง ๆ ของข้อมูลใน MIB ของอุปกรณ์นั้น ๆ

- GetResponse เป็น Message ที่ MA ใช้ในการส่งผลลัพธ์กลับมาให้ MS จากการที่ MS ได้ทำการส่ง Message GetRequest, getNextRequest, SetRequest ไปให้

- Trap เป็น Message ที่ MA ส่งไปให้ MS เพื่อรายงานเหตุการณ์หรือปัญหาที่เกิดขึ้น

- InformRequest ใน InformRequest PDU จะส่งจาก Manager ตัวหนึ่งไปยังตัวอื่น ๆ ที่อยู่ไกลออกไป เพื่อรับค่าของตัวแปรบางตัวจาก Agent ภายใต้การควบคุมของ Manager ที่อยู่ไกลออกไปนั้น ซึ่ง Manager ที่อยู่ไกลออกไปนั้นจะตอบสนองมาด้วย Response PDU

- Report ใน Report PDU ออกแบบมาเพื่อรายงานข้อผิดพลาดบางประเภทที่กว้าง Manager ด้วยกัน

SNMP เป็นโพรโทคอลในระดับ Application Layer โดยจะใช้โพรโทคอล UDP ของ TCP/IP ใน Layer ที่ 4 โดยอาศัยพอร์ต หมายเลข 161 ในการติดต่อแบบที่ 1 – 4 (GetRequest, getNextRequest, SetRequest, GetResponse) และพอร์ตหมายเลข 162 ใช้ในการติดต่อแบบที่ 5 (Trap) โดย UDP มีลักษณะการติดต่อแบบ Connectionless ดังนั้น กล่าวได้ว่า SNMP ไม่ได้จัดการสร้างกระบวนการการติดต่อระหว่าง Agent กับ Manager อย่างแข็งแรง เพราะ ข้อมูลที่รับส่งจัดว่าเป็น Message ที่ใช้ระหว่างติดต่อสื่อสาร ไม่ใช่ Information ในการทำงานของผู้ใช้ อีกประการคือ UDP จะรับส่งข้อมูลได้อย่างรวดเร็ว และใช้ทรัพยากรของระบบเครือข่ายเพียงเล็กน้อย ซึ่งจะส่งผลกระทบต่อการใช้งานของผู้ใช้งานในระบบเครือข่ายเล็กน้อย อย่างไรก็ตาม UDP ไม่ใช่วิธีการรับส่งข้อมูลที่ดี เพราะขาดความมีประสิทธิภาพในเรื่องของความน่าเชื่อถือของการรับส่งข้อมูล ดังที่แสดงในภาพที่ 2.2

TCP/IP Stack

OSI Model

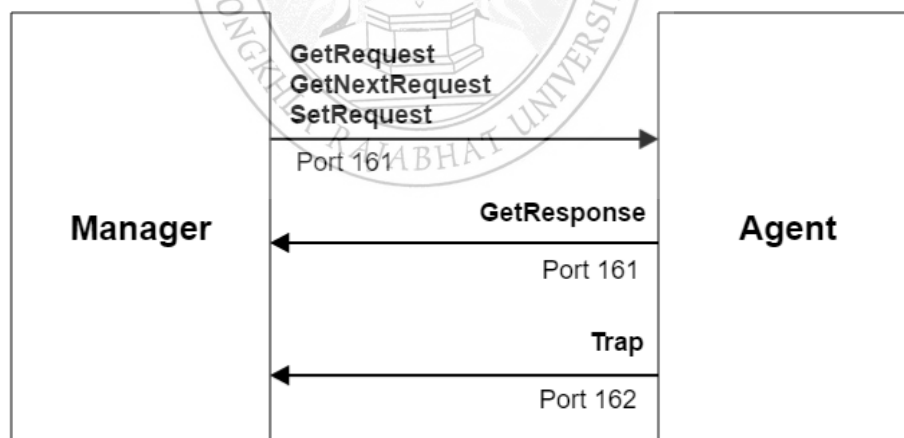
Process Layer (FTP, Telnet, SNMP)		Application Layer
		Presentation Layer
Host-to-Host Layer (TCP)		Session Layer
		Transport Layer
Internet Layer (IP)		Network Layer
Network Interface (IEEE 802.3, 802.5)		Data Link Layer
		Physical Layer

ภาพที่ 2.2 การอ้างอิง TCP กับ SNMP

จากภาพที่ 2.2 เป็นการรับส่งข้อมูลใน SNMP ที่ใช้ UDP ในการรับส่ง ซึ่งทำให้การรับส่งข้อมูลไม่มีประสิทธิภาพมากนัก

#### 1.4 การรับและส่ง SNMP Message

ตัวจัดการ (Manager) และ ตัวแทน (Agent) สามารถติดต่อสื่อสารกันได้โดยมีการรับและส่ง SNMP Message รูปแบบต่าง ๆ ระหว่าง Manager และ Agent ดังที่แสดงในภาพที่ 2.2



ภาพที่ 2.3 การรับส่ง Message ระหว่าง Manager และ Agent

จากภาพที่ 2.3 การทำงานของ SNMP ประกอบด้วย 2 ส่วน คือ ส่วนตัวจัดการ Manager และ ส่วนตัวแทน Agent การทำงานของ SNMP จะเริ่มจากการที่ตัวจัดการส่ง Message ไปให้ตัวแทนซึ่งกำลังรอรับ Message ที่ผ่านพอร์ตหมายเลข 162 หากมี Message GetRequest หรือ SetRequest เข้ามาตัวแทนก็จะตอบสนองโดยการส่ง Message GetResponse กลับไป



Agent มีส่วนประกอบสำคัญสองส่วน คือ Protocol Engine และ Management Information Base หรือเรียกย่อ ๆ ว่า MIB โดย Protocol Engine ทำหน้าที่ประมวลผลคำสั่งที่ได้รับมาจาก NMS เช่น การรับคำสั่ง การถอดรหัสคำสั่ง การทำงานตามคำสั่ง และการให้ค่าแก่คำสั่ง ส่วน MIB เป็นฐานข้อมูลที่เก็บข้อมูลสารสนเทศเพื่อการจัดการ จะเก็บตัวแปร และค่ากำหนดการทำงานของอุปกรณ์นั้น ๆ เรียกว่า Managed Object และแต่ละ Object จะมีชื่อเฉพาะเรียกว่า Object Identifier หรือเรียกย่อ ๆ ว่า OID โดยมีการกำหนดชื่อ แบบของข้อมูล สิทธิการเข้าถึง คำอธิบายลักษณะ และค่าของข้อมูล ตามข้อกำหนดที่เรียกว่า Structure of Management Information หรือ เรียกย่อ ๆ ว่า SMI

#### 4. SMS Tools 3

SMS Tools เป็น SMS Gateway ที่ใช้ส่งข้อความสั้น ๆ โดยผ่าน GSM Modem หรือ โทรศัพท์เคลื่อนที่ที่สามารถส่งข้อความสั้น โดยใช้ Text File ที่อยู่ใน Spool Directory พิเศษ โปรแกรมจะตรวจสอบ Directory นี้แล้วจะส่ง File อัตโนมัติ และรับข้อความที่เข้ามาใน Directory อื่นเป็น Text File หรือ Binary Message และยังสามารถส่ง Ring Tone Message ยังสามารถส่ง WAP Push Message ถึง WAP/MMS ในโทรศัพท์เคลื่อนที่อย่างมีประสิทธิภาพ

โปรแกรมยังสามารถรัน SMS Daemon และจะถูกรันโดยอัตโนมัติเมื่อ OS เริ่มทำงานมีประสิทธิภาพสูง และสามารถใช้ซิมการ์ดที่รองรับระบบ GSM ได้สูงสุดถึง 64 ซิมการ์ด และง่ายในการจัดการระบบ โปรแกรมสามารถรันโปรแกรมภายนอกได้ ด้วย Script จาก Event ที่เกิดขึ้นจากการรับส่งข้อความใหม่สำเร็จ และยังสามารถยังสามารถตรวจสอบปัญหาของโปรแกรม ภายในโปรแกรมเหล่านี้จะบรรทุกเหตุการณ์ขัดข้องลงใน Text File ที่ส่งมาและออกไปได้

SMS Tools สามารถรันบน Microsoft Windows (ด้วย Cygwin) และ Unix รวมไปถึง Solaris, BSD, FreeBSD และ Linux Software

#### 5. Perl Script

##### ประวัติภาษา Perl

ภาษา Perl ย่อมาจากคำว่า Practical Extraction and Report Language ถูกพัฒนาขึ้นโดยนายแลร์รี วอลล์ (Larry Wall) ในปี 1996 เพื่อใช้งานกับระบบปฏิบัติการ Unix ซึ่งในขณะนั้นการพัฒนาแอปพลิเคชันบนระบบ Unix เป็นเรื่องที่ยุ่งยากและต้องมีความรู้ความเข้าใจในภาษาโปรแกรมหลายภาษา เพราะในขณะนั้นภาษาโปรแกรมแต่ละภาษาจะทำงานในเครื่องใดเครื่องหนึ่งโดยเฉพาะ การจะทำงานออกมาสักงานหนึ่งจะต้องใช้โปรแกรมแทบทุกภาษาที่มีในขณะนั้น อย่างเช่นในส่วนของการทำงานกับไฟล์ต้องใช้ภาษา awk หรือ grep และแก้ไขไฟล์ด้วยภาษา sed เป็นต้น

ภาษา Perl ถูกสร้างขึ้นมาเพื่อทำงานทุกส่วนเสร็จสมบูรณ์ในตัวเองไม่ต้องไปเรียกใช้ภาษาอื่นอีก โดยที่ภาษา Perl ได้รวบรวมเอาข้อดีของภาษาต่างๆ ในขณะนั้นเข้ามาไว้ด้วยกัน อย่างเช่น ภาษา grep/awk, sh, sed และ C ทำให้การสร้างแอปพลิเคชันบนระบบปฏิบัติการ Unix เป็นไปได้ อย่างมีประสิทธิภาพมากขึ้น และที่สำคัญภาษา Perl ถูกออกแบบมาให้ใช้งานง่าย

ในปัจจุบันภาษา Perl สามารถใช้งานกับระบบปฏิบัติการที่หลากหลายไม่ว่าจะเป็น Unix, Linux, MVS, VMS, OS/2, Amiga และ Windows นอกจากนี้ด้วยความสำคัญที่สำคัญของภาษา Perl ที่สามารถทำงานได้หลายด้าน เป็นเหตุให้ตัวภาษา Perl ถูกนำมาใช้เขียนโปรแกรมบนอินเทอร์เน็ตประเภทต่างๆ เช่น เว็บ E-Commerce

### ข้อดีของภาษา Perl

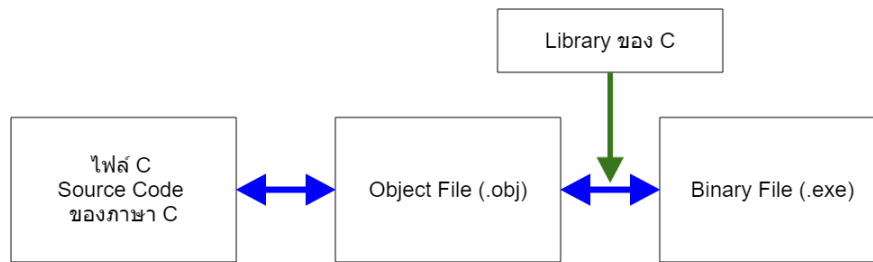
สำหรับข้อดีหลักๆ ที่สำคัญของการใช้งานภาษา Perl พอสรุปได้ดังนี้

1. ง่ายต่อการเรียนรู้ โดยภาษา Perl มีโครงสร้างของภาษาไม่ซับซ้อนเข้าใจง่าย ซึ่งโครงสร้างภาษา Perl จะคล้ายกับภาษา C มาก เพราะภาษา Perl สร้างขึ้นมาโดยใช้ภาษา C ทำให้ผู้ที่คุ้นเคยภาษา C อยู่แล้วใช้งานภาษา Perl ได้ไม่ยาก นอกจากนี้โดยตัวภาษาเองมีความยืดหยุ่นสูงทำให้การจัดการกับงานด้านข้อความ และ Text File ได้เป็นอย่างดี
2. ไม่ต้องเสียค่าใช้จ่ายใดๆ ทั้งสิ้น เพราะตัวแปลภาษา Perl อยู่ภายใต้ลิขสิทธิ์ GNU
3. ใช้ได้หลายแพลตฟอร์ม ในช่วงแรกภาษา Perl ถูกออกแบบใช้งานกับระบบ Unix อยู่ก็จริง แต่ในปัจจุบันได้มีการพัฒนาตัวแปลภาษา Perl ให้สามารถใช้กับระบบปฏิบัติการอื่นๆ อาทิเช่น Linux, Windows 95/98/ME, Windows NT, Windows 2000, OS/2, Macintosh
4. ภาษา Perl ถูกสร้างขึ้นโดยได้รวบรวมเอาส่วนดีของภาษาต่างๆ เข้ามาไว้ด้วยกัน อาทิเช่น ภาษา C, grep/awk, sh, sed
5. ภาษา Perl เป็นภาษาประเภท Server side Script คือการทำงานของภาษา Perl จะทำงานด้านฝั่ง Server แล้วส่งผลลัพธ์กลับมาฝั่ง Client ทำให้มีความปลอดภัยสูง

### หลักการทำงานของตัวแปลภาษา Perl

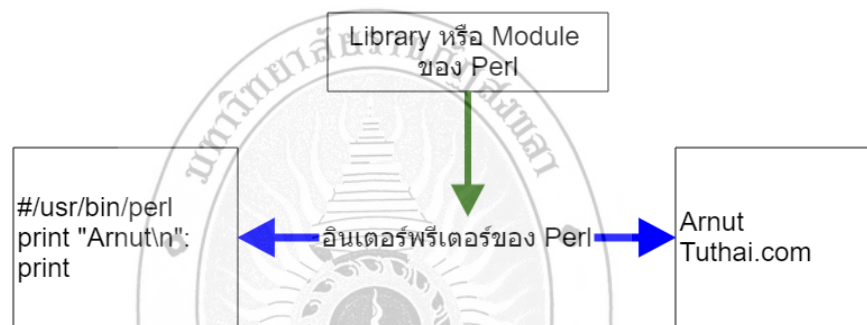
โดยลักษณะของตัวแปลภาษานั้นแบ่งได้ 2 ประเภทใหญ่ ๆ คือ

1. คอมไพเลอร์ (Compiler) เป็นตัวแปลภาษาสำหรับภาษา C, C++, Pascal การทำงานก็คือจะตรวจสอบความผิดพลาดของโค้ดคำสั่งตั้งแต่ต้นจนจบก่อน หรือเรียกว่าการคอมไพล์ ถ้าไม่มีข้อผิดพลาดก็จะทำการแปลโค้ดคำสั่งของเราให้เป็นไฟล์นามสกุล .obj (object file) จากนั้นก็ทำการแปลไฟล์ .obj ให้เป็นไบนารีไฟล์ .exe เพื่อทำงานต่อไป ดังตัวอย่างการทำงานของคอมไพเลอร์ภาษา C ดังรูป



ภาพที่ 2.4 คอมไพเลอร์ภาษา C

2. อินเตอร์พรีเตอร์ (Interpreter) จะทำงานเป็นบรรทัดต่อบรรทัด คือ อ่านโค้ดคำสั่งมาบรรทัดหนึ่งแล้วก็ทำงานให้ผลออกมาเลย ดังแสดงในรูป



ภาพที่ 2.5 อินเตอร์พรีเตอร์ (Interpreter)

จากภาพที่ 2.5 ตัวอย่างในกรณีที่มีการเรียกใช้ฟังก์ชันจากไลบรารี (Library) หรือโมดูล (Module) ของภาษา Perl อินเตอร์พรีเตอร์ของภาษา Perl ก็จะไปทำการเรียกฟังก์ชันเหล่านั้นให้ทำงานแล้วจึงแสดงผลการทำงานออกมา

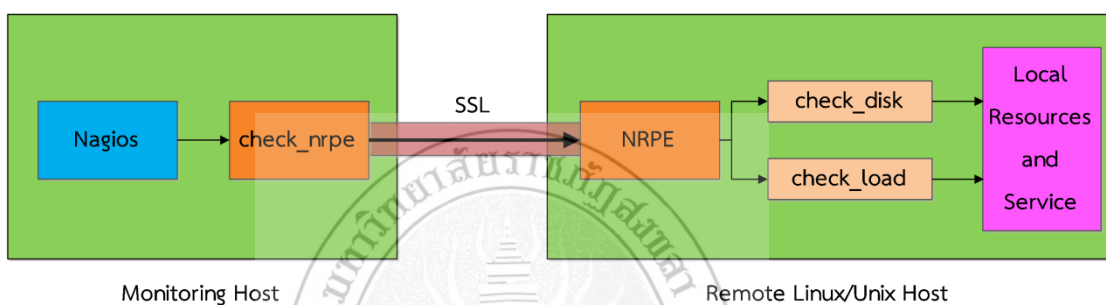
ในส่วนของประสิทธิภาพการทำงานนั้นตัวแปลภาษาแบบคอมไพเลอร์จะทำงานได้เร็วกว่าตัวแปลภาษาและอินเตอร์พรีเตอร์ เพราะโค้ดคำสั่งถูกคอมไพล์และลิงค์โดยตัวแปลภาษาแบบคอมไพเลอร์ผ่านแล้วได้เป็นไฟล์ .exe ออกมา จากนั้นก็เป็นขั้นตอนการทำงานอย่างเดียว

## 6. PNP

เป็นโอเพนซอร์สประเภทหนึ่งที่ทำกร่วมกับ RRDTool สำหรับการสร้างกราฟ เพื่อใช้สำหรับการวิเคราะห์ และติดตามการทำงานของระบบ ไม่ว่าจะเป็นสถิติด้านการใช้งาน CPU, Memory, จำนวนผู้ใช้งานในเครือข่าย, อัตราการรับ/ส่งข้อมูลผ่านเครือข่าย เก็บข้อมูลจาก Nagios มาแสดงเป็นกราฟเพื่อแสดงสถานะการให้บริการ

## 7. NRPE

โดยปกติแล้วการ Monitor จะเป็นการรันผ่าน Plug in ที่ Nagios Server ไปยังเครื่องที่ต้องการจะ Monitor โดยส่วนมากเป็นการส่ง Message ไป และรับรอ Response กลับมา เช่น Check\_Ping , Check\_HTTP แต่การ Monitor บางอย่างนั้นไม่สามารถใช้วิธีนี้ได้ เช่น Check\_Load , Check\_Disk ซึ่ง Plug in เหล่านี้จะสามารถทำได้ในเครื่อง Local เท่านั้น จึงจำเป็นต้องนำ NRPE มาช่วยในการ Monitor โดยการทำงานของ NRPE โดยภาพรวมจะมีการทำงานดังนี้



ภาพที่ 2.6 การทำงานของ NRPE

NRPE (Nagios Remote Plugin Executor) คือ Plug in ใช้สำหรับ monitor local resources บนเครื่อง client ที่เป็นระบบ linux ซึ่งบางอย่างไม่สามารถ monitor โดยใช้ SNMP v1,v2,v3 ได้ จึงใช้ NRPE มา monitor เช่น การ monitor CPU , Memory , Storage เป็นต้น ซึ่งจะสามารถอธิบายการทำงานได้ 4 ขั้นตอน ดังนี้

1. Nagios ทำการ Execute Check\_NRPE Plug in และระบุ Service ที่ต้องการ Monitor
2. Check\_NRPE Plug in จะทำการติดต่อกับ NRPE Daemon ที่ฝั่งของ Remote Linux Host บน SSL-Protected Connection
3. NRPE Daemon จะเรียกคำสั่งที่ Nagios Plug in ได้จัดเตรียมไว้เพื่อที่จะไปตรวจสอบ และนำข้อมูลที่ Nagios ต้องการออกมา
4. ผลลัพธ์จากการตรวจสอบจะถูก NRPE Daemon ส่งกลับไปยัง Check\_NRPE Plug in และส่งผลการตรวจสอบกลับไปยัง Nagios

## 8. OID

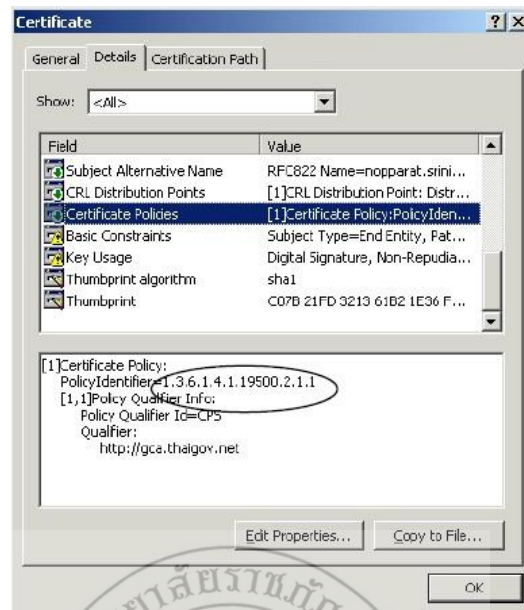
OID หรือ Object Identifier คือ ชุดตัวเลขที่ใช้ระบุอ้างอิงวัตถุ (object) และบ่งบอกถึงความเป็นหนึ่ง ของวัตถุใดๆ เป็นกลไกการระบุวัตถุที่ใช้กันอย่างกว้างขวาง มีความยืดหยุ่นในการใช้งาน สำหรับใช้ในการตั้งชื่อ ชนิดของวัตถุแนวคิด หรือ สิ่งใดๆ เพื่อให้มีความชัดเจน ไม่ซ้ำกันทั่วโลก และเพื่อใช้ระบุอ้างอิงวัตถุนั้น ๆ

แนวคิดของ OID เริ่มต้นขึ้นในช่วงทศวรรษ 1980 ตอนต้น โดยมีพื้นฐานจากความต้องการการระบุวัตถุใดๆ ให้มีความชัดเจน และเพื่อให้รองรับการสื่อสารผ่านระบบคอมพิวเตอร์ในทุก ๆ รูปแบบ สหภาพโทรคมนาคมระหว่างประเทศ ภาคการกำหนดมาตรฐานโทรคมนาคม (International Telecommunication Union Telecommunication Standardization Sector: ITU-T) และ องค์การระหว่างประเทศว่าด้วยการมาตรฐาน / คณะกรรมาธิการระหว่างประเทศว่าด้วยมาตรฐานสาขาอิเล็กทรอนิกส์ (International Organization for Standardization / International Electrotechnical Commission: ISO/IEC) จึงได้ร่วมกัน พัฒนามาตรฐานที่เกี่ยวข้องกับ OID ขึ้นในปี ค.ศ. 1984

การทำงานของ OID มีพื้นฐานจากโครงสร้างของชื่อแบบมีลำดับชั้นและสามารถเพิ่มขยายได้ไม่จำกัด หรือที่เรียกว่า “OID tree” ทั้งนี้ข้อเสนอแนะของ ITU-T (ITU-T Recommendations) และมาตรฐานสากล ของ ISO/IEC ส่วนใหญ่ได้กำหนดให้การระบุถึงสิ่งใดๆ จะต้องมีความชัดเจน ไม่คลุมเครือ สำหรับการจัดสรร หมายเลข OID นั้น มีวิธีที่ไม่ซับซ้อน กล่าวคือ มีหน่วยงานรับผิดชอบเพียงหมายเลย OID (Registration Authority: RA) ทำหน้าที่รับผิดชอบและบริหารหมายเลข OID ภายใต้งิ่งที่ได้รับการจัดสรร

1) OID ส่วนที่กำหนดเป็นค่าแน่นอนภายในมาตรฐานใดมาตรฐานหนึ่ง เป็น OID ที่สามารถนำมาใช้ได้เลยภายในขอบเขตของมาตรฐานนั้น ๆ เช่น {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanism(5) pkix(7) keyPurpose(3) ClientAuth(2)} คือ OID ที่อ้างอิงเพื่อกำหนดจุดประสงค์การใช้งานใบรับรอง ว่าใบรับรองใช้เพื่อทำ SSL Client Authenticate ซึ่งเป็น OID ที่กำหนดอยู่ใน RFC 3280

2) OID ส่วนที่เป็นเพียงโครงสร้างพื้นฐาน สามารถออกแบบโครงสร้างกำหนดสิทธิ์ย่อย หรือ ขอบเขตของการใช้งาน ภายใต้อัตลักษณ์ในการจัดการข้อมูลที่อยู่ในเอกสาร X.660 โดยขอบเขตการใช้งาน OID จะขึ้นอยู่กับนโยบาย และการจัดการของผู้มีสิทธิ์ในการจัดการข้อมูล



ภาพที่ 2.7 ผลใบรับรองอิเล็กทรอนิกส์

ภาพที่ 2.7 ตัวอย่างใบรับรองอิเล็กทรอนิกส์ซึ่งแสดงให้เห็นถึงโครงสร้างข้อมูลที่มีการใช้ OID ที่กำหนดอยู่ใน RFC 3280 คือ ชื่อ Field “Certificate Policy” และค่าของ Field (Value) คือ ข้อมูลที่ใช้ในการอ้างอิง Certificate Policy ซึ่งจากตัวอย่างได้มีการกำหนดเป็น OID ไว้ด้วยแต่ OID ส่วนนี้เป็น OID ที่ไม่ได้กำหนดในมาตรฐานใด ๆ หรือเป็น OID ที่จำเป็นต้องจดทะเบียนภายใต้ขอบเขตของนโยบายของผู้ใช้ หรือ ผู้รับจดทะเบียน

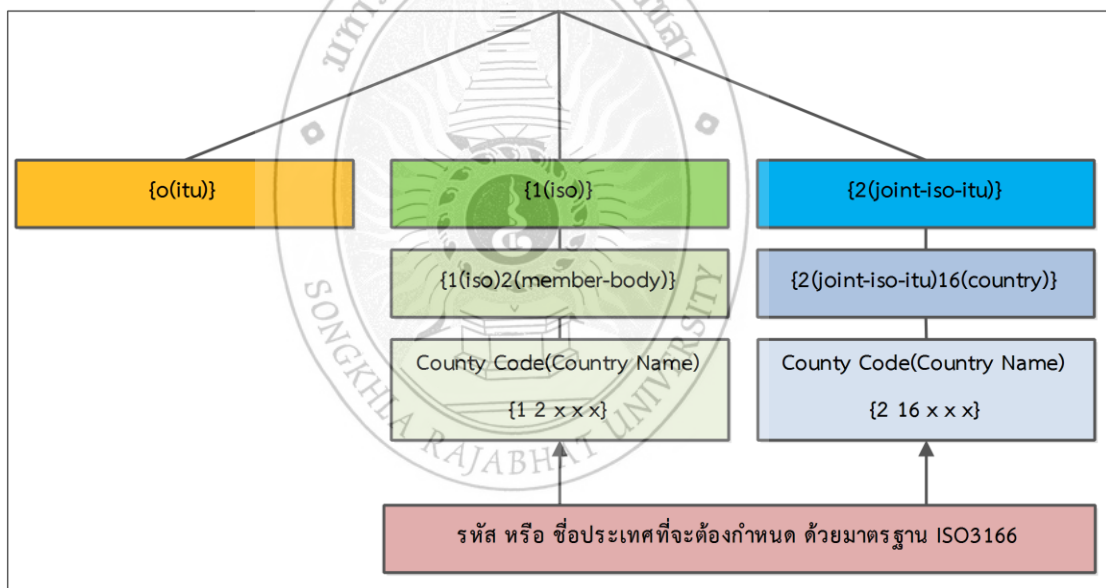
สำหรับการนำ OID มาใช้งาน นอกจากการนำมาใช้ในเรื่องของใบรับรองอิเล็กทรอนิกส์แล้ว ยังสามารถนำมาใช้กับเรื่องอื่น ๆ ได้ เช่น การอ้างอิงชื่อประเทศ หรือ ชื่อหน่วยงาน การอ้างอิง Attribute สำหรับ LDAP Directory การอ้างอิงเอกสารมาตรฐาน การอ้างอิงสำหรับโปรโตคอล SNMP (Simple Network Management Protocol) เป็นต้น

Object Identifier (OID) เป็นวิธีการอ้างอิงข้อมูลโดยที่ OID มีโครงสร้างการอ้างอิงแบบต้นไม้ (Tree) ที่เป็นลำดับขั้น ซึ่งเป็นไปตามมาตรฐานของ Abstract Syntax Notation One (ASN.1) ซึ่ง ASN.1 เป็นมาตรฐานเพื่อให้แอปพลิเคชัน หรือโปรโตคอลสามารถแลกเปลี่ยนข้อมูลกันได้

การนำ OID มาใช้ร่วมกับเรื่องของใบรับรองอิเล็กทรอนิกส์นั้น จำเป็นต้องมีการวางโครงสร้างของ OID ให้สามารถบริหารจัดการได้ง่าย เพื่อรองรับการ Interoperability กับนานาประเทศในอนาคต

โหนด Top Level ของ OID มี 3 โหนด คือ {itu-t(0)} , {iso(1)} และ {joint-iso-itu-t(2)}

1. {itu-t(0)} เป็นโหนดของ International Telecommunication Union (ITU-T) ซึ่งเป็นโหนดที่สงวนไว้ใช้สำหรับเอกสารอ้างอิงส่วน ITU ดังนั้นการจดหมายเลข OID สำหรับประเทศไม่สามารถจดภายใต้กิ่งนี้ได้
2. {iso(1)} เป็นโหนดของ International Organization for Standardization (ISO) ซึ่งประเทศต่าง ๆ สามารถจดทะเบียน OID สำหรับประเทศภายใต้กิ่งนี้ได้ โดยสามารถจดได้ที่ {iso(1) member-body(2)}
3. {joint-iso-itu-t(2)} เป็นโหนดร่วมกันระหว่าง International Telecommunication Union (ITU-T) และ International Organization for Standardization (ISO) ซึ่งเหมาะกับประเทศที่มีหน่วยงานได้ ISO และหน่วยงานด้าน ITU ภายในประเทศ ซึ่งประเทศต่าง ๆ สามารถจดทะเบียน OID สำหรับประเทศภายใต้กิ่งนี้ได้ โดยสามารถจดได้ที่ {joint-iso-itu-t(2) country(16)}



ภาพที่ 2.8 โครงสร้างของ OID แต่ละโหนด

จากมาตรฐาน ITU-T X.660 ซึ่งเป็นเอกสารมาตรฐานที่เกี่ยวข้องกับ OID ระบุว่า การจดทะเบียน OID สำหรับประเทศขึ้นกับการพิจารณาของแต่ละประเทศ ซึ่งถ้าประเทศใดต้องการที่จะจดทะเบียน OID และประเทศนั้นมีหน่วยงานที่เกี่ยวข้องกับ ISO National Body (ประเทศไทย : สำนักงานมาตรฐานผลิตภัณฑ์อุตสาหกรรม) หรือ Country's ITU-T Administration (ประเทศไทย : คณะกรรมการกิจการโทรคมนาคมแห่งชาติ) หรือ มีทั้ง 2 หน่วยงานดังกล่าว ประเทศนั้นควรจดทะเบียนหมายเลข OID ประเทศภายใต้กิ่งที่ 1 ได้เช่น ขึ้นอยู่กับการพิจารณาของประเทศนั้น ๆ

### แนวคิดพื้นฐานของ OID

OID มีพื้นฐานบนโครงสร้างของชื่อแบบมีลำดับชั้นและมีโครงสร้างเป็น Tree สามารถระบุหรืออ้างอิง ชื่อจากชื่อโหนดระดับบนสุด (Top level node) ใน OID tree ผ่านทางกิ่ง (arc) ซึ่งเชื่อมต่อไปยังโหนดลูก ต่อๆ กันไป โดยไม่จำกัดความลึก แต่ละโหนดจะถูกกำหนดชื่อเป็นตัวเลขหรือตัวอักษรและสามารถมีโหนดลูกได้ไม่จำกัด (ยกเว้น root จะมีโหนดลูกเพียง 3 โหนด)

การระบุหรืออ้างอิงทำได้โดยการระบุตัวเลขหรือตัวอักษรของโหนดบนเส้นทาง (path) จาก root ถึง โหนดนั้นๆ ทั้งนี้หน่วยงานรับผิดชอบ (Registration Authority: RA) จะเป็นผู้ทำหน้าที่กำหนด / จัดสรร RA ในลำดับถัดไป และต่อๆ ไป โดยไม่จำกัดความลึก ดังนั้น OID tree คือ โครงสร้างแบบมีลำดับชั้นของ RA นั้นเอง

### 9. RRDTool

RRD เป็นคำศัพท์ที่ย่อมาจากคำว่า Round Robin Database ซึ่งเป็นระบบที่มีการจัดเก็บ และแสดงผลข้อมูลตามช่วงเวลาที่มีความต่อเนื่อง ยกตัวอย่างข้อมูลประเภทนี้ เช่น ปริมาณการใช้ Bandwidth ของเครือข่าย อุณหภูมิของห้องคอมพิวเตอร์ Server Load Average เป็นต้น ซึ่งข้อมูลจะมีการเก็บอย่างกระชับแน่นอน จะไม่มีการขยายเพิ่มขึ้นแม้ว่าจะมีเวลาเพิ่มขึ้นก็ตาม นอกจากนี้ RRDTool ยังมีกระบวนการพิเศษที่ทำการรวมข้อมูลดิบกับข้อมูลที่ได้เก็บไว้แล้ว กระบวนการนี้จะทำให้ข้อมูลที่จัดเก็บมีขนาดเล็กลงเพื่อที่จะประหยัดพื้นที่ในการจัดเก็บ และยังมีฟังก์ชันในการรวมข้อมูล อาทิเช่น AVERAGE , MAXIMUM , MINIMUM และ LAST

คุณสมบัติพิเศษของ RRDTool คือ มีฟังก์ชันที่ใช้ในการสร้างกราฟมาให้ ซึ่งสามารถนำไปใช้งานร่วมกับ Web Server ได้ เพื่อให้ผู้ใช้งานสามารถเรียกข้อมูลกราฟผ่านทาง Browser ได้ทุกประเภท การสร้างกราฟของ RRDTool นั้นมีคุณสมบัติหลากหลายประการ เช่น การนำข้อมูลหลายชิ้นมารวมไว้ในกราฟเดียว สนับสนุนการทำ Autoscaling และ Logarithmic ในแกน y การนำข้อมูลมาต่อกันในรูปแบบ Stack และการแสดงค่า Minimum , Average , Maximum เป็นต้น เพื่อให้กราฟออกมาในแบบที่สวยงามอีกด้วย

### 10. DNS

DNS คือ Domain Name System (DNS ได้มีการประกาศใน RFC 1034 – RFC 1035 RFC1034 กล่าวถึงการแนะนำ DNS, การใช้อีเมล, การสนับสนุนที่อยู่ของโฮสต์, โพรโทคอล และเซิร์ฟเวอร์ที่ได้ใช้ชื่อโดเมนเนม ส่วน RFC1035 กล่าวถึงการดำเนินการและข้อมูลจำเพาะ) เป็นระบบแปลงชื่อเว็บเป็นหมายเลข IP Address โดยมีโครงสร้างฐานข้อมูลแบบลำดับชั้น เพื่อใช้เก็บข้อมูล ที่เรียกค้นได้อย่างรวดเร็ว



ระบบ DNS ทำหน้าที่แปลงข้อมูลชื่อ และหมายเลข IP Address หรือสามารถทำกลับกันได้ นอกจากนี้ยังมีฟังก์ชันเพิ่มเติมอื่นๆ อีก เช่น แจกชื่ออีเมลเซิร์ฟเวอร์ในโดเมนที่รับผิดชอบ ด้วยระบบ DNS จะมีการกำหนด Name Space ที่มีกฎเกณฑ์อย่างชัดเจน มีกลไกการเก็บข้อมูลเป็นฐานข้อมูลแบบกระจาย ทำงานในลักษณะของ Client Server

### การทำงานของระบบ DNS

การทำงานของระบบชื่อโดเมนนั้น เริ่มต้นจากเครื่องคอมพิวเตอร์ซึ่งทำหน้าที่เป็น DNS Server ซึ่งทำงานด้วยซอฟต์แวร์พิเศษชื่อว่า BIND ที่ทำหน้าที่รับส่งข้อมูลระหว่าง DNS Server กับเครื่องคอมพิวเตอร์แต่เครื่องผ่าน DNS Protocol เมื่อมีคำร้องขอให้สืบค้นหมายเลข IP Address ใดๆก็ตาม คำตอบที่ DNS Server จะมีให้ต่อคำร้องขอหนึ่งๆ นั้นขึ้นกับว่า DNS Server นั้นเป็น DNS Server ประเภทใด ซึ่ง DNS Server สามารถแบ่งได้ 3 ประเภท คือ

1. Name Resolvers โดยเครื่อง Client ที่ต้องการสอบถามหมายเลขไอพีเรียกว่า Resolver ซึ่งซอฟต์แวร์ที่ทำหน้าที่เป็น Resolvers นั้นจะถูกสร้างมากับแอปพลิเคชันหรือเป็น Library ที่มีอยู่ใน Client

2. Domain Name Space เป็นฐานข้อมูลของ DNS ซึ่งมีโครงสร้างเป็น Tree หรือเป็นลำดับชั้น แต่ละโหนดคือ โดเมนโดยสามารถมีโดเมนย่อย (Sub Domain) ซึ่งจะใช้จุดในการแบ่งแยก

3. Name Servers เป็นคอมพิวเตอร์ที่รันโปรแกรมจัดการฐานข้อมูลบางส่วนของ DNS โดย Name Server จะตอบการร้องขอทันที โดยการหาข้อมูลตัวเอง หรือส่งต่อการร้องขอไปยัง Name Server อื่น ซึ่งถ้า Name Server มีข้อมูลของส่วนโดเมนแสดงว่า Server นั้นเป็นเจ้าของโดเมนเรียกว่า Authoritative แต่ถ้าไม่มีเรียกว่า Non-Authoritative

### ข้อจำกัดของระบบ DNS

1. รับรู้เฉพาะตัวอักษรละติน (ASCII character set)
2. ใน RFC 1035 ระบุว่าสัญลักษณ์ที่ใช้ได้ในโดเมนเนม คือ
  - 2.1 ตัวอักษร a ถึง z (case insensitive)
  - 2.2 เลข 0 – 9
  - 2.3 เครื่องหมายยัติภังค์ (-)

## 11. GSM

GSM ย่อมาจาก Global System for Mobile communication ซึ่งเป็นระบบโทรศัพท์เคลื่อนที่ระบบดิจิทัลระบบหนึ่ง ซึ่งใช้กันอย่างกว้างขวางทั่วโลก โดยทั่วโลกมีผู้ใช้งานถึงกว่าพันล้านคน และมีใช้ในกว่า 190 ประเทศ โดยเฉพาะในยุโรป ซึ่งถือเป็นมาตรฐานของยุโรปไปโดยปริยาย สำหรับประเทศไทย เราก็ใช้ระบบ GSM กันอย่างกว้างขวางเช่นกัน โดยมีส่วนแบ่งการตลาดสูงที่สุดเช่นกัน ดังนั้นผู้ใช้ระบบ GSM จึงสามารถทำ International Roaming ได้ง่าย โดยใช้เครื่องเติมของตัวเอง

GSM นั้นใช้เทคนิคการส่งข้อมูลแบบ TDMA (Time Division Multiple Access) หรือการแบ่งการเข้าถึงข้อมูลหลาย ๆ ชุดตามช่วงเวลา ซึ่งเป็นเทคนิคที่ได้รับความนิยมที่สุดในระบบการสื่อสารไร้สาย โดยกระบวนการเริ่มจากการเปลี่ยนสัญญาณเสียงให้เป็นดิจิทัล และ บีบอัดข้อมูลจากนั้นก็ส่งสัญญาณที่บีบอัดไปพร้อมข้อมูลของผู้ใช้งาน โดยผู้ใช้งานจะได้รับการจองลำดับสัญญาณไว้ในลำดับเต็ม หรือสล็อตเต็มตลอดเวลาใน timeslot โดยมากจะใช้ช่องสัญญาณในย่านความถี่ 900 MHz และ 1800 MHz ในการส่งข้อมูล ซึ่งจะใช้ Bandwidth รวมประมาณ 25 MHz จากนั้นซอยช่องสัญญาณในการส่งออกเป็นช่องย่อย ๆ โดยมีความกว้างช่องสัญญาณ (Bandwidth) 200 KHz สำหรับ 1 ช่องสัญญาณ (Channel) ดังนั้นโดยส่วนมากจำนวนช่องสัญญาณของ GSM จะมีได้ประมาณ 124 ช่องสัญญาณ โดยใน 1 ช่องสัญญาณจะรองรับผู้ใช้งานได้ 8 คน โดยแบ่งเป็น 8 timeslot ดังแสดงในรูปด้านล่าง โดยความเร็วในการสื่อสารที่ทำได้สูงสุดสำหรับ 8 timeslots ต่อช่อง ก็คือ 270.833 Kbit/s

โดยสรุป หลักการทำงานของระบบ GSM คือ เสียงที่เราพูดจะถูกกรองโดย Filter จากนั้นจะแปลงเป็นสัญญาณดิจิทัล และทำการบีบอัดข้อมูลเพื่อลดจำนวนข้อมูลที่ต้องส่ง ก่อนจะแบ่งสัญญาณที่เราพูดนี้ออกเป็นส่วนย่อย ๆ โดยแต่ละส่วนย่อย ๆ จะมีข้อมูลผู้ใช้ติดไปกับชุดข้อมูลทุกชุด สำหรับใน 1 ช่องสัญญาณ จะมีทั้งหมด 8 timeslot นั้นหมายความว่ารองรับผู้ใช้ได้ 8 คนพร้อม ๆ กัน

## 12. งานวิจัยที่เกี่ยวข้อง

12.1 งานวิจัยเรื่อง “ระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่ายและเครื่องบริการ โปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา” โดยนางสาวชัชวานี ดนกล้าเต๊ะและนางสาวสนธญา บิลหวัน มหาวิทยาลัยราชภัฏสงขลา, 2554

โดยโครงการนี้มีการจัดทำระบบตรวจสอบสถานะการให้บริการของอุปกรณ์เครือข่าย และเครื่องบริการภายในโปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา เป็นการใช้

Open Source ซึ่งแกนหลักคือ Nagios ซึ่งเป็น Open Source ใช้ในการตรวจสอบสถานะการทำงาน และตรวจสอบทรัพยากรของอุปกรณ์ในเครือข่ายได้ โดยมีการนำโพรโตคอล SNMP เข้ามาใช้ในการเรียกดูข้อมูลรายละเอียดของอุปกรณ์ในเครือข่าย และได้มีการเชื่อมโยงการทำงานกับโทรศัพท์มือถือเพื่อให้สามารถส่งข้อความสั้น (Short Message Service) ไปยังผู้ดูแลระบบ

#### ความสามารถของระบบ

- ตรวจสอบการทำงานของอุปกรณ์เครือข่ายและเครื่องบริการผ่าน Web Interface ได้ผ่านระบบเครือข่ายเดียวกัน
- ระบบทำการส่งข้อความสั้นไปยังผู้ดูแลระบบ เมื่ออุปกรณ์บนเครือข่ายหยุดให้บริการ ผ่านทางโทรศัพท์เคลื่อนที่ไปยังผู้ดูแลระบบ
- แสดงข้อมูลการให้บริการ และข้อมูลทรัพยากรเครื่อง ในรูปแบบกราฟ

12.2 งานวิจัยเรื่อง “ระบบมอนิเตอร์ริงกราฟฟิกผ่านเว็บเบราว์เซอร์” โดยนายเกียรติศักดิ์ แสงอรุณ สาขาวิศวกรรมเครือข่าย มหาวิทยาลัยเทคโนโลยีมหานคร, 2554

โดยโครงการนี้ทำขึ้นเพื่อคอยตรวจสอบปริมาณข้อมูลที่วิ่งเข้าออกผ่านอุปกรณ์เกตเวย์ หรือสวิตช์เพื่อให้ทราบว่ามีการใช้งานเครือข่ายมากที่สุด และน้อยที่สุดในช่วงเวลาใด โดยเฉพาะปริมาณกราฟฟิกนั้นมีสูงกว่าเส้นทางการสื่อสารหรือไม่ โดยมีการพัฒนาระบบ Monitor Traffic ของเครือข่ายภายในศูนย์ Network Operation Center ให้มีประสิทธิภาพโดยใช้ Cacti Traffic Grapher ซึ่งเป็นซอฟต์แวร์แบบ Open Source

#### ความสามารถของระบบ

- สามารถตรวจสอบข้อมูลที่ไหลผ่านในระบบ และบริหารจัดการกราฟฟิกได้อย่างเหมาะสม
- แสดงปริมาณกราฟฟิกในรูปแบบของกราฟ สามารถเลือกดูแบบรายวัน รายเดือน รายปีได้
- สามารถส่งการแจ้งเตือนเมื่อระบบในเครือข่ายมีปัญหาผ่านจดหมายอิเล็กทรอนิกส์ (E-Mail)

12.3 งานวิจัยเรื่อง “ระบบการตรวจสอบการให้บริการเว็บและจดหมายอิเล็กทรอนิกส์กรณีศึกษา บริษัท ซอฟต์แวร์ จำกัด (มหาชน)” โดยนายอานนท์ มุ่งยอดกลาง สาขาวิชาความมั่นคงทางระบบสารสนเทศ คณะวิทยาการและเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีมหานคร, 2557

โดยโครงการนี้ได้จัดทำขึ้นเพื่อเป็นระบบตรวจสอบสถานะการให้บริการของบริษัท ซอฟต์แวร์ จำกัด (มหาชน) ซึ่งระบบนี้ได้นำ Nagios ซึ่งเป็นซอฟต์แวร์แบบ Open Source และ

โครงการนี้ได้นำมาใช้สำหรับการตรวจสอบสถานะการให้บริการของ Web Server และ Email Server

#### ความสามารถของระบบ

- ตรวจสอบความพร้อมใช้ของระบบ
- มีการแจ้งเตือนผ่าน Email เมื่อ เกิดปัญหาหรือข้อผิดพลาดในการทำงานของระบบ

### สรุปงานวิจัยที่ได้จากการศึกษา

จากการศึกษางานวิจัยดังกล่าวข้างต้น เป็นการศึกษางานวิจัยที่สอดคล้องกับงานวิจัยที่ทางคณะผู้จัดทำได้ดำเนินการ ซึ่งสามารถสรุปได้ดังนี้

1. รู้ และเข้าใจหลักการทำงานของ Nagios ซึ่งเป็นแกนหลักที่ใช้ในโครงการนี้ รวมถึงการนำ Plug in และส่วนเสริมของ Nagios คือ Nagios Plug-In และ NRPE มาผนวกใช้งานกับการเรียกดูข้อมูลรายละเอียดการตรวจสอบสถานะต่าง ๆ ของอุปกรณ์เครือข่าย
2. รู้ถึงหลักการทำงานของโพรโตคอล SNMP ที่ใช้ในการเรียกดูข้อมูล และทราบถึงการนำโพรโตคอล SNMP มาประยุกต์กับ Nagios
3. ทราบถึงหลักการของการเขียนคำสั่งเพื่อติดต่อระหว่างโทรศัพท์เคลื่อนที่กับคอมพิวเตอร์ที่ต้องการเพื่อให้สามารถส่งข้อความไปยังโทรศัพท์ปลายทางได้

## บทที่ 3

### การวิเคราะห์และออกแบบระบบ

#### ความต้องการของระบบ

ระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่ายและเครื่องบริการ เป็นระบบที่ช่วยอำนวยความสะดวกในการตรวจสอบสถานะการหยุดให้บริการ และเพิ่มความสามารถในการเข้าตรวจสอบอุปกรณ์บนเครือข่ายได้อย่างรวดเร็ว โดยการส่งข้อความสั้น (SMS : Short Message Service) แจ้งเตือนไปยังผู้ดูแลระบบหากอุปกรณ์บนเครือข่ายเกิดสถานะการหยุดให้บริการพร้อมทั้งสามารถตรวจสอบสถานะของระบบต่าง ๆ ผ่านทาง Web Page โดยผู้ดูแลระบบสามารถดูข้อมูลสถานะการหยุดให้บริการของอุปกรณ์เครือข่ายทั้งปัจจุบันและย้อนหลัง ในรูปแบบเอกสาร ตาราง และกราฟ

#### นิยามคำศัพท์พร้อมความหมายที่เกี่ยวข้อง

1. ผู้ดูแลระบบ หมายถึง ผู้ที่สามารถจัดการข้อมูลทั้งหมดของระบบ
2. เครื่องบริการ หมายถึง Server
  - 2.1 เครื่องบริการ Nagios หมายถึง เครื่องที่ติดตั้งบนระบบปฏิบัติการ Linux Debian 7 (Version Wheezy) สำหรับทำหน้าที่เป็น Nagios Server เพื่อตรวจสอบอุปกรณ์ และเครื่องบริการอื่น ๆ บนเครือข่ายที่ต้องการจะตรวจสอบ
  - 2.2 เครื่องบริการ printer.1 หมายถึง เครื่องพิมพ์งานเอกสารที่อยู่ภายในโปรแกรมวิชาคอมพิวเตอร์
  - 2.3 เครื่องบริการ printer.4 หมายถึง เครื่องพิมพ์งานเอกสารที่อยู่ภายในโปรแกรมวิชาคอมพิวเตอร์
  - 2.4 เครื่องบริการ AP-8-402[.2] หมายถึง เครื่องอุปกรณ์ที่ทำหน้าที่กระจายสัญญาณอินเทอร์เน็ตไร้สาย (Wireless Internet)
  - 2.5 เครื่องบริการ AP-8-402[.3] หมายถึง เครื่องอุปกรณ์ที่ทำหน้าที่กระจายสัญญาณอินเทอร์เน็ตไร้สาย (Wireless Internet)
  - 2.6 เครื่องบริการ server8407 หมายถึง เครื่องบริการที่ถูกติดตั้งไว้ที่ห้อง 8 -407 ซึ่งเครื่องบริการนี้เป็นเครื่องบริการที่ใช้ในการจัดเก็บบัญชีผู้ใช้ของโปรแกรมวิชาคอมพิวเตอร์

2.7 เครื่องบริการ gateway-8402 หมายถึง เครื่องบริการที่ให้บริการควบคุมการใช้งานเครือข่ายอินเทอร์เน็ตโปรแกรมวิชาคอมพิวเตอร์ในการติดต่อกับเครือข่ายภายนอก ซึ่งได้ถูกติดตั้งอยู่ที่ห้อง 8-402

2.8 เครื่องบริการ VMServer.15 หมายถึง เครื่องบริการจำลอง ซึ่งได้ถูกติดตั้งระบบปฏิบัติการ Linux

2.9 เครื่องบริการ VMServer.16 หมายถึง เครื่องบริการจำลอง ซึ่งได้ถูกติดตั้งระบบปฏิบัติการ Linux และ มีการให้บริการแชร์ไฟล์ (Samba)

