



การตรวจสอบเชิงป้องกันโดยใช้แบชเชลล์สคริปต์
PREVENTIVE MONITORING BY BASH SHELL SCRIPT

นายคุณาธิป สุนทรพัชรคุณ

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

พ.ศ. 2561

การตรวจสอบเชิงป้องกันโดยใช้แบชเชลล์สคริปต์
PREVENTIVE MONITORING BY BASH SHELL SCRIPT

นายคุณาธิป สุนทรพัชรคุณ

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีไทย - ญี่ปุ่น
ปีการศึกษา 2561

คณะกรรมการสอบ

..... ประธานกรรมการสอบ

(อาจารย์ อติศักดิ์ เสือสมิง)

..... กรรมการสอบ

(ผู้ช่วยศาสตราจารย์ ดร.ประจักษ์ เจริญ)

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.นรังสรรค์ วิไลสกุลยง)

..... ประธานสหกิจศึกษาสาขาวิชา

(อาจารย์ สติลา ชีวภิกดาการ)

ลิขสิทธิ์ของสถาบันเทคโนโลยีไทย - ญี่ปุ่น

หัวข้อ	การตรวจสอบเชิงป้องกัน โดยใช้แบบจำลองสคริปต์
ผู้เขียน	นายคุณาธิป สุนทรพัชรคุณ
คณะวิชา	เทคโนโลยีสารสนเทศ สาขาวิชาเทคโนโลยีสารสนเทศ (IT)
อาจารย์ที่ปรึกษา	อาจารย์นรังสรรค์ วิไลสกุลยง
พนักงานที่ปรึกษา	นายภานุพงษ์ วงษ์ประดิษฐ์
ชื่อบริษัท	บริษัทเอ-โฮสต์จำกัด
ประเภทธุรกิจ	ให้บริการทางด้าน Oracle Product, IBM Product และ Hosting Service

บทสรุป

จากการศึกษาเซลล์สคริปต์ซึ่งเป็นภาษาโปรแกรมชนิดหนึ่งที่ใช้ในการเขียนเซลล์สคริปต์สำหรับระบบปฏิบัติการยูนิกซ์หรือลินุกซ์โดยในส่วนของงานที่ได้รับมอบหมายคือการเขียนเซลล์สคริปต์ให้สามารถตรวจสอบระบบและแจ้งเตือนอีเมลเมื่อเกิดเหตุการณ์ขึ้นบนเครื่องเซิร์ฟเวอร์ เช่น ซีพียูมีการทำงานที่หนักเกินไปหรือหน่วยความจำบนเครื่องเซิร์ฟเวอร์ใกล้จะเต็ม จะมีการส่งอีเมลเป็นการแจ้งเตือนผู้ดูแลระบบเพื่อป้องกันความเสี่ยงที่จะเกิดขึ้นในอนาคต

จากการที่ได้เขียนเซลล์สคริปต์ได้มีการศึกษาการใช้งานของระบบปฏิบัติการลินุกซ์และการใช้งานคำสั่งต่างๆในเซลล์สคริปต์โดยมีการเลือกใช้เซลล์สคริปต์สกุลแบบ หรือ แบบเซลล์สคริปต์ ซึ่งประโยชน์ที่ได้รับจากการทำสหกิจในครั้งนี้คือ ได้นำความรู้ที่ได้จากการทำงานสนับสนุนรุ่นพี่ในบริษัทและการศึกษาหาความรู้เซลล์สคริปต์จากช่องทางต่างๆ มาใช้ในการทำงานจริง เช่น การแก้ไขอีเมลภายในบริษัทเมื่ออีเมลไม่สามารถส่งเข้าส่งออกได้และติดตั้งซอฟต์แวร์เบื้องต้นให้กับคอมพิวเตอร์หรือ โน้ตบุ๊กของบริษัท อีกทั้งยังรู้จักความรับผิดชอบในหน้าที่ของตนเองนับว่าเป็นประสบการณ์ที่สำคัญในการพัฒนาตนเอง

Project's name	Preventive Monitoring by Bash Shell Script
Writer	Mr. Kunathip Suntornpatcharakun
Faculty	Faculty of Information Technology Program Information Technology
Faculty Advisor	Mr. Narungsun Wilaisakoolyong
Job Supervisor	Mr. Parnupong Wongpradit
Company's name	A-HOST Company Limited
Business Type	Oracle Product Service, IBM Product Service and Hosting Service

Summary

Based on the study of Shell Scripts. Shell Script is a programming language used to write Scripts for Unix or Linux. The assigned task is to write a shell script to monitor the system and alert the email when server has action or event such as when cpu is overworking or memory of server is near full the script will send an email to server administrator for prevent future risk.