2.10 เครื่องบริการ VMServer.18 หมายถึง เครื่องบริการจำลอง ซึ่งได้ถูกติดตั้งระบบปฏิบัติการ Linux

2.11 เครื่องบริการ WinServer.19 หมายถึง เครื่องบริการที่ถูกติดตั้งระบบปฏิบัติการ Windows Server สำหรับรองรับ Database

2.12 เครื่องบริการ webcomsci หมายถึง เครื่องบริการที่ทำหน้าที่เป็นเว็บไซต์ของโปรแกรมวิชา คอมพิวเตอร์

2.13 เครื่องบริการ Cacti หมายถึง เครื่องบริการที่เป็นเครื่องโครงงานของนักศึกษา

3. เครือข่าย หมายถึง กลุ่มคอมพิวเตอร์ หรืออุปกรณ์สื่อสารชนิดต่างๆ ที่เชื่อมต่อเพื่อให้ผู้ใช้ในเครือข่าย สามารถติดต่อสื่อสารแลกเปลี่ยนข้อมูลและใช้อุปกรณ์ต่าง ๆ ร่วมกัน

4. GSM หมายถึง มาตรฐานของเทคโนโลยีการสื่อสารที่ให้บริการโทรศัพท์เคลื่อนที่

5. Secure Shell หมายถึง คือ โปรแกรมสำหรับล็อกอินและรันคำสั่งที่เครื่องปลายทางได้โดยไม่ต้องไปใช้งานที่หน้าจอคอนโซลของเครื่อง

6. Aircard (GSM Modem) หมายถึง อุปกรณ์ที่ถูกบรรจุซิมการ์ดโทรศัพท์ไว้ภายใน โดยถูกเชื่อมต่อกับเครื่องบริการ Nagios ซึ่งจะทำหน้าที่ส่งข้อความสั้น (SMS : Short Message Service) ไปยังหมายเลขโทรศัพท์ของผู้ดูแลระบบ

7. Pooling หมายถึง การที่เครื่องบริการร้องขอข้อมูลจากอุปกรณ์ในเครือข่าย ซึ่งอุปกรณ์ในเครือข่ายนั้นจะตอบกลับด้วยการส่งข้อมูลกลับมาตามที่เครื่องบริการที่เป็นฝ่ายร้องขอ

## การวิเคราะห์และออกแบบระบบ

การวิเคราะห์ และออกแบบระบบตรวจสอบสถานะการหยุดให้บริการอุปกรณ์เครือข่าย และเครื่องบริการศึกษาการแจ้งเตือนสถานะการหยุดให้บริการอุปกรณ์เครือข่าย โดยการแจ้งเตือนผ่านทางโทรศัพท์เคลื่อนที่โดยใช้ข้อความสั้น (Short Message Service) และแสดงสถานะการทำงานผ่าน Web Page

1. .แผนผังการทำงานของระบบตรวจสอบสถานะการหยุดให้บริการอุปกรณ์เครือข่ายและเครื่องบริการ

1.1 ผู้ดูแลระบบ เข้าสู่ระบบ โดยการระบุตัวตนด้วย Username และ Password

1.2 ระบบจะตรวจสอบ Username และ Password หากไม่ถูกต้อง จะต้องระบุตัวตนด้วย Username และ Password ใหม่

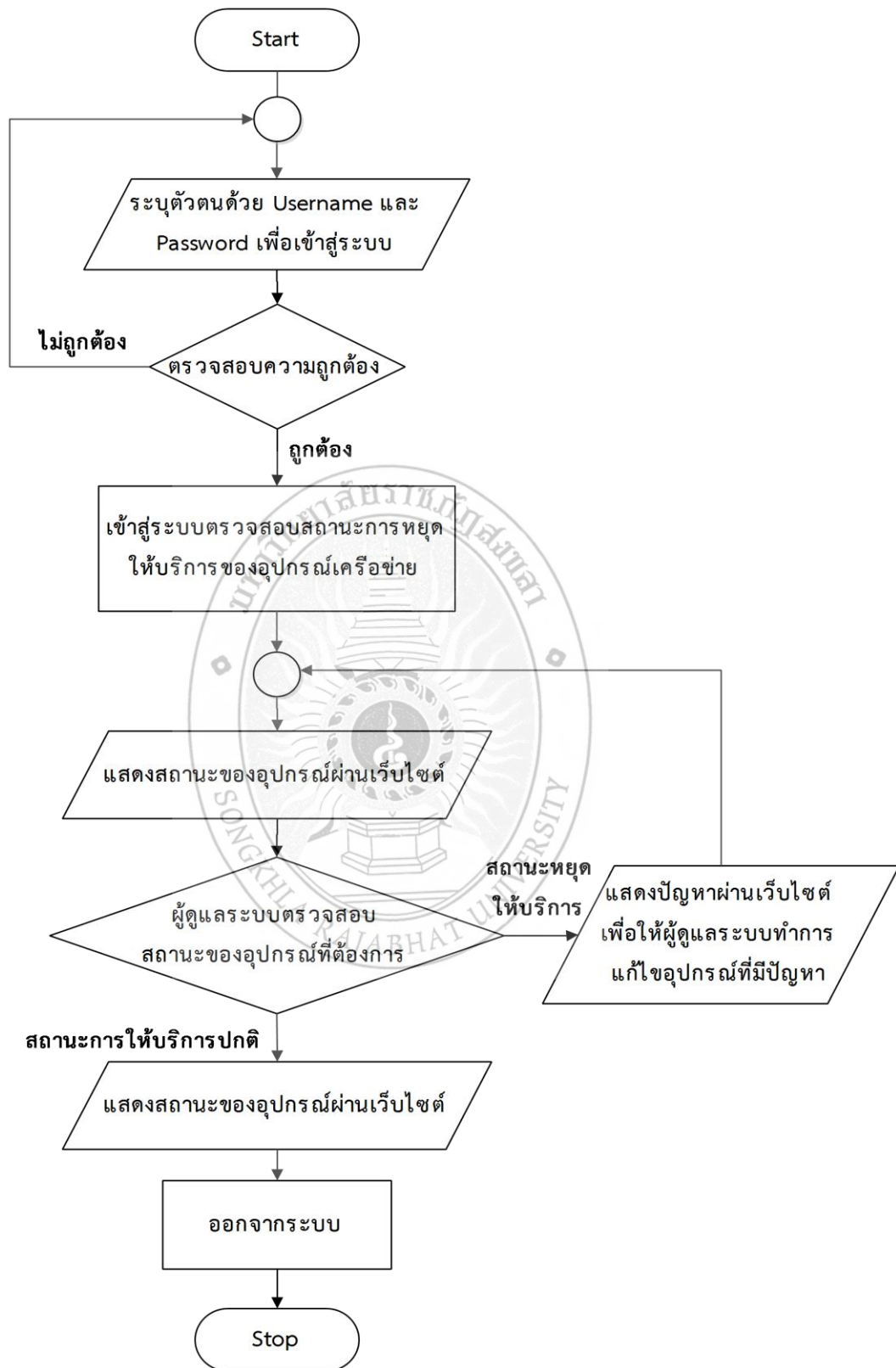
1.3 ถ้า Username และ Password ถูกต้องผู้ดูแลระบบสามารถเข้าสู่ระบบตรวจสอบสถานะของอุปกรณ์บนเครือข่าย โดยแสดงสถานะการให้บริการของเครือข่ายผ่านเว็บไซต์

1.4 ถ้าอุปกรณ์บนเครือข่ายหยุดให้บริการ ระบบจะส่งข้อความสั้น (SMS) แจ้งเตือนสถานะการหยุดให้บริการของอุปกรณ์บนเครือข่ายไปยังโทรศัพท์เคลื่อนที่ของผู้ดูแลระบบ

1.5 เมื่อผู้ดูแลระบบได้รับการแจ้งเตือน ทำให้สามารถแก้ไขอุปกรณ์เครือข่ายหยุดให้บริการได้อย่างทัน่วงที

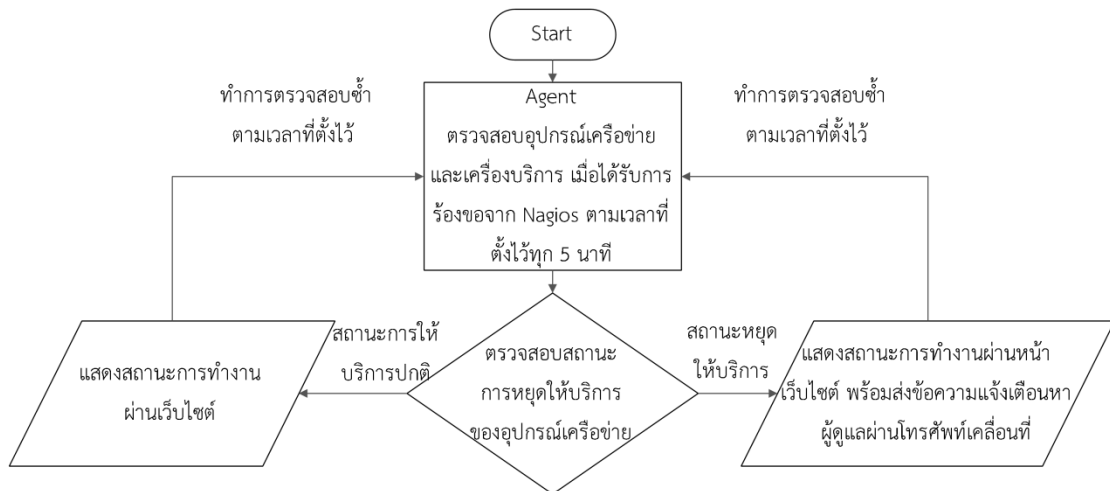
1.6 ข้อมูลสถานะอุปกรณ์เครือข่ายที่ถูกแก้ไขแล้วจะถูกแสดงสถานะบนเว็บไซต์ เช่นเดียวกับการตรวจสอบสถานะของอุปกรณ์เครือข่ายที่ไม่หยุดให้บริการ





ภาพที่ 3.1 ขั้นตอนการทำงานของระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่ายและเครื่อง  
บริการ



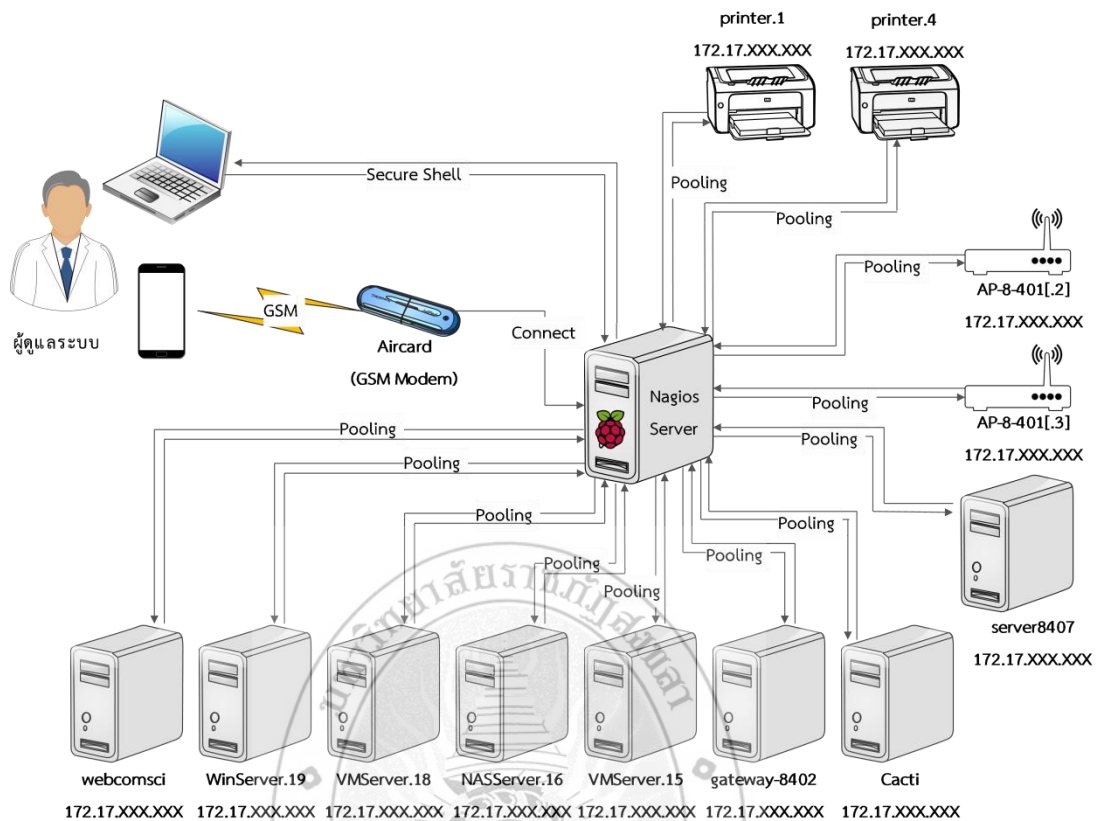


ภาพที่ 3.2 ขั้นตอนการทำงานของ Agent ในการตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย และเครื่องบริการ

ในการบริการและจัดการเครือข่ายต้องใช้อุปกรณ์ต่างๆที่มีส่วนของการทำงานร่วมกับระบบจัดการเครือข่าย ซึ่งเราเรียกว่าเอเจนต์ (Agents) เอเจนต์เป็นส่วนของซอฟต์แวร์ที่อยู่ในอุปกรณ์ต่างๆ ที่เชื่อมอยู่ในเครือข่ายโดยมีคอมพิวเตอร์หลักในระบบหนึ่งเครื่องเป็นตัวจัดการและบริหารเครือข่าย หรือเรียกว่า NMS-Network Management System

2. โครงสร้างกระบวนการทำงานของระบบตรวจสอบสถานะการให้บริการของอุปกรณ์บนเครือข่ายและเครื่องบริการ

โครงสร้างกระบวนการทำงานของระบบตรวจสอบการให้บริการของอุปกรณ์เครือข่ายและเครื่องบริการและจะส่ง SMS แจ้งเตือนเมื่อเกิดสถานะการหยุดให้บริการของอุปกรณ์เครือข่าย ผ่านระบบ GSM ไปยังโทรศัพท์เคลื่อนที่ของผู้ดูแลระบบ และผู้ดูแลระบบสามารถตรวจสอบสถานะผ่านเครือข่ายไร้สายเดียวกันได้ตลอดเวลา



ภาพที่ 3.3 โครงสร้างกระบวนการทำงานของระบบตรวจสอบสถานะการให้บริการของอุปกรณ์  
เครือข่ายและเครื่องบริการ

## 2.1 ส่วนประกอบหลักของโครงสร้างกระบวนการทำงานของระบบ

2.1.1 Nagios Server หน้าที่เป็นเครื่องบริการสำหรับตรวจสอบสถานะการทำงาน  
ของอุปกรณ์บนเครือข่ายที่ต้องการจะตรวจสอบ

2.1.2 printer.1 ทำหน้าที่สำหรับพิมพ์เอกสารในสำนักงานโปรแกรมวิชา  
คอมพิวเตอร์

2.1.3 printer.4 ทำหน้าที่สำหรับพิมพ์เอกสารในสำนักงานโปรแกรมวิชา  
คอมพิวเตอร์

2.1.4 AP-8-401[.2] ทำหน้าที่กระจายสัญญาณอินเทอร์เน็ตไร้สาย

2.1.5 AP-8-406[.3] ทำหน้าที่กระจายสัญญาณอินเทอร์เน็ตไร้สาย

2.1.6 server8407 ทำหน้าที่ในการจัดเก็บบัญชีผู้ใช้ของโปรแกรมวิชาคอมพิวเตอร์

2.1.7 gateway-8402 ทำหน้าที่ควบคุมการใช้งานเครือข่ายอินเทอร์เน็ตโปรแกรม  
วิชาคอมพิวเตอร์ในการติดต่อกับเครือข่ายภายนอก

2.1.8 VMServer.15 เครื่องบริการจำลอง ซึ่งติดตั้งระบบปฏิบัติการ Linux

2.1.9 NASServer.16 เครื่องบริการจำลอง ซึ่งได้ถูกติดตั้งระบบปฏิบัติการ Linux และ มีการให้บริการแชร์ไฟล์ (Samba)

2.1.10 VMServer.18 เครื่องบริการจำลอง ซึ่งติดตั้งระบบปฏิบัติการ Linux

2.1.11 WinServer.19 เครื่องบริการที่ถูกติดตั้งระบบปฏิบัติการ Windows Server ทำหน้าที่สำหรับรองรับ Database

2.1.12 Webcomsci ทำหน้าที่ให้บริการเว็บ และข่าวสารต่าง ๆ ของโปรแกรมวิชาคอมพิวเตอร์

2.1.13 Cacti ทำหน้าที่เป็นเครื่องตรวจสอบสถานะการให้บริการของอุปกรณ์ในเครือข่าย โดยใช้ SNMP ในการตรวจสอบสถานะ

2.1.14 ระบบ GSM เป็นมาตรฐานของระบบเครือข่ายโทรศัพท์เคลื่อนที่

2.1.15 Aricard ทำหน้าที่เป็น GSM Modem ใช้สำหรับส่งข้อความแจ้งเตือนสถานะการทำงานของอุปกรณ์ไปยังผู้ดูแลระบบ

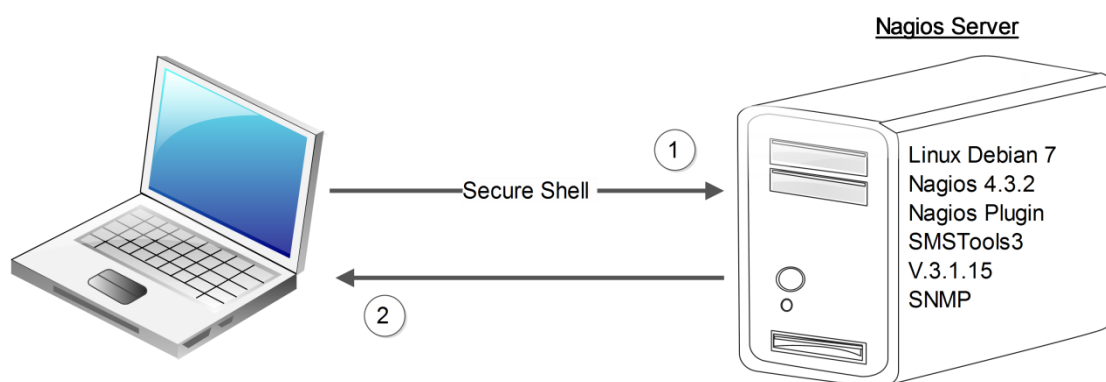
3. กระบวนการทำงานของระบบตรวจสอบสถานะการให้บริการของอุปกรณ์เครือข่ายและเครื่องบริการ ในส่วนต่าง ๆ ดังนี้

3.1 การ Remote เพื่อติดตั้ง Package บนเครื่องบริการ

3.1.1 เส้นที่ 1 แสดงการ Remote โดยใช้ Secure Shell เพื่อล็อกอินเข้าไปยัง Nagios Server

3.1.2 หลังจากนั้นทำการติดตั้ง Package Nagios 4.3.2, Nagios Plug in, SMSTools3 V.3.1.15, โพรโตคอล SNMP เพื่อทำเป็นเครื่อง Nagios Server

3.1.3 เส้นที่ 2 เป็นการดูข้อมูลการติดตั้ง Package บนเครื่องบริการ Nagios ทางหน้าจอคอมพิวเตอร์ หรือ โน้ตบุ๊ก

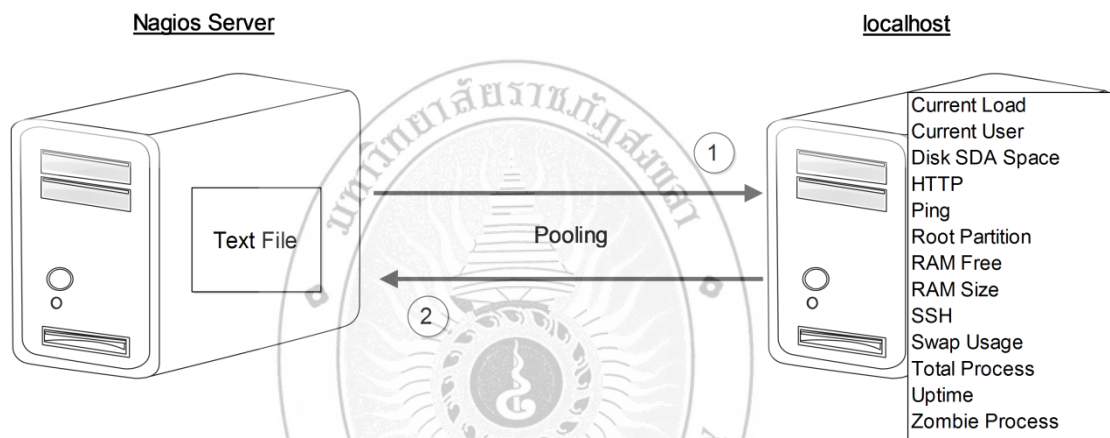


ภาพที่ 3.4 การ Remote เพื่อติดตั้ง Package บนเครื่อง Nagios Server

3.2 การตรวจสอบสถานะบน Localhost ซึ่งเป็น Nagios Server โดยใช้การ Monitoring เก็บข้อมูล

3.2.1 เส้นที่ 1 ตรวจสอบสถานะโดยใช้การ Monitoring ในเครื่อง Nagios Server เก็บข้อมูล และสถานะการทำงานของเครื่อง Nagios Server

3.2.2 เส้นที่ 2 นำข้อมูล Current Load, Current User, Disk SDA Space, HTTP, Ping, Root Partition, RAM Free, RAM Size, SSH, Swap Usage, Total Process, Uptime และ Zombie Process ที่ได้จากการตรวจสอบ มาเก็บไว้เป็น Text File ในเครื่อง Nagios Server เพื่อใช้แสดงสถานะการทำงาน



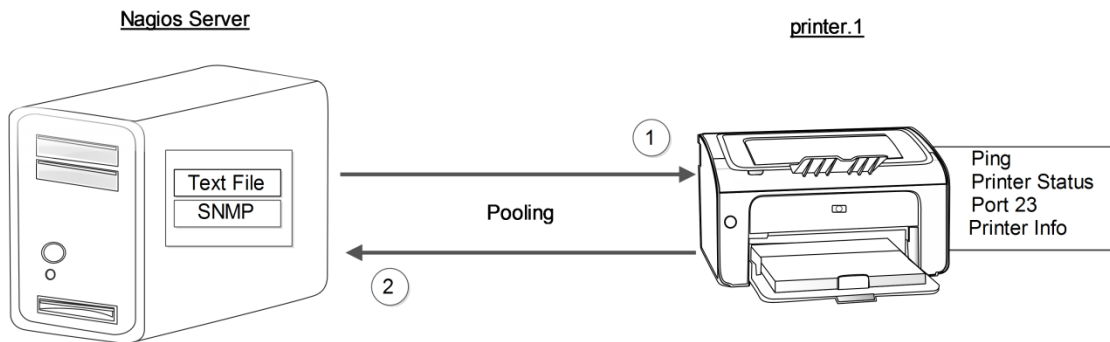
ภาพที่ 3.5 การตรวจสอบสถานะบน Localhost ซึ่งเป็น Nagios Server โดยใช้การ Monitoring

3.3 การตรวจสอบสถานะบน printer.1 ซึ่งเป็นอุปกรณ์เครือข่าย โดยการใช้ SNMP ในการเก็บข้อมูล

3.3.1 เส้นที่ 1 แสดงการตรวจสอบสถานะโดยใช้ SNMP ซึ่งทำงานอยู่บน Nagios Server

3.3.2 SNMP จะทำการส่ง Agent ไปเก็บข้อมูล และสถานะการทำงานของ printer.1 ที่มีโปรโตคอล SNMP อยู่ในอุปกรณ์

3.3.3 เส้นที่ 2 Agent จะนำข้อมูลสถานะ Ping, Printer Status, Port 23, Printer Info กลับมาเก็บไว้ในฐานข้อมูลในรูปแบบ Text File เพื่อใช้ในการแสดงสถานะการทำงาน



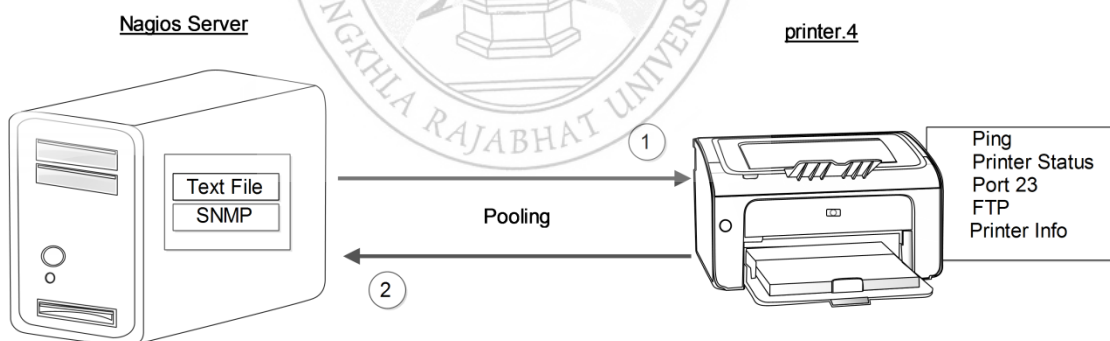
ภาพที่ 3.6 การตรวจสอบสถานะบน printer.1 ซึ่งเป็นอุปกรณ์เครือข่าย โดยใช้ โพรโตคอล SNMP

3.4 การตรวจสอบสถานะบน printer.4ซึ่งเป็นอุปกรณ์เครือข่าย โดยการใช้ SNMP ในการเก็บข้อมูล

3.4.1 เส้นที่ 1 แสดงการตรวจสอบสถานะโดยใช้ SNMP ซึ่งทำงานอยู่บน Nagios Server

3.4.2 SNMP จะทำการส่ง Agent ไปเก็บข้อมูล และสถานะการทำงานของ printer.4 ที่มีโพรโตคอล SNMP อยู่ในอุปกรณ์

3.4.3 เส้นที่ 2 Agent จะนำข้อมูลสถานะ Ping, Printer Status, Port 23, Printer Info และ FTP กลับมาเก็บไว้ในฐานข้อมูลในรูปแบบ Text File เพื่อใช้ในการแสดงสถานะการทำงาน

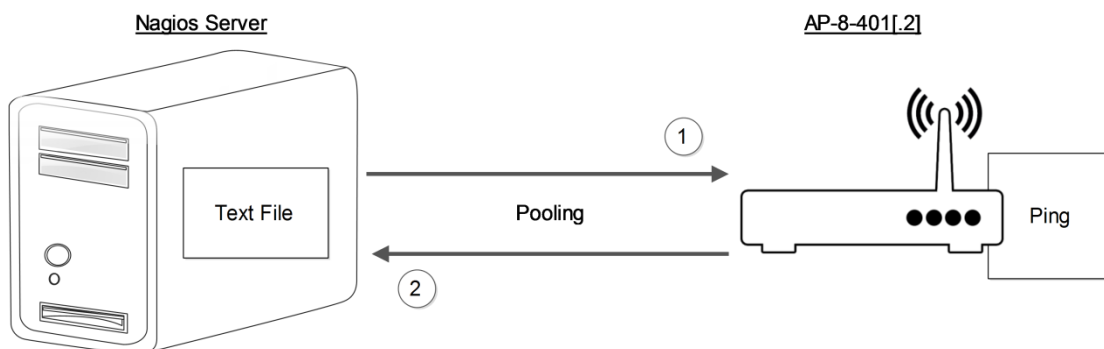


ภาพที่ 3.7 การตรวจสอบสถานะบน printer.4 ซึ่งเป็นอุปกรณ์เครือข่าย โดยใช้ โพรโตคอล SNMP

3.5 การตรวจสอบสถานะบน AP-8-401[.2] โดยใช้การ Monitoring เพื่อเก็บข้อมูล

3.5.1 เส้นที่ 1 ตรวจสอบสถานะโดยใช้การ Monitoring จากเครื่อง Nagios Server ไปตรวจสอบเครื่องโดยใช้การ Ping ไปยัง AP-8-401[.2]

3.5.2 เส้นที่ 2 จะนำข้อมูลการตอบสนอง Ping ที่ได้จากการตรวจสอบมาเก็บไว้ในฐานข้อมูลในรูปแบบ Text File ในเครื่อง Nagios Server เพื่อใช้ในการแสดงผลสถานะการทำงาน

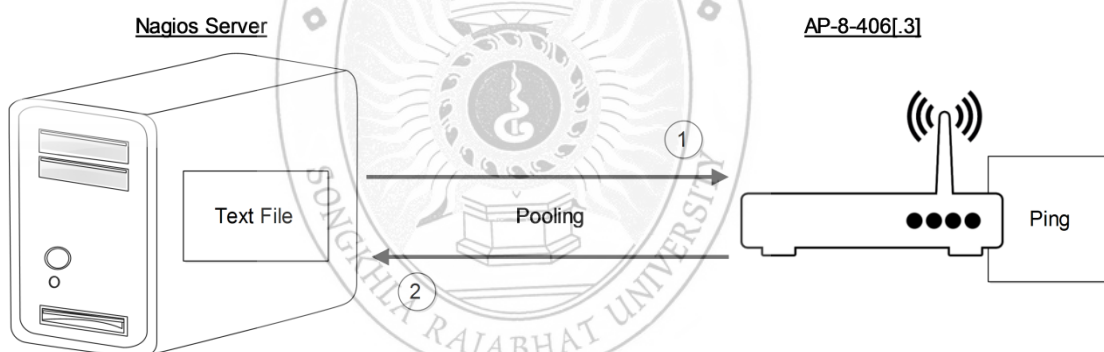


ภาพที่ 3.8 การตรวจสอบสถานะบน AP-8-401[.2] โดยใช้การ Monitoring ไปเก็บข้อมูล

3.6 การตรวจสอบสถานะบน AP-8-406[.3] โดยใช้การ Monitoring เพื่อเก็บข้อมูล

3.6.1 เส้นที่ 1 ตรวจสอบสถานะโดยใช้การ Monitoring จากเครื่อง Nagios Server ไปตรวจสอบเครื่องโดยใช้การ Ping ไปยัง AP-8-406[.3]

3.6.2 เส้นที่ 2 จะนำข้อมูลการตอบสนอง Ping ที่ได้จากการตรวจสอบมาเก็บไว้ในฐานข้อมูลในรูปแบบ Text File ในเครื่อง Nagios Server เพื่อใช้ในการแสดงผลสถานะการทำงาน

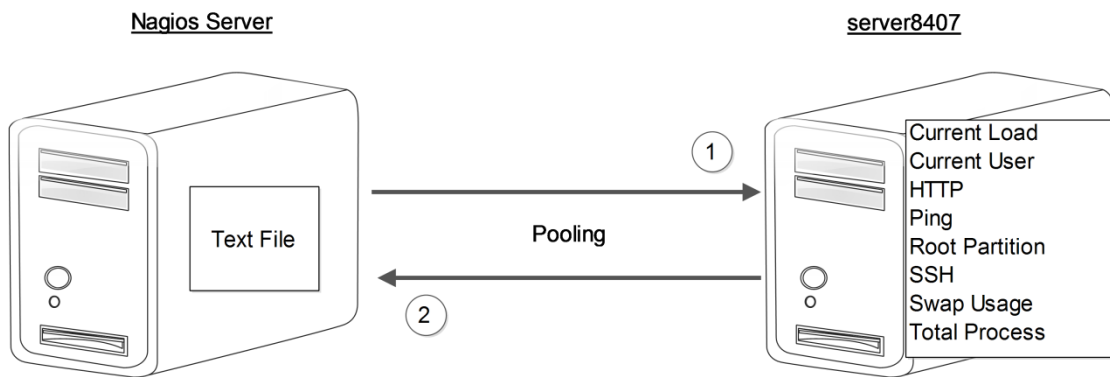


ภาพที่ 3.9 การตรวจสอบสถานะบน AP-8-406[.3] โดยใช้การ Monitoring ไปเก็บข้อมูล

3.7 การตรวจสอบสถานะบน server8407 โดยใช้การ Monitoring เพื่อเก็บข้อมูล

3.7.1 เส้นที่ 1 ตรวจสอบสถานะโดยการ Monitoring จากเครื่อง Nagios Server ไปเก็บข้อมูล และสถานะการทำงานของ server8407

3.7.2 เส้นที่ 2 นำข้อมูล Current Load, Current User, HTTP, Ping, Root Partition, SSH, Swap Usage และ Total Process ที่ได้จากการตรวจสอบ มาเก็บไว้เป็น Text File ในเครื่อง Nagios Server เพื่อใช้แสดงผลสถานะการทำงาน

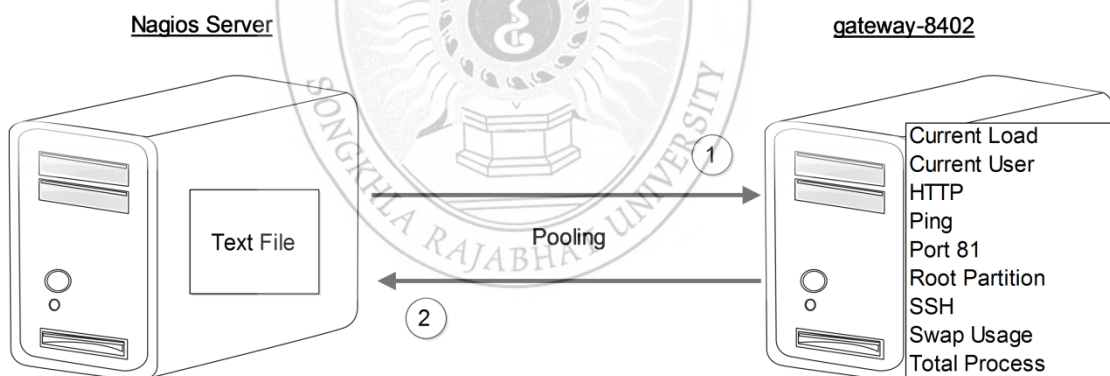


ภาพที่ 3.10 การตรวจสอบสถานะบน server8407 โดยการใช้การ Monitoring ไปเก็บข้อมูล

3.8 การตรวจสอบสถานะบน gateway-8402 โดยการใช้การ Monitoring เพื่อเก็บข้อมูล

3.8.1 เส้นที่ 1 ตรวจสอบสถานะโดยการ Monitoring จากเครื่อง Nagios Server ไปเก็บข้อมูล และสถานะการทำงานของ gateway-8402

3.8.2 เส้นที่ 2 นำข้อมูล Current Load, Current User, HTTP, Ping, Port 81, Root Partition, SSH, Swap Usage และ Total Process ที่ได้จากการตรวจสอบ มาเก็บไว้เป็น Text File ในเครื่อง Nagios Server เพื่อใช้แสดงสถานะการทำงาน



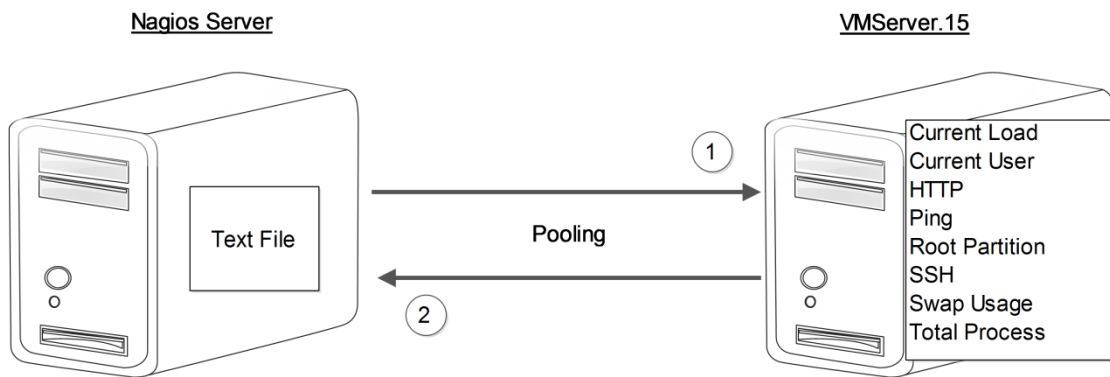
ภาพที่ 3.11 การตรวจสอบสถานะบน gateway-8402 โดยการใช้การ Monitoring ไปเก็บข้อมูล

3.9 การตรวจสอบสถานะบน VMServer.15 โดยการใช้การ Monitoring เพื่อเก็บข้อมูล

3.9.1 เส้นที่ 1 ตรวจสอบสถานะโดยการ Monitoring จากเครื่อง Nagios Server ไปเก็บข้อมูล และสถานะการทำงานของ VMServer.15

3.9.2 เส้นที่ 2 นำข้อมูล Current Load, Current User, HTTP, Ping, Root Partition, SSH, Swap Usage และ Total Process ที่ได้จากการตรวจสอบ มาเก็บไว้เป็น Text File ในเครื่อง Nagios Server เพื่อใช้แสดงสถานะการทำงาน



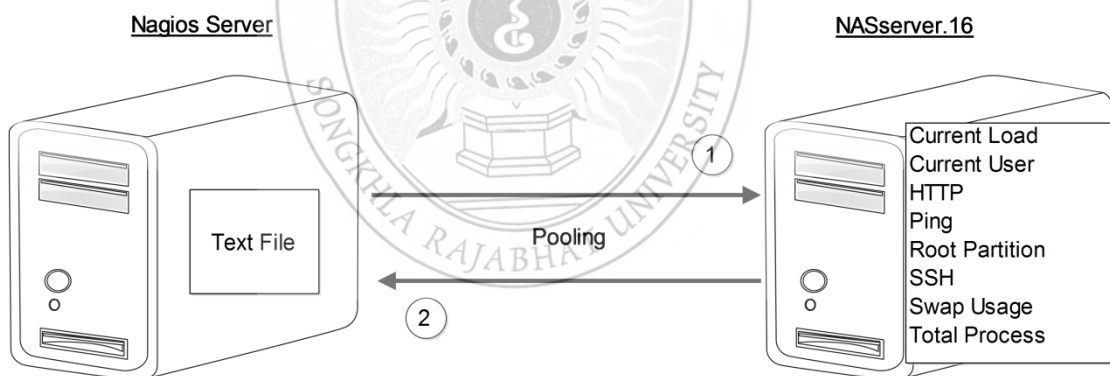


ภาพที่ 3.12 การตรวจสอบสถานะบน VMServer.15 โดยใช้การ Monitoring ไปเก็บข้อมูล

### 3.10 การตรวจสอบสถานะบน NASserver.16 โดยใช้การ Monitoring เพื่อเก็บข้อมูล

3.10.1 เส้นที่ 1 ตรวจสอบสถานะโดยการ Monitoring จากเครื่อง Nagios Server ไปเก็บข้อมูล และสถานะการทำงานของ NASserver.16

3.10.2 เส้นที่ 2 นำข้อมูล Current Load, Current User, HTTP, Ping, Root Partition, SSH, Swap Usage และ Total Process ที่ได้จากการตรวจสอบ มาเก็บไว้เป็น Text File ในเครื่อง Nagios Server เพื่อใช้แสดงสถานะการทำงาน



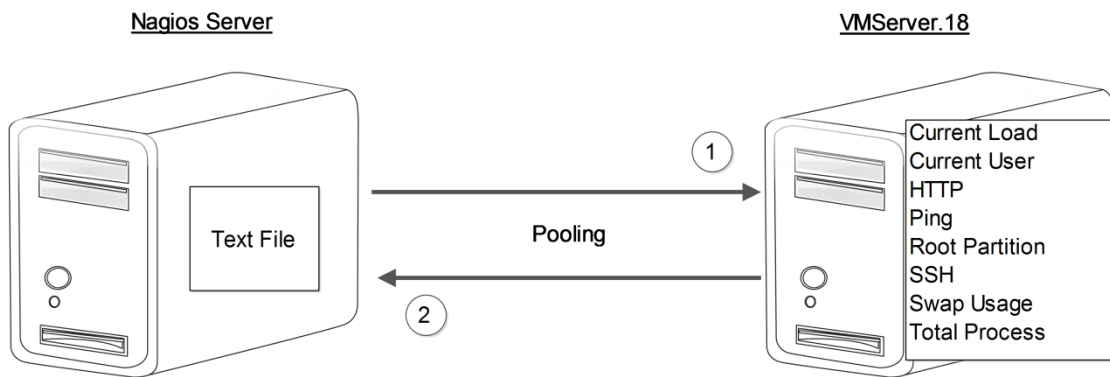
ภาพที่ 3.13 การตรวจสอบสถานะบน NASserver.16 โดยใช้การ Monitoring ไปเก็บข้อมูล

### 3.11 การตรวจสอบสถานะบน VMServer.18 โดยใช้การ Monitoring เพื่อเก็บข้อมูล

3.11.1 เส้นที่ 1 ตรวจสอบสถานะโดยการ Monitoring จากเครื่อง Nagios Server ไปเก็บข้อมูล และสถานะการทำงานของ VMServer.18

3.11.2 เส้นที่ 2 นำข้อมูล Current Load, Current User, HTTP, Ping, Root Partition, SSH, Swap Usage และ Total Process ที่ได้จากการตรวจสอบ มาเก็บไว้เป็น Text File ในเครื่อง Nagios Server เพื่อใช้แสดงสถานะการทำงาน



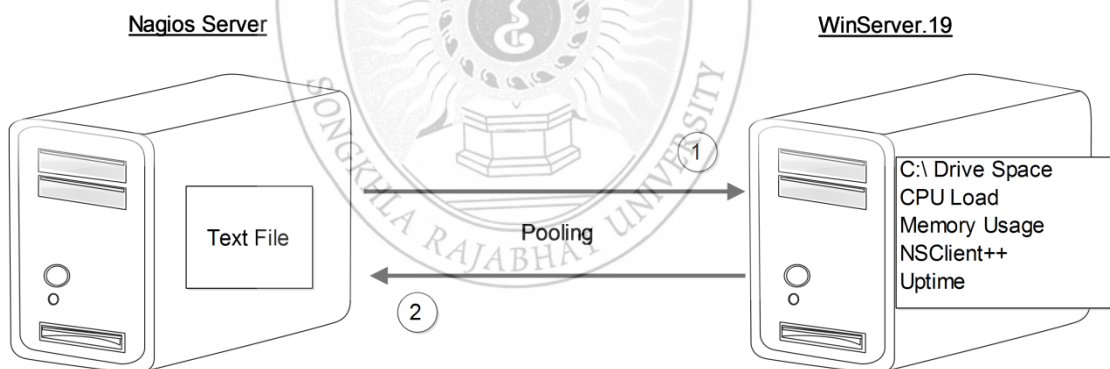


ภาพที่ 3.14 การตรวจสอบสถานะบน VMServer.18 โดยใช้การ Monitoring ไปเก็บข้อมูล

### 3.12 การตรวจสอบสถานะบน WinServer.19 โดยใช้การ Monitoring เพื่อเก็บข้อมูล

3.12.1 เส้นที่ 1 ตรวจสอบสถานะโดยการ Monitoring จากเครื่อง Nagios Server ไปเก็บข้อมูล และสถานะการทำงานของ WinServer.19

3.12.2 เส้นที่ 2 นำข้อมูล C:\ Drive Space, CPU Load, Memory Usage, NSClient++ และ Uptime ที่ได้จากการตรวจสอบ มาเก็บไว้เป็น Text File ในเครื่อง Nagios Server เพื่อใช้แสดงสถานะการทำงาน

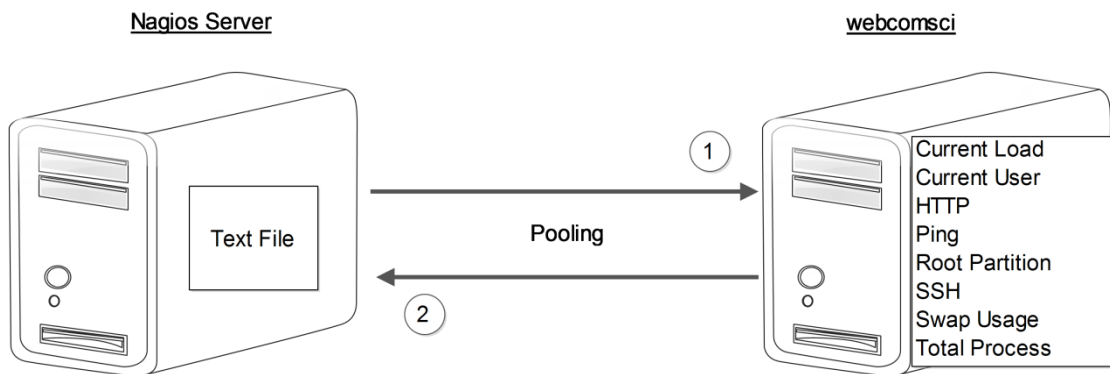


ภาพที่ 3.15 การตรวจสอบสถานะบน WinServer.19 โดยใช้การ Monitoring ไปเก็บข้อมูล

### 3.13 การตรวจสอบสถานะบน webcomsci โดยใช้การ Monitoring เพื่อเก็บข้อมูล

3.13.1 เส้นที่ 1 ตรวจสอบสถานะโดยการ Monitoring จากเครื่อง Nagios Server ไปเก็บข้อมูล และสถานะการทำงานของ webcomsci

3.13.2 เส้นที่ 2 นำข้อมูล Current Load, Current User, HTTP, Ping, Root Partition, SSH, Swap Usage และ Total Process ที่ได้จากการตรวจสอบ มาเก็บไว้เป็น Text File ในเครื่อง Nagios Server เพื่อใช้แสดงสถานะการทำงาน



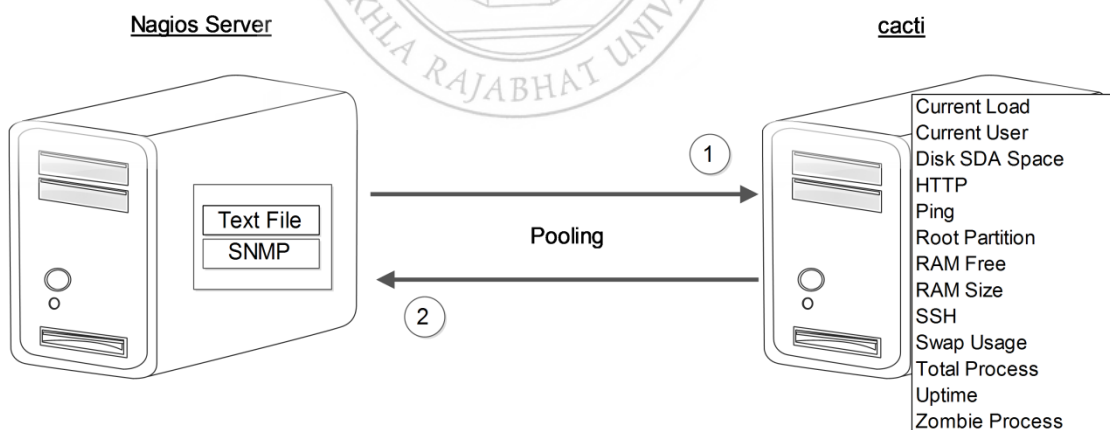
ภาพที่ 3.16 การตรวจสอบสถานะบน webcomsci โดยใช้การ Monitoring ไปเก็บ

### 3.14 การตรวจสอบสถานะบน cacti โดยใช้โปรโตคอล SNMP เพื่อเก็บข้อมูล

3.14.1 เส้นที่ 1 แสดงการตรวจสอบสถานะโดยใช้ SNMP ซึ่งทำงานอยู่บน Nagios Server

3.14.2 SNMP จะทำการส่ง Agent ไปเก็บข้อมูล และสถานะการทำงานของ cacti ที่ทำการติดตั้งและได้เปิดใช้งานโปรโตคอล SNMP อยู่ในอุปกรณ์

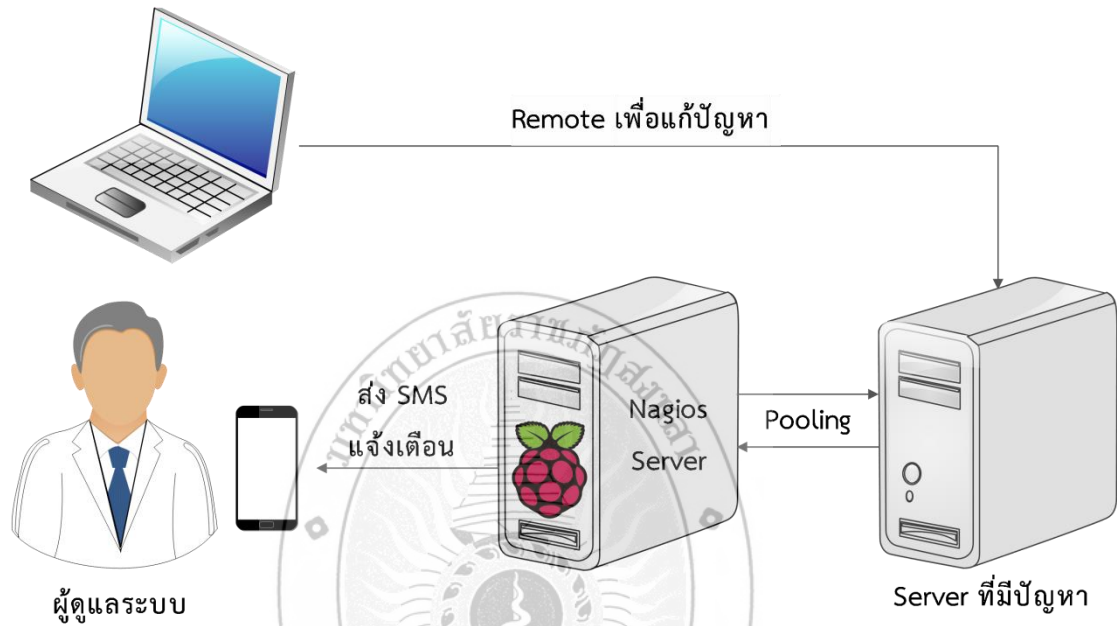
3.14.3 เส้นที่ 2 Agent จะนำข้อมูล Current Load, Current User, Disk SDA Space, HTTP, Ping, Root Partition, RAM Free, RAM Size, SSH, Swap Usage, Total Process, Uptime และ Zombie Process กลับมาเก็บไว้ในฐานข้อมูลในรูปแบบ Text File เพื่อใช้แสดงสถานะการทำงาน



ภาพที่ 3.17 การตรวจสอบสถานะของ cacti โดยใช้โปรโตคอล SNMP

4. การเข้าตรวจสอบสถานะเพื่อแก้ไขข้อมูลที่มีปัญหาการหยุดให้บริการของอุปกรณ์เครือข่ายของผู้ดูแลระบบ

4.1 ผู้ดูแลระบบสามารถแก้ไขเครื่องบริการที่มีปัญหาโดยการล็อกอินไปยังเครื่องที่มีปัญหาเพื่อตั้งค่าให้เครื่องบริการที่มีปัญหาสามารถใช้งานได้



ภาพที่ 3.18 การแก้ไขเครื่องบริการที่มีปัญหา

5. การเก็บข้อมูลในฐานข้อมูลจะเก็บเป็น Text File ซึ่งจะจัดเก็บข้อมูลสถานะเป็นตามเวลา และข้อมูลรายละเอียดต่างๆที่เกิดขึ้นกับเครื่องบริการ

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Swap by NRPE	OK	08-23-2017 14:02:13	1d 3h 16m 23s	1/3	SWAP OK - 100% free (99 MB out of 99 MB)
	Total Processes	OK	08-23-2017 14:08:29	1d 3h 15m 6s	1/4	PROCS OK: 67 processes with STATE = RSZDT
	Total Processes by NRPE	OK	08-23-2017 14:04:43	1d 3h 13m 53s	1/3	PROCS OK: 117 processes
	Uptime by SNMP	OK	08-23-2017 14:05:59	1d 3h 12m 37s	1/3	SNMP OK - 34131969
	Zombie Processus by NRPE	OK	08-23-2017 14:00:58	1d 3h 17m 38s	1/3	PROCS OK: 0 processes with STATE = Z
server6407	Current Load	OK	08-23-2017 14:08:29	11d 3h 15m 17s	1/3	OK - load average: 0.06, 0.10, 0.13
	Current Users	OK	08-23-2017 13:59:44	11d 3h 14m 25s	1/3	USERS OK - 0 users currently logged in
	HTTP	OK	08-23-2017 14:07:31	0d 8h 1m 5s	1/3	HTTP OK: HTTP/1.1 200 OK - 363 bytes in 0.002 second response time
	PING	CRITICAL	08-23-2017 14:02:18	11d 3h 9m 1s	3/3	CRITICAL - Plugin timed out
	Root Partition	OK	08-23-2017 14:03:33	1d 3h 15m 3s	1/3	DISK OK - free space: / 10321 MB (72.79% inode=87%):
	SSH	OK	08-23-2017 14:04:48	11d 3h 20m 25s	1/3	SSH OK - OpenSSH_6.4 (protocol 2.0)
	Swap Usage	OK	08-23-2017 14:06:03	1d 3h 12m 33s	1/3	SWAP OK - 100% free (99 MB out of 99 MB)
	Total Processes	OK	08-23-2017 14:07:18	1d 3h 11m 18s	1/3	PROCS OK: 66 processes with STATE = RSZDT
webcomsci	Current Load	OK	08-23-2017 13:58:34	1d 3h 10m 3s	1/3	OK - load average: 0.03, 0.08, 0.12
	Current Users	OK	08-23-2017 13:59:48	11d 3h 16m 54s	1/3	USERS OK - 0 users currently logged in
	HTTP	OK	08-23-2017 14:04:10	0d 5h 44m 26s	1/3	HTTP OK: HTTP/1.1 302 Found - 276 bytes in 0.002 second response time
	PING	CRITICAL	08-23-2017 14:02:22	11d 3h 5m 7s	3/3	CRITICAL - Plugin timed out
	Root Partition	OK	08-23-2017 14:03:37	11d 3h 14m 16s	1/3	DISK OK - free space: / 10321 MB (72.79% inode=87%):
	SSH	OK	08-23-2017 14:04:52	0d 5h 53m 44s	1/3	SSH OK - OpenSSH_5.3 (protocol 2.0)
	Swap Usage	OK	08-23-2017 14:06:07	1d 3h 12m 29s	1/3	SWAP OK - 100% free (99 MB out of 99 MB)
	Total Processes	OK	08-23-2017 14:07:24	1d 3h 11m 13s	1/3	PROCS OK: 67 processes with STATE = RSZDT

ภาพที่ 3.19 การเก็บข้อมูลเป็น Text File

## 5.1 การแสดงการเก็บข้อมูลเป็น Text File ในลักษณะการตรวจสอบการ Ping ของอุปกรณ์

[1503051972] Warning: Check of service 'Total Processes' on host 'webcomsci' timed out after 68.007s!

[1503051979] Warning: Check of service 'Current Load' on host 'NASServer.16' timed out after 61.008s!

[1503052009] Warning: Check of service 'Swap Usage' on host 'cacti' timed out after 67.185s!

[1503052024] Warning: Check of service 'Current Load' on host 'localhost' timed out after 60.009s!

[1503053464] Warning: The check of service 'PING' on host 'NASServer.16' looks like it was orphaned (results never came back; last\_check=1503052167;

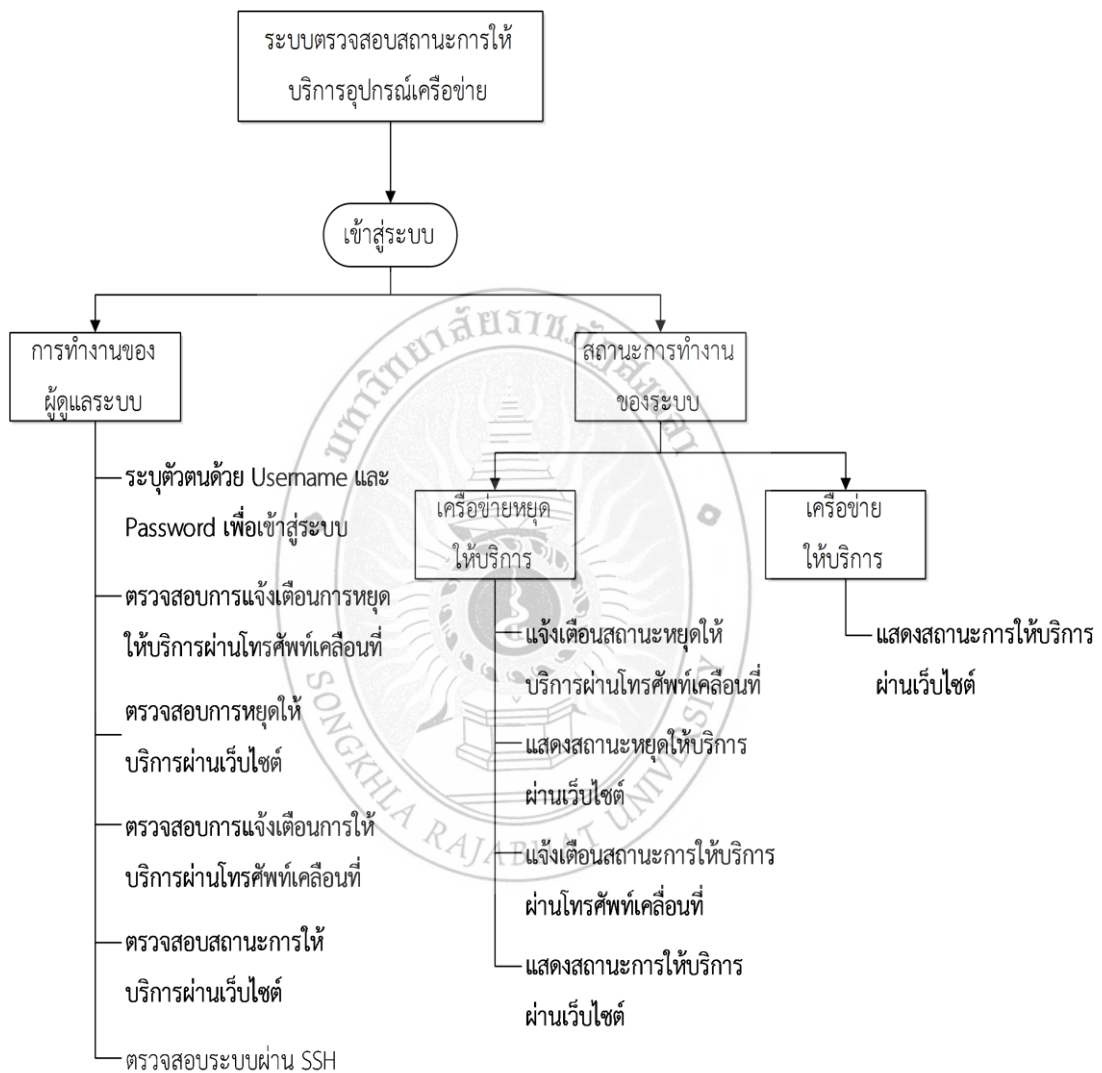
## การออกแบบระบบ

### 1.แนวคิดในการออกแบบระบบ

จากปัญหาที่ได้กล่าวไว้ข้างต้นผู้จัดทำโครงการฯจึงได้เล็งเห็นถึงการพัฒนาระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่ายและเครื่องบริการเป็นการทำงานในรูปแบบของ Client Server โดยระบบจะทำการตรวจสอบอุปกรณ์ในเครือข่ายโปรแกรมวิชาคอมพิวเตอร์ หากตรวจสอบสถานะแล้วพบว่ามึอุปกรณ์ที่มีสถานะหยุดให้บริการ ระบบจะทำการส่งข้อความสั้น (SMS) แจ้งไปยังผู้ดูแลระบบ เพื่อให้ผู้ดูแลระบบสามารถทราบปัญหาได้ทัน่วงที

## 2. โครงสร้างหน้าจอ

สามารถแสดงโครงสร้างหน้าจอ (Interface Structure Diagram) ในส่วนการออกแบบเมนู ตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่ายและเครื่องบริการ ได้ดังรูปแผนภาพต้นไม้ (Tree) แสดงได้ดังนี้



ภาพที่ 3.20 โครงสร้างหน้าจอของระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย

## ส่วนประกอบหลักของระบบ

### 1. หลักการ Monitoring ของ Nagios

การตรวจสอบว่าการทำงานของ Host ว่าทำงานอยู่นั้นก็สามารถสรุปได้ว่า Service ที่อยู่บน Host ยังเปิดให้บริการ ดังนั้น Nagios จึงตรวจสอบสถานะของ Service ในส่วนนี้ แต่การตรวจสอบจะทำได้ยากมากขึ้นถ้า Host ที่ต้องการตรวจสอบต้องติดต่อผ่านจากอุปกรณ์หลายๆประเภท เช่น Router เพราะกรณีที่ Router มีปัญหาแต่ Host และ Service ยังคงทำงานอยู่ อาจทำให้ผู้ดูแลระบบได้รับข้อสรุปที่ผิดพลาดจาก Nagios ได้ ถ้า Config Nagios ไม่ถูกต้อง

#### ขั้นตอนการตรวจสอบของ Nagios

ในการตรวจสอบเริ่มจากการตรวจสอบ Service ถ้าตรวจสอบ Service แล้วพบว่ายังเป็นปกติ Nagios จะไม่ตรวจสอบอื่นๆ แต่ถ้าตรวจสอบพบว่า Service ไม่ทำงาน Nagios จะทำการตรวจสอบ Host เพื่อหาสาเหตุว่าให้ได้ว่าที่ Service ไม่ทำงานเกิดสาเหตุมาจาก Host หรือไม่ ถ้าตรวจสอบพบว่าเป็นที่ Host ก็จะตรวจสอบที่ Parent Host อีกครั้ง เพื่อดูสถานะของ Network

#### วิธีตรวจสอบมี 2 วิธี ดังนี้

1. การตรวจสอบอย่างสม่ำเสมอ ตามช่วงเวลาที่กำหนดในตัวแปร Check\_Interval โดยวิธีการนี้ มีข้อเสียคือ ถ้ากำหนด Interval น้อยจนเกินไปจะทำให้ Nagios ส่ง Package ไปตรวจสอบบ่อย และอาจทำให้ประสิทธิภาพการทำงานลดน้อยลง ควรกำหนดให้เหมาะสม โดยตามที่ Nagios แนะนำ คือ Check\_Interval 5 หมายถึง ตรวจสอบทุก ๆ 5 นาที

2. ใช้ Ping-Based ใช้ในการตรวจสอบข้อดีก็คือ ผู้ดูแลระบบจะได้ข้อมูลเกี่ยวกับ Response Time ,Package Loss ด้วย เพื่อใช้ในการวิเคราะห์ถึงเบาะแสที่ก่อให้เกิดปัญหาได้ง่ายขึ้น

#### สถานะของ Host หรือ Service

ค่าสถานะที่ได้รับจาก Nagios มี 4 ค่า คือ 0 (OK), 1 (WARNING), 2 (CRITICAL), 3 (UNKNOWN)

- กรณี ของสถานะ UNKNOWN สาเหตุจะมาจาก Config ค่าผิด ดังนั้นถ้า Nagios คืนค่านี้มาผู้ดูแลระบบควรทำการตรวจสอบค่า Configuration ที่กำหนดไว้ในระบบอีกครั้ง

- กรณี ของสถานะ WARNING มีค่าเท่ากับ DOWN คือ Host มีสถานะการหยุดให้บริการ

- กรณี ของสถานะ CRITICAL มีค่าเท่ากับ UNREACHABLE คือ ไม่สามารถเข้าถึง Host ได้ แต่ยังไม่สามารถสรุปได้ว่า Host มีสถานะ Down เนื่องจากการตรวจสอบพบว่า Parent Down อยู่

## 2. ความปลอดภัยของ Nagios Server

Nagios Server ใช้ในการตรวจสอบอุปกรณ์เครือข่ายและบริการมีประโยชน์สำหรับผู้ดูแลระบบ แต่ในทางกลับกันอาจจะเป็นช่องโหว่ให้กับระบบได้อีกด้วย ความปลอดภัยของ Nagios Server เป็นเรื่องที่สำคัญเพราะ Nagios Server จะบรรจุด้วย Configuration Information ที่สำคัญๆขององค์กรซึ่งประกอบไปด้วย รายละเอียดของ Hosts และรายละเอียดของ Services อาทิ IP Address, Service ต่างๆรายละเอียดเหล่านี้เป็นสิ่งที่มีความสำคัญมากสำหรับ Attacker ที่จะหา Weak Point ดังนั้น Nagios Server จะต้องแน่ใจว่าจะไม่มีช่องโหว่ให้ Attacker เข้ามาบุกรุกได้

### 2.1. ข้อเสนอแนะในการดูแล Nagios Server ขั้นพื้นฐาน

ควรทำการติดตั้งเฉพาะ Package และ Components ที่จำเป็นทำการ Update ระบบ โดยการ Patching Update หรือ Workaround (Apt-Get หรือ Yum) ทำการลบ User และ Group ต่าง ๆ ที่ไม่มีความจำเป็นในการ Log In เข้ามาพร้อมกับการใช้ Password ที่มี Strong Word ทำการลบ Process Daemon หรือ Service ที่ไม่มีความจำเป็นออกจากระบบ ทำการติดตั้ง Firewall (Iptables) ให้ทำการ Handle Traffic ทั้งขาเข้าและขาออก เพิ่มระบบความปลอดภัยให้แก่ Nagios Server ด้วยการเปิดใช้ Security-Enhanced Linux (SELinux)

### 2.2. ไม่ควร Run Nagios โดยใช้ User ที่เป็น Root

Run Nagios ให้ทำงานผู้ดูแลระบบจำเป็นต้องทราบขั้นตอนการติดตั้ง Nagios ได้ทำการสร้าง User และ Group ในการทำงานไว้แล้วจึงไม่มีความจำเป็นต้องใช้ Root User ในการสั่งให้ Nagios ทำงาน

### 2.3. ควรระวังการเปิดใช้ External Commands

ต้องระมัดระวังการใช้ External Commands เนื่องจากคำสั่งนี้สามารถสั่ง Nagios Server ทำงานผ่านทางหน้า Web ของ Nagios ได้ เช่นการสั่ง การเปิด / ปิดระบบแจ้งเตือน ซึ่งโดยปกติแล้ว External\_Commands จะถูกปิดไว้ในไฟล์ Nagios Configuration ซึ่งมีคำสั่ง Check\_External\_Commands เป็นตัวควบคุม เมื่อ Check\_External\_Commands = 0 คือ การปิดการทำงานของคำสั่งนี้

### 2.4. ความปลอดภัยของ Web Console

ข้อที่ควรคำนึงถึงอีกด้านหนึ่งก็คือ ความปลอดภัยของ Web Console เนื่องจากการที่ Nagios Server จะทำการแสดงผลการตรวจสอบต่างๆออกมาผ่าน Web ทำให้ในหน้า Web Page บรรจุไปด้วยข้อมูลต่างๆที่สำคัญขององค์กรเป็นสิ่งที่มีความสำคัญมากสำหรับ Attacker อีกทั้งยังสามารถสั่งหรือควบคุม Nagios Server ได้อีกด้วย ผู้ดูแลระบบจึงควรทำการ Authentication และ Authorization เพื่อเป็นการกำหนดสิทธิการเข้าถึงและใช้งานให้กับ Nagios



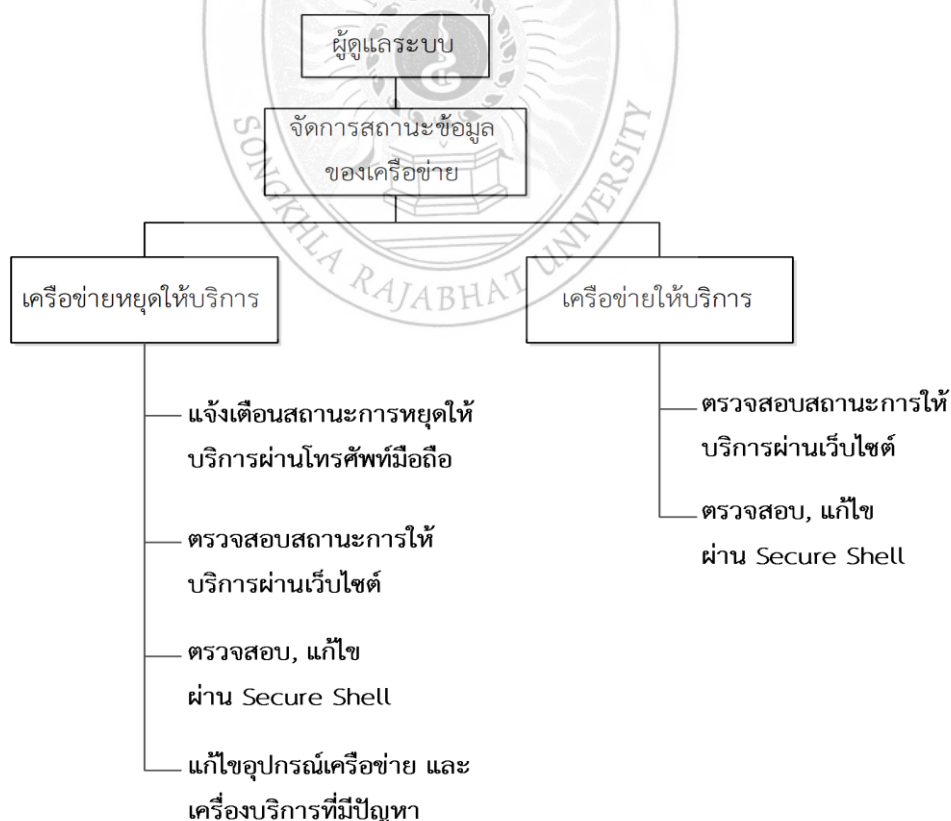
## บทที่ 4

### การพัฒนาระบบและการทดสอบ

จากการศึกษาระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย โปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา เพื่อนำมาออกแบบ และศึกษาสถานะการหยุดให้บริการของอุปกรณ์ของอุปกรณ์บนเครือข่ายที่สามารถแจ้งสถานะการทำงานไปยังผู้ดูแลระบบทันที เมื่อระบบหยุดให้บริการ ผู้จัดทำโครงการจึงได้ออกแบบ และพัฒนาระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย โปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา

#### โครงสร้างการดำเนินงาน

การออกแบบโครงสร้างการดำเนินงานของระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย ได้แบ่งโครงสร้างการดำเนินงาน โดยผู้ดูแลระบบจะเป็นผู้จัดการข้อมูลทั้งหมดของระบบ



ภาพที่ 4.1 โครงสร้างการดำเนินงานของผู้ดูแลระบบ



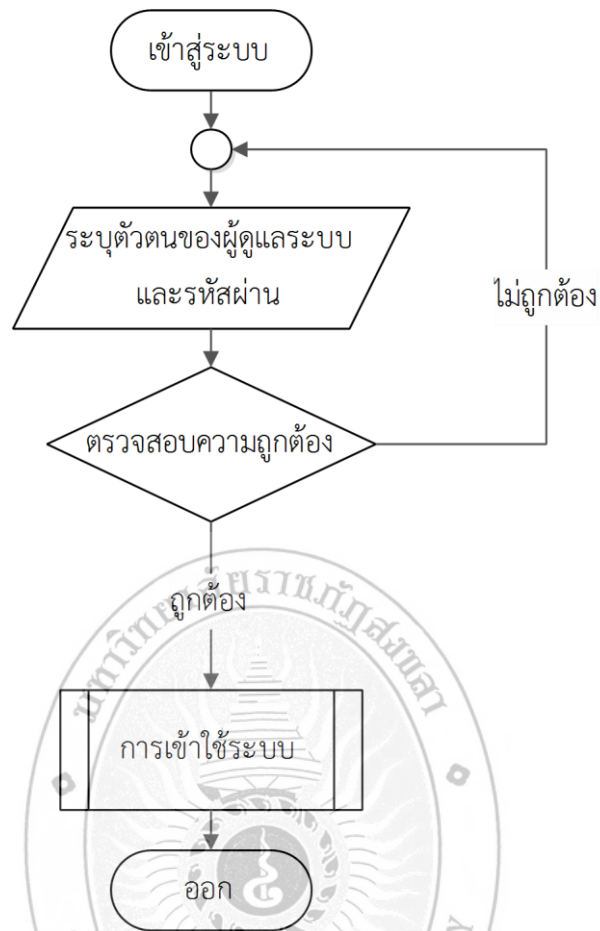
จากภาพที่ 4.1 ในการทำงานของระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย ใน ส่วนของผู้ดูแลระบบ สามารถจัดการข้อมูลของเครือข่ายดังนี้

- เครือข่ายหยุดให้บริการ ผู้ดูแลระบบทำหน้าที่ในการตรวจสอบสถานะการหยุดให้บริการ เพื่อให้ระบบสามารถกลับมาใช้งานได้ปกติ
- เครือข่ายให้บริการ ผู้ดูแลระบบทำหน้าที่เฝ้าระวังหากเครือข่ายเกิดสถานะการหยุด ให้บริการ เพื่อการแก้ไขการทำงานของอุปกรณ์เครือข่ายได้อย่างทันท่วงทีหากเกิดข้อผิดพลาด
- แจ้งเตือนสถานะการหยุดให้บริการผ่านโทรศัพท์เคลื่อนที่ ผู้ดูแลระบบจะได้รับข้อความแจ้งเตือนทันที หากอุปกรณ์ในเครือข่ายมีสถานะการหยุดให้บริการ
- ตรวจสอบสถานะการให้บริการผ่านเว็บไซต์ ผู้ดูแลระบบทำหน้าที่ตรวจสอบการทำงานของ เครือข่ายผ่านเว็บไซต์ได้ทั้งอุปกรณ์เครือข่ายที่หยุดให้บริการ และอุปกรณ์เครือข่ายที่ยังให้บริการอยู่
- ตรวจสอบ, แก้ไข ผ่าน Secure Shell ผู้ดูแลระบบทำหน้าที่ในการตรวจสอบสถานะการ ทำงานของเครือข่าย สามารถเพิ่ม ลบ แก้ไข อุปกรณ์เครือข่าย และเครื่องบริการต่าง ๆ
- แก้ไขอุปกรณ์เครือข่าย และเครื่องบริการที่มีปัญหา ผู้ดูแลระบบทำหน้าที่แก้ไขสถานะการ หยุดให้บริการของอุปกรณ์เครือข่าย และเครื่องบริการที่มีปัญหา

## หลักการทำงานของโปรแกรม

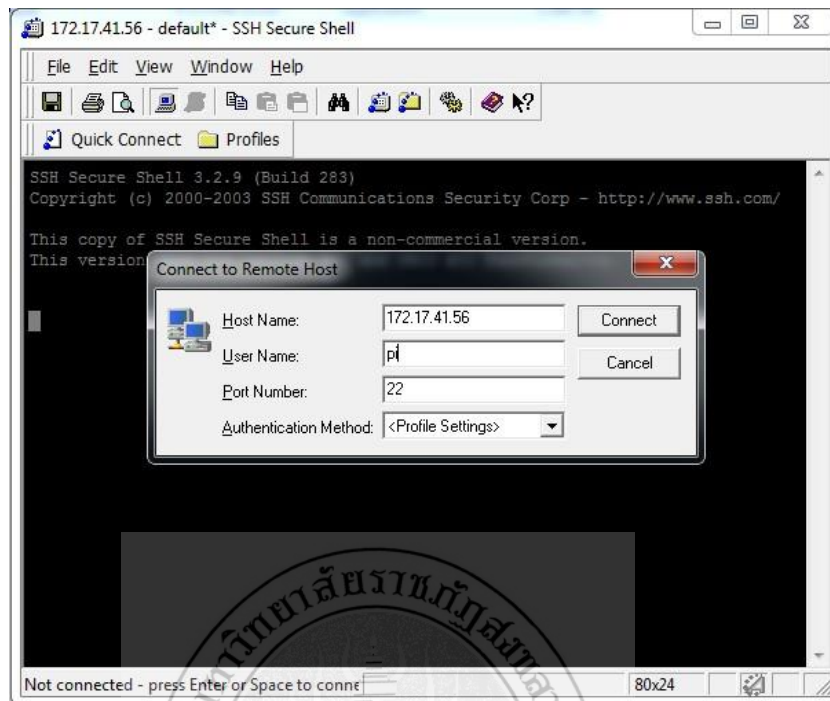
### 1. การเข้าใช้งานระบบของผู้ดูแล

จากโครงสร้างการดำเนินงานของระบบตรวจสอบสถานะการให้บริการของอุปกรณ์ เครือข่าย การดำเนินงานในแต่ละส่วนประกอบด้วย การนำข้อมูลเข้าในรูปแบบหน้าจอของโปรแกรม การดำเนินงานต่าง ๆ ตามสิทธิ์ และหน้าที่ของผู้ดูแลระบบ และการแสดงผลในรูปแบบการทำงานของ โปรแกรมในลักษณะการแสดงผ่านเว็บไซต์ หลักการทำงานของระบบในส่วนของการเข้าใช้งาน ของผู้ดูแลระบบสามารถเขียนในรูปแบบของแผนภูมิสายงาน (Flowchart) ได้ดังนี้



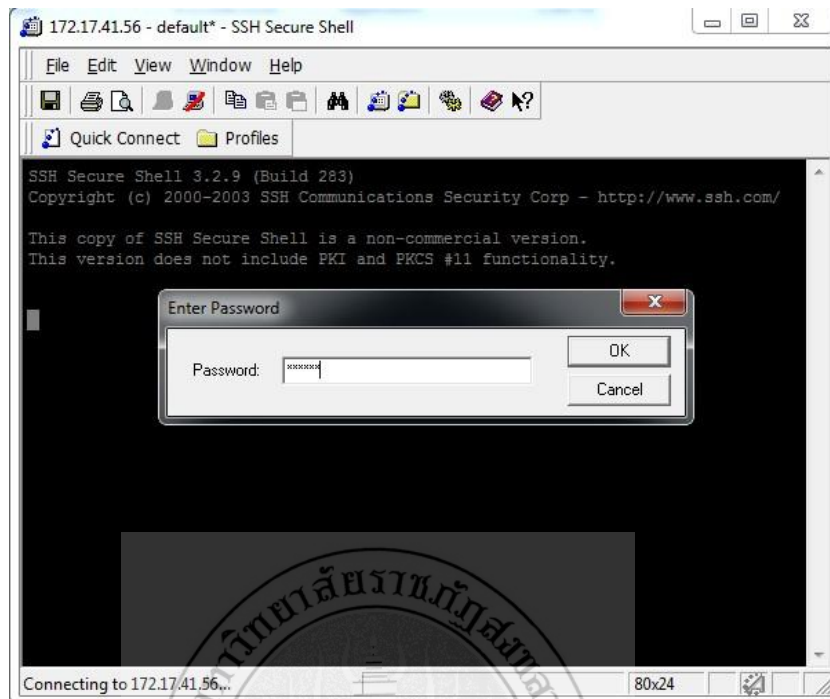
ภาพที่ 4.2 โครงสร้างการเข้าใช้งานระบบของผู้ดูแลระบบ

การทำงานของระบบในส่วนของกรเข้าใช้ระบบของผู้ดูแลระบบนั้นสามารถเข้าใช้ระบบได้โดยการ Login เข้าสู่ระบบ ซึ่งทั้งหมดของระบบผู้ดูแลระบบจะเป็นผู้จัดการข้อมูล ดังนี้ จัดการข้อมูลของเครือข่าย ตรวจสอบสถานะการทำงานของเครือข่ายทั้งเครือข่ายที่ให้บริการและหยุดให้บริการ แก้ไขอุปกรณ์ เครือข่ายที่หยุดให้บริการ เพิ่ม ลบ อุปกรณ์ที่ต้องการตรวจสอบ และแก้ไขคำสั่งการทำงานของระบบ โดยการใช้โปรแกรม Secure Shell บนคอมพิวเตอร์ และแสดงสถานะการณ์ทำงานของอุปกรณ์ที่ตรวจสอบผ่านเว็บไซต์ การ Login เข้าสู่ระบบของผู้ดูแลระบบโดยการใช้โปรแกรม Secure Shell มีดังนี้

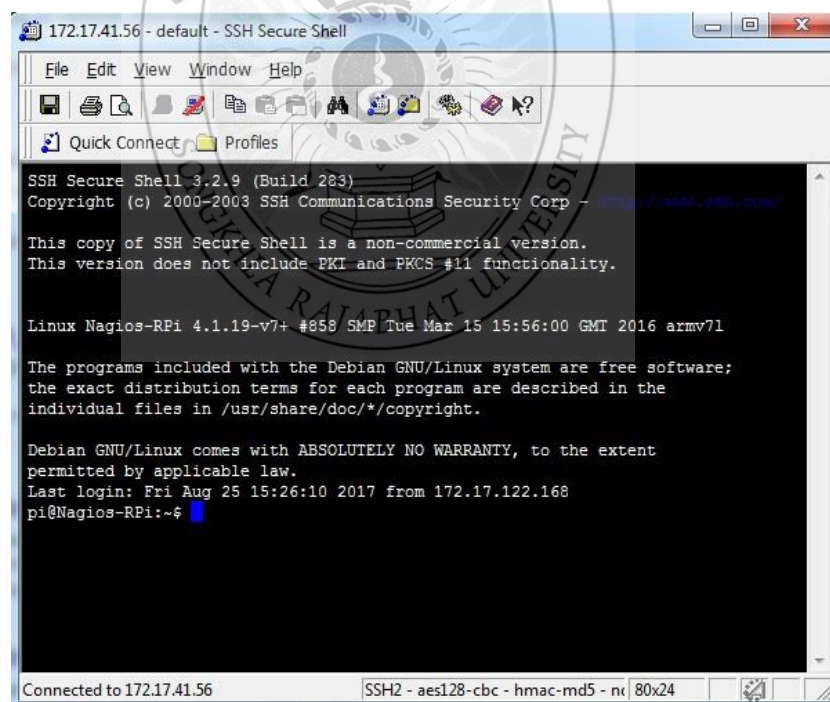


ภาพที่ 4.3 การเข้าใช้ระบบของผู้ดูแลระบบ

จากภาพที่ 4.3 การเข้าใช้ระบบของผู้ดูแลระบบในรูปแบบการใช้งานโปรแกรม Secure Shell โดยการป้อนคำสั่งการใช้งานผ่านคอมพิวเตอร์ การเข้าใช้จะต้องระบุเครื่องบริการที่เป็น Nagios Server ชื่อของผู้ดูแลระบบที่ต้องการเข้าใช้งาน และระบุการทำงานของระบบซึ่งจะทำงานในพอร์ตที่ 22 เนื่องจากมีความปลอดภัย



ภาพที่ 4.4 ป้อนรหัสผ่านเพื่อเข้าใช้งานระบบของผู้ดูแลระบบ



ภาพที่ 4.5 หน้าการทำงานของโปรแกรม Secure Shell

จากภาพที่ 4.5 หน้าการทำงานของโปรแกรม Secure Shell เพื่อใช้ในการเข้าถึงเครื่องบริการที่ใช้เป็นเครื่องตรวจสอบสถานะการทำงานของอุปกรณ์บนเครือข่าย

## 2. การติดตั้งโปรแกรมที่ใช้สำหรับตรวจสอบอุปกรณ์บนเครือข่าย ดังนี้

2.1 Nagios Core คือ ซอฟต์แวร์ประยุกต์ (Application) ที่ใช้ตรวจสอบผ่าน Web Application เพื่อใช้ตรวจสอบสถานะการทำงานของ Host และ Service ที่ต้องการ

2.2 Nagios Plugin เป็นโปรแกรมที่ใช้ร่วมกับ Nagios Server ซึ่งโปรแกรมนี้จะประกอบไปด้วยข้อมูลที่ออกแบบไว้เพื่อการ Monitoring ระบบต่าง ๆ

2.3 Net-SNMP เป็นโพรโทคอลที่อยู่ในระดับชั้นการประยุกต์ และเป็นส่วนหนึ่งของชุดโพรโทคอล TCP/IP ในการบริการ และจัดการเครือข่ายต้องใช้อุปกรณ์ต่าง ๆ มีส่วนการทำงานร่วมกับระบบจัดการเครือข่าย

2.4 PNP เป็นโอเพนซอร์สประเภทหนึ่งที่ทำางร่วมกับ RRDTOOL สำหรับสร้างกราฟเพื่อใช้สำหรับการวิเคราะห์ และติดตามการทำงานของระบบ

2.5 RRDTOOL คือ ฟังก์ชันที่ใช้ในการสร้างกราฟ ซึ่งสามารถนำไปใช้ร่วมกับ Web Server เพื่อให้ผู้ใช้งานสามารถเรียกข้อมูลกราฟผ่านทาง Web Browser ได้

2.6 SMSTools3 เป็น SMS Gateway ที่ใช้ในการส่งข้อความสั้น (SMS) โดยผ่าน GSM Modem ร่วมไปถึงโทรศัพท์ที่สามารถส่งข้อความสั้นได้ โดยใช้ Text File ที่อยู่ใน Spool Directory

2.7 NRPE คือ ส่วนเสริมของ Nagios Core ที่ใช้สำหรับการ Monitoring ซึ่งจะเสริมความสามารถในการตรวจสอบอุปกรณ์เครือข่ายได้

### การพัฒนาโปรแกรม

การทำงานของระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย พัฒนาขึ้นมาโดยการใช้ Nagios Core เป็น Package หลักในการตรวจสอบ Host และ Service โดยผู้ดูแลระบบทำการ Config ผ่าน Secure Shell โดยใช้ Shell Script เพื่อสะดวกในการเรียกใช้ และแก้ไข การทำงานของอุปกรณ์บนเครือข่ายที่ต้องการจะตรวจสอบ ต้องมีไฟล์สำคัญดังต่อไปนี้

1. ตัวอย่างไฟล์ที่ใช้ในการ Config เพื่อให้ระบบตรวจสอบสถานะการทำงานของอุปกรณ์บนเครือข่าย

คำสั่งการกำหนดค่าในไฟล์ Nagios เพื่อตรวจสอบสถานะการทำงานของอุปกรณ์บนเครือข่าย

<pre>root@Nagios-RPi:~# cd /usr/local/nagios/</pre>	<p>เป็นการเข้าไปยังไฟล์ของ Nagios ที่ได้ทำการติดตั้งไว้แล้ว เพื่อตั้งค่าให้ระบบตรวจสอบ Host และ Service</p>
---	---

<pre>root@Nagios-RPi:~# cd /usr/local/nagios/etc/objects/</pre>	<p>เป็นการเข้าไปยังโฟลเดอร์ที่เก็บไฟล์ที่ใช้สำหรับการ Config ซึ่งภายในโฟลเดอร์นี้จะมีไฟล์ที่เก็บค่าในการ Config ดังนี้ commands.cfg, contacts.cfg, localhost.cfg, printer.cfg, switch.cfg, templates.cfg, timeperiods.cfg และ windows.cfg</p>
---	---

ตารางที่ 4.1 ไฟล์ Config เพื่อตรวจสอบการให้บริการของอุปกรณ์บนเครือข่าย

ชื่อไฟล์	คำอธิบาย
contacts.cfg	เป็นไฟล์ที่ใช้สำหรับการ Config เพื่อให้ระบบติดต่อกับผู้ดูแลระบบ
commands.cfg	เป็นไฟล์ที่ใช้สำหรับการ Config คำสั่งในการเรียกใช้การงานการตรวจสอบสถานะของอุปกรณ์ที่ต้องการตรวจสอบ
localhost.cfg	เป็นไฟล์ที่ใช้สำหรับการ Config เพื่อให้ระบบตรวจสอบสถานะของอุปกรณ์บนเครือข่ายที่ใช้งานระบบปฏิบัติการ Linux
printer.cfg	เป็นไฟล์ที่ใช้สำหรับการ Config เพื่อให้ระบบตรวจสอบสถานะของอุปกรณ์บนเครือข่ายที่เป็นเครื่องพิมพ์เอกสาร
switch.cfg	เป็นไฟล์ที่ใช้สำหรับการ Config เพื่อให้ระบบตรวจสอบสถานะของอุปกรณ์บนเครือข่ายที่เป็น Switch, Router และ Access Point
templates.cfg	เป็นไฟล์ที่ใช้สำหรับเก็บตัวอย่างการ Config อุปกรณ์แต่ละประเภท
timeperiods.cfg	เป็นไฟล์ที่ใช้สำหรับการเก็บตัวอย่าง Config เวลาที่คอยกำหนดการตรวจสอบอุปกรณ์
windows.cfg	เป็นไฟล์ที่ใช้สำหรับการ Config เพื่อให้ระบบตรวจสอบสถานะของอุปกรณ์บนเครือข่ายที่เป็น Windows Server

2. ตัวอย่างไฟล์ contacts.cfg ที่ใช้สำหรับการ Config เพื่อให้ระบบส่งข้อความไปยังโทรศัพท์เคลื่อนที่ของผู้ดูแลระบบ

```
root@Nagios-RPi:~# nano /usr/local/nagios/etc/objects/contacts.cfg
```

```
ส่วนที่ 1 # CONTACTS
define contact{
    contact_name          nagiosadmin
    use                   generic-contact
    alias                 Nagios Admin
    email                 satun.finalzol@gmail.com
    pager                +66815400283
    service_notification_commands  notify-service-by-email notify-service-by-sms
    host_notification_commands    notify-host-by-email notify-host-by-sms
    service_notification_period    24x7
    host_notification_period       24x7
    service_notification_options   c,r
    host_notification_options      d,r
}

ส่วนที่ 2 CONTACT GROUPS
define contactgroup{
    contactgroup_name     admins
    alias                 Nagios Administrators
    members               nagiosadmin
}
```

ส่วนที่ 1 คือ การกำหนดค่าเพื่อเชื่อมต่อให้ระบบส่งข้อความไปยังโทรศัพท์เคลื่อนที่ของผู้ดูแลระบบ รวมถึงเวลาในการแจ้งเตือน และแจ้งเตือนเมื่อเกิดเหตุการณ์ใด

ส่วนที่ 2 คือ การกำหนดสิทธิ์ของผู้ดูแลระบบในการใช้งานระบบ

สำหรับรายละเอียดของการ Config ที่ปรากฏในตัวอย่างไฟล์ contacts.cfg สามารถอธิบายได้ดังนี้

ตารางที่ 4.2 รายละเอียดของการ Config ไฟล์ contacts.cfg ที่นำมาใช้งานในระบบ

ชื่อ	คำอธิบาย
contact_name	เป็นการกำหนดชื่อผู้ใช้ ที่เข้าใช้งานในระบบ
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าตีฟอลต์จากเทมเพลตการติดต่อทั่วไป
alias	เป็นการกำหนดชื่อเต็มของชื่อผู้ใช้
email	เป็นการกำหนดที่อยู่อีเมลสำหรับการติดต่อ
pager	เป็นการกำหนดหมายเลขโทรศัพท์ที่ต้องการติดต่อ
service_notification_commands	เป็นการกำหนด และเรียกใช้คำสั่งเพื่อให้แจ้งเตือนเกี่ยวกับ Service ผ่านช่องทางการติดต่อที่ระบุไว้ใน email และ pager
host_notification_commands	เป็นการกำหนด และเรียกใช้คำสั่งเพื่อให้แจ้งเตือนเกี่ยวกับ Host ผ่านช่องทางการติดต่อที่ระบุไว้ใน email และ pager
service_notification_period	เป็นการกำหนดเวลาที่ให้ระบบทำการแจ้งเตือนเกี่ยวกับ Service
host_notification_period	เป็นการกำหนดเวลาที่ให้ระบบทำการแจ้งเตือนเกี่ยวกับ Host
service_notification_options	เป็นการกำหนดการแจ้งเตือนเมื่อ Service มีสถานะตรงกับที่ระบุไว้
host_notification_options	เป็นการกำหนดการแจ้งเตือนเมื่อ Host มีสถานะตรงกับที่ระบุไว้
contactgroup_name	เป็นการกำหนดชื่อกลุ่มผู้ใช้ ที่เข้าใช้งานระบบ
members	เป็นการกำหนดสมาชิกที่เข้าใช้งานในระบบ

3. ตัวอย่างไฟล์ localhost.cfg ใช้สำหรับ Config เพื่อให้ระบบสามารถตรวจสอบ Host และ Service ที่ใช้ระบบปฏิบัติการ Linux

```
root@Nagios-RPi:~# nano /usr/local/nagios/etc/objects/localhost.cfg
```

```
ส่วนที่ 1 # HOST DEFINITION
```

```
define host{
    use                linux-server
```



```

host_name      localhost
alias          localhost
address        127.0.0.1
icon_image     nagios.gif
statusmap_image nagios.gd2
}

```

### ส่วนที่ 2 HOST GROUP DEFINITION

```

define hostgroup{
    hostgroup_name linux-servers
    alias          Linux Servers
    members        localhost,cacti,server8407,webcomsci,
                  gateway-8402,VMServer.18,
                  VMServer.15,NASServer.16
}

```

### ส่วนที่ 3 # SERVICE DEFINITIONS

# Define a service to "ping" the local machine

```

define service{
    use                local-service,nagiosgraph
    host_name          localhost
    service_description PING
    check_command      check_ping!100.0,20%!500.0,60%
}

```

### ส่วนที่ 4 # Define a service to check the disk space of the root partition

```

define service{
    use                local-service,nagiosgraph
    host_name          localhost
    service_description Root Partition
    check_command      check_local_disk!20%!10%!/
}

```

```
}
```

ส่วนที่ 5 Define a service to check the number of currently logged in

```
define service{
    use                local-service,nagiosgraph
    host_name          localhost
    service_description    Current Users
    check_command      check_local_users!20!50
}
```

ส่วนที่ 6 # Define a service to check the number of currently running procs

```
define service{
    use                local-service,nagiosgraph
    host_name          localhost
    service_description    Total Processes
    check_command      check_local_procs!250!400!RSZDT
}
```

ส่วนที่ 7 # Define a service to check the load on the local machine.

```
define service{
    use                local-service,nagiosgraph
    host_name          localhost
    service_description    Current Load
    check_command      check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}
```

ส่วนที่ 8 # Define a service to check the swap usage the local machine.

```
define service{
    use                local-service,nagiosgraph
    host_name          localhost
    service_description    Swap Usage
```

```

    check_command    check_local_swap!20!10
}

```

ส่วนที่ 9 # Define a service to check SSH on the local machine.

```

define service{
    use                local-service,nagiosgraph
    host_name          localhost
    service_description SSH
    check_command      check_ssh
    notifications_enabled 1
}

```

ส่วนที่ 10 # Define a service to check HTTP on the local machine.

```

define service{
    use                local-service,nagiosgraph
    host_name          localhost
    service_description HTTP
    check_command      check_http
    notifications_enabled 1
}

```

# ในส่วนที่ 11 - 15 เป็นส่วนที่ใช้การตรวจสอบโดยใช้โปรโตคอล SNMP

ส่วนที่ 11 # check Swap Size

```

define service{
    use                generic-service,nagiosgraph
    host_name          localhost
    service_description Swap Size by SNMP
    check_command      snmp_SwapSize!-C public
}

```

ส่วนที่ 12 # check Swap Free

```
define service{
    use                generic-service,nagiosgraph
    host_name          localhost
    service_description  Swap Free by SNMP
    check_command      snmp_SwapFree!-C public
}
```

ส่วนที่ 13 # check RAM Size

```
define service{
    use                generic-service,nagiosgraph
    host_name          localhost
    service_description  RAM Size by SNMP
    check_command      snmp_RamSize!-C public
}
```

ส่วนที่ 14 #check RAM Free

```
define service{
    use                generic-service,nagiosgraph
    host_name          localhost
    service_description  RAM Free by SNMP
    check_command      snmp_RamFree!-C public
}
```

ส่วนที่ 15 # check Uptime

```
define service{
    use                generic-service,nagiosgraph
    host_name          localhost
    service_description  Uptime by SNMP
    check_command      snmp_Uptime!-C public
}
```

# ในส่วนที่ 16 – 22 เป็นส่วนที่ใช้การตรวจสอบโดยใช้ NRPE

ส่วนที่ 16 # check Disk Space SDA

```
define service{
    use                generic-service,nagiosgraph
    host_name          localhost
    service_description Disk Space SDA by NRPE
    check_command      check_nrpe!check_sda
}
```

ส่วนที่ 17 # check Current Users

```
define service{
    use                generic-service,nagiosgraph
    host_name          localhost
    service_description Current Users by NRPE
    check_command      check_nrpe!check_users
}
```

ส่วนที่ 18 # check Total Processes

```
define service{
    use                generic-service,nagiosgraph
    host_name          localhost
    service_description Total Processes by NRPE
    check_command      check_nrpe!check_total_procs
}
```

ส่วนที่ 19 # check Current Load

```
define service{
    use                generic-service,nagiosgraph
    host_name          localhost
    service_description Current Load by NRPE
    check_command      check_nrpe!check_load
}
```

```
}

```

ส่วนที่ 20 # check Swap

```
define service{
    use                generic-service,nagiosgraph
    host_name          localhost
    service_description Swap by NRPE
    check_command      check_nrpe!check_swap
}

```

ส่วนที่ 21 # check Zombie Processes

```
define ervice{
    use                generic-service,nagiosgraph
    host_name          localhost
    service_description Zombie Processes by NRPE
    check_command      check_nrpe!check_zombie_procs
}

```

ส่วนที่ 22 # check Root Partition

```
define service{
    use                generic-service,nagiosgraph
    host_name          localhost
    service_description Root Partition by NRPE
    check_command      check_nrpe!check_root
}

```

# ในส่วนที่ 23 – 28 เป็นการตรวจสอบสถานะโดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ในการตรวจสอบ

ส่วนที่ 23 # CPU Load

```
define service {
    use                generic-service,nagiosgraph

```

```

host_name          Cacti
service_description Load By SNMPv1
check_command      check_snmp_load_v1!nets!4,3,3!8,5,5!-C
                  public
}

```

#### ส่วนที่ 24 # Interface

```

define service {
    use          generic-service,nagiosgraph
    host_name    Cacti
    service_description eth0 Status by SNMP
    check_command check_snmp_int_v1!-C public!eth0!-k -w
                200,400 -c 0,600
}

```

#### ส่วนที่ 25 # Memory

```

define service {
    use          generic-service,nagiosgraph
    host_name    Cacti
    service_description Memory By SNMPv1
    check_command check_snmp_mem_v1!-N!95,60!99,90!-C
                  public
}

```

#### ส่วนที่ 26 # Process

```

define service {
    use          generic-service,nagiosgraph
    host_name    Cacti
    service_description Service Crontab By SNMPv1
    check_command check_snmp_process_v1!crond!-C public!0!0
}

```

ส่วนที่ 27 # Storage Verification swap

```
define service {
    use                generic-service,nagiosgraph
    host_name          Cacti
    service_description  Verification swap linux By SNMPv1
    check_command       check_snmp_storage_v1!Swap!60!90!-C
public
}
```

ส่วนที่ 28 # Storage disk all

```
define service {
    use                generic-service,nagiosgraph
    host_name          Cacti
    service_description  Verification / By SNMPv1
    check_command       check_snmp_storage_v1!"^/$"!80!90!-C public
}
```

ส่วนที่ 1 ไฟล์ localhost.cfg กำหนด Host เพื่อเป็นการระบุอุปกรณ์บนเครือข่ายที่ต้องการตรวจสอบสถานะการทำงาน

**ตารางที่ 4.3** รายละเอียดการกำหนด Host ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะ
alias	เป็นการกำหนดนามแฝงของอุปกรณ์ที่ต้องการตรวจสอบสถานะ
address	เป็นการระบุที่อยู่ของอุปกรณ์ที่ต้องการตรวจสอบสถานะ (IP Address)



ซึ่งระบบจะทำการตรวจสอบสถานะการทำงานของอุปกรณ์บนเครือข่ายที่ใช้ระบบปฏิบัติการ Linux ดังนี้

**ตารางที่ 4.4** รายชื่ออุปกรณ์ที่ใช้ระบบปฏิบัติการ Linux ที่ระบบจะทำการตรวจสอบสถานะ

host_name	address
localhost	172.17.XXX.XXX
server8407	172.17.XXX.XXX
gateway-8402	172.17.XXX.XXX
VMServer.15	172.17.XXX.XXX
NASserver.16	172.17.XXX.XXX
VMServer.18	172.17.XXX.XXX
cacti	172.17.XXX.XXX
webcomsci	172.17.XXX.XXX

ส่วนที่ 2 ไฟล์ localhost.cfg กำหนด Host Group เพื่อจัดกลุ่มของ Host ที่เป็นประเภทเดียวกัน

```
define hostgroup{
    hostgroup_name linux-servers
    alias Linux Servers
    members localhost,cacti,server8407,webcomsci,
    gateway-8402,VMServer.18,
    VMServer.15,NASServer.16
}
```

**ตารางที่ 4.5** รายละเอียดการกำหนดค่าของ Host Group ในระบบ

ชื่อ	คำอธิบาย
hostgroup_name	เป็นการกำหนดชื่อของกลุ่ม Host
alias	เป็นการกำหนดนามแฝงของ Host Group
members	เป็นการกำหนดสมาชิกของกลุ่มที่ต้องการจะให้อยู่ใน Host Group



```

use                generic-service,nagiosgraph
host_name          VMServer.18
service_description PING
check_command      check_ping!100.0,20%!500.0,60%
}

use                generic-service,nagiosgraph
host_name          VMServer.15
service_description PING
check_command      check_ping!100.0,20%!500.0,60%
}

use                generic-service,nagiosgraph
host_name          NASServer.16
service_description PING
check_command      check_ping!100.0,20%!500.0,60%
}