I learn Linux operation and system command in Shell Scripts by Bash Shell Scripts. Cooperative Education Profits is work experience and knowledge of Shell Scripts. That I can apply it to life example edit email system and beginner software install for computer or notebook. Moreover I gain duty responsibility that important for self-improvement.

กิตติกรรมประกาศ

การปฏิบัติงานสหกิจศึกษา ณ บริษัท เอ-โฮสต์ จำกัด ตลอดระยะเวลาในการปฏิบัติงานมาทั้งหมด 4 เดือนทำให้ได้รับความรู้และประสบการณ์การทำงานต่างๆ ที่มีคุณค่าและรายงานผลการดำเนินการวิชาสหกิจศึกษานี้ จะไม่สำเร็จลุล่วงหากปราศจากความอนุเคราะห์และการสนับสนุน ของบุคคลเหล่านี้ ซึ่งผู้ปฏิบัติงานสหกิจศึกษาใคร่ขอกราบขอบพระคุณอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณอาจารย์รังสรรค์ วิไลสกุลยง ซึ่งเป็นอาจารย์ที่ปรึกษาคอยดูแลและให้คำปรึกษา ที่สละเวลาว่างของการปฏิบัติสหกิจศึกษา

ขอขอบพระคุณพีธีรัตน์ ชัยพรวิดี (พี่ฝน) ตำแหน่ง System Engineer Manager ที่สละเวลาให้คำแนะนำ ข้อคิด และความรู้ในการทำงาน

ขอขอบพระคุณพีรวัสส์ บัณฑิตยารักษ์ (พี่จ๊อบ) ตำแหน่ง Supervisory System Engineer ที่สละเวลา คอยให้คำปรึกษาและดูแลงานต่างๆ และความรู้ในการปฏิบัติงานที่สามารถนำไปปรับแก้ไขในการประกอบอาชีพในอนาคต

ขอขอบพระคุณพีโพรจน์ ทองดี (พี่นัท) และพีภาณุพงษ์ วงษ์ประดิษฐ์ (พี่พีท) ตำแหน่ง System Engineer ที่คอยให้คำปรึกษา คำแนะนำ งานแก้ไขปัญหาเฉพาะหน้า

ขอขอบพระคุณพีรณกฤต อธิไมยยะ (พี่คลาส) ตำแหน่ง System Engineer ที่คอยให้คำปรึกษา คำแนะนำ การเขียน Bash Shell Script และความรู้เกี่ยวกับ Linux ตลอดระยะเวลาที่ปฏิบัติงาน

นอกจากนี้ขอขอบคุณเพื่อนร่วมงานสหกิจศึกษาในสถานประกอบการเดียวกันนี้ ที่ไม่ได้เอ่ยนามในการทำงานร่วมกัน และช่วยสร้างบรรยากาศในการทำงานให้สนุกสนานเป็นกันเอง