```

ตารางที่ 4.6 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการ Ping ในไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะการ Ping
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะการ Ping
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะการตอบสนองการ Ping ของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 4 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบพื้นที่ว่างของฮาร์ดดิสก์ของอุปกรณ์บนเครือข่าย ดังนี้

```

use                generic-service,nagiosgraph
host_name          localhost
service_description Root Partition
check_command      check_local_disk!20%!10%!/
}

use                generic-service,nagiosgraph
host_name          cacti
service_description Root Partition
check_command      check_local_disk!20%!10%!/
}

use                generic-service,nagiosgraph
host_name          server8407
service_description Root Partition
check_command      check_local_disk!20%!10%!/
}

use                generic-service,nagiosgraph
host_name          webcomsci
service_description Root Partition
check_command      check_local_disk!20%!10%!/
}

use                generic-service,nagiosgraph
host_name          gateway-8402
service_description Root Partition
check_command      check_local_disk!20%!10%!/
}

```

```

use                generic-service,nagiosgraph
host_name          VMServer.18
service_description Root Partition
check_command      check_local_disk!20%!10%!/
}

use                generic-service,nagiosgraph
host_name          VMServer.15
service_description Root Partition
check_command      check_local_disk!20%!10%!/
}

use                generic-service,nagiosgraph
host_name          NASServer.16
service_description Root Partition
check_command      check_local_disk!20%!10%!/
}

```

**ตารางที่ 4.7** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบพื้นที่ว่างของฮาร์ดดิสก์ในไฟล์  
localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าตีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบพื้นที่ว่างของฮาร์ดดิสก์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบพื้นที่ว่างของฮาร์ดดิสก์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบพื้นที่ว่างในฮาร์ดดิสก์ ของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 5 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบจำนวนผู้ใช้งานของอุปกรณ์บนเครื่องข่าย ดังนี้

```

use                generic-service,nagiosgraph
host_name          localhost
service_description Current Users
check_command      check_local_users!20!50
}

use                generic-service,nagiosgraph
host_name          cacti
service_description Current Users
check_command      check_local_users!20!50
}

use                generic-service,nagiosgraph
host_name          server8407
service_description Current Users
check_command      check_local_users!20!50
}

use                generic-service,nagiosgraph
host_name          webcomsci
service_description Current Users
check_command      check_local_users!20!50
}

use                generic-service,nagiosgraph
host_name          gateway-8402
service_description Current Users
check_command      check_local_users!20!50
}

```

```

use                generic-service,nagiosgraph
host_name          VMServer.18
service_description Current Users
check_command      check_local_users!20!50
}

use                generic-service,nagiosgraph
host_name          VMServer.15
service_description Current Users
check_command      check_local_users!20!50
}

use                generic-service,nagiosgraph
host_name          NASServer.16
service_description Current Users
check_command      check_local_users!20!50
}

```

**ตารางที่ 4.8** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบจำนวนผู้ใช้งานของอุปกรณ์เครือข่าย ในไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบจำนวนผู้ใช้งานของอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบจำนวนผู้ใช้งานของอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบจำนวนผู้ใช้งานของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 6 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบสถานะการประมวลผลทั้งหมดของอุปกรณ์บนเครือข่าย ดังนี้

```

use                generic-service,nagiosgraph
host_name          localhost
service_description Total Processes
check_command      check_local_procs!250!400!RSZDT
}

use                generic-service,nagiosgraph
host_name          cacti
service_description Total Processes
check_command      check_local_procs!250!400!RSZDT
}

use                generic-service,nagiosgraph
host_name          server8407
service_description Total Processes
check_command      check_local_procs!250!400!RSZDT
}

use                generic-service,nagiosgraph
host_name          webcomsci
service_description Total Processes
check_command      check_local_procs!250!400!RSZDT
}

use                generic-service,nagiosgraph
host_name          gateway-8402
service_description Total Processes
check_command      check_local_procs!250!400!RSZDT
}

```



```

use                generic-service,nagiosgraph
host_name          VMServer.18
service_description Total Processes
check_command      check_local_procs!250!400!RSZDT
}

use                generic-service,nagiosgraph
host_name          VMServer.15
service_description Total Processes
check_command      check_local_procs!250!400!RSZDT
}

use                generic-service,nagiosgraph
host_name          NASServer.16
service_description Total Processes
check_command      check_local_procs!250!400!RSZDT
}

```

**ตารางที่ 4.9** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการประมวลผลทั้งหมดของอุปกรณ์เครือข่าย ในไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะการประมวลผลทั้งหมดอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะการประมวลผลทั้งหมดอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะการประมวลผลทั้งหมดของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 7 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบสถานะการโหลดของอุปกรณ์บน  
เครือข่าย ดังนี้

```

use                                generic-service,nagiosgraph
host_name                          localhost
service_description                 Current Load
check_command                       check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

use                                generic-service,nagiosgraph
host_name                          cacti
service_description                 Current Load
check_command                       check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

use                                generic-service,nagiosgraph
host_name                          server8407
service_description                 Current Load
check_command                       check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

use                                generic-service,nagiosgraph
host_name                          webcomsci
service_description                 Current Load
check_command                       check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

use                                generic-service,nagiosgraph
host_name                          gateway-8402
service_description                 Current Load
check_command                       check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

```

```

use                generic-service,nagiosgraph
host_name          VMServer.18
service_description Current Load
check_command      check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

use                generic-service,nagiosgraph
host_name          VMServer.15
service_description Current Load
check_command      check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

use                generic-service,nagiosgraph
host_name          NASServer.16
service_description Current Load
check_command      check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

```

**ตารางที่ 4.10** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการโหลดของอุปกรณ์ ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าตีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะการโหลดของอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะการโหลดของอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะการโหลดของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 8 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบสถานะการแลกเปลี่ยนการทำงานของอุปกรณ์บนเครือข่าย ดังนี้

```

use                generic-service,nagiosgraph
host_name          localhost
service_description  Swap Usage
check_command      check_local_swap!20!10
}

use                generic-service,nagiosgraph
host_name          cacti
service_description  Swap Usage
check_command      check_local_swap!20!10
}

use                generic-service,nagiosgraph
host_name          server8407
service_description  Swap Usage
check_command      check_local_swap!20!10
}

use                generic-service,nagiosgraph
host_name          webcomsci
service_description  Swap Usage
check_command      check_local_swap!20!10
}

use                generic-service,nagiosgraph
host_name          gateway-8402
service_description  Swap Usage
check_command      check_local_swap!20!10
}

```

```

use                generic-service,nagiosgraph
host_name          VMServer.18
service_description Swap Usage
check_command      check_local_swap!20!10
}

use                generic-service,nagiosgraph
host_name          VMServer.15
service_description Swap Usage
check_command      check_local_swap!20!10
}

use                generic-service,nagiosgraph
host_name          NASServer.16
service_description Swap Usage
check_command      check_local_swap!20!10
}

```

**ตารางที่ 4.11** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการแลกเปลี่ยนการทำงานของอุปกรณ์ ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะสถานะการแลกเปลี่ยนการทำงานของอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะสถานะการแลกเปลี่ยนการทำงานของอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะการแลกเปลี่ยนการทำงานของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 9 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบสถานะ SSH ของอุปกรณ์บนเครือข่าย ดังนี้

```
use          generic-service,nagiosgraph
host_name    localhost
service_description  SSH
check_command  check_ssh
notifications_enabled  1
}
```

```
use          generic-service,nagiosgraph
host_name    cacti
service_description  SSH
check_command  check_ssh
notifications_enabled  1
}
```

```
use          generic-service,nagiosgraph
host_name    server8407
service_description  SSH
check_command  check_ssh
notifications_enabled  1
}
```

```
use          generic-service,nagiosgraph
host_name    webcomsci
service_description  SSH
check_command  check_ssh
notifications_enabled  1
}
```

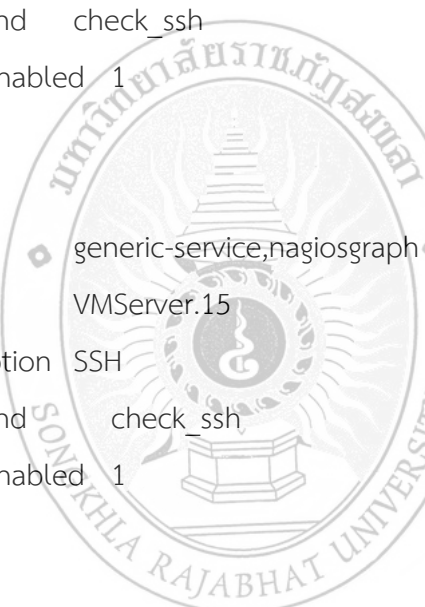
```
use          generic-service,nagiosgraph
```

```
host_name      gateway-8402
service_description SSH
check_command  check_ssh
notifications_enabled 1
}

use            generic-service,nagiosgraph
host_name      VMServer.18
service_description SSH
check_command  check_ssh
notifications_enabled 1
}

use            generic-service,nagiosgraph
host_name      VMServer.15
service_description SSH
check_command  check_ssh
notifications_enabled 1
}

use            generic-service,nagiosgraph
host_name      NASServer.16
service_description SSH
check_command  check_ssh
notifications_enabled 1
}
```

The image contains a watermark of the Sonkhla Rajabhat University logo. The logo is circular and features a central emblem with a stupa and a caduceus-like symbol. The text 'SONKHLA RAJABHAT UNIVERSITY' is written around the bottom half of the circle, and Thai text is written around the top half.

ตารางที่ 4.12 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะ SSH ของอุปกรณ์บนเครือข่าย  
ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะ SSH
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะ SSH
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะ SSH ของอุปกรณ์บนเครือข่ายที่ผู้ดูแลกำหนด
notifications_enabled	เป็นการกำหนดคำสั่งการเปิดใช้งานการแจ้งเตือน

ส่วนที่ 10 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบสถานะ HTTP ของอุปกรณ์บนเครือข่าย ดังนี้

```

use          generic-service,nagiosgraph
host_name    localhost
service_description HTTP
check_command check_http
notifications_enabled 1
}

use          generic-service,nagiosgraph
host_name    cacti
service_description HTTP
check_command check_http
notifications_enabled 1
}

use          generic-service,nagiosgraph
host_name    server8407
service_description HTTP

```




```
check_command check_http
notifications_enabled 1
}

use generic-service,nagiosgraph
host_name webcomsci
service_description HTTP
check_command check_http
notifications_enabled 1
}

use generic-service,nagiosgraph
host_name gateway-8402
service_description HTTP
check_command check_http
notifications_enabled 1
}

use generic-service,nagiosgraph
host_name VMServer.18
service_description HTTP
check_command check_http
notifications_enabled 1
}

use generic-service,nagiosgraph
host_name VMServer.15
service_description HTTP
check_command check_http
notifications_enabled 1
}
```

The image contains a large, semi-transparent watermark of the Songkhla Rajabhat University logo. The logo is circular and features a central emblem with a caduceus (a staff with two snakes) and a sunburst. The text 'SONGKHLA RAJBHAT UNIVERSITY' is written around the bottom half of the circle, and Thai text is written around the top half.

```

use          generic-service,nagiosgraph
host_name    NASServer.16
service_description HTTP
check_command check_http
notifications_enabled 1
}

```

**ตารางที่ 4.13** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะ HTTP ของอุปกรณ์บน  
เครือข่ายไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะ HTTP
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะ HTTP
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะ HTTP ของอุปกรณ์บนเครือข่ายที่ผู้ดูแลกำหนด
notifications_enabled	เป็นการกำหนดคำสั่งการเปิดใช้งานการแจ้งเตือน

ส่วนที่ 11 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบขนาดของการแลกเปลี่ยนการทำงานของอุปกรณ์บนเครือข่าย โดยใช้โปรโตคอล SNMP ดังนี้

```

use          generic-service,nagiosgraph
host_name    localhost
service_description Swap Size by SNMP
check_command snmp_SwapSize!-C public
}

use          generic-service,nagiosgraph
host_name    cacti
service_description Swap Size by SNMP
check_command snmp_SwapSize!-C public

```

```
}

```

**ตารางที่ 4.14** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบขนาดของการแลกเปลี่ยนการทำงานของอุปกรณ์บนเครือข่าย โดยใช้โปรโตคอล SNMP ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าตีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบขนาดของการแลกเปลี่ยนการทำงาน
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบขนาดของการแลกเปลี่ยนการทำงาน
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบขนาดของการแลกเปลี่ยนการทำงานของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 12 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบพื้นที่ว่างของการแลกเปลี่ยนการทำงานของอุปกรณ์บนเครือข่าย โดยใช้โปรโตคอล SNMP ดังนี้

```

use          generic-service,nagiosgraph
host_name    localhost
service_description  Swap Free by SNMP
check_command snmp_SwapFree!-C public
}

use          generic-service,nagiosgraph
host_name    cacti
service_description  Swap Free by SNMP
check_command snmp_SwapFree!-C public
}

```

**ตารางที่ 4.15** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบพื้นที่ว่างของการแลกเปลี่ยนการทำงานของอุปกรณ์บนเครือข่าย โดยใช้โปรโตคอล SNMP ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะพื้นที่ว่างของการแลกเปลี่ยนการทำงาน
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะพื้นที่ว่างของการแลกเปลี่ยนการทำงาน
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบพื้นที่ว่างของการแลกเปลี่ยนการทำงานของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 13 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบขนาดของหน่วยความจำหลักของอุปกรณ์บนเครือข่าย โดยใช้โปรโตคอล SNMP ดังนี้

use	generic-service,nagiosgraph
host_name	localhost
service_description	RAM Size by SNMP
check_command	snmp_RamSize!-C public
}	
use	generic-service,nagiosgraph
host_name	cacti
service_description	RAM Size by SNMP
check_command	snmp_RamSize!-C public
}	

**ตารางที่ 4.16** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบขนาดของหน่วยความจำหลักของอุปกรณ์บนเครือข่าย โดยใช้โปรโตคอล SNMP ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบขนาดของหน่วยความจำหลัก
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบขนาดของหน่วยความจำหลัก
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบขนาดของหน่วยความจำหลักของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 14 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบพื้นที่ว่างของหน่วยความจำหลักของอุปกรณ์บนเครือข่าย โดยใช้โปรโตคอล SNMP ดังนี้

use	generic-service,nagiosgraph
host_name	localhost
service_description	RAM Free by SNMP
check_command	snmp_RamFree!-C public
}	
use	generic-service,nagiosgraph
host_name	cacti
service_description	RAM Free by SNMP
check_command	snmp_RamFree!-C public
}	

**ตารางที่ 4.17** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบพื้นที่ว่างของหน่วยความจำหลักของอุปกรณ์บนเครือข่าย โดยใช้โปรโตคอล SNMP ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบพื้นที่ว่างของหน่วยความจำหลัก
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบพื้นที่ว่างของหน่วยความจำหลัก
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบพื้นที่ว่างของหน่วยความจำหลักของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 15 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบระยะเวลาทำงานของอุปกรณ์บนเครือข่าย โดยใช้โปรโตคอล SNMP ดังนี้

use	generic-service,nagiosgraph
host_name	localhost
service_description	Uptime by SNMP
check_command	snmp_Uptime!-C public
}	
use	generic-service,nagiosgraph
host_name	cacti
service_description	Uptime by SNMP
check_command	snmp_Uptime!-C public
}	

**ตารางที่ 4.18** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบระยะเวลาทำงานของอุปกรณ์บน  
เครือข่าย โดยใช้โปรโตคอล SNMP ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบระยะเวลาทำงาน
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบระยะเวลาทำงาน
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบระยะเวลาทำงานของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 16 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบพื้นที่ว่างฮาร์ดดิสก์ของอุปกรณ์บนเครือข่าย ดังนี้

```
define service{
    use          generic-service,nagiosgraph
    host_name    localhost
    service_description  Disk Space SDA by NRPE
    check_command check_nrpe!check_sda
}

define service{
    use          generic-service,nagiosgraph
    host_name    cacti
    service_description  Disk Space SDA by NRPE
    check_command check_nrpe!check_sda
}
```

ตารางที่ 4.19 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบพื้นที่ว่างฮาร์ดดิสก์ของอุปกรณ์บน  
เครือข่าย ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบพื้นที่ว่างฮาร์ดดิสก์ของอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบพื้นที่ว่างฮาร์ดดิสก์ของอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบพื้นที่ว่างฮาร์ดดิสก์ของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 17 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบผู้ใช้งานอุปกรณ์บนเครือข่ายโดยใช้ NRPE ดังนี้

use	generic-service,nagiosgraph
host_name	localhost
service_description	Current Users by NRPE
check_command	check_nrpe!check_users
}	
use	generic-service,nagiosgraph
host_name	cacti
service_description	Current Users by NRPE
check_command	check_nrpe!check_users
}	



**ตารางที่ 4.20** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบผู้ใช้งานอุปกรณ์บนเครือข่าย โดยใช้ NRPE ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบผู้ใช้งานอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบผู้ใช้งานอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบผู้ใช้งานของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 18 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบการประมวลผลทั้งหมดของอุปกรณ์บนเครือข่าย โดยใช้ NRPE ดังนี้

use	generic-service,nagiosgraph
host_name	localhost
service_description	Total Processes by NRPE
check_command	check_nrpe!check_total_procs
}	
use	generic-service,nagiosgraph
host_name	cacti
service_description	Total Processes by NRPE
check_command	check_nrpe!check_total_procs
}	

**ตารางที่ 4.21** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบการประมวลผลทั้งหมดของอุปกรณ์บนเครือข่าย โดยใช้ NRPE ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะการประมวลผล

ชื่อ	คำอธิบาย
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะการประมวลผลทั้งหมด
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบการประมวลผลทั้งหมดของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 19 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบการไหลของอุปกรณ์บนเครือข่าย โดยใช้ NRPE ดังนี้

```

use          generic-service,nagiosgraph
host_name    localhost
service_description    Current Load by NRPE
check_command    check_nrpe!check_load
}

use          generic-service,nagiosgraph
host_name    cacti
service_description    Current Load by NRPE
check_command    check_nrpe!check_load
}

```

**ตารางที่ 4.22** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบการไหลของอุปกรณ์บนเครือข่าย โดยใช้ NRPE ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะการไหลของอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะการไหลของอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบการไหลของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 20 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบการแลกเปลี่ยนการทำงานของอุปกรณ์บนเครือข่าย โดยใช้ NRPE ดังนี้

```

use          generic-service,nagiosgraph
host_name    localhost
service_description  Swap by NRPE
check_command  check_nrpe!check_swap
}

use          generic-service,nagiosgraph
host_name    cacti
service_description  Swap by NRPE
check_command  check_nrpe!check_swap
}

```

**ตารางที่ 4.23** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบการแลกเปลี่ยนการทำงานของอุปกรณ์บนเครือข่าย โดยใช้ NRPE ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะการแลกเปลี่ยนการทำงาน
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะการแลกเปลี่ยนการทำงาน
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบการแลกเปลี่ยนการทำงานของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 21 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบการประมวลผลที่สิ้นสุดแล้ว แต่ยังคงค้างอยู่ในระบบของอุปกรณ์บนเครือข่าย โดยใช้ NRPE ดังนี้

```

use          generic-service,nagiosgraph
host_name    localhost
service_description  Zombie Processes by NRPE

```

```

check_command      check_nrpe!check_zombie_procs
}

use                generic-service,nagiosgraph
host_name          cacti
service_description  Zombie Processes by NRPE
check_command      check_nrpe!check_zombie_procs
}

```

**ตารางที่ 4.24** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบการประมวลผลที่สิ้นสุดแล้ว แต่ยังคงค้างอยู่ในระบบของอุปกรณ์บนเครือข่าย โดยใช้ NRPE ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะการประมวลผลที่สิ้นสุดแล้ว แต่ยังคงค้างอยู่ในระบบ
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะการประมวลผลที่สิ้นสุดแล้ว แต่ยังคงค้างอยู่ในระบบ
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบการประมวลผลที่สิ้นสุดแล้ว แต่ยังคงค้างอยู่ในระบบของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 22 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบพื้นที่ว่างในฮาร์ดดิสก์ของอุปกรณ์บนเครือข่าย โดยใช้ NRPE ดังนี้

```

use                generic-service,nagiosgraph
host_name          localhost
service_description  Swap Size by SNMP
check_command      snmp_SwapSize!-C public
}

use                generic-service,nagiosgraph
host_name          cacti

```

```

service_description    Swap Size by SNMP
check_command          snmp_SwapSize!-C public
}

```

**ตารางที่ 4.25** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบพื้นที่ว่างในฮาร์ดดิสก์ของอุปกรณ์บนเครือข่าย โดยใช้ NRPE ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบพื้นที่ว่างในฮาร์ดดิสก์ของอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบพื้นที่ว่างในฮาร์ดดิสก์ของอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบพื้นที่ว่างของฮาร์ดดิสก์ของอุปกรณ์ที่ผู้ดูแลกำหนด

ส่วนที่ 23 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบสถานะการโหลดของหน่วยประมวลผลกลางของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ในการตรวจสอบ ดังนี้

```

define service {
    use                generic-service,nagiosgraph
    host_name          Cacti
    service_description    Load By SNMPv1
    check_command      check_snmp_load_v1!nets!4,3,3!8,5,5!-C public
}

```

**ตารางที่ 4.26** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการไหลของหน่วยประมวลผลกลางของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะการไหลของหน่วยประมวลผลกลางของอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะการไหลของหน่วยประมวลผลกลางของอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะการไหลของหน่วยประมวลผลกลางของอุปกรณ์

ส่วนที่ 24 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบสถานะการทำงานของ eth0 ของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ในการตรวจสอบ ดังนี้

```
define service {
    use                generic-service,nagiosgraph
    host_name          Cacti
    service_description eth0 Status by SNMP
    check_command      check_snmp_int_v1!-C public!eth0!-k -w 200,400 -c
                        0,600
}
```

**ตารางที่ 4.27** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการทำงานของ eth0 ของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะการทำงานของ eth0 ของอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะการทำงานของ eth0 ของอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะการทำงานของ eth0 ของอุปกรณ์

ส่วนที่ 25 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบสถานะการทำงานของหน่วยความจำหลักของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ในการตรวจสอบ ดังนี้

define service {	
use	generic-service,nagiosgraph
host_name	Cacti
service_description	Memory By SNMPv1
check_command	check_snmp_mem_v1!-N!95,60!99,90!-C public
}	

**ตารางที่ 4.28** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการทำงานของหน่วยความจำหลักของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะการทำงานของหน่วยความจำหลักของอุปกรณ์

ชื่อ	คำอธิบาย
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะการทำงานของหน่วยความจำหลักของอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะการทำงานของหน่วยความจำหลักของอุปกรณ์

ส่วนที่ 26 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบสถานะการทำงานของการทำงานของประมวลผลบริการของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ในการตรวจสอบ ดังนี้

```
define service {
    use                generic-service,nagiosgraph
    host_name          Cacti
    service_description Service Crontab By SNMPv1
    check_command      check_snmp_process_v1!crond!-C public!0!0
}
```

**ตารางที่ 4.29** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการทำงานของการทำงานของประมวลผลบริการของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะการทำงานของการทำงานของประมวลผลบริการของอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะการทำงานของการทำงานของประมวลผลบริการของอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะการทำงานของการทำงานของประมวลผลบริการของอุปกรณ์

ส่วนที่ 27 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบการแลกเปลี่ยนพื้นที่การทำงานของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ในการตรวจสอบ ดังนี้



```

define service {
    use                generic-service,nagiosgraph
    host_name          Cacti
    service_description  Verification swap linux By SNMPv1
    check_command       check_snmp_storage_v1!Swap!60!90!-C public
}

```

**ตารางที่ 4.30** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบการแลกเปลี่ยนพื้นที่การทำงานของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบการแลกเปลี่ยนพื้นที่การทำงานของอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบการแลกเปลี่ยนพื้นที่การทำงานของอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบการแลกเปลี่ยนพื้นที่การทำงานของอุปกรณ์

ส่วนที่ 28 ไฟล์ localhost.cfg กำหนดให้ระบบตรวจสอบพื้นที่ว่างภายใต้ / ของอุปกรณ์บนเครือข่าย โดยใช้ Perl Script ซึ่งใช้โปรโตคอล SNMP ในการตรวจสอบ ดังนี้

```

define service {
    use                generic-service,nagiosgraph
    host_name          Cacti
    service_description  Verification swap linux By SNMPv1
    service_description  Verification / By SNMPv1
    check_command       check_snmp_storage_v1!"^/$"!80!90!-C public
}

```

**ตารางที่ 4.31** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบพื้นที่ว่างภายใต้ / ของอุปกรณ์บน  
เครือข่าย โดยใช้ Perl Script ซึ่งใช้โพรโทคอล SNMP ไฟล์ localhost.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบพื้นที่ว่างภายใต้ / ของอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบพื้นที่ว่างภายใต้ / ของอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบพื้นที่ว่างภายใต้ / ของอุปกรณ์

สำหรับรายละเอียดของการ Config ที่ปรากฏในตัวอย่างไฟล์ localhost.cfg โดยยกตัวอย่างคำสั่งที่มาจากเฉพาะบรรทัดที่ไม่ถูก Comment ด้วย “#” มีดังนี้

**ตารางที่ 4.32** คำศัพท์ในไฟล์ localhost.cfg ที่ใช้ในการ Config

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะ
alias	เป็นการกำหนดนามแฝงของอุปกรณ์ที่ต้องการตรวจสอบสถานะ
address	เป็นการกำหนดที่อยู่ของอุปกรณ์ที่ต้องการตรวจสอบสถานะ
hostgroup	เป็นการกำหนดกลุ่มให้อุปกรณ์ที่ต้องการตรวจสอบสถานะ
members	เป็นการกำหนดสมาชิกกลุ่มของอุปกรณ์ที่ต้องการตรวจสอบสถานะ
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะ
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะ
notifications_enabled	เป็นการกำหนดเพื่อเปิดการแจ้งเตือนการตรวจสอบสถานะ

4. ตัวอย่างไฟล์ printer.cfg ใช้ในการ Config เพื่อให้ระบบตรวจสอบสถานะของเครื่องพิมพ์เอกสารที่อยู่ภายในโปรแกรมวิชาคอมพิวเตอร์

```
root@Nagios-RPi:~# nano /usr/local/nagios/etc/objects/printer.cfg
```

```
ส่วนที่ 1 # HOST DEFINITIONS
define host {
    use                generic-printer
    host_name          printer.4
    alias              HP LaserJet Printer 3015
    address            172.17.XXX.XXX
    hostgroups         network-printers
    icon_image         hp-printer40.png
    statusmap_image    hp-printer40.gd2
}

ส่วนที่ 2 # HOST GROUP DEFINITIONS
define hostgroup{
    hostgroup_name     network-printers
    alias              Network Printers
}

ส่วนที่ 3 # check Printer Status
define service {
    use                generic-service,nagiosgraph
    host_name          printer.4
    service_description Printer Status
    check_command      check_hpjd!-C public
    normal_check_interval 10
    retry_check_interval 1
}

ส่วนที่ 4 # check Ping
```

```

define service {
    use                generic-service,nagiosgraph
    host_name          printer.4
    service_description PING
    check_command      check_ping!3000.0,80%!5000.0,100%
    normal_check_interval 10
    retry_check_interval 1
}

```

ส่วนที่ 5 # check TCP Port 23

```

define service {
    use                generic-service,nagiosgraph
    host_name          printer.4
    service_description Port 23
    check_command      check_port!23
    normal_check_interval 10
    retry_check_interval 1
}

```

ส่วนที่ 6 # check Printer Info By SNMP

```

define service {
    use                generic-service,nagiosgraph
    host_name          printer.4
    service_description Printer Info
    check_command      check_printer_info!public!hp
    normal_check_interval 10
    retry_check_interval 1
}

```

ส่วนที่ 7 # check FTP Service

```

define service {

```

```

use          generic-service,nagiosgraph
host_name    HPLJP3015[.4]
service_description  FTP
check_command      check_ftp
normal_check_interval  10
retry_check_interval  1
}

```

ส่วนที่ 1 ไฟล์ printer.cfg กำหนด Host เพื่อเป็นการระบุเครื่องพิมพ์เอกสารบนเครือข่ายที่ต้องการตรวจสอบสถานะการทำงาน

**ตารางที่ 4.33** รายละเอียดการกำหนด Host ของเครื่องพิมพ์เอกสาร ในไฟล์ printer.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของเครื่องพิมพ์เอกสารที่ต้องการตรวจสอบสถานะ
alias	เป็นการกำหนดนามแฝงของเครื่องพิมพ์เอกสารที่ต้องการตรวจสอบสถานะ
address	เป็นการระบุที่อยู่ของเครื่องพิมพ์เอกสารที่ต้องการตรวจสอบสถานะ (IP Address)
hostgroup	เป็นการกำหนดกลุ่มของอุปกรณ์ที่เป็นเครื่องพิมพ์เอกสาร

ซึ่งระบบจะทำการตรวจสอบสถานะการทำงานของเครื่องพิมพ์เอกสารบนเครือข่าย ดังนี้

**ตารางที่ 4.34** รายชื่อเครื่องพิมพ์เอกสารที่ระบบจะทำการตรวจสอบสถานะ

host_name	address
printer.1	172.17.XXX.XXX
printer.4	172.17.XXX.XXX

ส่วนที่ 2 ไฟล์ printer.cfg กำหนด Host Group เพื่อจัดกลุ่มของ Host ที่เป็นประเภทเดียวกัน

```
define hostgroup{
    hostgroup_name    network-printers
    alias              Network Printers
}
```

ตารางที่ 4.35 รายละเอียดการกำหนดค่าของ Host Group ของเครื่องพิมพ์เอกสารในระบบ

ชื่อ	คำอธิบาย
hostgroup_name	เป็นการกำหนดชื่อของกลุ่ม Host
alias	เป็นการกำหนดนามแฝงของ Host Group

ส่วนที่ 3 ไฟล์ printer.cfg กำหนดให้ระบบตรวจสอบสถานะของเครื่องพิมพ์เอกสาร โดยการใช้โปรโตคอล SNMP ดังนี้

```
define service {
    use                generic-service,nagiosgraph
    host_name          printer.1
    service_description Printer Status
    check_command      check_hpjd!-C public
    normal_check_interval 10
    retry_check_interval 1
}
```

```
define service {
    use                generic-service,nagiosgraph
    host_name          printer.4
    service_description Printer Status
    check_command      check_hpjd!-C public
    normal_check_interval 10
    retry_check_interval 1
}
```

ตารางที่ 4.36 รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะของเครื่องพิมพ์เอกสาร โดยการ  
ใช้โปรโตคอล SNMP ไฟล์ printer.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของเครื่องพิมพ์เอกสารที่ต้องการตรวจสอบสถานะของเครื่องพิมพ์เอกสาร
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะของเครื่องพิมพ์เอกสาร
ชื่อ	คำอธิบาย
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะของเครื่องพิมพ์เอกสารที่ผู้ดูแลกำหนด
normal_check_interval	เป็นการกำหนดให้ระบบตรวจสอบสถานะทุก ๆ 10 นาที หากเครื่องพิมพ์เอกสารทำงานปกติ
retry_check_interval	เป็นการกำหนดให้ระบบตรวจสอบสถานะทุก ๆ นาทีจนกว่าจะมีการระบุสถานะได้

ส่วนที่ 4 ไฟล์ printer.cfg กำหนดให้ระบบตรวจสอบสถานะการตอบสนองการ Ping ไปยังเครื่องพิมพ์เอกสาร ดังนี้

```
define service {
    use                generic-service,nagiosgraph
    host_name          printer.1
    service_description Ping
    check_command      check_ping!3000.0,80%!5000.0,100%
    normal_check_interval 10
    retry_check_interval 1
}

define service {
    use                generic-service,nagiosgraph
```

```

host_name      printer.4
service_description  Ping
check_command  check_ping!3000.0,80%!5000.0,100%
normal_check_interval  10
retry_check_interval  1
}

```

**ตารางที่ 4.37** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการตอบสนองการ Ping ไปยังเครื่องพิมพ์เอกสาร ไฟล์ printer.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของเครื่องพิมพ์เอกสารที่ต้องการตรวจสอบสถานะการตอบสนองการ Ping ของเครื่องพิมพ์เอกสาร
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะการตอบสนองการ Ping ของเครื่องพิมพ์เอกสาร
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะการตอบสนองการ Ping ของเครื่องพิมพ์เอกสารที่ผู้ดูแลกำหนด
normal_check_interval	เป็นการกำหนดให้ระบบตรวจสอบสถานะทุก ๆ 10 นาที หากเครื่องพิมพ์เอกสารทำงานปกติ
retry_check_interval	เป็นการกำหนดให้ระบบตรวจสอบสถานะทุก ๆ นาทีจนกว่าจะมีการระบุสถานะได้

ส่วนที่ 5 ไฟล์ printer.cfg กำหนดให้ระบบตรวจสอบสถานะของพอร์ตที่ 23 ของเครื่องพิมพ์เอกสาร ดังนี้

```

define service {
    use          generic-service,nagiosgraph
    host_name    printer.1
    service_description  Port 23
    check_command  check_port!23
}

```



```

normal_check_interval    10
retry_check_interval     1
}

define service {
    use                    generic-service,nagiosgraph
    host_name              printer.4
    service_description    Port 23
    check_command          check_port!23
    normal_check_interval  10
    retry_check_interval   1
}

```

**ตารางที่ 4.38** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะของพอร์ตที่ 23 ของเครื่องพิมพ์เอกสาร ไฟล์ printer.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของเครื่องพิมพ์เอกสารที่ต้องการตรวจสอบสถานะของพอร์ตที่ 23 ของเครื่องพิมพ์เอกสาร
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะของพอร์ตที่ 23 ของเครื่องพิมพ์เอกสาร
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะของพอร์ตที่ 23 ของเครื่องพิมพ์เอกสารที่ผู้ดูแลกำหนด
normal_check_interval	เป็นการกำหนดให้ระบบตรวจสอบสถานะทุก ๆ 10 นาที หากเครื่องพิมพ์เอกสารทำงานปกติ
retry_check_interval	เป็นการกำหนดให้ระบบตรวจสอบสถานะทุก ๆ นาทีจนกว่าจะมีการระบุสถานะได้

ส่วนที่ 6 ไฟล์ printer.cfg กำหนดให้ระบบตรวจสอบข้อมูล และรายละเอียดของเครื่องพิมพ์เอกสาร โดยใช้โปรโตคอล SNMP ดังนี้

```
define service {
    use                generic-service,nagiosgraph
    host_name          printer.1
    service_description Printer Info V.2
    check_command      check_printer_info2!v2!hp
    normal_check_interval 10
    retry_check_interval 1
}

define service {
    use                generic-service,nagiosgraph
    host_name          printer.4
    service_description Printer Info
    check_command      check_printer_info!public!hp
    normal_check_interval 10
    retry_check_interval 1
}
```

**ตารางที่ 4.39** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบข้อมูล และรายละเอียดของเครื่องพิมพ์เอกสาร โดยใช้โปรโตคอล SNMP ไฟล์ printer.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของเครื่องพิมพ์เอกสารที่ต้องการตรวจสอบข้อมูล และรายละเอียดของเครื่องพิมพ์เอกสาร
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจข้อมูล และรายละเอียดของเครื่องพิมพ์เอกสาร

ชื่อ	คำอธิบาย
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบข้อมูล และรายละเอียดของเครื่องพิมพ์เอกสารที่ผู้ดูแลกำหนด
normal_check_interval	เป็นการกำหนดให้ระบบตรวจสอบสถานะทุก ๆ 10 นาที หากเครื่องพิมพ์เอกสารทำงานปกติ
retry_check_interval	เป็นการกำหนดให้ระบบตรวจสอบสถานะทุก ๆ นาทีจนกว่าจะมีการระบุสถานะได้

ส่วนที่ 7 ไฟล์ printer.cfg กำหนดให้ระบบตรวจสอบสถานะการทำงานของบริการ FTP ของเครื่องพิมพ์เอกสาร ดังนี้

```
define service {
    use                generic-service,nagiosgraph
    host_name          printer.4
    service_description  FTP
    check_command      check_ftp
    normal_check_interval 10
    retry_check_interval 1
}
```

**ตารางที่ 4.40** รายละเอียดการกำหนดคำสั่งสำหรับตรวจสอบสถานะการทำงานของบริการ FTP ของเครื่องพิมพ์เอกสาร ไฟล์ printer.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของเครื่องพิมพ์เอกสารที่ต้องการตรวจสอบสถานะบริการ FTP
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะบริการ FTP
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะบริการ FTP ของเครื่องพิมพ์เอกสารที่ผู้ดูแลกำหนด

ชื่อ	คำอธิบาย
normal_check_interval	เป็นการกำหนดให้ระบบตรวจสอบสถานะทุก ๆ 10 นาที หากเครื่องพิมพ์เอกสารทำงานปกติ
retry_check_interval	เป็นการกำหนดให้ระบบตรวจสอบสถานะทุก ๆ นาทีจนกว่าจะมีการระบุสถานะได้

สำหรับรายละเอียดของการ Config ที่ปรากฏในตัวอย่างไฟล์ printer.cfg โดยยกตัวอย่าง คำศัพท์มาเฉพาะบรรทัดที่ไม่ถูก Comment ด้วย “#” มีดังนี้

**ตารางที่ 4.41** คำศัพท์ในไฟล์ printer.cfg ที่ใช้ในการ Config

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของเครื่องพิมพ์เอกสารที่ต้องการตรวจสอบสถานะ
alias	เป็นการกำหนดนามแฝงของเครื่องพิมพ์เอกสารที่ต้องการตรวจสอบสถานะ
address	เป็นการกำหนดที่อยู่ของเครื่องพิมพ์เอกสารที่ต้องการตรวจสอบสถานะ
hostgroup	เป็นการกำหนดกลุ่มให้เครื่องพิมพ์เอกสารที่ต้องการตรวจสอบสถานะ
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะ
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะ
normal_check_interval	เป็นการกำหนดเวลาที่จะให้ระบบตรวจสอบสถานะ
retry_check_interval	เป็นการกำหนดเวลาที่จะให้ระบบตรวจสอบซ้ำเมื่ออุปกรณ์มีสถานะผิดปกติ

5. ตัวอย่างไฟล์ commands.cfg ใช้ในการ Config เพื่อให้ระบบเรียกคำสั่งต่าง ๆ เพื่อการตรวจสอบสถานะ

```
root@Nagios-RPi:~# nano /usr/local/nagios/etc/objects/commands.cfg
```

```
# คำสั่งตรวจสอบการมีอยู่ของ Host
# 'check-host-alive' command definition
define command{
    command_name    check-host-alive
```

```

command_line    $USER1$/check_ping -H $HOSTADDRESS$ -w
                3000.0,80% -c 5000.0,100% -p 5
    }

# คำสั่งตรวจสอบพื้นที่ว่างของฮาร์ดดิสก์
# 'check_local_disk' command definition
define command{
    command_name    check_local_disk
    command_line    $USER1$/check_disk -w $ARG1$ -c $ARG2$ -p $ARG3$
}

# คำสั่งตรวจสอบการไหลของ CPU
# 'check_local_load' command definition
define command{
    command_name    check_local_load
    command_line    $USER1$/check_load -w $ARG1$ -c $ARG2$
}

# คำสั่งตรวจสอบการประมวลผล
# 'check_local_procs' command definition
define command{
    command_name    check_local_procs
    command_line    $USER1$/check_procs -w $ARG1$ -c $ARG2$ -s
                    $ARG3$
}

# คำสั่งตรวจสอบผู้ใช้
# 'check_local_users' command definition
define command{
    command_name    check_local_users
    command_line    $USER1$/check_users -w $ARG1$ -c $ARG2$
}

```

```

    }

# คำสั่งตรวจสอบการแลกเปลี่ยนการทำงาน
# 'check_local_swap' command definition
define command{
    command_name    check_local_swap
    command_line    $USER1$/check_swap -w $ARG1$ -c $ARG2$
}

# คำสั่งตรวจสอบพอร์ตการทำงานตามที่ใช้กำหนด
# Check Port Open command definition
define command{
    command_name    check_port
    command_line    /usr/local/nagios/libexec/check_port.pl -p $ARG1$ -h
                    $HOSTADDRESS$ -c 1.5 -w 1.0 -v
}

# คำสั่งตรวจสอบ NRPE
# .check_nrpe. command definition
define command{
    command_name    check_nrpe
    command_line    /usr/lib/nagios/plugins/check_nrpe -H
                    $HOSTADDRESS$ -t 30 -c $ARG1$
}

# คำสั่งตรวจสอบ Service FTP
# 'check_ftp' command definition
define command{
    command_name    check_ftp
    command_line    $USER1$/check_ftp -H $HOSTADDRESS$ $ARG1$
}

```

```
# คำสั่งตรวจสอบเครื่องพิมพ์เอกสารของ HP
# 'check_hpjd' command definition
define command{
    command_name    check_hpjd
    command_line     $USER1$/check_hpjd -H $HOSTADDRESS$ $ARG1$
}

# คำสั่งตรวจสอบ โดยใช้โปรโตคอล SNMP
# 'check_snmp' command definition
define command{
    command_name    check_snmp
    command_line     $USER1$/check_snmp -H $HOSTADDRESS$ $ARG1$
}

# คำสั่งตรวจสอบข้อมูลของเครื่องพิมพ์เอกสาร โดยใช้โปรโตคอล SNMP กรณีที่เป็น V.1
# 'check printer info' command definition BY SNMP V1 or Public
define command{
    command_name    check_printer_info
    command_line     /usr/local/nagios/libexec/check_printer_info.pl -H
                    $HOSTADDRESS$ -C $ARG1$ -T $ARG2$
}

# คำสั่งตรวจสอบข้อมูลของเครื่องพิมพ์เอกสาร โดยใช้โปรโตคอล SNMP กรณีที่เป็น V.2 และ V.3
# 'check pinter info' command definition BY SNMP V2 or V3
define command{
    command_name    check_printer_info2
    command_line     /usr/local/nagios/libexec/check_printer_info2.pl -H
                    $HOSTADDRESS$ -$ARG1$ -T $ARG2$
}
```

```
# คำสั่งตรวจสอบขนาดของการแลกเปลี่ยนการทำงาน โดยใช้โปรโตคอล SNMP
define command{
    command_name    snmp_SwapSize
    command_line    $USER1$/check_snmp -o .1.3.6.1.4.1.2021.4.3.0 -H
                    $HOSTADDRESS$ $ARG1$
}

# คำสั่งตรวจสอบพื้นที่ว่างของการแลกเปลี่ยนการทำงาน โดยใช้โปรโตคอล SNMP
define command{
    command_name    snmp_SwapFree
    command_line    $USER1$/check_snmp -o .1.3.6.1.4.1.2021.4.4.0 -H
                    $HOSTADDRESS$ $ARG1$
}

# คำสั่งตรวจสอบขนาดของ RAM โดยใช้โปรโตคอล SNMP
define command{
    command_name    snmp_RamSize
    command_line    $USER1$/check_snmp -o .1.3.6.1.4.1.2021.4.5.0 -H
                    $HOSTADDRESS$ $ARG1$
}

# คำสั่งตรวจสอบพื้นที่ว่างของ RAM โดยใช้โปรโตคอล SNMP
define command{
    command_name    snmp_RamFree
    command_line    $USER1$/check_snmp -o .1.3.6.1.4.1.2021.4.11.0 -H
                    $HOSTADDRESS$ $ARG1$
}

# คำสั่งตรวจสอบเวลาทั้งหมดที่อุปกรณ์ทำงาน โดยใช้โปรโตคอล SNMP
define command{
    command_name    snmp_Uptime
```



```

    command_line    $USER1$/check_snmp -o .1.3.6.1.2.1.1.3.0 -H
                    $HOSTADDRESS$ $ARG1$
}

# คำสั่งตรวจสอบการไหลการทำงานของ โดยใช้โพรโทคอล SNMP V.1
# check_snmp_load_v1 : using snmpv1 (or v2 depending on $USER7$ value)
define command{
    command_name    check_snmp_load_v1
    command_line    $USER1$/check_snmp_load.pl -H $HOSTADDRESS$
                    $USER7$ -T $ARG1$ -w $ARG2$ -c $ARG3$ $ARG4$
}

# คำสั่งตรวจสอบการไหลการทำงานของ โดยใช้โพรโทคอล SNMP V.3
# check_snmp_load_v3 : using snmpv3
define command{
    command_name    check_snmp_load_v3
    command_line    $USER1$/check_snmp_load.pl -H $HOSTADDRESS$
                    $USER8$ -T $ARG1$ -w $ARG2$ -c $ARG3$ $ARG4$
}

# คำสั่งตรวจสอบ Interface โดยใช้โพรโทคอล SNMP V.1
# check_snmp_int_v1 : using snmpv1
define command{
    command_name    check_snmp_int_v1
    command_line    $USER1$/check_snmp_int.pl -H $HOSTADDRESS$
                    $USER7$ -n $ARG1$ $ARG2$
}

# คำสั่งตรวจสอบ Interface โดยใช้โพรโทคอล SNMP V.3
# check_snmp_int_v3 : using snmpv3
define command{

```

```

    command_name    check_snmp_int_v3
    command_line    $USER1$/check_snmp_int.pl -H $HOSTADDRESS$
                   $USER8$ -n $ARG1$ $ARG2$
}

# คำสั่งตรวจสอบหน่วยความจำหลัก โดยใช้โปรโตคอล SNMP V.1
# check_snmp_mem V.1
define command{
    command_name    check_snmp_mem_v1
    command_line    $USER1$/check_snmp_mem.pl -H $HOSTADDRESS$
                   $USER7$ $ARG1$ -w $ARG2$ -c $ARG3$ $ARG4$
}

# คำสั่งตรวจสอบหน่วยความจำหลัก โดยใช้โปรโตคอล SNMP V.1
# check_snmp_mem V.3
define command{
    command_name    check_snmp_mem_v3
    command_line    $USER1$/check_snmp_mem.pl -H $HOSTADDRESS$
                   $USER8$ $ARG1$ -w $ARG2$ -c $ARG3$ $ARG4$
}

# คำสั่งตรวจสอบการประมวลผล โดยใช้โปรโตคอล SNMP V.1
# check_snmp_process
define command{
    command_name    check_snmp_process_v1
    command_line    $USER1$/check_snmp_process.pl -H
                   $HOSTADDRESS$ $USER7$ -n $ARG1$ -w $ARG2$ -c
                   $ARG3$ $ARG4$
}

# คำสั่งตรวจสอบหน่วยเก็บข้อมูล โดยใช้โปรโตคอล SNMP V.1

```

```
#check_snmp_storage V1
define command{
    command_name    check_snmp_storage_v1
    command_line     $USER1$/check_snmp_storage.pl -H $HOSTADDRESS$
                    $USER7$ -m $ARG1$ -w $ARG2$ -c $ARG3$ $ARG4$
}

# คำสั่งตรวจสอบหน่วยเก็บข้อมูล โดยใช้โพรโทคอล SNMP V.3
#check_snmp_storage V3
define command{
    command_name    check_snmp_storage_v3
    command_line     $USER1$/check_snmp_storage.pl -H $HOSTADDRESS$
                    $USER8$ -m $ARG1$ -w $ARG2$ -c $ARG3$ $ARG4$
}

# คำสั่งตรวจสอบการทำงานของ Service HTTP
# 'check_http' command definition
define command{
    command_name    check_http
    command_line     $USER1$/check_http -I $HOSTADDRESS$ $ARG1$
}

# คำสั่งตรวจสอบการทำงานของ Service SSH
# 'check_ssh' command definition
define command{
    command_name    check_ssh
    command_line     $USER1$/check_ssh $ARG1$ $HOSTADDRESS$
}

# คำสั่งตรวจสอบการตอบสนองโดยการ Ping
# 'check_ping' command definition
```

```

define command{
    command_name    check_ping
    command_line     $USER1$/check_ping -H $HOSTADDRESS$ -w $ARG1$
                    -c $ARG2$ -p 5
}

# คำสั่งตรวจสอบอุปกรณ์ที่เป็นระบบปฏิบัติการ Windows Server
# 'check_nt' command definition
define command{
    command_name    check_nt
    command_line     $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -s
                    564234 -v $ARG1$ $ARG2$
}

```

**ตารางที่ 4.42** คำศัพท์จากไฟล์ commands.cfg ใช้ในการ Config เพื่อให้ระบบเรียกคำสั่งสำหรับการตรวจสอบการทำงานต่าง ๆ

คำศัพท์	คำอธิบาย
command_name	เป็นการกำหนดชื่อคำสั่งสำหรับการเรียกใช้งาน
command_line	เป็นการกำหนดที่อยู่ในการเรียกไฟล์คำสั่ง และกำหนดจำนวน Argument ที่ใช้ในคำสั่ง

6. ตัวอย่างไฟล์ windows.cfg ที่ Config เพื่อให้ระบบตรวจสอบ Host ที่ใช้ระบบปฏิบัติการ Windows Server

```
root@Nagios-RPi:~# nano /usr/local/nagios/etc/objects/windows.cfg
```

```

ส่วนที่ 1 # HOST DEFINITIONS
define host{
    use                windows-server
    host_name          WinServer.19
    alias              Windows App Server
    address            172.17.XXX.XXX
}

```

## ส่วนที่ 2 # HOST GROUP DEFINITIONS

```
define hostgroup{
    hostgroup_name    windows-servers
    alias              Windows Servers
    members            win-as-19
}
```

## ส่วนที่ 3 # check Version of NSClient++

```
define service{
    use                generic-service,nagiosgraph
    host_name          win-as-19
    service_description NSClient++ Version
    check_command      check_nt!CLIENTVERSION
}
```

## ส่วนที่ 4 # check Uptime

```
define service{
    use                generic-service,nagiosgraph
    host_name          win-as-19
    service_description Uptime
    check_command      check_nt!UPTIME
}
```

## ส่วนที่ 5 # check CPU Load

```
define service{
    use                generic-service,nagiosgraph
    host_name          win-as-19
    service_description CPU Load
    check_command      check_nt!CPULOAD!-l 5,80,90
}
```

ส่วนที่ 6 # check Memory Usage

```
define service{
    use                generic-service,nagiosgraph
    host_name          win-as-19
    service_description Memory Usage
    check_command      check_nt!MEMUSE!-w 80 -c 90
}
```

ส่วนที่ 7 # check C:\ Space

```
define service{
    use                generic-service,nagiosgraph
    host_name          win-as-19
    service_description C:\ Drive Space
    check_command      check_nt!USEDISKSPACE!-l c -w 80 -c 90
}
```

ส่วนที่ 1 ไฟล์ windows.cfg กำหนด Host เพื่อเป็นการระบุอุปกรณ์ที่ใช้ระบบปฏิบัติการ Windows Server บนเครือข่ายที่ต้องการตรวจสอบสถานะการทำงาน

**ตารางที่ 4.43** รายละเอียดการกำหนด Host ของเครื่องพิมพ์เอกสาร ในไฟล์ windows.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ใช้ระบบปฏิบัติการ Windows Server ที่ต้องการตรวจสอบสถานะ
alias	เป็นการกำหนดนามแฝงของอุปกรณ์ที่ใช้ระบบปฏิบัติการ Windows Server ที่ต้องการตรวจสอบสถานะ
address	เป็นการระบุที่อยู่ของอุปกรณ์ที่ใช้ระบบปฏิบัติการ Windows Server ที่ต้องการตรวจสอบสถานะ (IP Address)

ซึ่งระบบจะทำการตรวจสอบสถานะการทำงานของเครื่องพิมพ์เอกสารบนเครือข่าย ดังนี้

**ตารางที่ 4.44** รายชื่อเครื่องพิมพ์เอกสารที่ระบบจะทำการตรวจสอบสถานะ

host_name	address
WinServer.19	172.17.XXX.XXX

ส่วนที่ 2 ไฟล์ windows.cfg กำหนด Host Group เพื่อจัดกลุ่มของอุปกรณ์

```
define hostgroup{
    hostgroup_name    windows-servers
    alias              Windows Servers
    members           win-as-19
}
```

**ตารางที่ 4.45** รายละเอียดการกำหนดค่าของ Host Group ในระบบ

ชื่อ	คำอธิบาย
hostgroup_name	เป็นการกำหนดชื่อของกลุ่ม Host
alias	เป็นการกำหนดนามแฝงของ Host Group
members	เป็นการกำหนดสมาชิกของกลุ่มที่ต้องการจะให้อยู่ใน Host Group

ส่วนที่ 3 ไฟล์ windows.cfg เป็นการกำหนดให้ระบบตรวจสอบสถานะบริการ และรุ่นของ NSClient++ ที่ได้ทำการติดตั้งลงในเครื่อง Windows Server

```
use                generic-service,nagiosgraph
host_name          win-as-19
service_description NSClient++ Version
check_command      check_nt!CLIENTVERSION
}
```

**ตารางที่ 4.46** รายละเอียดการกำหนดคำสั่งให้ระบบตรวจสอบสถานะบริการ และรุ่นของ NSClient++

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะบริการ และรุ่นของ NSClient++ ในเครื่อง Windows Server
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในกาตรวจสอบสถานะบริการ และรุ่นของ NSClient++ ในเครื่อง Windows Server
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะบริการ และรุ่นของ NSClient++ ในเครื่อง Windows Server

ส่วนที่ 4 ไฟล์ windows.cfg เป็นการกำหนดให้ระบบตรวจสอบเวลาในการทำงานของ Windows Server

use	generic-service,nagiosgraph
host_name	win-as-19
service_description	Uptime
check_command	check_nt!UPTIME
	}

**ตารางที่ 4.47** รายละเอียดการกำหนดคำสั่งให้ระบบตรวจสอบเวลาในการทำงานของ Windows Server

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบเวลาในการทำงานของ Windows Server
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในกาตรวจสอบเวลาในการทำงานของ Windows Server



ชื่อ	คำอธิบาย
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบเวลาในการทำงานของ Windows Server

ส่วนที่ 5 ไฟล์ windows.cfg เป็นการกำหนดให้ระบบตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลาง

```

use          generic-service,nagiosgraph
host_name    win-as-19
service_description CPU Load
check_command check_nt!CPULOAD!-l 5,80,90
}

```

**ตารางที่ 4.48** รายละเอียดการกำหนดคำสั่งให้ระบบตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลาง

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าตีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลาง
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในกาตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลาง
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลาง

ส่วนที่ 6 ไฟล์ windows.cfg เป็นการกำหนดให้ระบบตรวจสอบสถานะการใช้หน่วยความจำหลัก

```

use          generic-service,nagiosgraph
host_name    win-as-19
service_description Memory Usage
check_command check_nt!MEMUSE!-w 80 -c 90

```

```
}

```

**ตารางที่ 4.49** รายละเอียดการกำหนดคำสั่งให้ระบบตรวจสอบสถานะสถานะการใช้หน่วยความจำหลัก

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะสถานะการใช้หน่วยความจำหลัก
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในกาตรวจสอบสถานะการใช้หน่วยความจำหลัก
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะการใช้หน่วยความจำหลัก

ส่วนที่ 7 ไฟล์ windows.cfg เป็นการกำหนดให้ระบบตรวจสอบพื้นที่ว่างใน C:\

```
use generic-service,nagiosgraph
host_name win-as-19
service_description C:\ Drive Space
check_command check_nt!USEDISKSPACE!-l c -w 80 -c 90
}
```

**ตารางที่ 4.50** รายละเอียดการกำหนดคำสั่งให้ระบบตรวจสอบพื้นที่ว่างใน C:\

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบพื้นที่ว่างใน C:\
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในกาตรวจสอบพื้นที่ว่างใน C:\
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบพื้นที่ว่างใน C:\

สำหรับรายละเอียดของการ Config ที่ปรากฏในตัวอย่างไฟล์ windows.cfg โดยยกตัวอย่าง คำศัพท์มาเฉพาะบรรทัดที่ไม่ถูก Comment ด้วย “#” มีดังนี้

ตารางที่ 4.51 คำศัพท์ในไฟล์ windows.cfg ที่ใช้ในการ Config

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของเครื่อง Windows Server ที่ต้องการตรวจสอบสถานะ
alias	เป็นการกำหนดนามแฝงของเครื่อง Windows Server ที่ต้องการตรวจสอบสถานะ
address	เป็นการกำหนดที่อยู่ของเครื่อง Windows Server ที่ต้องการตรวจสอบสถานะ
hostgroup	เป็นการกำหนดกลุ่มให้เครื่อง Windows Server ที่ต้องการตรวจสอบสถานะ
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะ
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะ

7. ตัวอย่างไฟล์ switch.cfg ที่ใช้สำหรับการ Config เพื่อให้ระบบตรวจสอบอุปกรณ์ที่เป็นอุปกรณ์ประเภท Switch, Router แล Access Point ได้

```
root@Nagios-RPi:~# nano /usr/local/nagios/etc/objects/switch.cfg
```

```
ส่วนที่ 1 # HOST DEFINITIONS
```

```
define host{
    use                generic-switch
    host_name          AP-8-401[.2]
    alias              Access Point 8-401
    address            172.17.XXX.XXX
    hostgroups         accesspoints
}
```

```
ส่วนที่ 2 # HOST GROUP DEFINITIONS
```

```

define hostgroup{
    hostgroup_name    accesspoints
    alias              Network AP
}

ส่วนที่ 3 # SERVICE DEFINITIONS
define service{
    use                generic-service,nagiosgraph
    host_name          AP-8-401[.2]
    service_description    PING
    check_command      check_ping!200.0,20%!600.0,60%
    check_interval     5
    retry_interval     1
}

```

ส่วนที่ 1 ไฟล์ switch.cfg กำหนด Host เพื่อเป็นการระบุอุปกรณ์บนเครือข่ายที่ต้องการตรวจสอบสถานะการทำงาน

**ตารางที่ 4.52** รายละเอียดการกำหนด Host ไฟล์ switch.cfg

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะ
alias	เป็นการกำหนดนามแฝงของอุปกรณ์ที่ต้องการตรวจสอบสถานะ
address	เป็นการระบุที่อยู่ของอุปกรณ์ที่ต้องการตรวจสอบสถานะ (IP Address)

ซึ่งระบบจะทำการตรวจสอบสถานะการทำงานของอุปกรณ์บนเครือข่ายที่ใช้ระบบปฏิบัติการ Linux ดังนี้

ตารางที่ 4.53 รายชื่ออุปกรณ์ที่ระบบจะทำการตรวจสอบสถานะ

host_name	address
AP-8-401[.2]	172.17.XXX.XXX
AP-8-406[.3]	172.17.XXX.XXX

ส่วนที่ 2 ไฟล์ switch.cfg กำหนด Host Group เพื่อจัดกลุ่มของ Host ที่เป็นประเภทเดียวกัน

```
define hostgroup{
    hostgroup_name    accesspoints
    alias             Network AP
}
```

ตารางที่ 4.54 รายละเอียดการกำหนดค่าของ Host Group ในระบบ

ชื่อ	คำอธิบาย
hostgroup_name	เป็นการกำหนดชื่อการกลุ่ม Host
alias	เป็นการกำหนดนามแฝงของ Host Group

ส่วนที่ 3 ไฟล์ switch.cfg เป็นการกำหนดให้ระบบตรวจสอบสถานะการตอบสนองของอุปกรณ์ โดยการ Ping ไปยังอุปกรณ์

```
use                generic-service,nagiosgraph
host_name          AP-8-401[.2]
service_description    PING
check_command      check_ping!200.0,20%!600.0,60%
check_interval     5
retry_interval     1
}

use                generic-service,nagiosgraph
host_name          AP-8-406[.3]
service_description    PING
check_command      check_ping!200.0,20%!600.0,60%
check_interval     5
```

retry_interval	1
----------------	---

**ตารางที่ 4.55** รายละเอียดการกำหนดคำสั่งให้ระบบตรวจสอบสถานะการตอบสนองของอุปกรณ์โดยการ Ping ไปยังอุปกรณ์

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะการตอบสนองของอุปกรณ์โดยการ Ping ไปยังอุปกรณ์
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะการตอบสนองของอุปกรณ์โดยการ Ping ไปยังอุปกรณ์
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะการตอบสนองของอุปกรณ์โดยการ Ping ไปยังอุปกรณ์

สำหรับรายละเอียดของการ Config ที่ปรากฏในตัวอย่างไฟล์ switch.cfg โดยยกตัวอย่าง คำศัพท์มาเฉพาะบรรทัดที่ไม่ถูก Comment ด้วย “#” มีดังนี้

**ตารางที่ 4.56** คำศัพท์ในไฟล์ switch.cfg ที่ใช้ในการ Config

ชื่อ	คำอธิบาย
use	เป็นการกำหนดการรับค่า ซึ่งในระบบเป็นการรับค่าดีฟอลต์จากเทมเพลตการติดต่อทั่วไป
host_name	เป็นการกำหนดชื่อของอุปกรณ์ที่ต้องการตรวจสอบสถานะ
alias	เป็นการกำหนดนามแฝงของเครื่อง Windows Server ที่ต้องการตรวจสอบสถานะ
address	เป็นการกำหนดที่อยู่ของเครื่องอุปกรณ์ที่ต้องการตรวจสอบสถานะ
hostgroup	เป็นการกำหนดกลุ่มให้เครื่องอุปกรณ์ที่ต้องการตรวจสอบสถานะ
service_description	เป็นการกำหนดคำอธิบายของคำสั่งในการตรวจสอบสถานะ
check_command	เป็นการกำหนดคำสั่งที่ต้องการจะตรวจสอบสถานะ

### การเชื่อมต่อ Aircard เพื่อทำงานร่วมกับระบบ

การทำงานของระบบตรวจสอบสถานะการให้บริการของอุปกรณ์เครือข่าย ได้มีการนำ Aircard มาเชื่อมต่อกับการทำงานของระบบเพื่อให้ระบบมีการทำงานที่มีประสิทธิภาพมากขึ้นในการแจ้งเตือนการทำงานของอุปกรณ์ในเครือข่าย โดยทำให้ระบบสามารถส่งข้อความสั้น (SMS) ไปยังผู้ดูแลระบบเมื่อระบบตรวจพบการหยุดให้บริการของอุปกรณ์ในเครือข่าย และมีการแจ้งเตือนผ่านข้อความสั้นอีกครั้งเมื่อระบบกลับมาให้บริการตามปกติ โดยการตั้งค่าจะมีไฟล์สำคัญดังต่อไปนี้

#### ตารางที่ 4.57 ไฟล์ที่สำคัญในการ Config อุปกรณ์ Aircard

ชื่อไฟล์	คำอธิบาย
smsd.conf	เป็นไฟล์ที่ใช้สำหรับการตั้งค่าอุปกรณ์ Aircard เพื่อให้เครื่องที่ทำหน้าที่เป็น Nagios Server รู้จักกับอุปกรณ์ Aircard
commands.cfg	เป็นไฟล์ที่ใช้สำหรับเพิ่มคำสั่งเพื่อเรียกใช้งานคำสั่งในไฟล์ contacts.cfg
contacts.cfg	เป็นไฟล์ที่ใช้สำหรับการเรียกใช้งานคำสั่งสำหรับแจ้งเตือนด้วยข้อความสั้น (SMS)

1. ตัวอย่างไฟล์ smsd.conf ใช้สำหรับการตั้งค่าอุปกรณ์ Aircard เพื่อให้เครื่องที่ทำหน้าที่เป็น Nagios Server รู้จักกับอุปกรณ์ Aircard  
 root@Nagios-RPi:~# nano /etc/smsd.conf

```

devices = modem1
loglevel = 5
logfile = /var/log/sms/smsd.log
outgoing = /var/spool/sms/outgoing
checked = /var/spool/sms/checked
failed = /var/spool/sms/failed
incoming = /var/spool/sms/incoming
sent = /var/spool/sms/sent
[modem1]
device = /dev/gsm1-1
baudrate = 115200
incoming = yes
report = yes

```

```
queues = modem1
```

สำหรับรายละเอียดของการตั้งค่าให้ Aircard สามารถทำงานร่วมกับเครื่อง Nagios Server ที่ปรากฏในตัวอย่างไฟล์ smsd.conf นี้ โดยจะยกตัวอย่างมาเฉพาะบรรทัดที่ไม่ถูก Comment ด้วย “#”

**ตารางที่ 4.58** คำศัพท์ที่แสดงใช้ในการตั้งค่าอุปกรณ์ Aircard เข้ากับเครื่อง Nagios Server

คำศัพท์	คำอธิบาย
devices	เป็นการระบุชื่อของอุปกรณ์ที่เครื่อง Nagios Server มองเห็น
loglevel	เป็นการระบุระดับขั้นของการเก็บค่า Log
logfile	เป็นการบอกที่อยู่ และชื่อไฟล์ที่ใช้ในการเก็บค่า Log
outgoing	เป็นไดเรกทอรีที่ใช้เก็บข้อความที่ส่งออกไป
checked	เป็นไดเรกทอรีที่ใช้เก็บการตรวจสอบการทำงาน
failed	เป็นไดเรกทอรีที่ใช้เก็บข้อความที่ไม่สามารถส่งไปได้
incoming	เป็นไดเรกทอรีที่ใช้เก็บข้อความที่ได้รับมา
sent	เป็นไดเรกทอรีที่ใช้เก็บข้อความที่ส่งออกไป และถึงไปยังผู้ดูแลระบบแล้ว
baudrate	เป็นอัตราการรับส่งข้อมูล
report	เป็นการกำหนดว่าต้องการรับการรายงานสถานะของแต่ละข้อความที่ได้ส่ง

1.1. ตัวอย่างไฟล์ในไดเรกทอรี outgoing ซึ่งเป็นไดเรกทอรีที่ใช้เก็บข้อความที่ส่งออกไป

```
root@Nagios-RPi:~# ls /var/spool/sms/outgoing

send_0pAWAT  send_01Q6BO  send_1D3L5Y  send_2nZoi7
send_2XbK5q  send_3IXc6Y  send_3M84qM  send_3SK4mn
```

ตัวอย่างข้อมูลในไฟล์ที่อยู่ภายใต้ไดเรกทอรี outgoing

```
root@Nagios-RPi:~# nano /var/spool/sms/outgoing/send_3SK4mn

To: +66815400283
```



```
NAGIOS / Host: server8407 / State: DOWN / Info:(Host check timed out after 30.01
seconds) / Date:08-25-2017 09:00:15
```

1.2. ตัวอย่างไฟล์ในไดเรกทอรี failed ซึ่งเป็นไดเรกทอรีที่ใช้เก็บข้อความที่ไม่สามารถส่งไปถึงผู้ดูแลระบบ

```
root@Nagios-RPi:~# ls /var/spool/sms/failed

send_0H0rvy    send_4ATnBl    send_4BbqrA    send_4oNAqP
send_7qxCKe    send_8eK9Ts    send_8IP307    send_vNUHK1
```

ตัวอย่างข้อมูลในไฟล์ที่อยู่ภายใต้ไดเรกทอรี failed

```
root@Nagios-RPi:~# nano /var/spool/sms/failed/send_vNUHK1

To: +66815400283
Modem: modem1
IMSI: 520002024081598
Fail_reason: +CMS ERROR: 500 (Unknown error)
Failed: 17-08-11 12:31:51
PDU:
0031000B916618450082F30000FF73CEE031F99C825E20E47B4ED781EE65F1F8DD9E8FD
3A017684A0FD3CB3A10F17975825EA0A4DBFCD6A190EF391D344697C76B103DDD2E93
41EF3A1D1436D3CB72D00CE682CD40F3F2F8ED26CF53A0178818A69775305C2B066BC
960B11B2836D3C168BA580D

NAGIOS / Host: webcomsci / State: DOWN / Info:(Host check timed out after 30.03
seconds) / Date:08-10-2017 13:04:15
```

1.3. ตัวอย่างไฟล์ในไดเรกทอรี sent ซึ่งเป็นไดเรกทอรีที่ใช้เก็บข้อความที่ส่งออกไปและได้ส่งถึงไปยังผู้ดูแลระบบแล้ว

```
root@Nagios-RPi:~# ls /var/spool/sms/outgoing

send_0s5c7o    send_1BoyZX    send_1cLCXG    send_1VQ8E9
send_1zyzEc    send_3iydNu    send_3t712B    send_8ql8as
```

ตัวอย่างข้อมูลในไฟล์ที่อยู่ภายใต้ไดเรกทอรี sent

```
root@Nagios-RPi:~# nano /var/spool/sms/outgoing/send_3SK4mn

To: +66815400283
Modem: modem1
Sent: 17-08-11 10:57:16
Message_id: 191
IMSI: 520002024081598

NAGIOS / Host: AccessPoint8-401 / State: CRITICAL / Info:CRITICAL - Plugin timed out
while executing system call / Date:08-10-2017 04:58:30
```

2. ตัวอย่างไฟล์ commands.cfg ใช้สำหรับการกำหนดคำสั่งให้ระบบตรวจสอบสถานะของอุปกรณ์เครือข่ายทำการส่งข้อความสั้นมายังผู้ดูแลระบบ

```
root@Nagios-RPi:~# nano /usr/local/nagios/etc/objects/commands.cfg
```

```
# 'host-notify-by-sms' command definition
define command{
    command_name    notify-host-by-sms
    command_line    /usr/bin/printf "%b" "NAGIOS / Host: "$HOSTNAME$" /
                    State: $HOSTSTATE$ / Info:$HOSTOUTPUT$ /
                    Date:$SHORTDATETIME$" | /usr/local/bin/sendsms
                    $CONTACTPAGER$"
}
```

```
# 'notify-by-sms' command definition
define command{
    command_name    notify-service-by-sms
    command_line    /usr/bin/printf "%b" "NAGIOS / Host: "$HOSTALIAS$" /
                    State: $SERVICESTATE$ / Info:$SERVICEOUTPUT$ /
                    Date:$SHORTDATETIME$" | /usr/local/bin/sendsms
                    $CONTACTPAGER$
}
```

สำหรับรายละเอียดของการตั้งค่าให้สามารถทำงานร่วมกับระบบตรวจสอบสถานะอุปกรณ์เครือข่ายและทำการส่งข้อความสั้นมายังผู้ดูแลระบบ ที่ปรากฏในตัวอย่างไฟล์ commands.cfg นี้ โดยจะยกตัวอย่างมาเฉพาะบรรทัดที่ไม่ถูก Comment ด้วย “#”

**ตารางที่ 4.59** คำศัพท์ที่แสดงใช้ในการตั้งค่าคำสั่งในการส่งข้อความไปยังผู้ดูแลระบบ

คำศัพท์	คำอธิบาย
command_name	เป็นการกำหนดชื่อคำสั่งในการเรียกใช้งาน
command_line	เป็นการกำหนดคำสั่งที่จะใช้งาน

3. ตัวอย่างไฟล์ contacts.cfg ใช้สำหรับการเรียกคำสั่งให้ระบบทำการส่งข้อความสั้นมายังผู้ดูแลระบบ โดยที่ผู้ดูแลระบบต้องระบุตั้งในส่วนที่ 1 ของไฟล์ contacts.cfg

```
root@Nagios-RPi:~# nano /usr/local/nagios/etc/objects/contacts.cfg
```

```
define contact{
    contact_name    nagiosadmin
    use             generic-contact
    alias           Nagios Admin
    email           satun.finalzol@gmail.com
    pager           +66815400283
    service_notification_commands    notify-service-by-email notify-service-by-sms
    host_notification_commands       notify-host-by-email notify-host-by-sms
    service_notification_period      24x7
    host_notification_period         24x7
}
```

```

service_notification_options    c,r
host_notification_options      d,r
}

```

สำหรับรายละเอียดของการตั้งค่าให้สามารถทำงานร่วมกับระบบตรวจสอบสถานะอุปกรณ์เครือข่ายและทำการส่งข้อความสั้นมายังผู้ดูแลระบบ ที่ปรากฏในตัวอย่างไฟล์ `commands.cfg` นี้ โดยจะยกตัวอย่างมาเฉพาะบรรทัดที่ไม่ถูก Comment ด้วย “#”

ซึ่งสิ่งที่ผู้ดูแลระบบต้องระบุมีด้วยกัน 3 ส่วน ดังนี้

**ตารางที่ 4.60** การเรียกคำสั่งในระบบเพื่อการส่งข้อความสั้นไปยังผู้ดูแลระบบ

ชื่อ	คำอธิบาย
pager	เป็นการกำหนดหมายเลขโทรศัพท์ปลายทางที่ผู้ดูแลระบบต้องการให้ระบบส่งข้อความสั้นไป
service_notification_commands	เป็นการกำหนดการเรียกคำสั่งในการส่งข้อความสั้น หากบริการมีสถานะหยุดให้บริการ
host_notification_commands	เป็นการกำหนดการเรียกคำสั่งในการส่งข้อความสั้น หากอุปกรณ์มีสถานะหยุดให้บริการ

### การเพิ่มประสิทธิภาพให้ระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย

ระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย โปรแกรมวิชาคอมพิวเตอร์ มีไอเฟน ซอร์สอย่าง Nagios Core เป็นแกนหลักในการทำงาน ซึ่ง Nagios Core นั้นมีความยืดหยุ่นในการทำงาน และพัฒนาต่อ ซึ่งผู้ดูแลระบบสามารถที่จะติดตั้งส่วนเสริมอย่าง NRPE (Nagios Remote Plugin Executor) โดย NRPE นั้นสามารถตรวจสอบสถานะได้มากขึ้น เช่น การตรวจสอบ Zombie Process เป็นต้น และ NRPE นั้นมีความปลอดภัยมากกว่าการตรวจสอบแบบปกติ นอกจากการติดตั้ง NRPE และ ผู้ดูแลระบบยังสามารถเขียนโปรแกรมภาษาต่าง ๆ เพื่อนำมาเสริมประสิทธิภาพการทำงานได้อีกด้วย

#### 1. การ Config NRPE เพื่อทำงานร่วมกับระบบ

การ Config ให้ NRPE ให้ทำงานร่วมกับระบบนั้นผู้ดูแลระบบต้องติดตั้ง NRPE ในเครื่องบริการที่ผู้ดูแลระบบใช้เป็นเครื่อง Nagios Server ก่อน และเครื่องบริการที่ต้องการตรวจสอบสถานะจำเป็นต้องติดตั้ง NRPE ด้วย เมื่อผู้ดูแลระบบทำการติดตั้ง NRPE แล้ว สามารถ Config ได้ดังตัวอย่างต่อไปนี้

ตัวอย่างการ Config NRPE ในฝั่งเครื่อง Nagios Serverc และเครื่องบริการที่เป็นเครื่องปลายทาง ในส่วนของไฟล์ nrpe.cfg

```

root@Nagios-RPi:~# nano /usr/local/nagios/etc/nrpe.cfg

ส่วนที่ 1 การกำหนดการติดต่อกันของระบบ
server_port=5666
nrpe_user=nagios
nrpe_group=nagios
allowed_hosts=127.0.0.1, 172.17.XXX.XXX

ส่วนที่ 2 การกำหนดคำสั่งที่ต้องการตรวจสอบสถานะโดยใช้ NRPE
command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -w 15,10,5 -c 30,25,20
command[check_hda1]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p
/dev/hda1
command[check_zombie_procs]=/usr/lib/nagios/plugins/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib/nagios/plugins/check_procs -w 230 -c 250
command[check_mem]=/usr/lib/nagios/plugins/check_mem.pl -w 5 -c 2 -f -C
command[check_root]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p
/dev/mapper/ser$
command[check_swap]=/usr/lib/nagios/plugins/check_swap -w 20% -c 10%
command[check_procs_nmbd]=/usr/lib/nagios/plugins/check_procs -c 1:6 -C nmbd
command[check_procs_smbd]=/usr/lib/nagios/plugins/check_procs -c 1:6 -C smbd
command[check_sda]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p /dev/sda
command[check_sda1]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p
/dev/sda1
command[check_sda2]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p
/dev/sda2
command[check_root]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p
/dev/root

```

สำหรับรายละเอียดของการตั้งค่าให้ NRPE สามารถทำงานร่วมกับเครื่อง Nagios Server ที่ปรากฏในตัวอย่างไฟล์ nrpe.cfg นี้ โดยจะยกตัวอย่างมาเฉพาะบรรทัดที่ไม่ถูก Comment ด้วย “#”

ตารางที่ 4. 61 ส่วนที่ผู้ดูแลระบบต้อง Config ไฟล์ nrpe.cfg

คำศัพท์	คำอธิบาย
server_port	เป็นการกำหนดพอร์ตการทำงานของ NRPE
nrpe_user	เป็นการกำหนดชื่อผู้ใช้ของ NRPE
nrpe_group	เป็นการกำหนดกลุ่มชื่อผู้ใช้ของ NRPE
allowed_hosts	เป็นการกำหนด Host ที่อนุญาตให้ตรวจสอบโดย NRPE ได้
command[check_service]	เป็นส่วนของการกำหนดคำสั่งในการตรวจสอบสถานะ

ตัวอย่างการ Config NRPE ในฝั่งเครื่อง Nagios Server และเครื่องบริการที่เป็นเครื่องปลายทาง ในส่วนของไฟล์ nrpe

```

root@Nagios-RPi:~# nano /etc/xinetd.d/nrpe

service nrpe
{
    flags                = REUSE
    socket_type          = stream
    port                 = 5666
    wait                 = no
    user                 = nagios
    group                = nagios
    server               = /usr/local/nagios/bin/nrpe
    server_args          = -c /usr/local/nagios/etc/nrpe.cfg --inetd
    log_on_failure      += USERID
    disable              = no
    only_from            = 127.0.0.1, 172.17.XXX.XXX
}

```

สำหรับรายละเอียดของการตั้งค่าให้ NRPE สามารถทำงานร่วมกับเครื่อง Nagios Server ที่ปรากฏในตัวอย่างไฟล์ nrpe นี้ โดยจะยกตัวอย่างมาเฉพาะบรรทัดที่ไม่ถูก Comment ด้วย “#” ซึ่งส่วนที่ผู้ดูแลระบบต้องทำการ Config มีดังนี้

#### ตารางที่ 4.62 ส่วนที่ผู้ดูแลระบบต้อง Config ไฟล์ nrpe

คำศัพท์	คำอธิบาย
port	เป็นการกำหนดพอร์ตการทำงานของ NRPE
user	เป็นการกำหนดชื่อผู้ใช้ของ NRPE
group	เป็นการกำหนดกลุ่มชื่อผู้ใช้ของ NRPE
only_from	เป็นการกำหนด Host ที่อนุญาตให้ตรวจสอบโดย NRPE ได้

ทำการเปิด Port 5666 ซึ่งเป็น Port การทำงานของ NRPE ทั้งฝั่งของเครื่อง Nagios Server และเครื่องปลายทางที่ต้องการตรวจสอบ

```
root@Nagios-RPi:~# iptables -I INPUT -p tcp --destination-port 5666 -j ACCEPT
```

การทดสอบการทำงานของ NRPE ว่าสามารถทำงาน และเชื่อมต่อฝั่งเครื่องบริการปลายทางได้ หากสามารถเชื่อมต่อกันได้ ระบบจะแสดงรุ่นของ NRPE ตอบกลับมา

```
root@Nagios-RPi:~# cd /usr/lib/nagios/plugins/
root@Nagios-RPi: /usr/lib/nagios/plugins/# ./check_nrpe -H 172.17.XXX.XXX
NRPE v2.13
```

ตัวอย่างการใช้งาน NRPE ในการตรวจสอบสถานะการให้บริการของอุปกรณ์ ตัวอย่างการตรวจสอบ Zombie Process ของเครื่องบริการ cacti

```
use                generic-service,nagiosgraph
host_name          cacti
service_description  Zombie Processes by NRPE
check_command      check_nrpe!check_zombie_procs
}
```

## 2. การตรวจสอบสถานะการทำงานโดยใช้โปรแกรมภาษา Perl

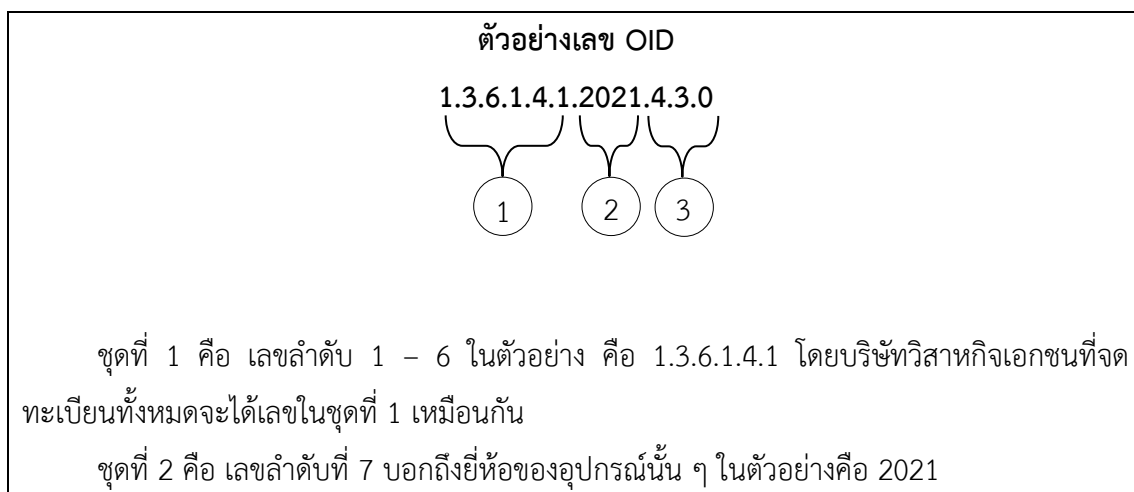
ในการตรวจสอบสถานะการให้บริการของอุปกรณ์เครือข่าย Nagios มีการเก็บไฟล์โปรแกรมที่ใช้ในการตรวจสอบสถานะบริการไว้ในไดเรกทอรี /usr/local/nagios/libexec ซึ่งหาก

ผู้ดูแลระบบต้องการที่จะเพิ่มประสิทธิภาพในการตรวจสอบการทำงาน สามารถเพิ่มไฟล์โปรแกรมคำสั่งเข้าไปในไดเรกทอรีนี้ได้ และเมื่อเพิ่มไฟล์เข้าไปในไดเรกทอรีนี้แล้ว ผู้ดูแลระบบจะต้องทำการเพิ่มคำสั่งในการเรียกใช้งานโปรแกรมคำสั่ง ในไฟล์ `commands.cfg` ที่อยู่ในไดเรกทอรี `/usr/local/nagios/etc/objects` ซึ่งไฟล์โปรแกรมภาษา Perl ที่ได้ทำการเพิ่มเข้าไปในระบบมีดังนี้

**ตารางที่ 4.63** ไฟล์ และรายละเอียดของไฟล์โปรแกรมภาษา Perl ที่ได้เพิ่มเข้าไปในระบบ

ชื่อไฟล์	คำอธิบาย
<code>check_snmp_int.pl</code>	เป็นไฟล์สำหรับการตรวจสอบสถานะการทำงานของ interface ด้วย SNMP
<code>check_snmp_load.pl</code>	เป็นไฟล์สำหรับการตรวจสอบสถานะการทำงานของ CPU ด้วย SNMP
<code>check_snmp_mem.pl</code>	เป็นไฟล์สำหรับการตรวจสอบสถานะการทำงานของหน่วยความจำหลักด้วย SNMP
<code>check_snmp_process.pl</code>	เป็นไฟล์สำหรับการตรวจสอบสถานะการทำงานของ Process ด้วย SNMP
<code>check_snmp_storage.pl</code>	เป็นไฟล์สำหรับการตรวจสอบสถานะการทำงานของฮาร์ดดิสก์ด้วย SNMP
<code>check_port.pl</code>	เป็นไฟล์สำหรับการตรวจสอบพอร์ตการทำงานของอุปกรณ์

โดยการทำงานของไฟล์โปรแกรมที่ตรวจสอบสถานะด้วย SNMP ต้องมีการระบุเลข OID ในโปรแกรม เพื่อเป็นการกำหนดว่าต้องการตรวจสอบอุปกรณ์ใด และตรวจสอบสถานะ หรือบริการใด





ชุดที่ 3 คือ เลขลำดับที่ 8 เป็นต้นไป บอกถึงว่าเป็นการตรวจสอบในส่วนใด ในตัวอย่าง คือ 4.3.0 กล่าวถึง Total Swap Size

### การตรวจสอบสถานะการทำงานของไฟล์ check\_snmp\_int.pl

```
root@Nagios-RPi:~# nano /usr/local/nagios/libexec/check_snmp_int.pl
```

```
# Help : ./check_snmp_int.pl -h

ส่วนที่ 1
# Nagios specific

my $TIMEOUT = 15;
my %ERRORS=('OK'=>0,'WARNING'=>1,'CRITICAL'=>2,'UNKNOWN'=>3,'DEPENDENT'=>4);

ส่วนที่ 2
# SNMP Datas

my $inter_table = '1.3.6.1.2.1.2.2.1';
my $index_table = '1.3.6.1.2.1.2.2.1.1';
my $descr_table = '1.3.6.1.2.1.2.2.1.2';
my $oper_table = '1.3.6.1.2.1.2.2.1.8';
my $admin_table = '1.3.6.1.2.1.2.2.1.7';
my $speed_table = '1.3.6.1.2.1.2.2.1.5';
my $speed_table_64 = '1.3.6.1.2.1.31.1.1.1.15';
my $in_octet_table = '1.3.6.1.2.1.2.2.1.10';
my $in_octet_table_64 = '1.3.6.1.2.1.31.1.1.1.6';
my $in_error_table = '1.3.6.1.2.1.2.2.1.14';
my $in_discard_table = '1.3.6.1.2.1.2.2.1.13';
my $out_octet_table = '1.3.6.1.2.1.2.2.1.16';
my $out_octet_table_64 = '1.3.6.1.2.1.31.1.1.1.10';
my $out_error_table = '1.3.6.1.2.1.2.2.1.20';
my $out_discard_table = '1.3.6.1.2.1.2.2.1.19';
```

```

ส่วนที่ 3
my %status =
(1=>'UP',2=>'DOWN',3=>'TESTING',4=>'UNKNOWN',5=>'DORMANT',6=>'NotPresent',7=>'
lowerLayerDown');

```

ส่วนที่ 1 คือ ลักษณะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการที่แสดงผ่านเว็บไซต์

**ตารางที่ 4.64** ไฟล์ check\_snmp\_int.pl สถานะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการที่แสดงผ่านเว็บไซต์

ชื่อ	คำอธิบาย
TIMEOUT = 15	การตรวจเช็คค่าในแต่ละครั้งภายใน 15 นาที
'OK'=>0	สถานะการทำงานเท่ากับหรือมากกว่า 0 คือ ปกติ มีการแสดงสถานะเป็นสีเขียว
'WARNING'=>1	สถานะการทำงานเท่ากับหรือมากกว่า 1 คือ เตือน มีการแสดงสถานะเป็นสีเหลือง
'CRITICAL'=>2	สถานะการทำงานเท่ากับหรือมากกว่า 2 คือ วิกฤติ มีการแสดงสถานะเป็นสีแดง
'UNKNOWN'=>3	สถานะการทำงานเท่ากับหรือมากกว่า 3 คือ ไม่รู้จัก มีการแสดงสถานะเป็นสีส้ม
'DEPENDENT'=>4	สถานะการทำงานเท่ากับหรือมากกว่า 4 คือ ขึ้นอยู่กับการทำงานของอุปกรณ์

ส่วนที่ 2 คือ แสดงรายละเอียดของการตรวจสอบ SNMP Datas เป็นการตรวจเช็คข้อกำหนดของข้อมูลที่ได้รับไว้เป็นเลข OID

**ตารางที่ 4.65** ไฟล์ check\_snmp\_int.pl สำหรับการตรวจสอบ SNMP Datas

ชื่อ	คำอธิบาย
my \$inter_table= '.1.3.6.1.2.1.2.2.1'	การจัดการข้อมูลในรูปแบบตารางสำหรับการเรียกใช้งาน

ชื่อ	คำอธิบาย
my \$index_table = '1.3.6.1.2.1.2.2.1.1'	ดัชนีของการจัดเก็บข้อมูลของอุปกรณ์เครือข่าย ในลักษณะตาราง
my \$descr_table = '1.3.6.1.2.1.2.2.1.2'	คำอธิบายของการจัดเก็บข้อมูลของอุปกรณ์ เครือข่ายในลักษณะตาราง
my \$oper_table = '1.3.6.1.2.1.2.2.1.8.'	การดำเนินงานปัจจุบันในลักษณะตาราง
my \$admin_table = '1.3.6.1.2.1.2.2.1.7.'	การทดสอบความต้องการในการใช้งานใน ลักษณะตาราง
my \$speed_table = '1.3.6.1.2.1.2.2.1.5.'	การประมาณค่าความเร็วของการดำเนินงาน
my \$speed_table_64 = '1.3.6.1.2.1.31.1.1.1.15.'	การประมาณการของแบนด์วิดธ์ในปัจจุบัน
my \$in_octet_table = '1.3.6.1.2.1.2.2.1.10.'	จำนวน octet
my \$in_octet_table_64 = '1.3.6.1.2.1.31.1.1.1.6.'	วัตถุที่มีค่าความจริงเมื่อรับแพ็กเก็ตทั้งหมด
my \$in_octet_table_64 = '1.3.6.1.2.1.31.1.1.1.6.'	จำนวนของแพ็กเก็ตขาเข้าที่มีข้อผิดพลาด
my \$in_discard_table = '1.3.6.1.2.1.2.2.1.13.'	แพ็กเก็ตขาเข้าที่ถูกทิ้งแม้ไม่มีข้อผิดพลาด
my \$out_octet_table = '1.3.6.1.2.1.2.2.1.16.'	จำนวนรวมของ octet ที่ส่งออก
my \$out_octet_table_64 = '1.3.6.1.2.1.31.1.1.1.10.'	จำนวนรวมของ octet ที่ส่งออก รวมทั้งอักขระ เป็นแบบ 64 บิต
my \$out_error_table = '1.3.6.1.2.1.2.2.1.20.'	จำนวนแพ็กเก็ตขาออกที่ไม่สามารถส่งได้ เนื่องจากข้อผิดพลาดสำหรับการเชื่อมต่อ
my \$out_discard_table = '1.3.6.1.2.1.2.2.1.19.'	แพ็กเก็ตขาออกที่ถูกทิ้งแม้ไม่มีข้อผิดพลาด

ตัวอย่างเลข OID สำหรับการตรวจสอบ SNMP Datas

'1.3.6.1.2.1.2.2.1'

เป็นการจัดการข้อมูลที่มีรายการสำหรับใช้งาน

'1.3.6.1.2.1.2.2.1.1'	ค่าที่ไม่ซ้ำกันและมากกว่าศูนย์สำหรับแต่ละInterface ค่าที่ได้รับจะติดกันตั้งแต่ 1 ค่า ซึ่งแต่ละส่วนย่อยจะติดอย่างน้อยจากหนึ่งของระบบการจัดการของหน่วยงานเครือข่ายเพื่อการเริ่มต้นต่อไปอีก
'1.3.6.1.2.1.2.2.1.2'	ค่าสตริงของข้อมูลต้นฉบับที่เกี่ยวข้องกับ Interface
'1.3.6.1.2.1.2.2.1.8.'	การดำเนินงานปัจจุบันของ Interface เพื่อแสดงให้เห็นว่ายังไม่มีแพ็คเก็ตในการดำเนินงาน
'1.3.6.1.2.1.2.2.1.7.'	การทดสอบความต้องการของ Interface ซึ่งให้เห็นว่ายังไม่มีแพ็คเก็ตในการดำเนินงานก็สามารถผ่านได้ เมื่อระบบมีการจัดการต้นทั้งหมดของการดำเนินการ
'1.3.6.1.2.1.2.2.1.5.'	การประมาณค่าของ Interface
'1.3.6.1.2.1.31.1.1.1.15.'	การประมาณการของแบนด์วิดท์ในปัจจุบันของ Interface จาก 1,000,000 บิตต่อวินาที ซึ่งค่าของ n เป็นความเร็วของ Interface
'1.3.6.1.2.1.2.2.1.10.'	จำนวน octet ที่ได้รับใน Interface รวมทั้งอักขระ
'1.3.6.1.2.1.31.1.1.1.6.'	วัตถุที่มีค่าเป็นเท็จ ถ้า Interface รับแพ็คเก็ต / เฟรมที่จำหน้าถึงวัตถุที่มีค่าของความจริง เมื่อรับแพ็คเก็ตทั้งหมด เฟรมที่จะส่งบนสื่อก็จะมีค่าของความจริง
'1.3.6.1.2.1.2.2.1.14.'	สำหรับการเชื่อมต่อแพ็คเก็ตที่มุ่งเน้นจำนวนของแพ็คเก็ตขาเข้าที่มีข้อผิดพลาดที่ป้องกันการถูกส่งไปยังโปรโตคอลที่สูงกว่า
'1.3.6.1.2.1.2.2.1.13.'	จำนวนแพ็คเก็ตขาเข้าที่ได้รับเลือกให้ถูกละทิ้งแม้ไม่มีข้อผิดพลาด ซึ่งได้รับการตรวจพบการป้องกันไม่ให้ถูกส่งไปยังชั้นโปรโตคอลที่สูงกว่า เหตุที่ต้องทิ้งเพราะแพ็คเก็ตอาจจะเพิ่มพื้นที่ว่างในบัฟเฟอร์
'1.3.6.1.2.1.2.2.1.16.'	จำนวนรวมของ octets ที่ส่งออกจากอินเทอร์เฟซรวมทั้งการกำหนดกรอบอักขระ
'1.3.6.1.2.1.31.1.1.1.10.'	จำนวนรวมของ octets ที่ส่งออกมาจาก รวมถึงการกำหนดกรอบอักขระวัตถุนี้ คือ รุ่น ifOutOctets 64 บิต
'1.3.6.1.2.1.2.2.1.20.'	จำนวนแพ็คเก็ตขาออกที่ไม่สามารถส่งได้เนื่องจากข้อผิดพลาด
'1.3.6.1.2.1.2.2.1.19.'	จำนวนแพ็คเก็ตขาออกที่ถูกเลือกให้ทิ้งแม้ว่าจะไม่มีการตรวจพบข้อผิดพลาดเพื่อป้องกันการส่งผ่าน

### ส่วนที่ 3 การแสดงสถานะการทำงานของอุปกรณ์ที่ตรวจสอบ

ตารางที่ 4.66 ไฟล์ check\_snmp\_int.pl สถานะของอุปกรณ์ที่ตรวจสอบ

ชื่อ	คำอธิบาย
Status	สถานะการทำงานของอุปกรณ์
1=>'UP'	สถานะการทำงานมากกว่าหรือเท่ากับ 1 คือ ทำงานปกติ
2=>'DOWN'	สถานะการทำงานมากกว่าหรือเท่ากับ 2 คือ หยุดทำงาน
3=>'TESTING'	สถานะการทำงานมากกว่าหรือเท่ากับ 3 คือ กำลังทดสอบ
4=>'UNKNOWN'	สถานะการทำงานมากกว่าหรือเท่ากับ 4 คือ ไม่รู้จัก
5=>'DORMANT'	สถานะการทำงานมากกว่าหรือเท่ากับ 5 คือ หยุดนิ่ง หรือชะงัก
6=>'NotPresent'	สถานะการทำงานมากกว่าหรือเท่ากับ 6 คือ ไม่อยู่
7=>'lowerLayerDown'	สถานะการทำงานมากกว่าหรือเท่ากับ 7 คือ การทำงานต่ำสุด

### การตรวจสอบสถานะการทำงานของไฟล์ check\_snmp\_load.pl

root@Nagios-RPi:~# nano /usr/local/nagios/libexec/check\_snmp\_load.pl

```
# Help : ./check_snmp_load.pl -h

use strict;
use Net::SNMP;
use Getopt::Long;

ส่วนที่ 1
# Nagios specific
my $TIMEOUT = 15;
my %ERRORS=('OK'=>0,'WARNING'=>1,'CRITICAL'=>2,'UNKNOWN'=>3,'DEPENDENT'=>4);

ส่วนที่ 2
# SNMP Datas
# Generic with host-ressource-mib
my $base_proc = "1.3.6.1.2.1.25.3.3.1"; # oid for all proc info
my $proc_id = "1.3.6.1.2.1.25.3.3.1.1"; # list of processors (product ID)
```

```
my $proc_load = "1.3.6.1.2.1.25.3.3.1.2"; # %time the proc was not idle over last
minute
```

### ส่วนที่ 3

```
# Linux load
```

```
my $linload_table= "1.3.6.1.4.1.2021.10.1"; # net-snmp load table
```

```
my $linload_name = "1.3.6.1.4.1.2021.10.1.2"; # text 'Load-1','Load-5', 'Load-15'
```

```
my $linload_load = "1.3.6.1.4.1.2021.10.1.3"; # effective load table
```

### ส่วนที่ 4

```
# Cisco cpu/load
```

```
my $cisco_cpu_5m = "1.3.6.1.4.1.9.2.1.58.0"; # Cisco CPU load (5min %)
```

```
my $cisco_cpu_1m = "1.3.6.1.4.1.9.2.1.57.0"; # Cisco CPU load (1min %)
```

```
my $cisco_cpu_5s = "1.3.6.1.4.1.9.2.1.56.0"; # Cisco CPU load (5sec %)
```

### ส่วนที่ 5

```
# Cisco catalyst cpu/load
```

```
my $ciscocata_cpu_5m = "1.3.6.1.4.1.9.9.109.1.1.1.5.9"; # Cisco CPU load (5min %)
```

```
my $ciscocata_cpu_1m = "1.3.6.1.4.1.9.9.109.1.1.1.3.9"; # Cisco CPU load (1min %)
```

```
my $ciscocata_cpu_5s = "1.3.6.1.4.1.9.9.109.1.1.1.4.9"; # Cisco CPU load (5sec %)
```

### ส่วนที่ 6

```
# Netscreen cpu/load
```

```
my $nsc_cpu_5m = "1.3.6.1.4.1.3224.16.1.4.0"; # NS CPU load (5min %)
```

```
my $nsc_cpu_1m = "1.3.6.1.4.1.3224.16.1.2.0"; # NS CPU load (1min %)
```

```
my $nsc_cpu_5s = "1.3.6.1.4.1.3224.16.1.3.0"; # NS CPU load (5sec %)
```

### ส่วนที่ 7

```
# AS/400 CPU
```

```
my $as400_cpu = "1.3.6.1.4.1.2.6.4.5.1.0"; # AS400 CPU load (10000=100%);
```

## ส่วนที่ 8

```
# Net-SNMP CPU
```

```
my $ns_cpu_idle = "1.3.6.1.4.1.2021.11.11.0"; # Net-snmp cpu idle
```

```
my $ns_cpu_user = "1.3.6.1.4.1.2021.11.9.0"; # Net-snmp user cpu usage
```

```
my $ns_cpu_system = "1.3.6.1.4.1.2021.11.10.0"; # Net-snmp system cpu usage
```

## ส่วนที่ 9

```
# Procurve CPU
```

```
my $procurve_cpu = "1.3.6.1.4.1.11.2.14.11.5.1.9.6.1.0"; # Procurve CPU Counter
```

## ส่วนที่ 10

```
# Nokia CPU
```

```
my $nokia_cpu = "1.3.6.1.4.1.94.1.21.1.7.1.0"; # Nokia CPU % usage
```

## ส่วนที่ 11

```
# Bluecoat Appliance
```

```
my $bluecoat_cpu = "1.3.6.1.4.1.3417.2.4.1.1.1.4.1"; # Bluecoat %cpu usage.
```

## ส่วนที่ 12

```
# Fortigate CPU
```

```
my $fortigate_cpu = ".1.3.6.1.4.1.12356.1.8.0"; # Fortigate CPU % usage
```

## ส่วนที่ 13

```
# Linkproof Appliance
```

```
my $linkproof_cpu = "1.3.6.1.4.1.89.35.1.55.0"; # CPU RE (Routing Engine Tasks)
```

## ส่วนที่ 14

```
# HP-UX cpu usage (thanks to krizb for the OIDs).
```

```
my $hpux_load_1_min="1.3.6.1.4.1.11.2.3.1.1.3.0";
```

```
my $hpux_load_5_min="1.3.6.1.4.1.11.2.3.1.1.4.0";
```

```
my $hpux_load_15_min="1.3.6.1.4.1.11.2.3.1.1.5.0";
```

ส่วนที่ 1 คือ ลักษณะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการที่แสดงผ่านเว็บไซต์

ตารางที่ 4.67 ไฟล์ check\_snmp\_load.pl สถานะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการที่แสดงผ่านเว็บไซต์

ชื่อ	คำอธิบาย
TIMEOUT = 15	การตรวจเช็คค่าในแต่ละครั้งภายใน 15 นาที
'OK'=>0	สถานะการทำงานเท่ากับหรือมากกว่า 0 คือ ปกติ มีการแสดงสถานะเป็นสีเขียว
'WARNING'=>1	สถานะการทำงานเท่ากับหรือมากกว่า 1 คือ เตือน มีการแสดงสถานะเป็นสีเหลือง
'CRITICAL'=>2	สถานะการทำงานเท่ากับหรือมากกว่า 2 คือ วิกฤติ มีการแสดงสถานะเป็นสีแดง
'UNKNOWN'=>3	สถานะการทำงานเท่ากับหรือมากกว่า 3 คือ ไม่รู้จัก มีการแสดงสถานะเป็นสีส้ม
'DEPENDENT'=>4	สถานะการทำงานเท่ากับหรือมากกว่า 4 คือ ขึ้นอยู่กับการทำงานของอุปกรณ์