นายคุณธิป สุนทรพัชรคุณ

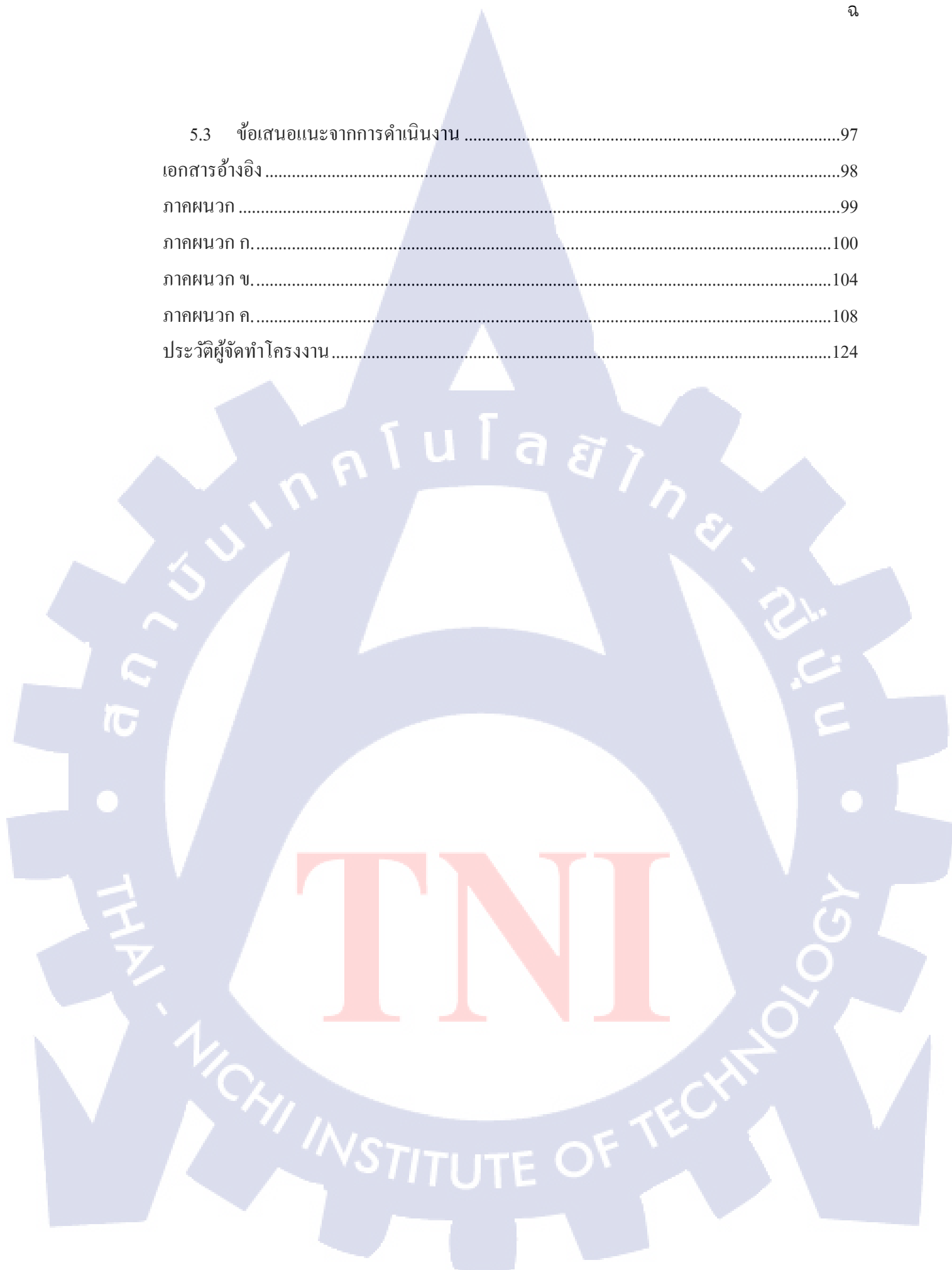
ผู้จัดทำ

สารบัญ

เรื่อง	หน้า
บทสรุป.....	ก
สารบัญ	ง
สารบัญภาพประกอบ	ช
สารบัญตาราง	๓
บทที่ 1	1
1.1 ชื่อและสถานที่ตั้งของสถานประกอบการ.....	1
1.2 ลักษณะธุรกิจของสถานประกอบการหรือการให้บริการขององค์กร.....	2
1.2.1 Hosting & Outsource Service	3
1.2.2 Oracle Core Technology Products and Advanced Services	4
1.2.3 Oracle Enterprise Performance Management (EPM) และ Hyperion Business Intelligence Products and Services	5
1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร	6
1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย	7
1.5 พนักงานที่ปรึกษาและตำแหน่งของพนักงานที่ปรึกษา.....	8
1.6 ระยะเวลาที่ปฏิบัติงาน.....	8
1.7 วัตถุประสงค์ของการปฏิบัติงานและ โครงการที่ได้รับมอบหมาย.....	8
1.8 ผลที่คาดว่าจะได้รับจากการปฏิบัติงานและ โครงการที่ได้รับมอบหมาย	8
บทที่ 2	9
2.1 ทฤษฎีที่เกี่ยวข้องในการทำงาน.....	9
2.1.1 Unix	9
2.1.2 Linux	11
2.1.3 Kernel.....	12
2.1.4 คำสั่ง Unix – Linux Command	14
2.1.5 Shell	19
2.1.6 Text Shell (CLI).....	20