ส่วนที่ 2 คือ แสดงรายละเอียดของการตรวจสอบสถานะทั่วไปด้วย MIB ที่ฝังอยู่ใน Host ที่ได้ระบุไว้เป็นเลข OID

ตารางที่ 4.68 ไฟล์ check\_snmp\_load.pl สำหรับการตรวจสอบสถานะทั่วไปด้วย MIB ที่ฝังอยู่ใน Host

ชื่อ	คำอธิบาย
my \$base_proc = "1.3.6.1.2.1.25.3.3.1"; # oid for all proc info	หน่วยประมวลผลที่มีอยู่ใน Host
my \$proc_id = "1.3.6.1.2.1.25.3.3.1.1"; # list of processors (product ID)	หมายเลขผลิตภัณฑ์
my \$proc_load = "1.3.6.1.2.1.25.3.3.1.2"; # %time the proc was not idle over last minute	ร้อยละของเวลาที่หน่วยประมวลผลไม่ว่าง



ตัวอย่างเลข OID สำหรับการตรวจสอบบนอุปกรณ์ Generic with host-resource-mib

"1.3.6.1.2.1.25.3.3.1" สำหรับหน่วยประมวลผลที่มีอยู่โดยฮาร์ดแวร์ hrDeviceIndex ในดัชนีที่แสดงถึงรายการใน hrDeviceTable ที่สอดคล้องกับ hrProcessorEntry เป็นตัวอย่างของวิธีการวัตถุในตารางนี้มีชื่อตัวอย่างของวัตถุ hrProcessorFrwID อาจเป็นชื่อ hrProcessorFrwID.3

"1.3.6.1.2.1.25.3.3.1.1" หมายเลขผลิตภัณฑ์ของเฟิร์มแวร์ที่เชื่อมโยงกับหน่วยประมวลผล

"1.3.6.1.2.1.25.3.3.1.2" ค่าเฉลี่ยร้อยละเวลาของหน่วยประมวลผลไม่ว่างสำหรับการใช้งานในกรณีที่ไม่จำเป็น

ส่วนที่ 3 คือ แสดงรายละเอียดของการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ที่เป็น Linux ที่ได้ระบุไว้เป็นเลข OID

**ตารางที่ 4.69** ไฟล์ check\_snmp\_load.pl สำหรับการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของ Linux

ชื่อ	คำอธิบาย
my \$linload_table= "1.3.6.1.4.1.2021.10.1"; # net-snmp load table	กำหนดให้การตรวจสอบมีการจัดเก็บข้อมูลแบบ ลำดับชั้น
my \$linload_name = "1.3.6.1.4.1.2021.10.1.2"; # text 'Load- 1','Load-5', 'Load-15'	การตรวจสอบชื่อ "Load-1" คือ ข้อมูลการโหลด ใน 1 นาที การตรวจสอบชื่อ "Load-5" คือ ข้อมูลการโหลด ใน 5 นาที การตรวจสอบชื่อ "Load-15" คือ ข้อมูลการ โหลดใน 15 นาที
my \$linload_load = "1.3.6.1.4.1.2021.10.1.3"; # effective load table	ค่าเฉลี่ยความเร็วของ CPU Load

ตัวอย่างเลข OID สำหรับการตรวจสอบบนอุปกรณ์ Linux

"1.3.6.1.4.1.2021.10.1"      การจัดเก็บข้อมูลแบบลำดับชั้นภายใต้ค่าเฉลี่ยความเร็วในการโหลด  
 "1.3.6.1.4.1.2021.10.1.2"      เป็นการตรวจสอบชื่อโดยใช้ "Load-1" คือ ข้อมูลการโหลดใน 1 นาที  
 การตรวจสอบชื่อ "Load-5" คือ ข้อมูลการโหลดใน 5 นาที การตรวจสอบชื่อ "Load-15" คือ ข้อมูล  
 การโหลดใน 15 นาที  
 "1.3.6.1.4.1.2021.10.1.3"      ค่าเฉลี่ยความเร็วในการโหลดเป็น

ส่วนที่ 4 คือ แสดงรายละเอียดของการตรวจสอบสถานะการทำงานของหน่วยประมวลผล  
 กลางของอุปกรณ์ยี่ห้อ Cisco ที่ได้ระบุไว้เป็นเลข OID

**ตารางที่ 4.70** ไฟล์ check\_snmp\_load.pl สำหรับการตรวจสอบสถานะการทำงานของหน่วย  
 ประมวลผลกลางของอุปกรณ์ยี่ห้อ Cisco

ชื่อ	คำอธิบาย
my \$cisco_cpu_5m = "1.3.6.1.4.1.9.2.1.58.0"; # Cisco CPU load (5min %)	การโหลดของ CPU ยี่ห้อ Cisco ในเวลา 5 นาที
my \$cisco_cpu_1m = "1.3.6.1.4.1.9.2.1.57.0"; # Cisco CPU load (1min %)	การโหลดของ CPU ยี่ห้อ Cisco ในเวลา 1 นาที
my \$cisco_cpu_5s = "1.3.6.1.4.1.9.2.1.56.0"; # Cisco CPU load (5sec %)	การโหลดของ CPU ยี่ห้อ Cisco ในเวลา 5 วินาที

ตัวอย่างเลข OID สำหรับการตรวจสอบบนอุปกรณ์ Cisco

"1.3.6.1.4.1.9.2.1.58.0"      ค่าเฉลี่ยในการทำงานของ CPU ใน 5 นาที  
 "1.3.6.1.4.1.9.2.1.57.0"      ค่าเฉลี่ยในการทำงานของ CPU ใน 1 นาที  
 "1.3.6.1.4.1.9.2.1.56.0"      ค่าเฉลี่ยในการทำงานของ CPU ใน 5 วินาที

ส่วนที่ 5 คือ แสดงรายละเอียดของการตรวจสอบสถานะการทำงานของหน่วยประมวลผล

**ตารางที่ 4.71** ไฟล์ check\_snmp\_load.pl สำหรับการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Cisco Catalyst

ชื่อ	คำอธิบาย
my \$ciscocata_cpu_5m = ".1.3.6.1.4.1.9.9.109.1.1.1.1.5.9"; # Cisco CPU load (5min %)	การโหลดของ CPU ยี่ห้อ Cisco Catalyst ใน เวลา 5 นาที
my \$ciscocata_cpu_1m = ".1.3.6.1.4.1.9.9.109.1.1.1.1.3.9"; # Cisco CPU load (1min %)	การโหลดของ CPU ยี่ห้อ Cisco Catalyst ใน เวลา 1 นาที
my \$ciscocata_cpu_5s = ".1.3.6.1.4.1.9.9.109.1.1.1.1.4.9"; # Cisco CPU load (5sec %)	การโหลดของ CPU ยี่ห้อ Cisco Catalyst ใน เวลา 5 วินาที

ตัวอย่างเลข OID สำหรับการตรวจสอบบนอุปกรณ์ Cisco Catalyst

".1.3.6.1.4.1.9.9.109.1.1.1.1.5.9"	ร้อยละของ CPU โยไม่รวมสถานะไม่ว่างในช่วงระยะเวลา 5 นาที
".1.3.6.1.4.1.9.9.109.1.1.1.1.3.9"	ร้อยละของ CPU โยไม่รวมสถานะไม่ว่างในช่วงระยะเวลา 1 นาที
".1.3.6.1.4.1.9.9.109.1.1.1.1.4.9"	ร้อยละของ CPU โยไม่รวมสถานะไม่ว่างในช่วงระยะเวลา 5 วินาที

ส่วนที่ 6 คือ แสดงรายละเอียดของการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Netscreen ที่ได้ระบุไว้เป็นเลข OID

**ตารางที่ 4.72** ไฟล์ check\_snmp\_load.pl สำหรับการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Netscreen

ชื่อ	คำอธิบาย
my \$nsc_cpu_5m = "1.3.6.1.4.1.3224.16.1.4.0"; # NS CPU load (5min %)	การโหลดของ CPU ยี่ห้อ Netscreen ในเวลา 5 นาที
my \$nsc_cpu_1m = "1.3.6.1.4.1.3224.16.1.2.0"; # NS CPU load (1min %)	การโหลดของ CPU ยี่ห้อ Netscreen ในเวลา 1 นาที
my \$nsc_cpu_5s = "1.3.6.1.4.1.3224.16.1.3.0"; # NS CPU load (5sec %)	การโหลดของ CPU ยี่ห้อ Netscreen ในเวลา 5 วินาที

ตัวอย่างเลข OID สำหรับการตรวจสอบบนอุปกรณ์ Netscreen
"1.3.6.1.4.1.3224.16.1.4.0" การโหลดของ CPU ในเวลา 5 นาที โดยคิดเป็นค่าเฉลี่ยร้อยละ
"1.3.6.1.4.1.3224.16.1.2.0" การโหลดของ CPU ในเวลา 1 นาที โดยคิดเป็นค่าเฉลี่ยร้อยละ
"1.3.6.1.4.1.3224.16.1.3.0" การโหลดของ CPU ในเวลา 5 วินาที โดยคิดเป็นค่าเฉลี่ยร้อยละ

ส่วนที่ 7 คือ แสดงรายละเอียดของการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ AS/400 ที่ได้ระบุไว้เป็นเลข OID

**ตารางที่ 4.73** ไฟล์ check\_snmp\_load.pl สำหรับการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ AS/400

ชื่อ	คำอธิบาย
my \$as400_cpu = "1.3.6.1.4.1.2.6.4.5.1.0"; # AS400 CPU load (10000=100%);	ร้อยละของการโหลดของ CPU ยี่ห้อ AS/400

ตัวอย่างเลข OID สำหรับการตรวจสอบบนอุปกรณ์ AS/400

"1.3.6.1.4.1.2.6.4.5.1.0" การโหลดของ CPU โดยคิดเป็นเปอร์เซ็นต์ เช่น 10000 เท่ากับ 100%

ส่วนที่ 8 คือ แสดงรายละเอียดของการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางที่ถูกใช้โดย Net-SNMP ที่ได้ระบุไว้เป็นเลข OID

**ตารางที่ 4.74** ไฟล์ check\_snmp\_load.pl สำหรับการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางที่ถูกใช้โดย Net-SNMP

ชื่อ	คำอธิบาย
my \$ns_cpu_idle = "1.3.6.1.4.1.2021.11.11.0"; # Net-snmp cpu idle	ร้อยละของเวลาที่ CPU ไม่ได้ถูกใช้งาน
my \$ns_cpu_user = "1.3.6.1.4.1.2021.11.9.0"; # Net-snmp user cpu usage	ร้อยละของเวลาที่ CPU ถูกใช้
my \$ns_cpu_system = "1.3.6.1.4.1.2021.11.10.0"; # Net-snmp system cpu usage	ร้อยละของเวลาที่ระบบใช้ CPU

ตัวอย่างเลข OID สำหรับการตรวจสอบบนอุปกรณ์ Net-SNMP CPU

"1.3.6.1.4.1.2021.11.11.0" เปอร์เซ็นต์เวลาที่ CPU ที่ไม่ได้ใช้งาน

"1.3.6.1.4.1.2021.11.9.0" เปอร์เซ็นต์ของเวลาที่ CPU ถูกใช้งาน

"1.3.6.1.4.1.2021.11.10.0" เปอร์เซ็นต์ของเวลาของ CPU ของระบบ

ส่วนที่ 9 คือ แสดงรายละเอียดของการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Procurve ที่ได้ระบุไว้เป็นเลข OID

**ตารางที่ 4.75** ไฟล์ check\_snmp\_load.pl สำหรับการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Procurve

ชื่อ	คำอธิบาย
my \$procurve_cpu = "1.3.6.1.4.1.11.2.14.11.5.1.9.6.1.0"; # Procurve CPU Counter	นับการทำงานของ CPU ในอุปกรณ์ยี่ห้อ HP Procurve

ตัวอย่างเลข OID สำหรับการตรวจสอบบนอุปกรณ์ Procurve	
"1.3.6.1.4.1.11.2.14.11.5.1.9.6.1.0"	นับการทำงานของ CPU ในอุปกรณ์ยี่ห้อ HP Procurve

ส่วนที่ 10 คือ แสดงรายละเอียดของการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Nokia ที่ได้ระบุไว้เป็นเลข OID

**ตารางที่ 4.76** ไฟล์ check\_snmp\_load.pl สำหรับการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Nokia

ชื่อ	คำอธิบาย
my \$nokia_cpu = "1.3.6.1.4.1.94.1.21.1.7.1.0"; # Nokia CPU % usage	ร้อยละของการใช้งาน CPU ในปัจจุบันของอุปกรณ์ยี่ห้อ Nokia

ตัวอย่างเลข OID สำหรับการตรวจสอบบนอุปกรณ์ Nokia	
"1.3.6.1.4.1.94.1.21.1.7.1.0"	ร้อยละของการใช้งาน CPU ในปัจจุบันของอุปกรณ์ยี่ห้อ Nokia

ส่วนที่ 11 คือ แสดงรายละเอียดของการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Bluecoat ที่ได้ระบุไว้เป็นเลข OID

**ตารางที่ 4.77** ไฟล์ check\_snmp\_load.pl สำหรับการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Bluecoat

ชื่อ	คำอธิบาย
my \$bluecoat_cpu = "1.3.6.1.4.1.3417.2.4.1.1.1.4.1"; # Bluecoat %cpu usage.	ร้อยละของการใช้งาน CPU ของอุปกรณ์ยี่ห้อ Bluecoat

ตัวอย่างเลข OID สำหรับการตรวจสอบบนอุปกรณ์ Bluecoat "1.3.6.1.4.1.3417.2.4.1.1.1.4.1" ร้อยละของการใช้งาน CPU ของอุปกรณ์ยี่ห้อ Bluecoat
---

ส่วนที่ 12 คือ แสดงรายละเอียดของการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Fortigate ที่ได้ระบุไว้เป็นเลข OID

**ตารางที่ 4.78** ไฟล์ check\_snmp\_load.pl สำหรับการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Fortigate

ชื่อ	คำอธิบาย
my \$fortigate_cpu = ".1.3.6.1.4.1.12356.1.8.0"; # Fortigate CPU % usage	ร้อยละของการใช้งาน CPU ของอุปกรณ์ยี่ห้อ Fortigate

ตัวอย่างเลข OID สำหรับการตรวจสอบบนอุปกรณ์ Fortigate ".1.3.6.1.4.1.12356.1.8.0" ร้อยละของการใช้งาน CPU ของอุปกรณ์ยี่ห้อ Fortigate
---

ส่วนที่ 13 คือ แสดงรายละเอียดของการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Linkproof ที่ได้ระบุไว้เป็นเลข OID

**ตารางที่ 4.79** ไฟล์ check\_snmp\_load.pl สำหรับการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ Linkproof

ชื่อ	คำอธิบาย
my \$linkproof_cpu= "1.3.6.1.4.1.89.35.1.55.0"; # CPU RE (Routing Engine Tasks)	การใช้ CPU และการใช้ทรัพยากรทางเครือข่ายของอุปกรณ์ยี่ห้อ Linkproof

ตัวอย่างเลข OID สำหรับการตรวจสอบการใช้ CPU และการใช้ทรัพยากรทางเครือข่ายของอุปกรณ์ยี่ห้อ Linkproof	
"1.3.6.1.4.1.89.35.1.55.0"	การใช้ CPU และการใช้ทรัพยากรทางเครือข่ายของอุปกรณ์ยี่ห้อ Linkproof

ส่วนที่ 14 คือ แสดงรายละเอียดของการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ HP-UX ที่ได้ระบุไว้เป็นเลข OID

**ตารางที่ 4.80** ไฟล์ check\_snmp\_load.pl สำหรับการตรวจสอบสถานะการทำงานของหน่วยประมวลผลกลางของอุปกรณ์ยี่ห้อ HP-UX

ชื่อ	คำอธิบาย
my \$hpux_load_1_min = "1.3.6.1.4.1.11.2.3.1.1.3.0";	จำนวนงานโดยเฉลี่ยในช่วง 1 นาที * 100
my \$hpux_load_5_min = "1.3.6.1.4.1.11.2.3.1.1.4.0";	จำนวนงานโดยเฉลี่ยในช่วง 5 นาที * 100
my \$hpux_load_15_min = "1.3.6.1.4.1.11.2.3.1.1.5.0";	จำนวนงานโดยเฉลี่ยในช่วง 15 วินาที * 100



ตัวอย่างเลข OID สำหรับการตรวจสอบบนอุปกรณ์ HP-UX

```
"1.3.6.1.4.1.11.2.3.1.1.3.0" จำนวนงานโดยเฉลี่ยในช่วง 1 นาที * 100
"1.3.6.1.4.1.11.2.3.1.1.4.0" จำนวนงานโดยเฉลี่ยในช่วง 5 นาที * 100
"1.3.6.1.4.1.11.2.3.1.1.5.0" จำนวนงานโดยเฉลี่ยในช่วง 15 วินาที * 100
```

### การตรวจสอบสถานะการทำงานของไฟล์ check\_snmp\_mem.pl

```
root@Nagios-RPi:~# nano /usr/local/nagios/libexec/check_snmp_mem.pl
```

```
# Help : ./check_snmp_load.pl -h

use strict;
use Net::SNMP;
use Getopt::Long;

ส่วนที่ 1
# Nagios specific
#my $TIMEOUT = 15;
#my
%ERRORS=('OK'=>0,'WARNING'=>1,'CRITICAL'=>2,'UNKNOWN'=>3,'DEPENDENT'=>4);

ส่วนที่ 2
# SNMP Datas
# Net-snmp memory
my $nets_ram_free = "1.3.6.1.4.1.2021.4.6.0"; # Real memory free
my $nets_ram_total = "1.3.6.1.4.1.2021.4.5.0"; # Real memory total
my $nets_ram_cache = "1.3.6.1.4.1.2021.4.15.0"; # Real memory cached
my $nets_swap_free = "1.3.6.1.4.1.2021.4.4.0"; # swap memory free
my $nets_swap_total = "1.3.6.1.4.1.2021.4.3.0"; # Swap memory total
my @nets_oids =
($nets_ram_free,$nets_ram_total,$nets_swap_free,$nets_swap_total,$nets_ram_cach
e);
```

## ส่วนที่ 3

# Cisco

```

my $cisco_mem_pool    = "1.3.6.1.4.1.9.9.48.1.1.1"; # Cisco memory pool
my $cisco_index       = "1.3.6.1.4.1.9.9.48.1.1.1.2"; # memory pool name and index
my $cisco_valid       = "1.3.6.1.4.1.9.9.48.1.1.1.4"; # Valid memory if 1
my $cisco_used        = "1.3.6.1.4.1.9.9.48.1.1.1.5"; # Used memory
my $cisco_free        = "1.3.6.1.4.1.9.9.48.1.1.1.6"; # Free memory
# .1 : type, .2 : name, .3 : alternate, .4 : valid, .5 : used, .6 : free, .7 : max free

```

## ส่วนที่ 4

# HP Procurve

```

my $hp_mem_pool       = "1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1"; # HP memory pool
my $hp_mem_index     = "1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1.1"; # memory slot index
my $hp_mem_total     = "1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1.5"; # Total Bytes
my $hp_mem_free      = "1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1.6"; # Free Bytes
my $hp_mem_free_seg = "1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1.3"; # Free segments

```

ส่วนที่ 1 คือ ลักษณะสถานะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการที่แสดงผ่านเว็บไซต์

**ตารางที่ 4.81** ไฟล์ check\_snmp\_mem.pl สถานะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการที่แสดงผ่านเว็บไซต์

ชื่อ	คำอธิบาย
TIMEOUT = 15	การตรวจเช็คค่าในแต่ละครั้งภายใน 15 นาที
'OK'=>0	สถานะการทำงานเท่ากับหรือมากกว่า 0 คือ ปกติ มีการแสดงสถานะเป็นสีเขียว
'WARNING'=>1	สถานะการทำงานเท่ากับหรือมากกว่า 1 คือ เตือน มีการแสดงสถานะเป็นสีเหลือง
'CRITICAL'=>2	สถานะการทำงานเท่ากับหรือมากกว่า 2 คือ วิกฤติ มีการแสดงสถานะเป็นสีแดง

ชื่อ	คำอธิบาย
'UNKNOWN'=>3	สถานะการทำงานเท่ากับหรือมากกว่า 3 คือ ไม่รู้จัก มีการแสดงสถานะเป็นสีส้ม
'DEPENDENT'=>4	สถานะการทำงานเท่ากับหรือมากกว่า 4 คือ ขึ้นอยู่กับการทำงานของอุปกรณ์

ส่วนที่ 2 แสดงรายละเอียดของ check\_snmp\_mem.pl สำหรับการตรวจสอบ Net-SNMP Memory การเช็คหน่วยความจำหลักของอุปกรณ์ที่ติดตั้งระบบปฏิบัติการ Linux ที่ใช้ในระบบ

**ตารางที่ 4.82** ไฟล์ check\_snmp\_mem.pl สำหรับการตรวจสอบหน่วยความจำหลัก Net-SNMP Memory

ชื่อ	คำอธิบาย
my \$nets_ram_free = "1.3.6.1.4.1.2021.4.6.0";	หน่วยความจำหลักที่ยังว่าง
my \$nets_ram_total = "1.3.6.1.4.1.2021.4.5.0";	หน่วยความจำหลักทั้งหมดที่ใช้งาน
my \$nets_ram_cache = "1.3.6.1.4.1.2021.4.15.0";	หน่วยความจำหลักสำหรับเก็บข้อมูลที่เรียกใช้บ่อย
my \$nets_swap_free = "1.3.6.1.4.1.2021.4.4.0";	พื้นที่ว่างในหน่วยความจำหลักที่ใช้สำหรับการแลกเปลี่ยนการใช้งาน
my \$nets_swap_total = "1.3.6.1.4.1.2021.4.3.0";	หน่วยความจำหลักทั้งหมดที่มีการแลกเปลี่ยนการใช้งาน

ตัวอย่างเลข OID สำหรับการตรวจสอบ Net-SNMP Memory

- "1.3.6.1.4.1.2021.4.6.0" พื้นที่หน่วยความจำหลักที่มีว่างอยู่
- "1.3.6.1.4.1.2021.4.5.0" ขนาดของหน่วยความจำหลักทั้งหมด
- "1.3.6.1.4.1.2021.4.15.0" หน่วยความจำหลักทั้งหมดสำหรับเก็บข้อมูลที่เรียกใช้บ่อย
- "1.3.6.1.4.1.2021.4.4.0" พื้นที่ว่างในหน่วยความจำหลักที่ใช้สำหรับการแลกเปลี่ยนการใช้งาน
- "1.3.6.1.4.1.2021.4.3.0" หน่วยความจำหลักทั้งหมดที่มีการแลกเปลี่ยนการใช้งาน

ส่วนที่ 3 แสดงรายละเอียดของ check\_snmp\_mem.pl สำหรับการตรวจสอบหน่วยความจำหลักบนอุปกรณ์ยี่ห้อ Cisco

ตารางที่ 4.83 ไฟล์ check\_snmp\_mem.pl สำหรับการตรวจสอบหน่วยความจำหลัก Cisco

ชื่อ	คำอธิบาย
my \$cisco_mem_pool = "1.3.6.1.4.1.9.9.48.1.1.1";	การใช้งานหน่วยความจำหลักร่วมกันในอุปกรณ์ Cisco
my \$cisco_index = "1.3.6.1.4.1.9.9.48.1.1.1.2";	ชื่อในการใช้งานหน่วยความจำหลักในอุปกรณ์ Cisco
my \$cisco_valid = "1.3.6.1.4.1.9.9.48.1.1.1.4";	ค่าความถูกต้องในการใช้งานหน่วยความจำหลัก ในอุปกรณ์ Cisco
my \$cisco_used = "1.3.6.1.4.1.9.9.48.1.1.1.5";	ปริมาณการใช้งานหน่วยความจำหลักในอุปกรณ์ Cisco
my \$cisco_free = "1.3.6.1.4.1.9.9.48.1.1.1.6";	พื้นที่ว่างของหน่วยความจำหลักในอุปกรณ์ Cisco

ตัวอย่างเลข OID สำหรับการตรวจสอบหน่วยความจำหลักบนอุปกรณ์ Cisco
"1.3.6.1.4.1.9.9.48.1.1.1" การใช้งานหน่วยความจำหลักร่วมกันในอุปกรณ์
"1.3.6.1.4.1.9.9.48.1.1.1.2" ชื่อในการใช้งานหน่วยความจำหลักในอุปกรณ์
"1.3.6.1.4.1.9.9.48.1.1.1.4" ค่าความถูกต้องในการใช้งานหน่วยความจำหลักในอุปกรณ์
"1.3.6.1.4.1.9.9.48.1.1.1.5" ปริมาณการใช้งานหน่วยความจำหลักในอุปกรณ์
"1.3.6.1.4.1.9.9.48.1.1.1.6" พื้นที่ว่างในหน่วยความจำหลักในอุปกรณ์

ส่วนที่ 4 แสดงรายละเอียดของ check\_snmp\_mem.pl สำหรับการตรวจสอบหน่วยความจำหลักบนอุปกรณ์ยี่ห้อ HP Procurve

**ตารางที่ 4.84** ไฟล์ check\_snmp\_mem.pl สำหรับการตรวจสอบหน่วยความจำหลัก HP Procurve

ชื่อ	คำอธิบาย
my \$hp_mem_pool = "1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1";	การใช้งานหน่วยความจำหลักร่วมกันในอุปกรณ์ HP Procurve
my \$hp_mem_index = "1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1.1";	ค่าดัชนีในการใช้งานหน่วยความจำหลักของอุปกรณ์ HP Procurve
my \$hp_mem_total = "1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1.5";	หน่วยความจำหลักทั้งหมดในอุปกรณ์ HP Procurve
my \$hp_mem_free = "1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1.6";	พื้นที่ว่างของหน่วยความจำหลักในอุปกรณ์ HP Procurve
my \$hp_mem_free_seg = "1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1.3";	พื้นที่ว่างในส่วนของการแบ่งพื้นที่ของหน่วยความจำหลักในอุปกรณ์ HP Procurve

ตัวอย่างเลข OID สำหรับการตรวจสอบหน่วยความจำหลักบนอุปกรณ์ Cisco

"1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1"	การใช้งานหน่วยความจำหลักร่วมกันในอุปกรณ์
"1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1.1"	ค่าดัชนีในการใช้งานหน่วยความจำหลักของอุปกรณ์
"1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1.5"	หน่วยความจำหลักทั้งหมดในอุปกรณ์
"1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1.6"	พื้นที่ว่างของหน่วยความจำหลักในอุปกรณ์
"1.3.6.1.4.1.11.2.14.11.5.1.1.2.2.1.1.3"	พื้นที่ว่างในส่วนของการแบ่งพื้นที่ของหน่วยความจำหลักในอุปกรณ์

### การตรวจสอบสถานะการทำงานของไฟล์ check\_snmp\_process.pl

```
root@Nagios-RPi:~# nano /usr/local/nagios/libexec/check_snmp_process.pl
```

```
# Help : ./check_snmp_load.pl -h

use strict;
use Net::SNMP;
use Getopt::Long;

ส่วนที่ 1
# Nagios specific
my $TIMEOUT = 15;
my %ERRORS=('OK'=>0,'WARNING'=>1,'CRITICAL'=>2,'UNKNOWN'=>3,'DEPENDENT'=>4);

ส่วนที่ 2
# SNMP Datas
my $process_table= '1.3.6.1.2.1.25.4.2.1';
my $index_table = '1.3.6.1.2.1.25.4.2.1.1';
my $run_name_table = '1.3.6.1.2.1.25.4.2.1.2';
my $run_path_table = '1.3.6.1.2.1.25.4.2.1.4';
my $run_param_table = '1.3.6.1.2.1.25.4.2.1.5';
my $proc_mem_table = '1.3.6.1.2.1.25.5.1.1.2'; # Kbytes
my $proc_cpu_table = '1.3.6.1.2.1.25.5.1.1.1'; # Centi sec of CPU
my $proc_run_state = '1.3.6.1.2.1.25.4.2.1.7';
```

ส่วนที่ 1 คือ ลักษณะสถานะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการที่แสดงผ่านเว็บไซต์

**ตารางที่ 4.85** ไฟล์ check\_snmp\_process.pl สถานะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการที่แสดงผ่านเว็บไซต์

ชื่อ	คำอธิบาย
TIMEOUT = 15	การตรวจเช็คค่าในแต่ละครั้งภายใน 15 นาที
'OK'=>0	สถานะการทำงานเท่ากับหรือมากกว่า 0 คือ ปกติ มีการแสดงสถานะเป็นสีเขียว
'WARNING'=>1	สถานะการทำงานเท่ากับหรือมากกว่า 1 คือ เตือน มีการแสดงสถานะเป็นสีเหลือง
'CRITICAL'=>2	สถานะการทำงานเท่ากับหรือมากกว่า 2 คือ วิกฤติ มีการแสดงสถานะเป็นสีแดง
'UNKNOWN'=>3	สถานะการทำงานเท่ากับหรือมากกว่า 3 คือ ไม่รู้จัก มีการแสดงสถานะเป็นสีส้ม
'DEPENDENT'=>4	สถานะการทำงานเท่ากับหรือมากกว่า 4 คือ ขึ้นอยู่กับการทำงานของอุปกรณ์

ส่วนที่ 2 แสดงรายละเอียดของ check\_snmp\_process.pl สำหรับการตรวจสอบข้อกำหนดของข้อมูลในการจัดการของกระบวนการที่อุปกรณ์เครือข่ายใช้ในการแลกเปลี่ยน

**ตารางที่ 4.86** ไฟล์ check\_snmp\_process.pl สถานะการทำงานของกระบวนการบนเครื่องบริการ

ชื่อ	คำอธิบาย
my \$process_table= '1.3.6.1.2.1.25.4.2.1';	ตารางกระบวนการการทำงาน
\$index_table = '1.3.6.1.2.1.25.4.2.1.1';	ค่าดัชนีของการจัดเก็บข้อมูลของอุปกรณ์เครือข่ายในลักษณะตารางที่ไม่ซ้ำกัน
my \$run_name_table = '1.3.6.1.2.1.25.4.2.1.2';	ชื่อตารางที่เกี่ยวกับการทำงานของซอฟต์แวร์ที่ทำงานอยู่
my \$run_path_table = '1.3.6.1.2.1.25.4.2.1.4';	รายละเอียดของสถานที่ที่ใช้ในการจัดเก็บข้อมูล

ชื่อ	คำอธิบาย
my \$run_param_table = '1.3.6.1.2.1.25.4.2.1.5';	คำอธิบายของพารามิเตอร์
my \$proc_mem_table = '1.3.6.1.2.1.25.5.1.1.2';	จำนวนหน่วยความจำหลักทั้งหมดในระบบ
my \$proc_cpu_table = '1.3.6.1.2.1.25.5.1.1.1';	จำนวน ทรัพยากรที่ผู้ใช้ในกระบวนการ มีหน่วย เป็น centi-seconds ของ CPU ทั้งหมดของ ระบบ
my \$proc_run_state = '1.3.6.1.2.1.25.4.2.1.7';	สถานะของซอฟต์แวร์ที่มีการตั้งค่าที่ไม่ถูกต้อง และทำให้เกิดข้อผิดพลาด

ตัวอย่างเลข OID สำหรับการตรวจสอบสถานะการทำงานของกระบวนการ	
'1.3.6.1.2.1.25.4.2.1'	ซอฟต์แวร์ที่ทำงานบน Host จากการติดตั้งตารางซอฟต์แวร์ที่มีข้อมูล สำหรับซอฟต์แวร์ที่จัดเก็บไว้บน Host ทุกส่วนของซอฟต์แวร์ที่ทำงานจะพบว่าการติดตั้งตาราง ซอฟต์แวร์ที่เป็นจริงของซอฟต์แวร์ที่ถูกโหลดและทำงานจากแหล่งที่ไม่ใช่ Local
'1.3.6.1.2.1.25.4.2.1.1'	ค่าที่ไม่ซ้ำกันของซอฟต์แวร์ที่ทำงานบน Host
'1.3.6.1.2.1.25.4.2.1.2'	คำอธิบายเกี่ยวกับการทำงานของซอฟต์แวร์รวมทั้งผู้ผลิตที่รู้จักกัน ทั่วไป ถ้าซอฟต์แวร์นี้ถูกติดตั้งในพื้นที่เดียวกันก็จะเป็นสตริงที่สอดคล้องกัน
'1.3.6.1.2.1.25.4.2.1.4'	รายละเอียดของสถานที่ในการจัดเก็บข้อมูลในระยะยาว เช่น ดิสก์ ไดรฟ์
'1.3.6.1.2.1.25.4.2.1.5'	คำอธิบายของพารามิเตอร์ของซอฟต์แวร์เมื่อถูกโหลดในขั้นแรก
'1.3.6.1.2.1.25.5.1.1.2'	จำนวนทั้งหมดของหน่วยความจำหลักที่ระบบได้จัดสรรกับ กระบวนการ
'1.3.6.1.2.1.25.5.1.1.1'	จำนวน centi-seconds ของ CPU ทั้งหมดของระบบทรัพยากรที่ บริโภคโดยกระบวนการนี้ โปรดทราบว่าใน multi-processor ค่านี้อาจเพิ่มขึ้นโดย มากกว่าหนึ่ง centi-second ในหนึ่ง centi-second ของเวลาจริง
'1.3.6.1.2.1.25.4.2.1.7'	สถานะของการทำงานของซอฟต์แวร์สำหรับการตั้งค่าที่ไม่ถูกต้องทำ ให้ซอฟต์แวร์หยุดทำงานและยกเลิกการโหลด



### การตรวจสอบสถานะการทำงานของไฟล์ check\_snmp\_storage.pl

root@Nagios-RPi:~# nano /usr/local/nagios/libexec/check\_snmp\_storage.pl

```
# Help : ./check_snmp_load.pl -h

use strict;
use Net::SNMP;
use Getopt::Long;

ส่วนที่ 1
# Nagios specific
#my $TIMEOUT = 15;
#my %ERRORS =
('OK'=>0,'WARNING'=>1,'CRITICAL'=>2,'UNKNOWN'=>3,'DEPENDENT'=>4);

ส่วนที่ 2
# SNMP Datas
my $storage_table= '1.3.6.1.2.1.25.2.3.1';
my $storagetype_table = '1.3.6.1.2.1.25.2.3.1.2';
my $index_table = '1.3.6.1.2.1.25.2.3.1.1';
my $descr_table = '1.3.6.1.2.1.25.2.3.1.3';
my $size_table = '1.3.6.1.2.1.25.2.3.1.5';
my $used_table = '1.3.6.1.2.1.25.2.3.1.6';
my $alloc_units = '1.3.6.1.2.1.25.2.3.1.4';

ส่วนที่ 3
my %hrStorage;
$hrStorage{"Other"} = '1.3.6.1.2.1.25.2.1.1';
$hrStorage{"1.3.6.1.2.1.25.2.1.1"} = 'Other';
$hrStorage{"Ram"} = '1.3.6.1.2.1.25.2.1.2';
$hrStorage{"1.3.6.1.2.1.25.2.1.2"} = 'Ram';
$hrStorage{"VirtualMemory"} = '1.3.6.1.2.1.25.2.1.3';
```

```

$hrStorage{"1.3.6.1.2.1.25.2.1.3"} = 'VirtualMemory';
$hrStorage{"FixedDisk"} = '1.3.6.1.2.1.25.2.1.4';
$hrStorage{"1.3.6.1.2.1.25.2.1.4"} = 'FixedDisk';
$hrStorage{"RemovableDisk"} = '1.3.6.1.2.1.25.2.1.5';
$hrStorage{"1.3.6.1.2.1.25.2.1.5"} = 'RemovableDisk';
$hrStorage{"FloppyDisk"} = '1.3.6.1.2.1.25.2.1.6';
$hrStorage{"1.3.6.1.2.1.25.2.1.6"} = 'FloppyDisk';
$hrStorage{"CompactDisk"} = '1.3.6.1.2.1.25.2.1.7';
$hrStorage{"1.3.6.1.2.1.25.2.1.7"} = 'CompactDisk';
$hrStorage{"RamDisk"} = '1.3.6.1.2.1.25.2.1.8';
$hrStorage{"1.3.6.1.2.1.25.2.1.8"} = 'RamDisk';
$hrStorage{"FlashMemory"} = '1.3.6.1.2.1.25.2.1.9';
$hrStorage{"1.3.6.1.2.1.25.2.1.9"} = 'FlashMemory';
$hrStorage{"NetworkDisk"} = '1.3.6.1.2.1.25.2.1.10';
$hrStorage{"1.3.6.1.2.1.25.2.1.10"} = 'NetworkDisk';

```

ส่วนที่ 1 คือ ลักษณะสถานะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการที่แสดงผ่านเว็บไซต์

**ตารางที่ 4.87** ไฟล์ check\_snmp\_storage.pl สถานะการทำงานของอุปกรณ์เครือข่าย และเครื่องบริการที่แสดงผ่านเว็บไซต์

ชื่อ	คำอธิบาย
TIMEOUT = 15	การตรวจเช็คค่าในแต่ละครั้งภายใน 15 นาที
'OK'=>0	สถานะการทำงานเท่ากับหรือมากกว่า 0 คือ ปกติ มีการแสดงสถานะเป็นสีเขียว
'WARNING'=>1	สถานะการทำงานเท่ากับหรือมากกว่า 1 คือ เตือน มีการแสดงสถานะเป็นสีเหลือง
'CRITICAL'=>2	สถานะการทำงานเท่ากับหรือมากกว่า 2 คือ วิกฤติ มีการแสดงสถานะเป็นสีแดง
'UNKNOWN'=>3	สถานะการทำงานเท่ากับหรือมากกว่า 3 คือ ไม่รู้จัก มีการแสดงสถานะเป็นสีส้ม

ชื่อ	คำอธิบาย
'DEPENDENT'=>4	สถานะการทำงานเท่ากับหรือมากกว่า 4 คือ ขึ้นอยู่กับการทำงานของอุปกรณ์

ส่วนที่ 2 แสดงรายละเอียดของ check\_snmp\_storage.pl สำหรับการตรวจสอบข้อมูลที่เกี่ยวข้องกับหน่วยจัดเก็บข้อมูล

**ตารางที่ 4.88** ไฟล์ check\_snmp\_process.pl สถานะการทำงานเกี่ยวกับหน่วยจัดเก็บข้อมูลบนเครื่องบริการ ในการทำงานส่วน SNMP Datas

ชื่อ	คำอธิบาย
my \$storage_table= '1.3.6.1.2.1.25.2.3.1';	แหล่งเก็บข้อมูลของอุปกรณ์เครือข่ายจัดเก็บไว้ในลักษณะตาราง
my \$storagetype_table = '1.3.6.1.2.1.25.2.3.1.2';	ประเภทของการจัดเก็บข้อมูลของอุปกรณ์เครือข่ายในลักษณะตาราง
my \$index_table = '1.3.6.1.2.1.25.2.3.1.1';	ค่าดัชนีของการจัดเก็บข้อมูลของอุปกรณ์เครือข่ายในลักษณะตาราง
my \$descr_table = '1.3.6.1.2.1.25.2.3.1.3';	คำอธิบายของการจัดเก็บข้อมูลของอุปกรณ์เครือข่ายในลักษณะตาราง
my \$size_table = '1.3.6.1.2.1.25.2.3.1.5';	ขนาดของการจัดเก็บข้อมูลของอุปกรณ์เครือข่ายในลักษณะตาราง
my \$used_table = '1.3.6.1.2.1.25.2.3.1.6';	การใช้การจัดเก็บข้อมูลของอุปกรณ์เครือข่ายในลักษณะตาราง
my \$alloc_units = '1.3.6.1.2.1.25.2.3.1.4';	การเตรียมพื้นที่ในหน่วยความจำสำหรับอุปกรณ์เครือข่าย

ตัวอย่างเลข OID สำหรับการตรวจสอบ SNMP Datas

'1.3.6.1.2.1.25.2.3.1'	เป็นพื้นที่สำหรับจัดเก็บข้อมูลเชิงตรรกะชนิดหนึ่งใน Host
'1.3.6.1.2.1.25.2.3.1.2'	เป็นประเภทของการจัดเก็บข้อมูล
'1.3.6.1.2.1.25.2.3.1.1'	เป็นค่าที่ไม่ซ้ำกันสำหรับแต่ละพื้นที่จัดเก็บข้อมูลเชิงตรรกะที่มีอยู่ใน Host

'1.3.6.1.2.1.25.2.3.1.3'	คำอธิบายของประเภท และตัวอย่างของการจัดเก็บข้อมูล
'1.3.6.1.2.1.25.2.3.1.5.'	ขนาดของการจัดเก็บในหน่วยความจำ ซึ่งสามารถอนุญาตให้ตั้งค่าระยะไกลของขนาดพื้นที่จัดเก็บข้อมูล
'1.3.6.1.2.1.25.2.3.1.6.'	การจัดเก็บแทนโดยรายการที่มีการแบ่งส่วนในหน่วยของ hrStorage
'1.3.6.1.2.1.25.2.3.1.4.'	ขนาดของข้อมูลในการจัดสรรสำหรับการตรวจสอบบล็อก บัฟเฟอร์ หรือแพ็คเกจ

ส่วนที่ 3 แสดงรายละเอียดของ check\_snmp\_storage.pl สำหรับการตรวจสอบ Storage type definition – from /usr/share/snmp/mibs/HOST-RESOURCES-TYPES เป็นการตรวจสอบประเภทของการเก็บทรัพยากร Host

ตารางที่ 4.89 ไฟล์ check\_snmp\_process.pl สำหรับตรวจสอบ Storage Types

ชื่อ	คำอธิบาย
\$hrStorage{"Other"} = '1.3.6.1.2.1.25.2.1.1';	แหล่งอื่น ๆ ที่ใช้เก็บข้อมูลของอุปกรณ์
\$hrStorage{"1.3.6.1.2.1.25.2.1.1"} = 'Other';	การจัดเก็บข้อมูลของอุปกรณ์เครือข่ายจากแหล่งอื่น ๆ
\$hrStorage{"Ram"} = '1.3.6.1.2.1.25.2.1.2';	หน่วยความจำหลักที่ใช้ในการระบุประเภทของการจัดเก็บข้อมูล
\$hrStorage{"1.3.6.1.2.1.25.2.1.2"} = 'Ram';	ประเภทของการจัดเก็บข้อมูลที่ได้จากหน่วยความจำหลัก
\$hrStorage{"VirtualMemory"} = '1.3.6.1.2.1.25.2.1.3';	หน่วยความจำเสมือนที่ใช้ในการระบุประเภทของการจัดเก็บข้อมูล
\$hrStorage{"1.3.6.1.2.1.25.2.1.3"} = 'VirtualMemory';	ประเภทของการจัดเก็บข้อมูลที่ได้จากหน่วยความจำเสมือน
\$hrStorage{"FixedDisk"} = '1.3.6.1.2.1.25.2.1.4';	อุปกรณ์จัดเก็บข้อมูลแม่เหล็กที่การจัดเก็บข้อมูลไม่สามารถถอดออกได้
\$hrStorage{"1.3.6.1.2.1.25.2.1.4"} = 'FixedDisk';	การจัดเก็บข้อมูลไม่สามารถถอดออกได้จากอุปกรณ์จัดเก็บข้อมูลแบบแม่เหล็ก
\$hrStorage{"RemovableDisk"} = '1.3.6.1.2.1.25.2.1.5';	อุปกรณ์จัดเก็บข้อมูลแม่เหล็กที่การจัดเก็บข้อมูลที่สามารถถอดออกได้

ชื่อ	คำอธิบาย
\$hrStorage{"1.3.6.1.2.1.25.2.1.5"} = 'RemovableDisk';	การจัดเก็บข้อมูลที่สามารถถอดออกได้จาก อุปกรณ์จัดเก็บข้อมูลแบบแม่เหล็ก
\$hrStorage{"FloppyDisk"} = '1.3.6.1.2.1.25.2.1.6';	อุปกรณ์จัดเก็บข้อมูลแบบแม่เหล็กที่การจัดเก็บ ข้อมูลที่ใช้สำหรับการไม่หมุน
\$hrStorage{"1.3.6.1.2.1.25.2.1.6"} = 'FloppyDisk';	การจัดเก็บข้อมูลที่ใช้สำหรับการไม่หมุนของ อุปกรณ์จัดเก็บข้อมูลแบบแม่เหล็ก
\$hrStorage{"CompactDisk"} = '1.3.6.1.2.1.25.2.1.7';	อุปกรณ์จัดเก็บข้อมูลแบบออปติคอลลสำหรับ ข้อมูลที่ใช้สำหรับอ่านอย่างเดียว
\$hrStorage{"1.3.6.1.2.1.25.2.1.7"} = 'CompactDisk';	ข้อมูลที่ใช้สำหรับอ่านอย่างเดียวจากอุปกรณ์ จัดเก็บข้อมูลแบบออปติคอลล
\$hrStorage{"RamDisk"} = '1.3.6.1.2.1.25.2.1.8';	หน่วยความจำหลักที่ใช้สำหรับเก็บข้อมูลของ ระบบแฟ้มที่จะถูกเก็บ
\$hrStorage{"1.3.6.1.2.1.25.2.1.8"} = 'RamDisk';	การเก็บข้อมูลของระบบแฟ้มที่จะถูกเก็บจาก หน่วยความจำหลัก
\$hrStorage{"FlashMemory"} = '1.3.6.1.2.1.25.2.1.9';	หน่วยความจำแบบแฟลชที่ใช้สำหรับเก็บข้อมูล
\$hrStorage{"1.3.6.1.2.1.25.2.1.9"} = 'FlashMemory';	การจัดเก็บข้อมูลจากหน่วยความจำแบบแฟลช
\$hrStorage{"NetworkDisk"} = '1.3.6.1.2.1.25.2.1.10';	ระบบแฟ้มเครือข่ายที่ใช้ในการจัดเก็บข้อมูล
\$hrStorage{"1.3.6.1.2.1.25.2.1.10"} = 'NetworkDisk';	การจัดเก็บข้อมูลที่ใช้สำหรับระบบแฟ้มเครือข่าย

ตัวอย่างเลข OID สำหรับการตรวจสอบ Storage type definition – from  
/usr/share/snmp/mibs/HOST-RESOURCES-TYPES

'1.3.6.1.2.1.25.2.1.1' เป็นการระบุประเภทของการจัดเก็บข้อมูลที่นำมาใช้เมื่อไม่มีประเภทอื่นที่  
เหมาะสม

'1.3.6.1.2.1.25.2.1.2'	เป็นการระบุประเภทของการจัดเก็บข้อมูลที่ใช้สำหรับ RAM
'1.3.6.1.2.1.25.2.1.3'	เป็นการระบุประเภทของการจัดเก็บข้อมูลที่ใช้สำหรับหน่วยความจำ เสมือน การจัดเก็บชั่วคราวของหน่วยความจำสลับ หรือเพจ
'1.3.6.1.2.1.25.2.1.4'	เป็นการระบุประเภทของการจัดเก็บข้อมูลที่ไม่สามารถถอดออกได้ของ อุปกรณ์จัดเก็บข้อมูลแบบจานแม่เหล็กหมุน
'1.3.6.1.2.1.25.2.1.5'	เป็นการระบุประเภทของการจัดเก็บข้อมูลที่สามารถถอดออกได้ของ อุปกรณ์จัดเก็บข้อมูลแบบจานแม่เหล็กหมุน
'1.3.6.1.2.1.25.2.1.6'	เป็นการระบุประเภทของการจัดเก็บข้อมูลที่ใช้สำหรับอุปกรณ์จัดเก็บ ข้อมูลแบบจานแม่เหล็กไม่หมุน
'1.3.6.1.2.1.25.2.1.7'	เป็นการระบุประเภทของการจัดเก็บข้อมูลที่ใช้สำหรับอ่านอย่างเดียวของ อุปกรณ์จัดเก็บข้อมูลแบบออปติคอลล
'1.3.6.1.2.1.25.2.1.8'	เป็นการระบุประเภทของการจัดเก็บข้อมูลที่ใช้สำหรับระบบแฟ้มที่ถูก จัดเก็บไว้ใน RAM
'1.3.6.1.2.1.25.2.1.9'	เป็นการระบุประเภทของการจัดเก็บข้อมูลที่ใช้สำหรับหน่วยความจำแบบ แฟลช
'1.3.6.1.2.1.25.2.1.10'	เป็นการระบุประเภทของการจัดเก็บข้อมูลที่ใช้สำหรับระบบแฟ้มเครือข่าย

### การตรวจสอบการทำงานของระบบจากไฟล์

ในการทำงานของระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย โพรแกมวิชา  
คอมพิวเตอร์ นั้นสามารถดูข้อมูลเหตุการณ์การทำงาน (Log File) ที่เก็บการทำงานของระบบไว้ใน  
ไฟล์ ใน 1 ไฟล์จะเก็บการทำงานของระบบไว้ 1 วัน ซึ่งผู้ดูแลระบบสามารถเลือกดูได้ตามวันที่ต้องการ

```
root@Nagios-RPi:~# cd /usr/local/nagios/var/archives/
root@Nagios-RPi: /usr/local/nagios/var/archives/# ls
nagios-08-10-2017-00  nagios-08-11-2017-00  nagios-08-12-2017-00
nagios-08-13-2017-00  nagios-08-14-2017-00  nagios-08-15-2017-00
nagios-08-16-2017-00  nagios-08-17-2017-00  nagios-08-18-2017-00
```

ตัวอย่าง Log File ที่ได้เก็บเหตุการณ์การทำงานของระบบ ในวันที่ 18 สิงหาคม 2560

```

root@Nagios-RPi: /usr/local/nagios/var/archives/# nano nagios-08-18-2017-00
[1502902800] LOG ROTATION: DAILY
[1502902800] LOG VERSION: 2.0
[1502902800] CURRENT HOST STATE: AP-8-401[.2];UP;HARD;1;PING OK - Packet loss =
0%, RTA = 0.90 ms
[1502902800] CURRENT HOST STATE: AP-8-406[.3];UP;HARD;1;PING OK - Packet loss =
0%, RTA = 0.74 ms
[1502902800] CURRENT HOST STATE: NASServer.16;UP;HARD;1;PING OK - Packet loss
= 0%, RTA = 0.42 ms
[1502902800] CURRENT HOST STATE: VMServer.18;UP;HARD;1;PING OK - Packet loss =
0%, RTA = 0.67 ms
[1502902800] CURRENT HOST STATE: cacti;UP;HARD;1;PING OK - Packet loss = 0%,
RTA = 0.72 ms
[1502902800] CURRENT HOST STATE: gateway-8402;UP;HARD;1;PING OK - Packet loss
= 0%, RTA = 0.48 ms
[1502902800] CURRENT HOST STATE: localhost;UP;HARD;1;PING OK - Packet loss =
0%, RTA = 0.17 ms
[1502902800] CURRENT HOST STATE: server8407;DOWN;HARD;10;(Host check timed
out after 30.01 seconds)
[1502902800] CURRENT HOST STATE: webcomsci;DOWN;HARD;10;(Host check timed
out after 31.02 seconds)
[1502902800] CURRENT SERVICE STATE: AP-8-401[.2];PING;OK;HARD;1;PING OK -
Packet loss = 0%, RTA = 0.94 ms
[1502902800] CURRENT SERVICE STATE: AP-8-406[.3];PING;OK;HARD;1;PING OK - Packet
loss = 0%, RTA = 0.88 ms

```

วิธีการอ่าน Log File ในฐานข้อมูลข้างต้น เช่น บรรทัด [1502902800] CURRENT HOST STATE: gateway-8402;UP;HARD;1;PING OK - Packet loss = 0%, RTA = 0.48 ms หมายถึง เป็นการตรวจสอบสถานะของ Host ชื่อ gateway-8402 มีสถานะเป็น Up มีการตอบสนองการ Ping แปกเก็ตที่ส่งไปมีการสูญเสียเท่ากับ 0%

## การทดสอบระบบ

หลังจากที่ผู้พัฒนาได้ทำการพัฒนาระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่ายเสร็จแล้ว ผู้พัฒนาได้ทำการทดสอบการใช้งานระบบ โดยแบ่งออกเป็น 1 กลุ่ม คือ

1. ผู้ดูแลระบบจัดการข้อมูลการเข้าใช้งานระบบผ่านทางโปรแกรม Secure Shell โดยผู้ดูแลระบบจะกำหนดรหัสผ่านเริ่มต้นสำหรับเข้าใช้งานระบบผ่านทาง Nagios Browser ซึ่งมีหลักการทำงานดังนี้

1.1. การ Login เข้าสู่ระบบผ่านทางโปรแกรม Secure Shell ซึ่งมีขั้นตอนดังนี้

1.1.1. โดยผู้ดูแลระบบกำหนดรหัสผ่านเริ่มต้นการเข้าระบบ

1.1.2. ผู้ดูแลระบบเข้าสู่ระบบโดยการเข้าแบบ Root เพื่อความปลอดภัย

1.2. การเพิ่มข้อมูลอุปกรณ์ที่ต้องการตรวจสอบใน Nagios Server โดยมีขั้นตอนดังนี้

1.2.1. เข้าไปยังไดเรกทอรีที่เก็บไฟล์ที่ใช้ในการทำงานของ Nagios โดยใช้คำสั่ง `root@Nagios-RPi: cd /usr/local/nagios/etc/objects/` ซึ่งภายในจะมีไฟล์ดังนี้ `commands.cfg`, `contacts.cfg`, `localhost.cfg`, `printer.cfg`, `switch.cfg`, `templates.cfg`, `timeperiods.cfg` และ `windows.cfg` หลังจากนั้นผู้ใช้ทำการเปิดไฟล์ และป้อนอุปกรณ์ และบริการที่ต้องการตรวจสอบ เช่น หากผู้ใช้ต้องการตรวจสอบ Router เข้าไปยังไฟล์ `switch.cfg` ทำการป้อนที่อยู่ของอุปกรณ์ และบริการที่ต้องการตรวจสอบ เช่น Ping เป็นต้น

1.3. การตรวจสอบสถานะการทำงานของอุปกรณ์บนเครือข่ายผ่านทางเว็บไซต์ มีขั้นตอนดังนี้

1.3.1. เปิด Browser กรอก Address ของเครื่อง Nagios Server แล้วตามด้วย `/nagios` เช่น `172.17.XXX.XXX /nagios`

1.3.2. เข้าสู่ระบบโดยกรอกชื่อผู้ใช้ และรหัสผ่าน



ต้องมีการตรวจสอบความถูกต้อง

http://172.17.41.56 ต้องใช้ชื่อผู้ใช้และรหัสผ่าน

การเชื่อมต่อกับเว็บไซด์นี้ไม่ปลอดภัย

ชื่อผู้ใช้:

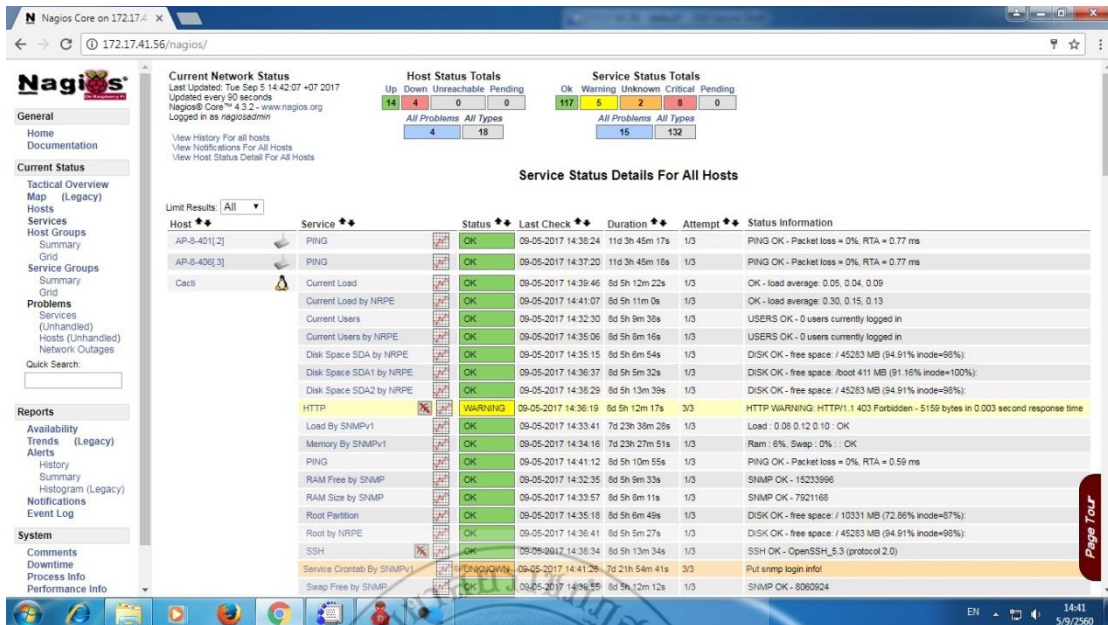
รหัสผ่าน:

ภาพที่ 4.6 หน้าจอสำหรับการ Login เพื่อตรวจสอบสถานะผ่านเว็บไซต์

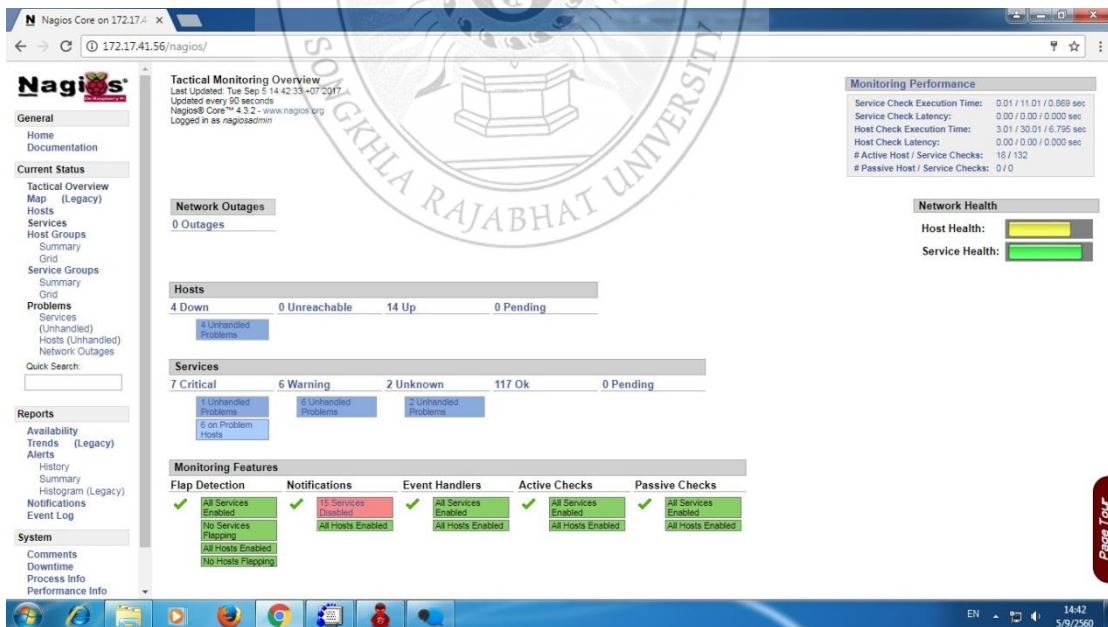
กรณีที่ 1 : ทดสอบการเข้าสู่ระบบ ในกรณีที่ระบุชื่อผู้ใช้ และรหัสผ่านไม่ถูกต้อง หรือไม่มีข้อมูล  
ผลการทดสอบ : ไม่สามารถเข้าสู่ระบบได้ เนื่องจากป้อนข้อมูลไม่ถูกต้อง ดังภาพที่ 4.1

ภาพที่ 4.7 หน้าเว็บหลักเมื่อกรอกชื่อผู้ใช้ และรหัสผ่านถูกต้อง

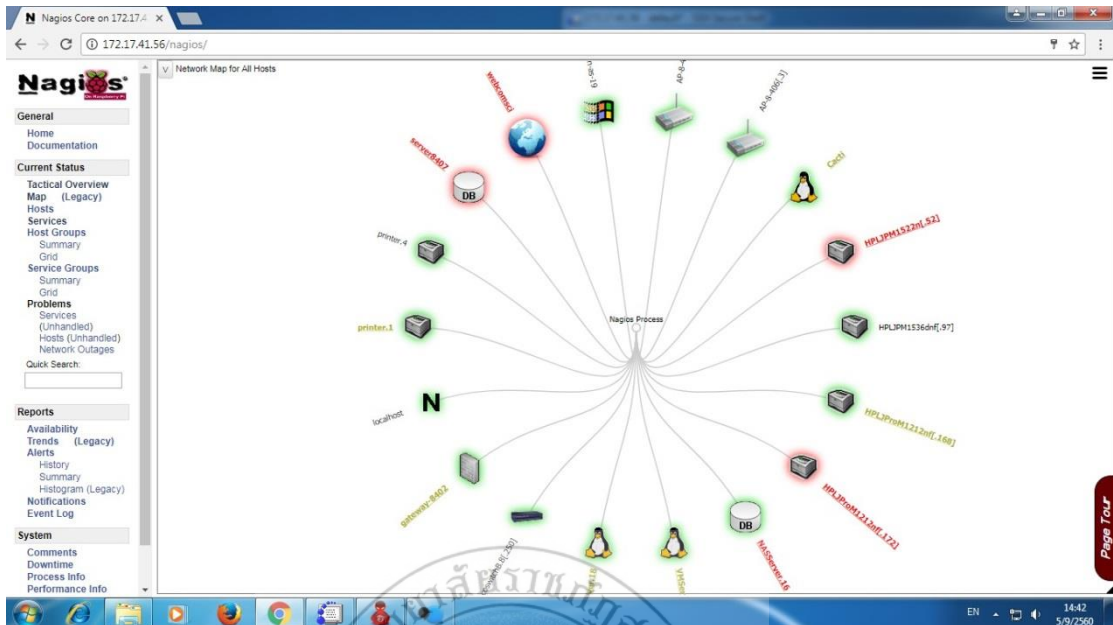
กรณีที่ 2 : ทดสอบการตรวจสอบสถานะการทำงานของอุปกรณ์บนเครือข่าย  
ผลการทดสอบ : มีแสดงผลการตรวจสอบการทำงานผ่านหน้าเว็บไซต์



ภาพที่ 4.8 เมนู Current Status ในเมนูย่อย Service แสดงการตรวจสอบสถานะของอุปกรณ์  
 เมนู Current Status ในเมนูย่อย Tactical Overview จะแสดงภาพรวมของการตรวจสอบสถานะ

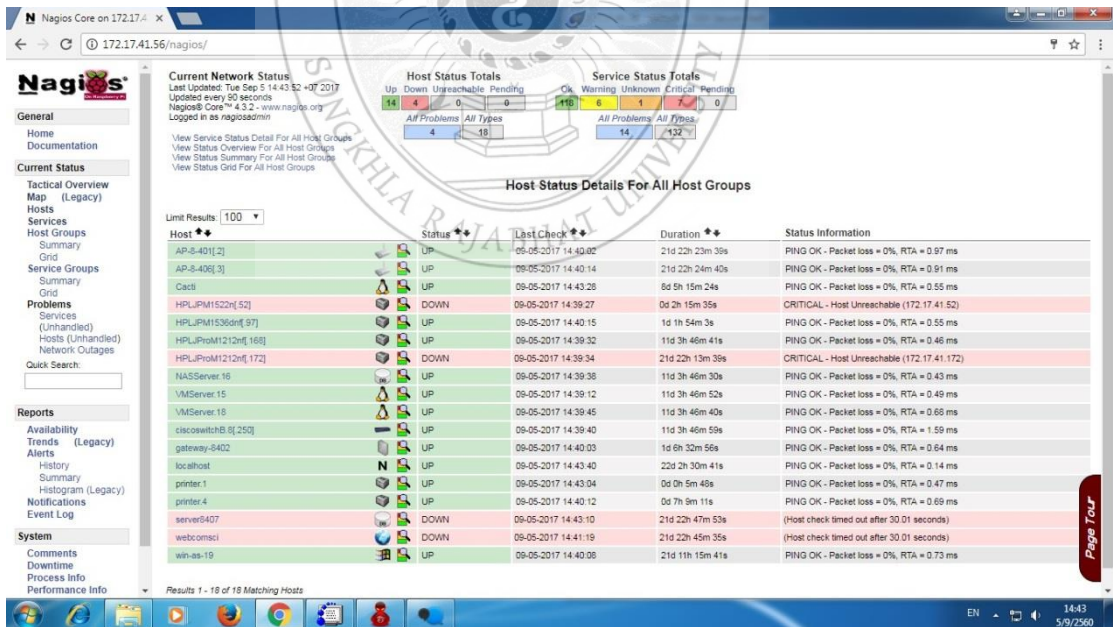


ภาพที่ 4.9 เมนู Current Status ในเมนูย่อย Tactical Overview  
 เมนู Current Status ในเมนูย่อย Map จะแสดงการตรวจสอบสถานะ ในรูปแบบแผนผังซึ่งสามารถเลือกรูปแบบได้หลายรูปแบบ



ภาพที่ 4.10 เมนู Current Status ในเมนูย่อย Map

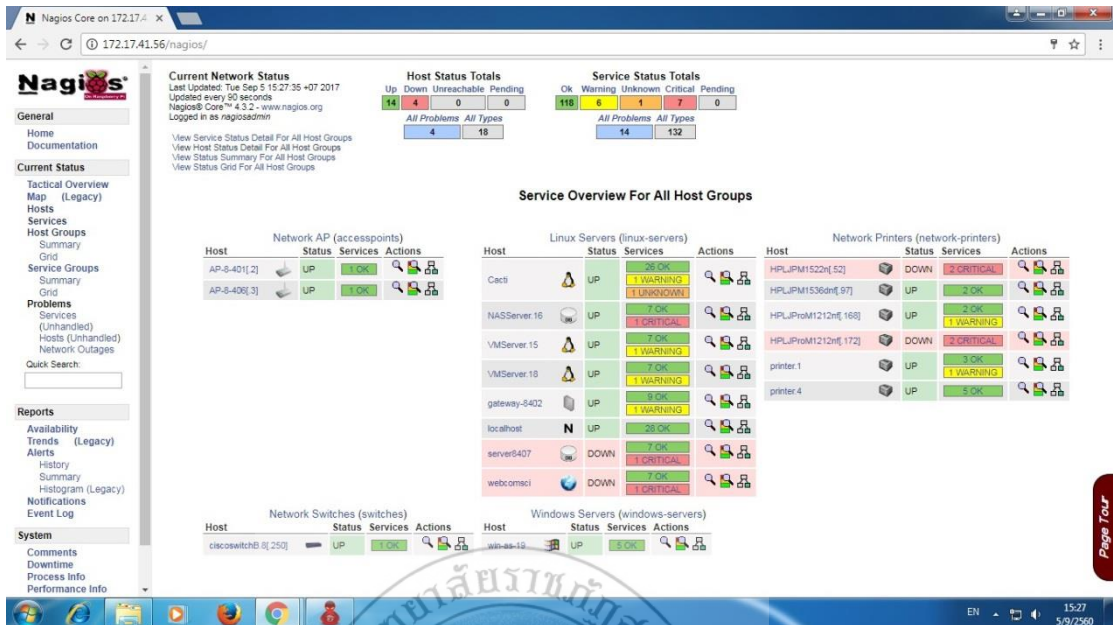
เมนู Current Status ในเมนูย่อย Hosts จะแสดงการตรวจสอบสถานะการทำงานของ Hosts ในแต่ละอุปกรณ์



ภาพที่ 4.11 เมนู Current Status ในเมนูย่อย Hosts

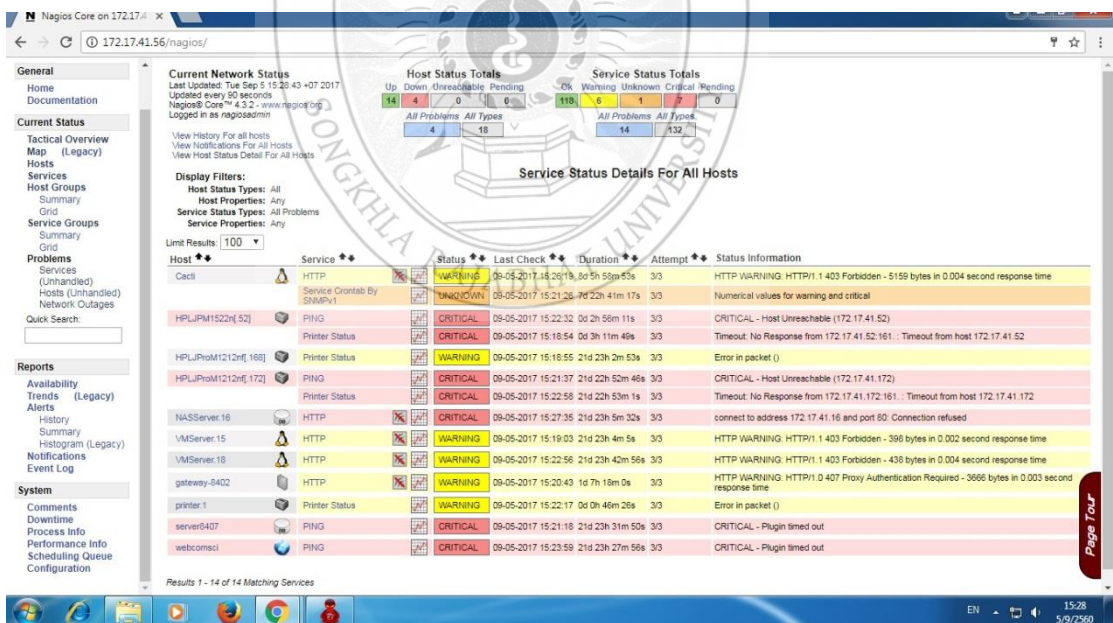
เมนู Current Status ในเมนูย่อย Host Groups จะแสดงการตรวจสอบสถานะการทำงานของ Host กลุ่ม ของ Host





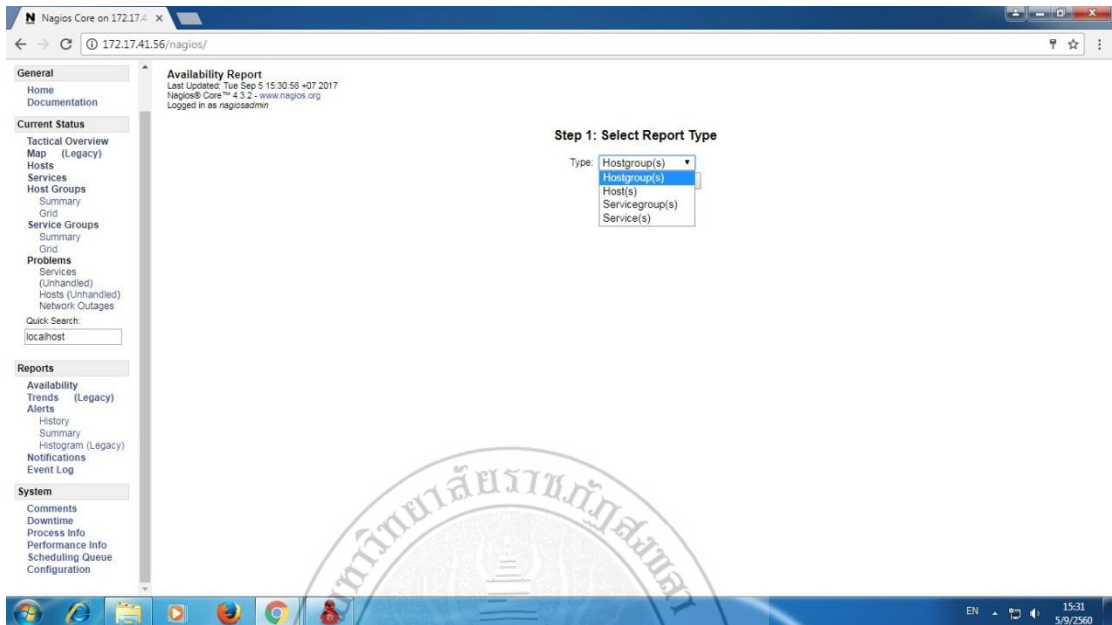
ภาพที่ 4.12 เมนู Current Status ในเมนูย่อย Host Groups

เมนู Current Status ในเมนูย่อย Problems จะแสดงการตรวจสอบสถานะการทำงานเฉพาะ Host ที่เกิดข้อผิดพลาด

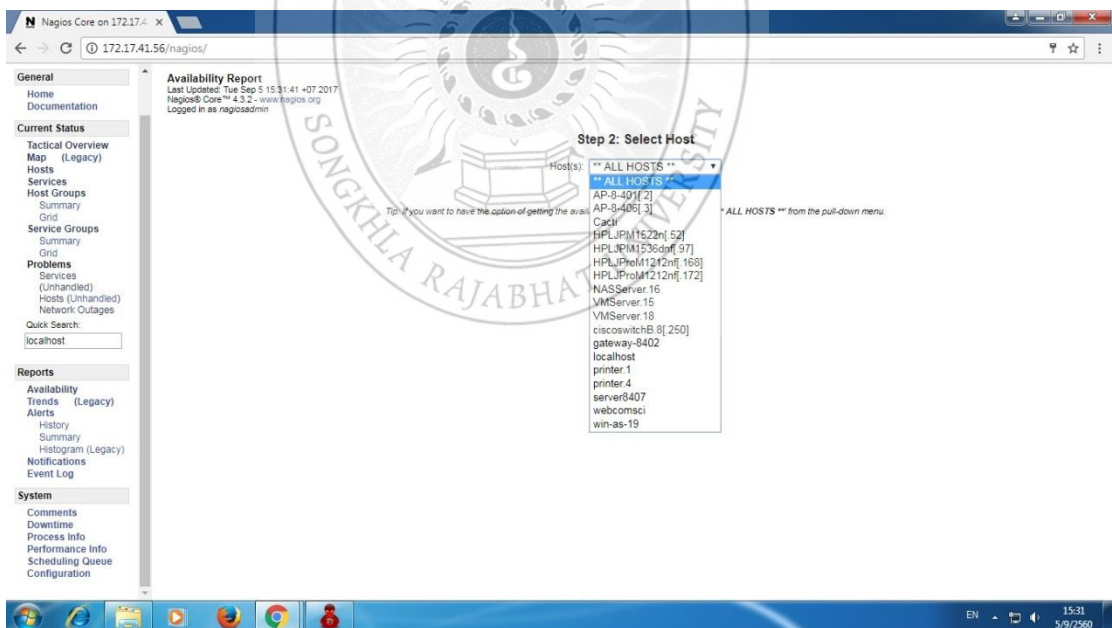


ภาพที่ 4.13 เมนู Current Status ในเมนูย่อย Problems

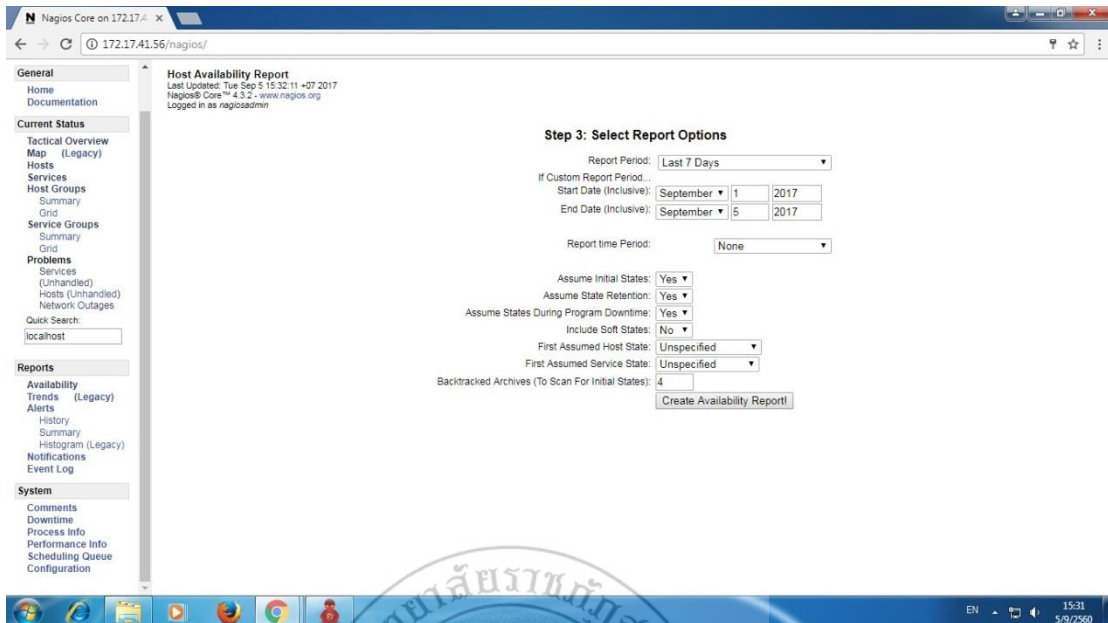
## การตรวจสอบสถานะการทำงานของอุปกรณ์ย้อนหลัง



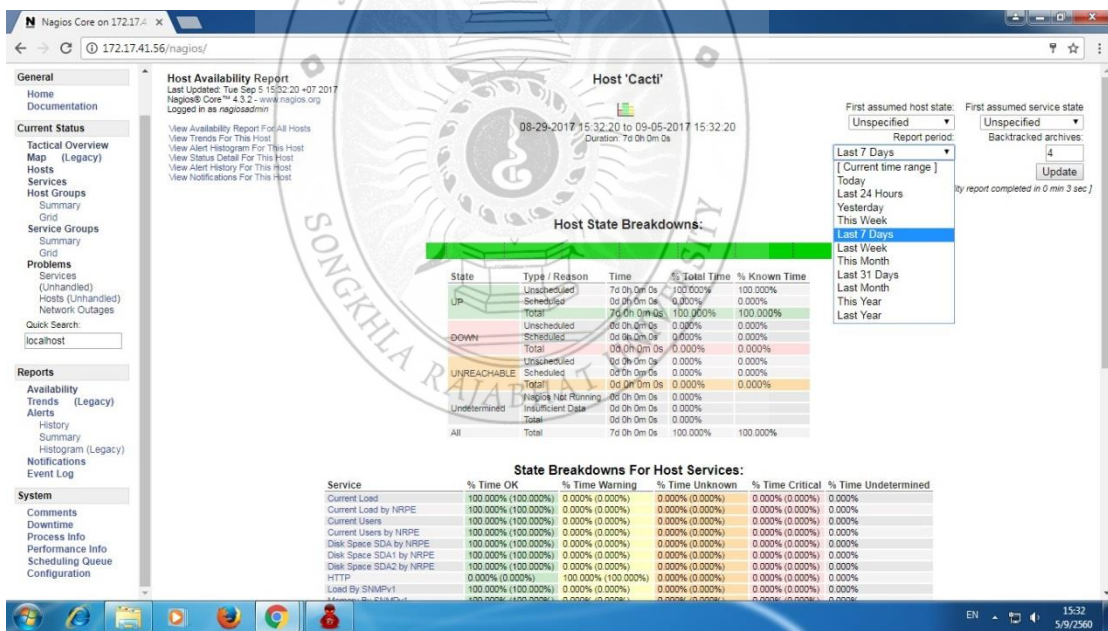
ภาพที่ 4.14 หน้าต่างสำหรับเลือกประเภทของข้อมูลที่ต้องการตรวจสอบย้อนหลัง



ภาพที่ 4.15 หน้าต่างสำหรับเลือกอุปกรณ์ที่ต้องการตรวจสอบย้อนหลัง

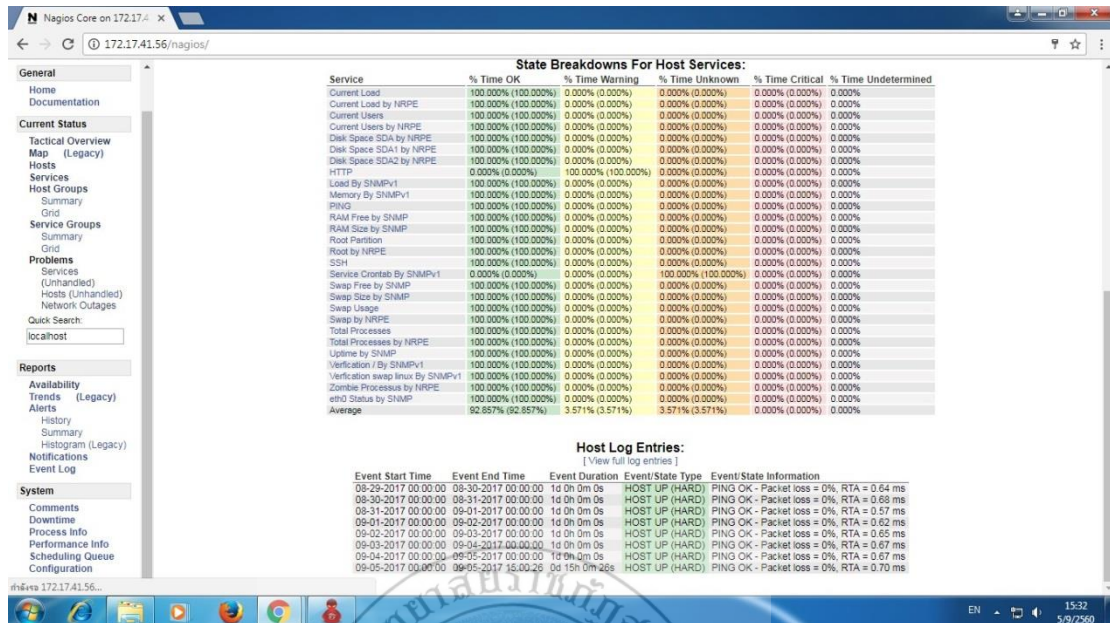


ภาพที่ 4.16 หน้าต่างสำหรับเลือกข้อมูลที่ต้องการให้แสดงการตรวจสอบย้อนหลัง



ภาพที่ 4.17 หน้าต่างการแสดงผลข้อมูลการตรวจสอบย้อนหลัง

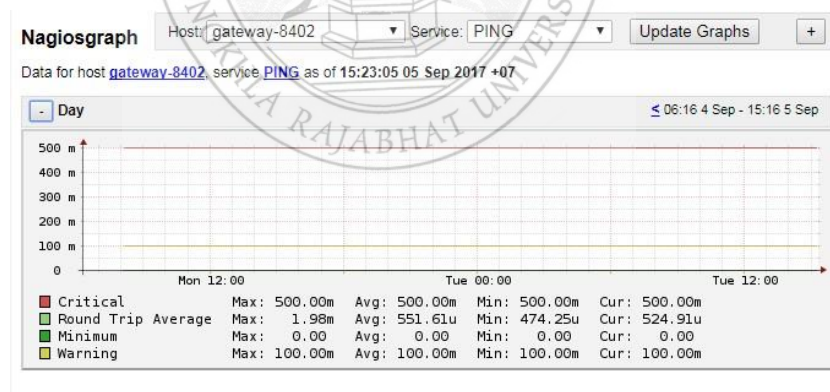




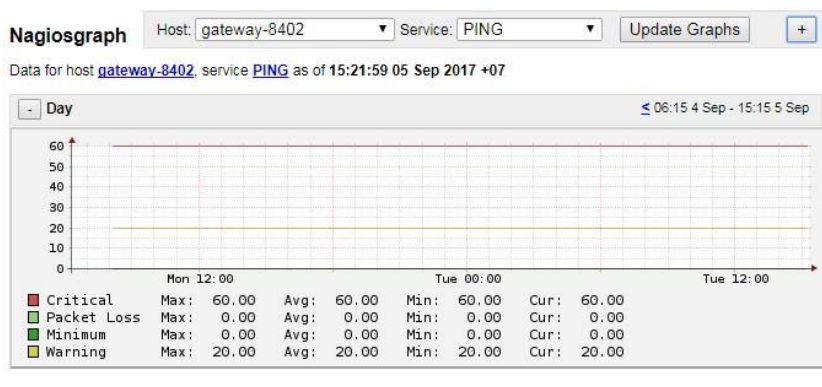
ภาพที่ 4.18 หน้าต่างการแสดงผลข้อมูลการตรวจสอบย้อนหลัง (ต่อ)

#### 1.4. การแสดงสถานะการทำงานในลักษณะของกราฟผ่านเว็บไซต์

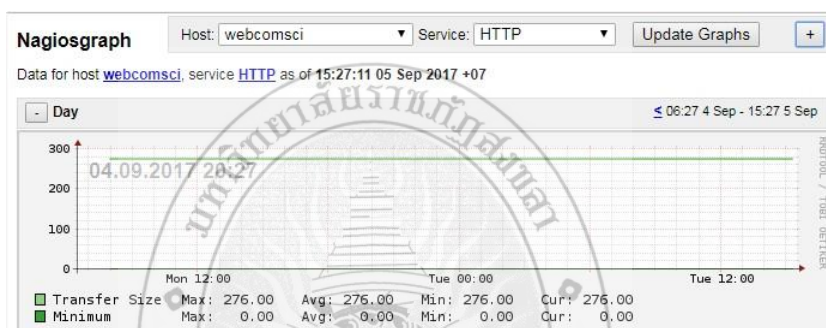
เนื่องจาก Nagios เป็น Open Source รองรับการใช้งานกราฟ PNP เพราะการวิเคราะห์ข้อมูลผลการดำเนินงานให้บริการโดย Plugin และเก็บข้อมูลเข้าสู่ฐานข้อมูลโดยอัตโนมัติ ซึ่งมีการติดตั้ง RRDTOol



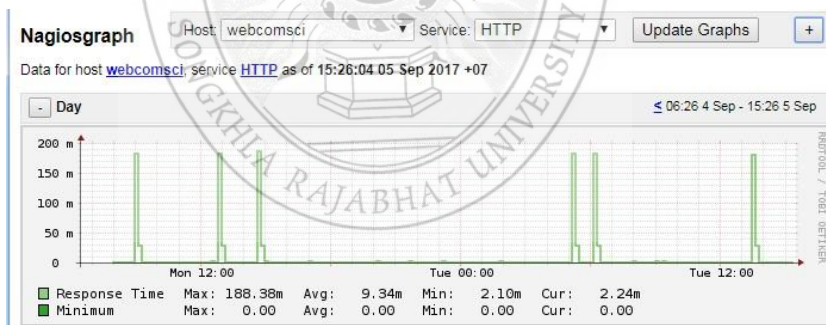
ภาพที่ 4.19 กราฟสถานะการให้บริการของ gateway-8402 โดยการ Ping การทำงานของอุปกรณ์



ภาพที่ 4.20 กราฟสถานะการให้บริการของ gateway-8402 เมื่อหยุดให้บริการ



ภาพที่ 4.21 กราฟสถานะการให้บริการของ webcomsci โดยการตรวจสอบ HTTP

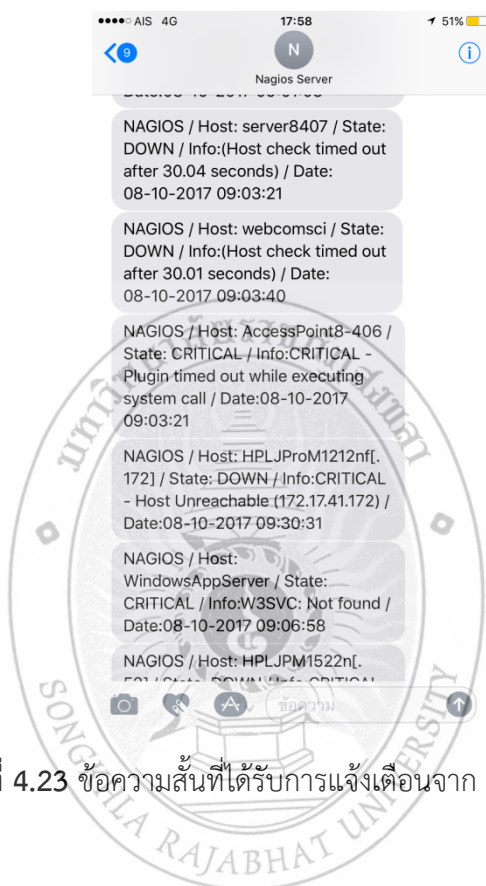


ภาพที่ 4.22 กราฟขนาดของสถานะการให้บริการของ webcomsci โดยการตรวจสอบ HTTP



1.5. การแจ้งเตือนการทำงานของระบบเมื่อเกิดสถานะหยุดการให้บริการ หรือเกิดข้อผิดพลาด

เมื่ออุปกรณ์เครือข่ายเกิดสถานะหยุดให้บริการ หรือเกิดข้อผิดพลาด เครื่องบริการ Nagios Server จะทำการส่งข้อความสั้น (SMS) แจ้งเตือนไปยังผู้ดูแลระบบ



ภาพที่ 4.23 ข้อความสั้นที่ได้รับการแจ้งเตือนจาก Nagios Server

## บทที่ 5

### บทสรุปและข้อเสนอแนะ

#### สรุปผลการดำเนินงาน

การดำเนินงานเป็นไปตามวัตถุประสงค์ที่ตั้งไว้ คือ สามารถเป็นต้นแบบในการตรวจสอบสถานะการหยุดให้บริการของอุปกรณ์และเครือข่าย กล่าวคือ

1. ระบบสามารถตรวจสอบสถานะการทำงานของอุปกรณ์บนเครือข่ายได้
2. สามารถตรวจสอบสถานะการทำงานของอุปกรณ์ผ่านทาง Web Page ได้
3. ระบบสามารถส่งข้อความสั้นไปยังผู้ดูแลระบบได้ หากอุปกรณ์ที่อยู่ในการตรวจสอบของระบบมีสถานะหยุดให้บริการ
4. ระบบสามารถแสดงกราฟข้อมูลการตรวจสอบทั้งปัจจุบัน และย้อนหลังของอุปกรณ์ที่ตรวจสอบได้
5. ผู้ดูแลระบบสามารถทำการเพิ่ม ลบ แก้ไข อุปกรณ์ที่ต้องการตรวจสอบได้
6. ผู้ดูแลระบบสามารถดูสถานะการให้บริการผ่านเครือข่ายเดียวกันได้

#### ปัญหาและอุปสรรคในการดำเนินงาน

ปัญหาในการเพิ่มอุปกรณ์บนเครือข่ายที่จะนำมาตรวจสอบ ผู้ดูแลระบบจะต้องทราบข้อมูลเกี่ยวกับอุปกรณ์ที่จะนำมาตรวจสอบ ว่าอุปกรณ์นั้นเป็นอุปกรณ์ประเภทใด ใช้ระบบปฏิบัติการใด และอุปกรณ์นั้นเปิดบริการไต่บ้าง สามารถตรวจสอบบริการใดได้บ้าง

#### ข้อจำกัดของระบบ

เนื่องจากระบบตรวจสอบสถานะการให้บริการของอุปกรณ์เครือข่าย สามารถตรวจสอบอุปกรณ์ได้เฉพาะอุปกรณ์ที่อยู่ในเครือข่ายเดียวกัน ซึ่งระบบที่ผู้จัดทำโครงการได้พัฒนาขึ้นนี้สามารถตรวจสอบได้เฉพาะอุปกรณ์ที่อยู่ในเครือข่ายของมหาวิทยาลัยราชภัฏสงขลาเท่านั้น

## ข้อเสนอแนะ

สำหรับผู้ที่มีความสนใจใน ระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย สามารถนำระบบนี้ไปพัฒนาต่อในส่วนของการเพิ่มอุปกรณ์ที่ต้องการตรวจสอบ และพัฒนาในด้านของการเขียนโปรแกรมภาษาต่าง ๆ ในการตรวจสอบสถานะอื่น ๆ นอกเหนือจากความสามารถเดิมของระบบ เพื่อให้ระบบมีความสามารถในการทำงานมากยิ่งขึ้น



## บรรณานุกรม

- ชี้ชวนี ดนล่าเต๊ะและสนธญา บิลหวัน. 2554. **ระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่ายและเครื่องให้บริการ โปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา.** สาขาวิชาวิทยาการคอมพิวเตอร์ โปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา
- เกียรติศักดิ์ แสงอรุณ. 2554. **ระบบมอนิเตอร์ริงกราฟฟิกผ่านเว็บเบราว์เซอร์.** สาขาวิศวกรรมเครือข่าย มหาวิทยาลัยเทคโนโลยีมหานคร
- อานนท์ มุ่งยอดกลาง. 2557. **ระบบการตรวจสอบการให้บริการเว็บและจดหมายอิเล็กทรอนิกส์กรณีศึกษา บริษัท ซอฟต์แวร์ จำกัด (มหาชน).** สาขาวิชาความมั่นคงทางระบบสารสนเทศ คณะวิทยาการและเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีมหานคร
- Installing Nagios Core From Source** (Online). <https://support.nagios.com/kb/article.php?id=96>, 16 พฤษภาคม 2560.
- Monitoring Network Printers** (Online). <https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/monitoring-printers.html>, 17 พฤษภาคม 2560
- How to install a Raspberry Pi SMS Server** (Online). <https://www.howtoforge.com/tutorial/how-to-install-raspberry-pi-sms-server/>, 25 พฤษภาคม 2560
- SNMP** (Online). <https://wiki.debian.org/SNMP>, 30 พฤษภาคม 2560
- configure Nagiosgraph with Nagios Core** (Online). <http://www.linuxfunda.com/2013/04/02/steps-to-configure-nagiosgraph-with-nagios-core/>, 13 มิถุนายน 2560
- Nagios SNMP checks** (Online). <http://www.thenocave.com/2012/10/29/nagios-snmp-checks/>, 12 กรกฎาคม 2560

## บรรณานุกรม (ต่อ)

**Plugins check SNMP for Nagios** (Online). [http://nagios.manubulon.com/index\\_commands.html](http://nagios.manubulon.com/index_commands.html), 20 กรกฎาคม 2560

**How To Install NRPE v3 From Source** (Online). <https://support.nagios.com/kb/article.php?id=515>, 25 กรกฎาคม 2560

**Set up NRPE with xinetd** (Online). <https://altinukshini.wordpress.com/2015/12/17/set-up-nrpe-v2-13-with-xinetd-in-ubuntu-server-14-04/>, 25 กรกฎาคม 2560

**Check Printer By SNMP** (Online). [https://exchange.nagios.org/directory/Plugins/Printing/check\\_printer\\_info-2Ept/details](https://exchange.nagios.org/directory/Plugins/Printing/check_printer_info-2Ept/details), 15 สิงหาคม 2560





## ภาคผนวก ก

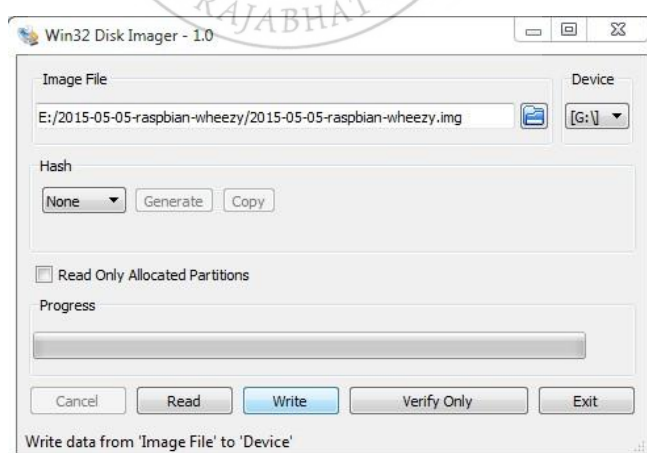
### สภาพแวดล้อมของระบบและการติดตั้ง

#### สภาพแวดล้อมของระบบ

ระบบตรวจสอบสถานะการให้บริการอุปกรณ์เครือข่าย โปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา ได้นำเครื่องคอมพิวเตอร์ขนาดเล็ก Raspberry Pi รุ่น 2 Model B และมีการใช้ระบบปฏิบัติการ Raspbian โดยมีแกนหลักเป็นระบบปฏิบัติการ Linux Debian ซึ่งได้ติดตั้งระบบปฏิบัติการนี้เป็นเครื่องบริการ Nagios และใช้โปรแกรม SSH Secure Shell client ในการ Remote เพื่อเข้าไป Configuration ในเครื่องบริการ Nagios โดยมีวิธีการติดตั้งและการใช้งานระบบดังต่อไปนี้

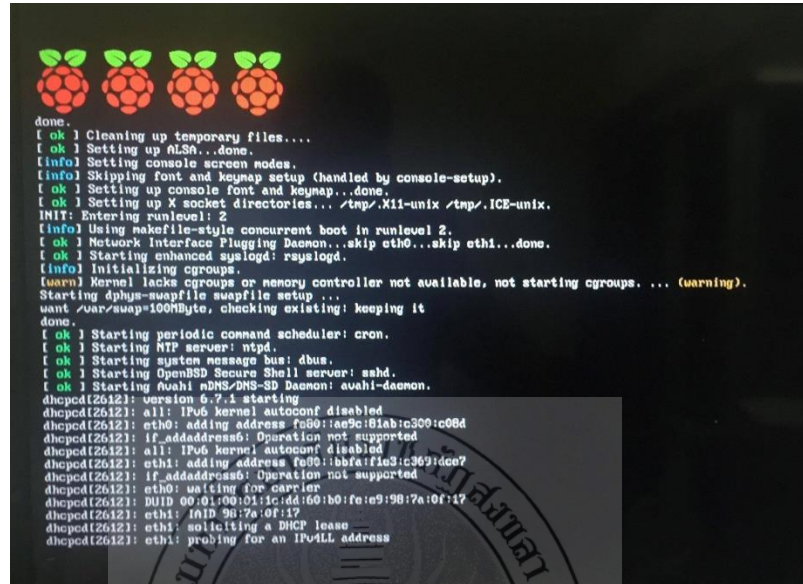
#### 1. การติดตั้งระบบปฏิบัติการ Raspbian (Linux Debian)

1. Download ระบบปฏิบัติการ Raspbian จากเว็บไซต์
2. ทำการเขียนไฟล์ Image ของระบบปฏิบัติการลงใน SD Card ที่ต้องการ โดยใช้โปรแกรม Win32DiskImage



ภาพที่ ก.1 เขียนไฟล์ Image ของระบบปฏิบัติการลงใน SD Card ด้วย Win32DiskImage

3. เมื่อทำขั้นตอนที่ 2 เสร็จแล้ว นำ SD Card ที่เขียนระบบปฏิบัติการแล้วมาใส่ในเครื่อง Raspberry Pi และทำการจ่ายไฟให้กับอุปกรณ์



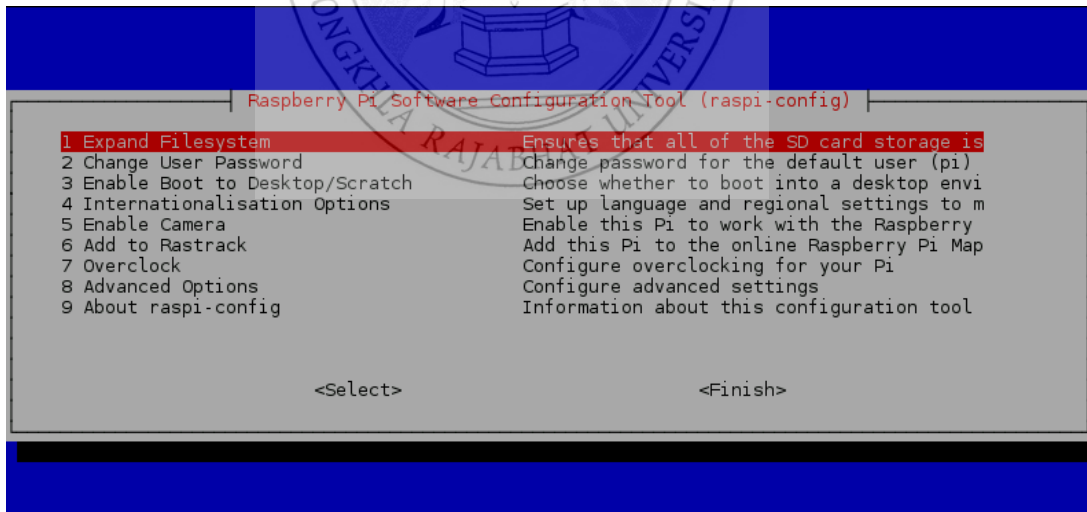
```

done.
[ ok ] Cleaning up temporary files...
[ ok ] Setting up ALSA...done.
[info] Setting console screen modes.
[info] Skipping font and keymap setup (handled by console-setup).
[ ok ] Setting up console font and keymap...done.
[ ok ] Setting up X socket directories... /tmp/.X11-unix /tmp/.ICE-unix.
INIT: Entering runlevel: 2
[info] Using makefile-style concurrent boot in runlevel 2.
[ ok ] Network Interface Plugging Daemon...skip eth0...skip eth1...done.
[ ok ] Starting enhanced syslogd: rsyslogd.
[info] Initializing cgroups.
[warn] Kernel lacks cgroups or memory controller not available, not starting cgroups. ... (warning).
Starting dhphys-swafpfile swafpfile setup ...
want /var/swap=100MByte, checking existing: keeping it
done.
[ ok ] Starting periodic command scheduler: cron.
[ ok ] Starting NTP server: ntpd.
[ ok ] Starting system message bus: dbus.
[ ok ] Starting OpenBSD Secure Shell server: sshd.
[ ok ] Starting Avahi: avahi-daemon: avahi-daemon.
dhcpcd[2612]: version 6.7.1 starting
dhcpcd[2612]: all: IPv6 kernel autoconf disabled
dhcpcd[2612]: eth0: adding address fe80::a09c:81ab:c309:c08d
dhcpcd[2612]: if_addaddress6: Operation not supported
dhcpcd[2612]: all: IPv6 kernel autoconf disabled
dhcpcd[2612]: eth1: adding address fe80::bbfa:ffe3:c369:dce7
dhcpcd[2612]: if_addaddress6: Operation not supported
dhcpcd[2612]: eth0: waiting for carrier
dhcpcd[2612]: DUID 00:01:000011:c:ad:60:b0:fe:e9:98:7a:0f:17
dhcpcd[2612]: eth1: IAD 98:7a:0f:17
dhcpcd[2612]: eth1: soliciting a DHCP lease
dhcpcd[2612]: eth1: probing for an IPv4LL address