2.1.7	Graphical Shell	20
2.1.8	Shell Script.....	21
2.1.9	การเขียน Shell Script	23
2.2	เทคโนโลยีที่ใช้ในการปฏิบัติงาน.....	35
2.2.1	โปรแกรม Oracle VM VirtualBox	35
2.2.2	โปรแกรม FileZilla	36
บทที่ 3		37
3.1	แผนงานปฏิบัติงาน	37
3.2	รายละเอียดงานที่นักศึกษาปฏิบัติในงานสหกิจศึกษา หรือรายละเอียดโครงการที่ได้รับมอบหมาย.....	38
3.3	ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงานหรือโครงการ.....	38
3.3.1	ศึกษาการใช้งาน โปรแกรมที่จำเป็นต่อการพัฒนา Shell Script.....	38
3.3.2	ทำความเข้าใจความต้องการของผู้ใช้และศึกษาข้อมูลที่มีความจำเป็น.....	38
3.3.3	ทดลองเขียน Shell Script เบื้องต้นและ ทดสอบใช้งาน Shell Script.....	40
3.3.4	เขียน Shell Script สำหรับ Monitor	40
3.3.5	เขียน Shell Script สำหรับแจ้งเตือน Email.....	44
3.3.6	แก้ไข Shell Script เมื่อพบข้อผิดพลาด	44
3.3.7	ทำคู่มือการติดตั้งและใช้งาน Script	44
3.3.8	ทำรูปเล่มโครงการและเตรียมการนำเสนอโครงการ	45
3.3.9	ช่วยสนับสนุนงานด้านต่างๆ และแก้ปัญหาภายในบริษัท	45
บทที่ 4		46
4.1	ขั้นตอนและผลการดำเนินงาน	46
4.1.1	ศึกษาและทำความเข้าใจความต้องการของผู้ใช้งาน	46
4.1.2	เขียน Shell Script และติดตั้งใช้งาน Shell Script ที่พัฒนา.....	46
4.2	ผลการวิเคราะห์ข้อมูล	95
4.3	วิจารณ์ข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์และจุดมุ่งหมายการปฏิบัติงานหรือการจัดทำโครงการ	95
บทที่ 5		96
5.1	สรุปผลการดำเนินงาน	96
5.2	แนวทางการแก้ไขปัญหา.....	97

5.3 ข้อเสนอแนะจากการดำเนินงาน	97
เอกสารอ้างอิง	98
ภาคผนวก	99
ภาคผนวก ก.....	100
ภาคผนวก ข.....	104
ภาคผนวก ค.....	108
ประวัติผู้จัดทำโครงการ.....	124



สารบัญภาพประกอบ

ภาพที่ 1. 1 สถานที่ประกอบการ A-HOST Company Limited.....	1
ภาพที่ 1. 2 รางวัลที่ A-HOST Company Limited ได้รับ.....	3
ภาพที่ 1. 3 คณะผู้บริหาร A-HOST Company Limited	6
ภาพที่ 2. 1 Logo Oracle VM VirtualBox	35
ภาพที่ 2. 2 Logo FileZilla	36
ภาพที่ 3. 1 ตัวอย่าง Website ที่ทำการหาข้อมูลเกี่ยวกับ Shell	39
ภาพที่ 3. 2 ตัวอย่าง Website ที่ทำการหาข้อมูลเกี่ยวกับ Shell Script.....	39
ภาพที่ 3. 3 ตัวอย่าง Website ที่ทำการหาข้อมูลเกี่ยวกับการ Shell Script เบื้องต้น	40
ภาพที่ 3. 4 ตัวอย่าง Shell Script ที่ใช้ในการแสดงรายละเอียดข้อมูล	40
ภาพที่ 3. 5 ตัวอย่างกระบวนการทำงานของ Script	41
ภาพที่ 3. 6 ตัวอย่างกระบวนการทำงานของ Script ในส่วน Server Information	42
ภาพที่ 3. 7 ตัวอย่างกระบวนการทำงานของ Script ในส่วน Backup Log File	42
ภาพที่ 3. 8 ตัวอย่างกระบวนการทำงานของ Script ในส่วน System Information	43
ภาพที่ 3. 9 ตัวอย่างกระบวนการทำงานของ Script ในส่วน Generate Monthly Report.....	43
ภาพที่ 4. 1 ตัวอย่างโปรแกรม FileZilla	47
ภาพที่ 4. 2 ตัวอย่างการโอนย้ายข้อมูล (ต่อภาพที่ 4.1).....	47
ภาพที่ 4. 3 ตัวอย่างหน้ารายละเอียด Shell Script สำหรับการ Monitor	48
ภาพที่ 4. 4 ตัวอย่างหน้าเมนูหลักของ Shell Script สำหรับการ Monitor (Main Menu).....	48
ภาพที่ 4. 5 ตัวอย่างเมนูสำหรับการ Monitor Hardware (Server Information)	49
ภาพที่ 4. 6 ตัวอย่างการ Monitor Hardware ในหัวข้อ CPU Info	50
ภาพที่ 4. 7 ตัวอย่างการ Monitor Hardware ในหัวข้อ Memory Info	50
ภาพที่ 4. 8 ตัวอย่างการ Monitor Hardware ในหัวข้อ OS Info	50
ภาพที่ 4. 9 ตัวอย่างการ Monitor Hardware ในหัวข้อ IP Address	51
ภาพที่ 4. 10 ตัวอย่างการ Monitor Hardware ในหัวข้อ Manipulate Disk Partition Table	51
ภาพที่ 4. 11 ตัวอย่างเมนูสำหรับ Monitor System (System Information)	52

ภาพที่ 4. 12 ตัวอย่างการ Monitor System ในหัวข้อ Free and Use Memory	52
ภาพที่ 4. 13 ตัวอย่างการ Monitor System ในหัวข้อ File System Disk Space Usage	52
ภาพที่ 4. 14 ตัวอย่างการ Monitor System ในหัวข้อ Percentage CPU/RAM	52
ภาพที่ 4. 15 ตัวอย่างการ Monitor System ในหัวข้อ System Uptime	52
ภาพที่ 4. 16 ตัวอย่างเมนูสำหรับการเข้าสู่ไฟล์ Backup (Backup Log File Menu)	53
ภาพที่ 4. 17 ตัวอย่างเมนูสำหรับเข้าไปสร้างไฟล์ข้อมูลเก็บค่าสถานะระบบแบบสรุป	54
ภาพที่ 4. 18 ตัวอย่างข้อความบอกสถานที่ที่เคยกำหนด	55
ภาพที่ 4. 19 ตัวอย่างข้อความบอกสถานที่ที่ไม่เคยมีการกำหนดสถานที่	55
ภาพที่ 4. 20 ตัวอย่างการออกจากการใช้ฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล	55
ภาพที่ 4. 21 ตัวอย่างผู้ใช้ไม่ได้พิมพ์ตัวเลขหรือเว้นช่องว่าง	56
ภาพที่ 4. 22 ตัวอย่างผู้ใช้พิมพ์ตัวอักษรหรือนอกเหนือจากตัวเลขที่กำหนด	56
ภาพที่ 4. 23 ตัวอย่างคำถามบันทึกไฟล์ข้อมูลหลังจาก Monitor Hardware	57
ภาพที่ 4. 24 ตัวอย่างหากมีการกำหนดสถานที่เก็บไฟล์ข้อมูลแล้วบันทึกไฟล์ข้อมูล	57
ภาพที่ 4. 25 ตัวอย่างหากไม่มีการกำหนดสถานที่เก็บไฟล์ข้อมูลแล้วบันทึกไฟล์ข้อมูล	57
ภาพที่ 4. 26 ตัวอย่างหากผู้ใช้พิมพ์ “n” หลังจากนั้น Enter	58
ภาพที่ 4. 27 ตัวอย่างหากผู้ใช้พิมพ์นอกเหนือ “n” หรือ “y” หลังจากนั้น Enter	58
ภาพที่ 4. 28 ตัวอย่างหากผู้ใช้เลือกพิมพ์เลข 6 จะเป็นการออกจากหน้าเมนูการ Monitor Hardware	58
ภาพที่ 4. 29 ตัวอย่างผู้ใช้ไม่ได้พิมพ์ตัวเลขหรือเว้นช่องว่าง	59
ภาพที่ 4. 30 ตัวอย่างผู้ใช้พิมพ์ตัวอักษรหรือนอกเหนือจากตัวเลขที่กำหนด	59
ภาพที่ 4. 31 ตัวอย่างคำถามบันทึกไฟล์ข้อมูลหลังจาก Monitor System	59
ภาพที่ 4. 32 ตัวอย่างหากมีการกำหนดสถานที่เก็บไฟล์ข้อมูลแล้วบันทึกไฟล์ข้อมูล	60
ภาพที่ 4. 33 ตัวอย่างหากไม่มีการกำหนดสถานที่เก็บไฟล์ข้อมูลแล้วบันทึกไฟล์ข้อมูล	60
ภาพที่ 4. 34 ตัวอย่างหากผู้ใช้พิมพ์ “n” หลังจากนั้น Enter	60
ภาพที่ 4. 35 ตัวอย่างหากผู้ใช้พิมพ์นอกเหนือ “n” หรือ “y” หลังจากนั้น Enter	60
ภาพที่ 4. 36 ตัวอย่างหากผู้ใช้เลือกพิมพ์เลข 5 จะเป็นการออกจากหน้าเมนูการ Monitor System	61
ภาพที่ 4. 37 ตัวอย่างผู้ใช้ไม่ได้พิมพ์ตัวเลขหรือเว้นช่องว่าง	61
ภาพที่ 4. 38 ตัวอย่างผู้ใช้ได้พิมพ์ตัวอักษรหรือนอกเหนือจากตัวเลขที่กำหนด	62
ภาพที่ 4. 39 ตัวอย่างข้อความบอกสถานที่ที่เคยกำหนด	62
ภาพที่ 4. 40 ตัวอย่างข้อความบอกสถานที่ที่ไม่เคยมีการกำหนดสถานที่	62

ภาพที่ 4. 41 ตัวอย่างการออกจากการใช้ฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล.....	63
ภาพที่ 4. 42 ตัวอย่างหากไม่มีการกำหนดสถานที่เก็บไฟล์ข้อมูลที่ Backup บน Server.....	63
ภาพที่ 4. 43 ตัวอย่างการออกจากการใช้งานหัวข้อที่ 4 โดยผู้ใช้ Enter.....	63
ภาพที่ 4. 44 ตัวอย่างการออกจากการใช้งานหัวข้อที่ 4 โดยผู้ใช้พิมพ์ “q” จากนั้น Enter.....	63
ภาพที่ 4. 45 ตัวอย่างการใช้งานฟังก์ชันในหัวข้อที่ 4 ของฟังก์ชันกำหนดสถานที่ที่ Server เก็บ ไฟล์ข้อมูลที่ Backup.....	64
ภาพที่ 4. 46 ตัวอย่างผลลัพธ์การใช้งานหัวข้อที่ 2-4 ของฟังก์ชันกำหนดสถานที่ที่ Server เก็บ ไฟล์ข้อมูลที่ Backup.....	64
ภาพที่ 4. 47 ตัวอย่างหากไม่มีการกำหนดสถานที่เก็บไฟล์ข้อมูลที่ Backup บน Server.....	64
ภาพที่ 4. 48 ตัวอย่างหากไม่มีการกำหนดสถานที่เก็บไฟล์ข้อมูลที่ Backup บน Server.....	65
ภาพที่ 4. 49 ตัวอย่างผลลัพธ์การใช้งานหัวข้อที่ 5 โดยมีข้อผิดพลาดของฟังก์ชันกำหนดสถานที่ที่ Server เก็บไฟล์ข้อมูลที่ Backup	65
ภาพที่ 4. 50 ตัวอย่างผลลัพธ์การใช้งานหัวข้อที่ 5 โดยไม่มีข้อผิดพลาดของฟังก์ชันกำหนดสถานที่ที่ Server เก็บไฟล์ข้อมูลที่ Backup	65
ภาพที่ 4. 51 ตัวอย่างผู้ใช้ไม่ได้พิมพ์ตัวเลขหรือเว้นช่องว่าง	66
ภาพที่ 4. 52 ตัวอย่างผู้ใช้ได้พิมพ์ตัวอักษรหรือนอกเหนือจากตัวเลขที่กำหนด	66
ภาพที่ 4. 53 ตัวอย่างข้อความบอกสถานที่