```

ภาพที่ ก.2 หน้าจอแสดงการ Boot ระบบปฏิบัติการ

4. ในการ Boot ระบบปฏิบัติการครั้งแรก จะแสดงหน้าต่างการตั้งค่าต่าง ๆ ของอุปกรณ์ Raspberry Pi ซึ่งผู้ใช้สามารถเรียกการตั้งค่าในภายหลังได้โดยใช้คำสั่ง `sudo raspi-config`

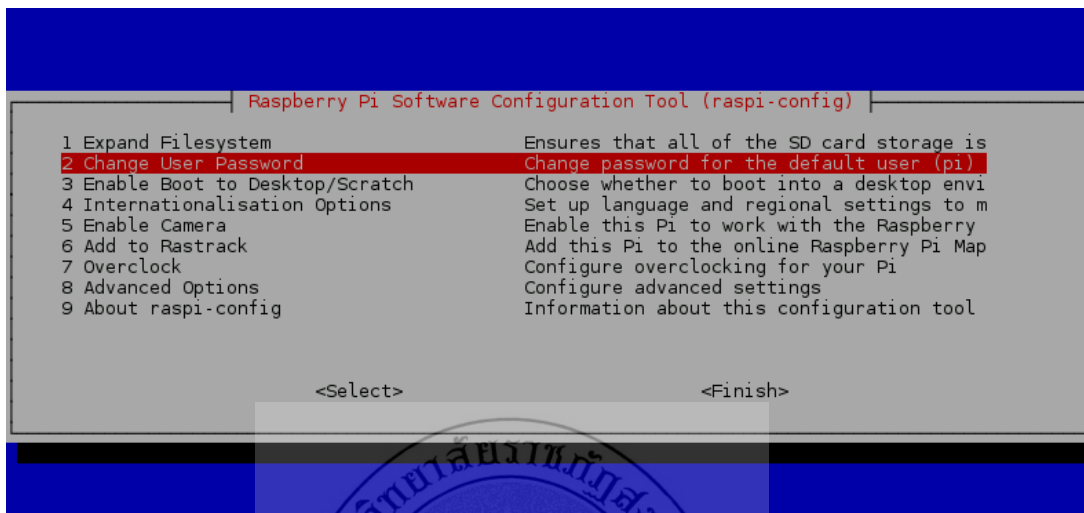


ภาพที่ ก.3 หน้าจอแสดงการตั้งค่าเริ่มต้นของอุปกรณ์ Raspberry Pi

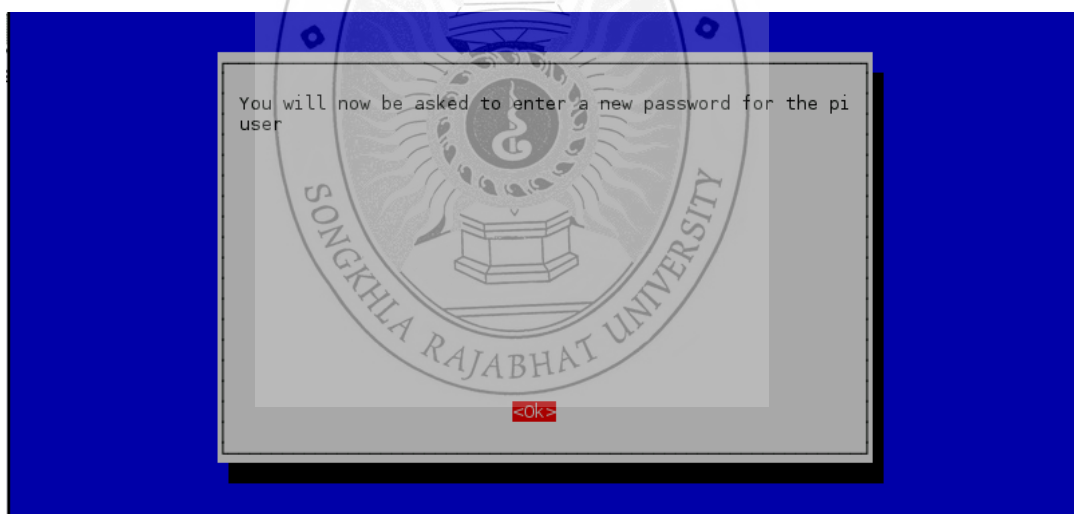




6. หัวข้อ Change User Password เป็นหัวข้อในการเปลี่ยน Password สำหรับ Login  
เข้าใช้งานเครื่อง Raspberry Pi



ภาพที่ ก.6 หัวข้อ Change User Password



ภาพที่ ก.7 ยืนยันความต้องการในการเปลี่ยน Password

```

4 heads, 16 sectors/track, 473728 cylinders, total 30318592 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xa6202af7

   Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1      8192        122879       57344    c   W95 FAT32 (LBA)
/dev/mmcblk0p2     122880     30318591    15097856   83   Linux

Command (m for help): The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
update-rc.d: using dependency based boot sequencing
update-rc.d: warning: default start runlevel arguments (2 3 4 5) do not match resize2fs_once Default-Start values (2 3 4 5 S)
update-rc.d: warning: default stop runlevel arguments (0 1 6) do not match resize2fs_once Default-Start values (none)
Enter new UNIX password:
Retype new UNIX password:

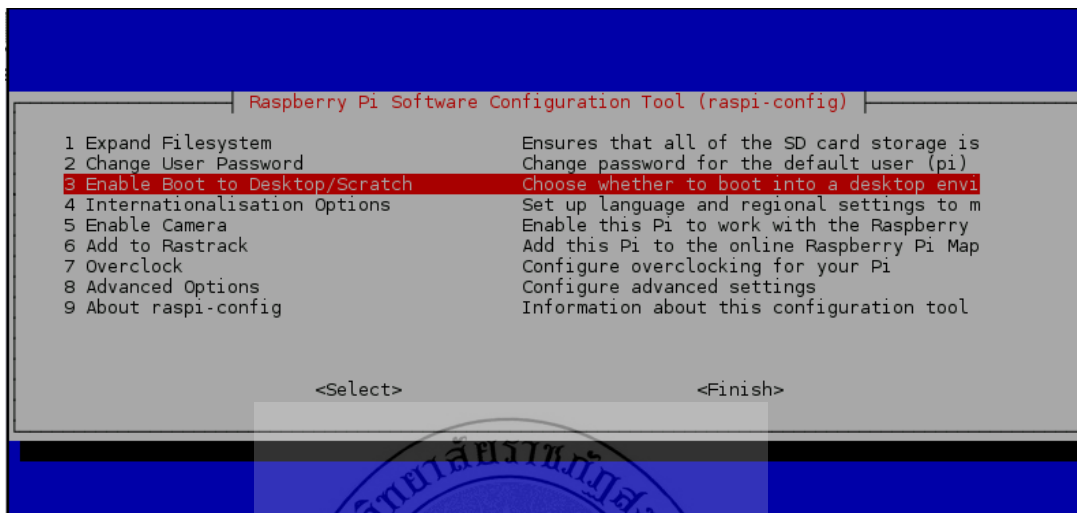
```

ภาพที่ ก.8 ยืนยัน Password ใหม่อีกครั้ง

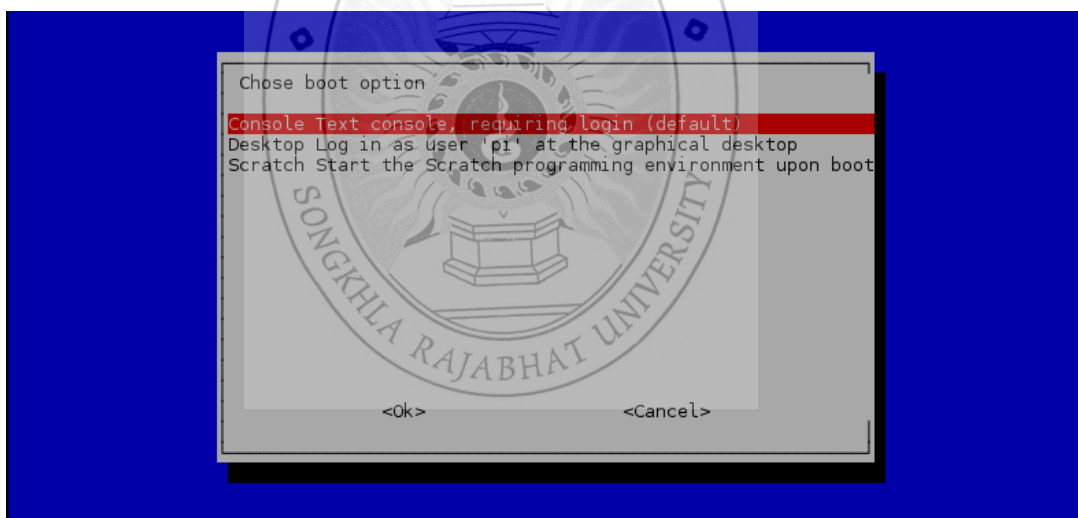


ภาพที่ ก.9 เมื่อตั้ง Password เรียบร้อยแล้ว

7. หัวข้อ Enable Boot to Desktop/Scratch ในส่วนนี้เป็นการกำหนดรูปแบบการ Boot ของ Raspberry Pi



ภาพที่ ก.10 หัวข้อ Enable Boot to Desktop/Scratch



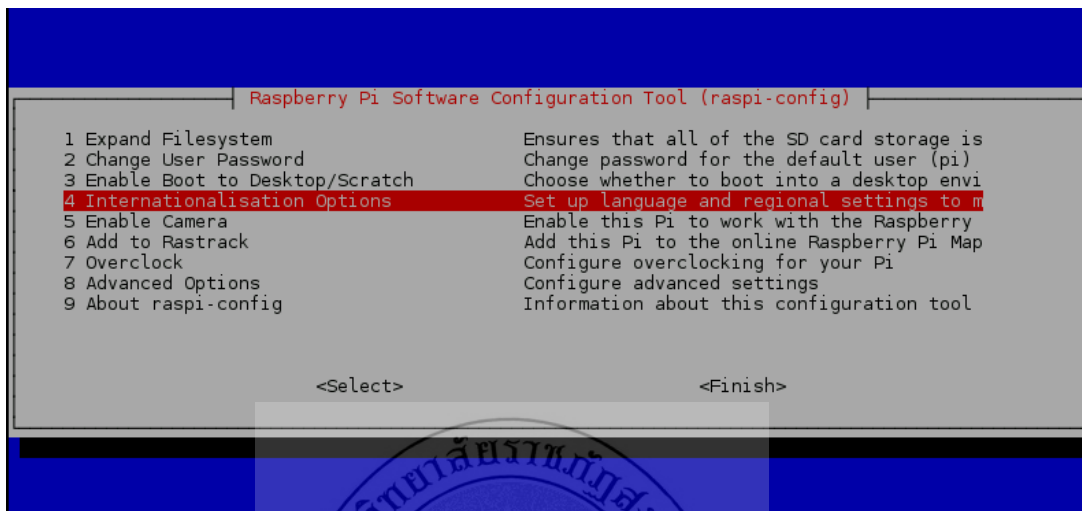
ภาพที่ ก.11 ตัวเลือกของรูปแบบการ Boot การทำงาน

จากภาพที่ ก.11 มีตัวเลือกของรูปแบบการ Boot สำหรับใช้งาน ดังนี้

- Console หมายถึง กำหนดให้ระบบเริ่มการทำงานโดยเป็นหน้าจอแบบ Text Mode
- Desktop หมายถึง กำหนดให้ระบบเริ่มการทำงานโดยเป็นหน้าจอแบบ Graphic
- Scratch หมายถึง ให้ระบบทำการเริ่มการทำงานของ Scratch ซึ่งเป็นโปรแกรมสำหรับ

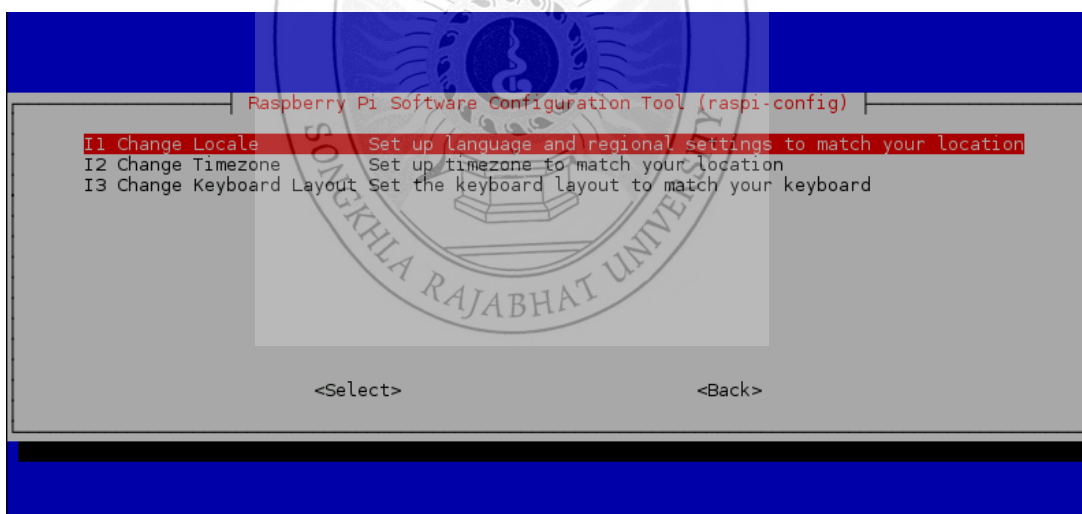
ฝึกเขียนโปรแกรมบน Raspberry Pi

8. หัวข้อ Internationalisation Options ในหัวข้อนี้เป็นการตั้งค่าเกี่ยวกับ เวลา ภาษา และรูปแบบ Keyboard

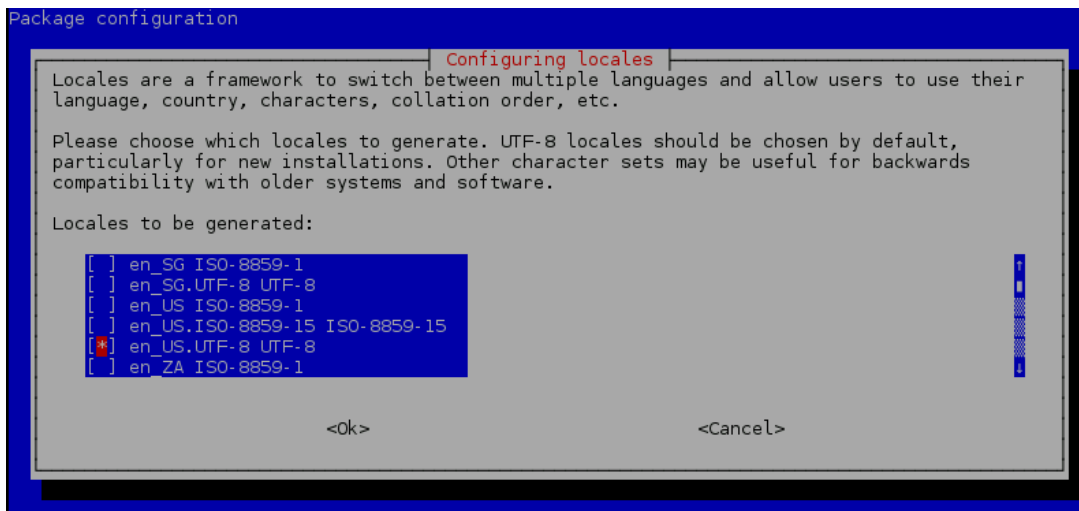


ภาพที่ ก.12 หัวข้อ Internationalisation Options

8.1 Change Locale เป็นการ set ภาษา เลือกเป็น en\_US.UTF-8

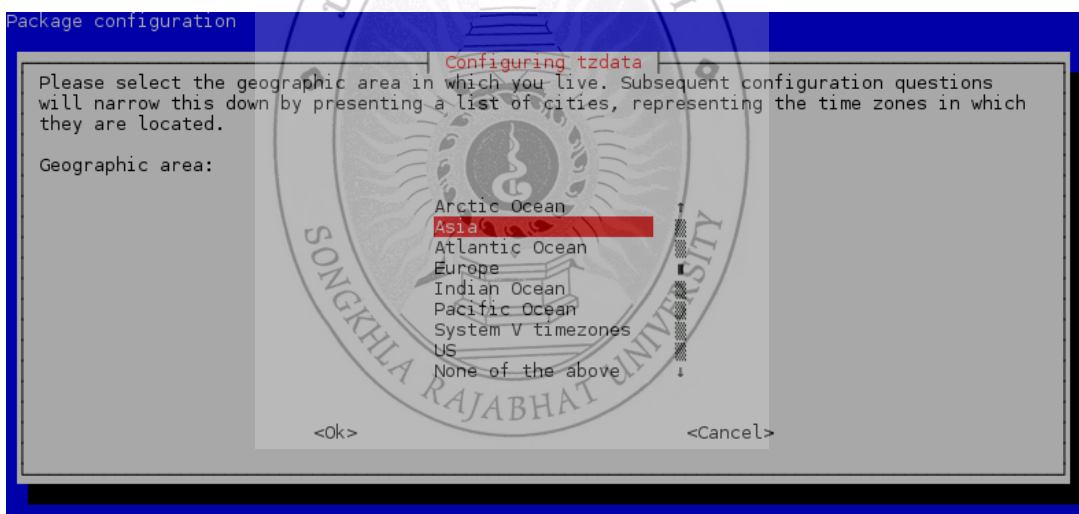


ภาพที่ ก.13 Change Locale

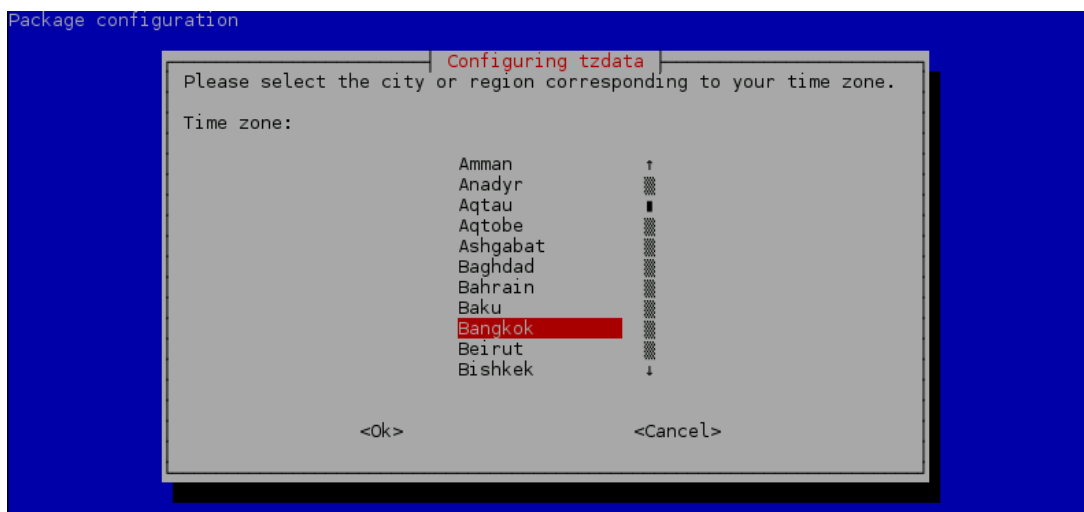


ภาพที่ ก.14 เลือกรูปแบบภาษาเป็น en\_US.UTF-8

## 8.2 Change Timezone เป็นการเลือกเขตเวลา Asia/Bangkok

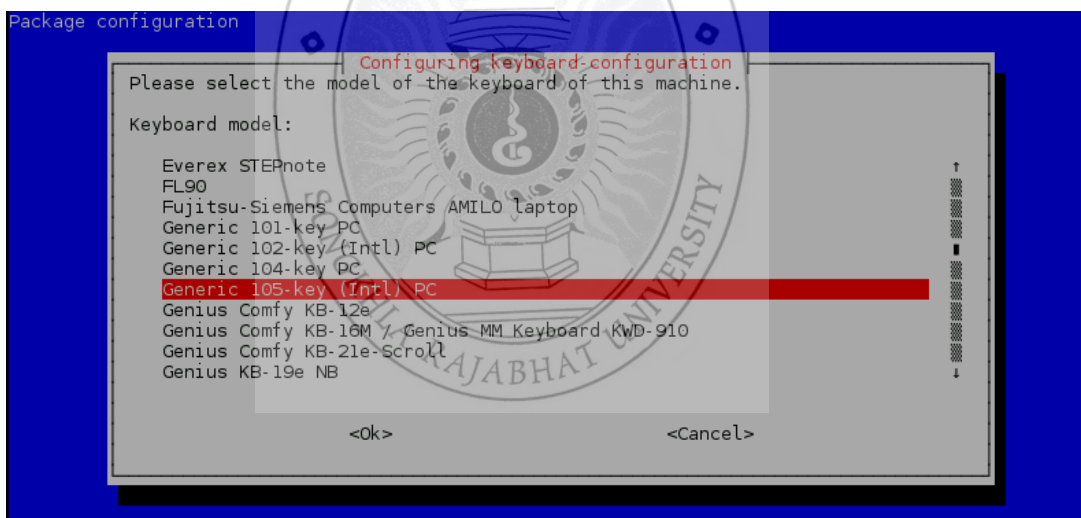


ภาพที่ ก.15 ในส่วน Geographic area เลือก Asia

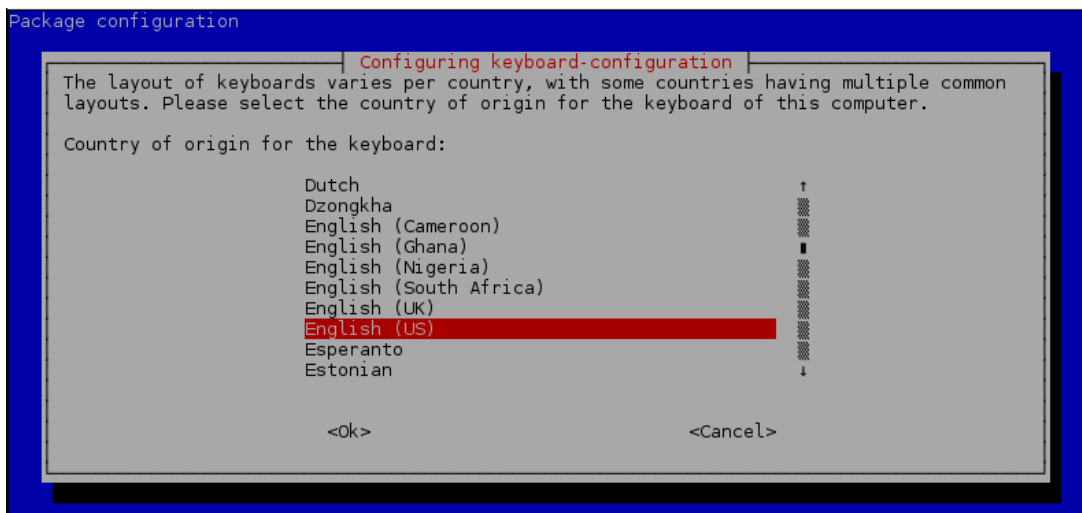


ภาพที่ ก.16 ในส่วน Time zone เลือก Bangkok

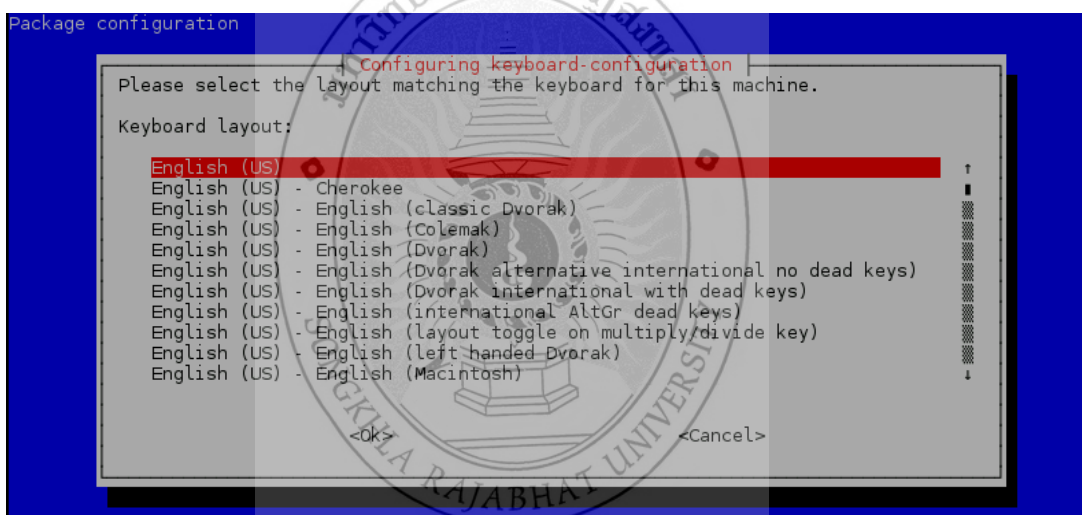
8.3 Change Keyboard Layout เป็นการเลือกรูปแบบการรับค่าการพิมพ์จากแป้นพิมพ์



ภาพที่ ก.17 ในส่วน Keyboard model เลือก Generic 105-key (Intl) PC



ภาพที่ ก.18 ในส่วน Country of origin for the keyboard เลือก English (US)



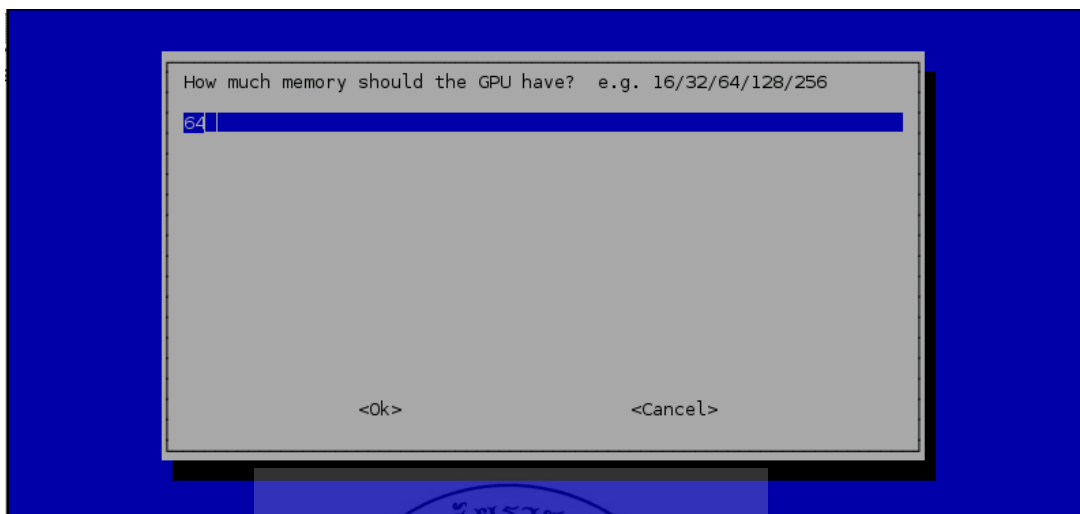
ภาพที่ ก.19 ในส่วน Keyboard layout เลือกเป็น English (US)





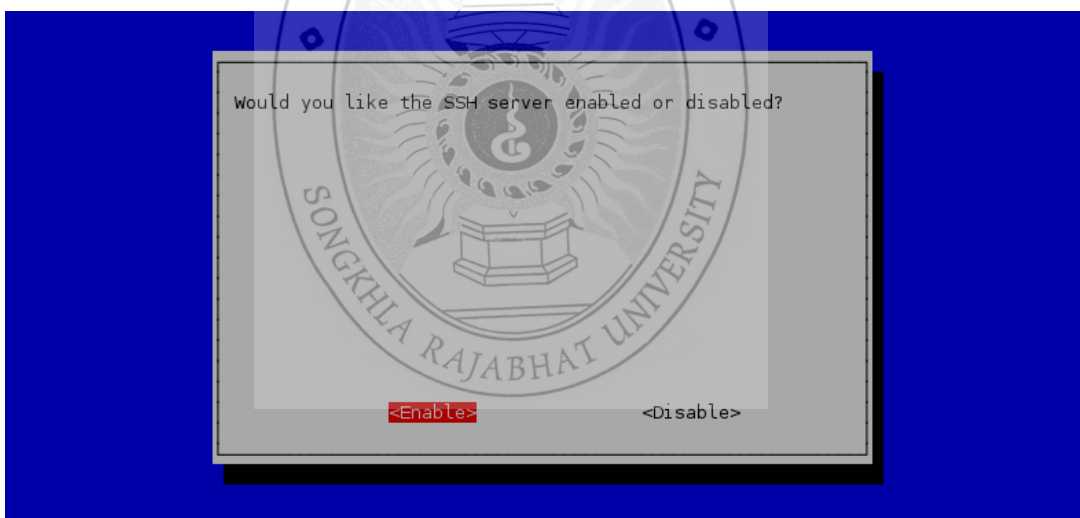


## 10.2 Memory Split เป็นตั้งค่าการแบ่งปันหน่วยความจำหลักไปให้กับ GPU



ภาพที่ ก.24 การแบ่งปันหน่วยความจำหลักไปให้กับ GPU

## 10.3 SSH เป็นการตั้งการเปิด / ปิด การทำงานของโปรโตคอล SSH



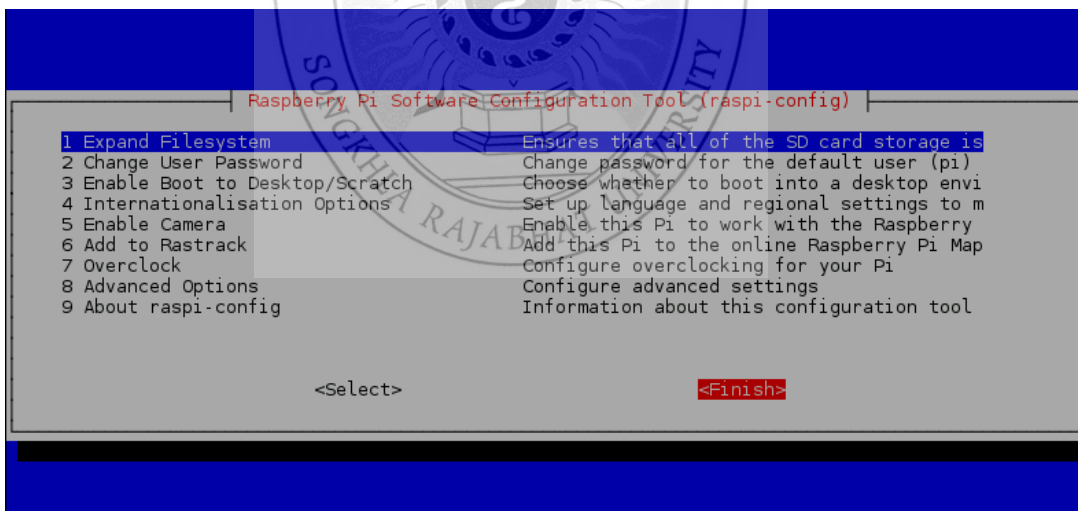
ภาพที่ ก.25 เปิดการทำงานของ SSH

10.4 Device Tree เป็นการตั้งการเปิด / ปิด การใช้งานการเก็บไบนารีของโครงสร้างของฮาร์ดแวร์

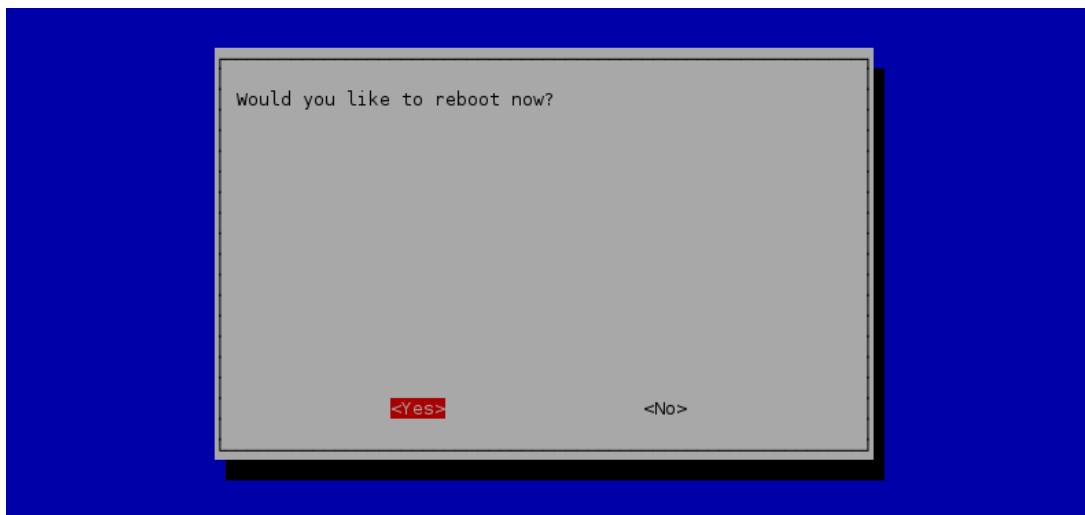


ภาพที่ ก.26 เปิดการทำงานของ Device Tree

11. เมื่อตั้งค่าตามข้างต้นเสร็จแล้ว เลือก Finish และระบบจะให้ถามผู้ใช้งานที่ต้องการ Reboot ระบบใหม่ หรือไม่ ให้ตอบ Yes



ภาพที่ ก.27 ตั้งค่าเสร็จแล้วให้เลือก Finish

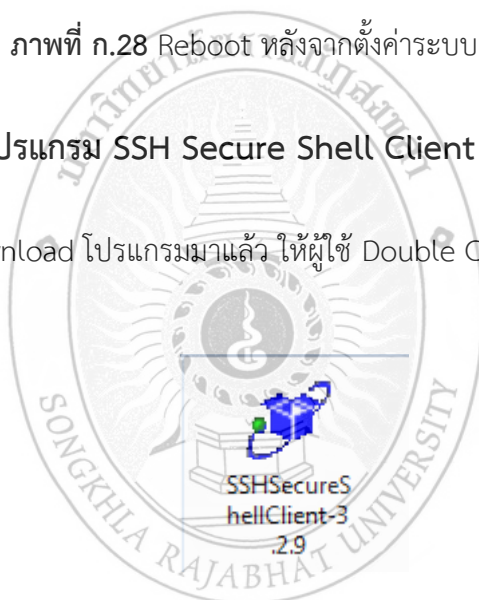


ภาพที่ ก.28 Reboot หลังจากตั้งค่าระบบเสร็จแล้ว

## 2. การติดตั้งโปรแกรม SSH Secure Shell Client

1. เมื่อ Download โปรแกรมมาแล้ว ให้ผู้ใช้ Double Click ที่ไฟล์ติดตั้งที่ได้ Download

มา



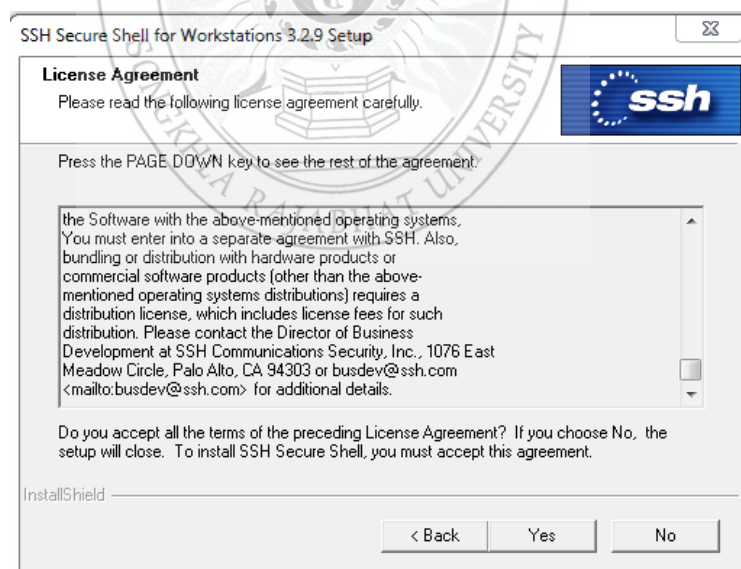
ภาพที่ ก.29 ไฟล์ติดตั้งของโปรแกรม SSH Secure Shell Client

2. ปรากฏหน้าต่างต้อนรับของโปรแกรม คลิก Next ดำเนินการติดตั้ง



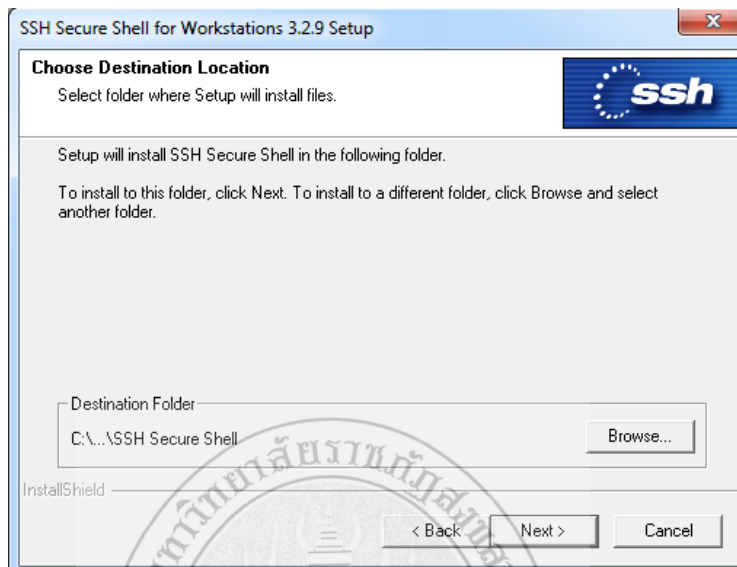
ภาพที่ ก.30 หน้าต่างต้อนรับของโปรแกรม SSH Secure Shell Client

3. ปรากฏหน้าต่างเงื่อนไขการใช้งานโปรแกรม (License) คลิก Yes เพื่อยอมรับ



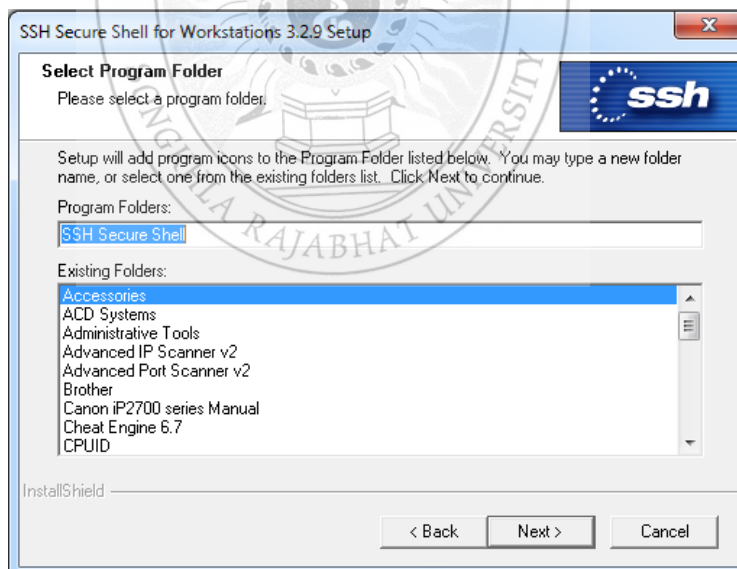
ภาพที่ ก.31 หน้าต่างเงื่อนไขการใช้งานโปรแกรม (License)

4. จะปรากฏหน้าต่างเลือก Folder สำหรับการติดตั้งโปรแกรม จากนั้นคลิก Next เพื่อไปยังขั้นตอนการติดตั้ง



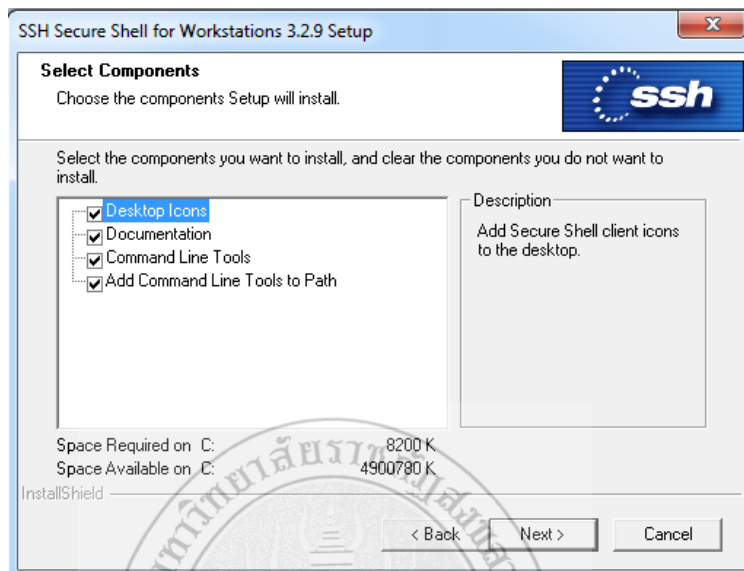
ภาพที่ ก.32 หน้าต่างเลือก Folder สำหรับการติดตั้งโปรแกรม

5. จะแสดงหน้าต่าง Select Program Folder ให้คลิก Next เพื่อไปยังขั้นตอนถัดไป



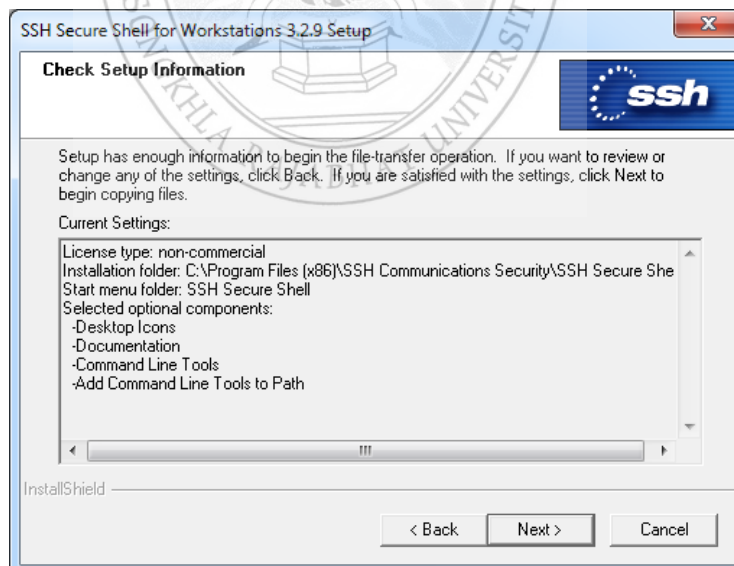
ภาพที่ ก.33 หน้าต่าง Select Program Folder

6. จะปรากฏหน้าต่าง Select Components เป็นการเลือกส่วนของโปรแกรมที่ต้องการติดตั้ง แล้วคลิก Next



ภาพที่ ก.34 หน้าต่าง Select Components เป็นการเลือกส่วนของโปรแกรมที่ต้องการติดตั้ง

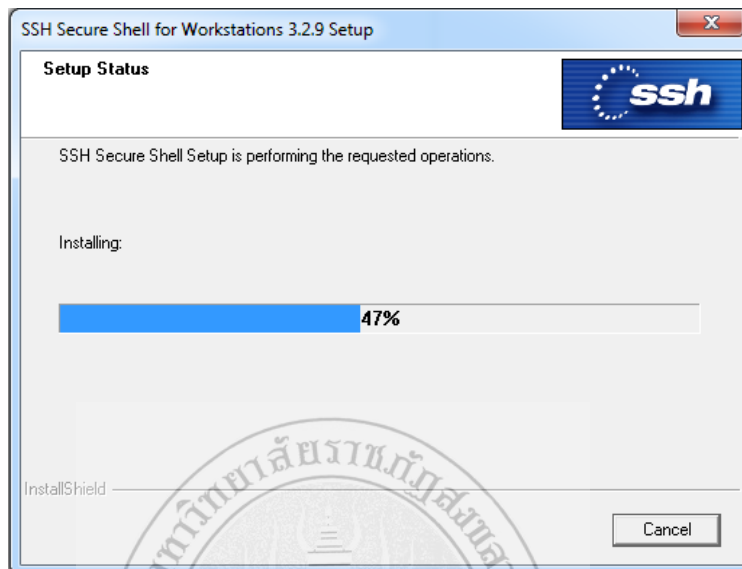
7. หลังจากนั้นจะแสดงหน้าต่าง Check Setup Information ให้คลิก Next เพื่อเริ่มการติดตั้ง



ภาพที่ ก.35 หน้าต่าง Check Setup Information

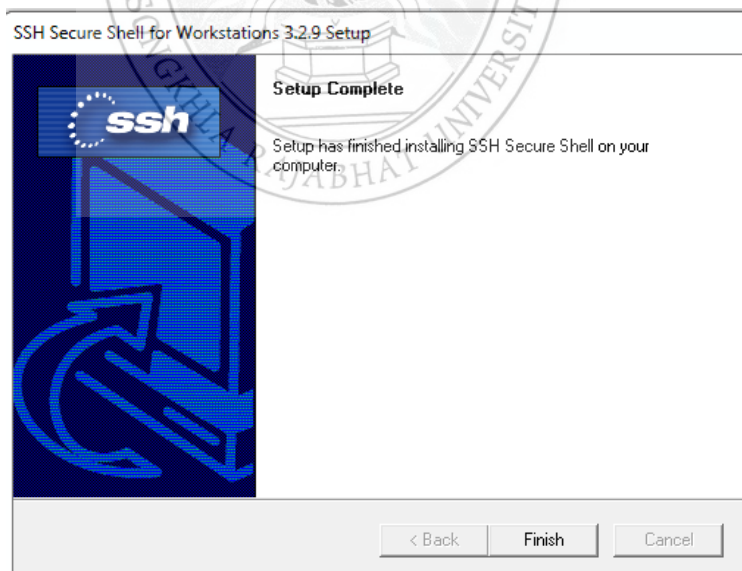


8. จะปรากฏหน้าต่าง Setup Status จะแสดงหน้าต่างสถานะของการติดตั้งโปรแกรม โดยต้องติดตั้งให้เสร็จจนครบ 100%



ภาพที่ ก.36 หน้าต่าง Setup Status

9. เมื่อติดตั้งจนครบ 100% แล้วจะแสดงหน้าต่าง Setup Complete หมายถึงการติดตั้งเสร็จสมบูรณ์ คลิก Finish เพื่อออกจากหน้าต่างการติดตั้ง



ภาพที่ ก.37 หน้าต่าง Setup Complete

## ประวัติผู้จัดทำโครงการ

ชื่อ - สกุล นายกฤษดา เบ็ญกาเสมอ

วันเดือนปี เกิด 28 เมษายน 2538

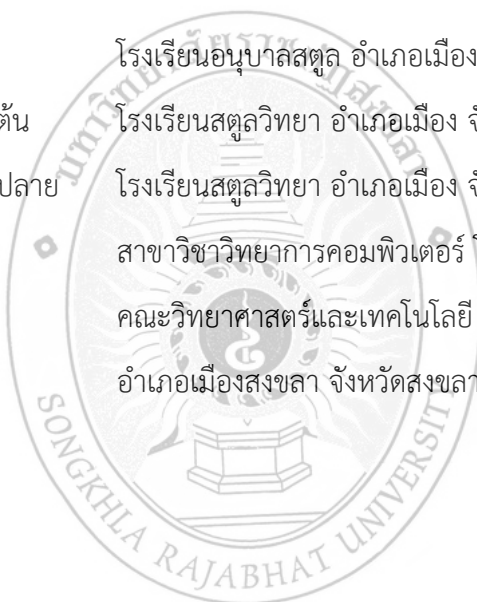
ที่อยู่ 434 หมู่ 1 ตำบลบ้านควน อำเภอเมือง จังหวัดสตูล 91140

โทรศัพท์ 08-1540-0283

อีเมล satun.finalzol@gmail.com

### ประวัติการศึกษา

ประถมศึกษา	โรงเรียนอนุบาลสตูล อำเภอเมือง จังหวัดสตูล
มัธยมศึกษาตอนต้น	โรงเรียนสตูลวิทยา อำเภอเมือง จังหวัดสตูล
มัธยมศึกษาตอนปลาย	โรงเรียนสตูลวิทยา อำเภอเมือง จังหวัดสตูล
ปริญญาตรี	สาขาวิชาวิทยาการคอมพิวเตอร์ โปรแกรมวิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสงขลา อำเภอเมืองสงขลา จังหวัดสงขลา



## ประวัติผู้จัดทำโครงการ

ชื่อ – สกุล นายพงษ์นรินทร์ พงศาปาน

วันเดือนปี เกิด 11 มีนาคม 2538

ที่อยู่ 188/164 หมู่ 6 ตำบลรูสะมิแล อำเภอเมือง จังหวัดปัตตานี 94000

โทรศัพท์ 08-0546-6696

อีเมลล์ stoneza001@hotmail.co.th

### ประวัติการศึกษา

ประถมศึกษา	โรงเรียนอนุบาลดอกบัว อำเภอเมือง จังหวัดปัตตานี
มัธยมศึกษาตอนต้น	โรงเรียนเดชะปัตตนิยานุกูล อำเภอเมือง จังหวัดปัตตานี
มัธยมศึกษาตอนปลาย	โรงเรียนเดชะปัตตนิยานุกูล อำเภอเมือง จังหวัดปัตตานี
ปริญญาตรี	สาขาวิชาวิทยาการคอมพิวเตอร์ โปรแกรมวิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสงขลา อำเภอเมืองสงขลา จังหวัดสงขลา

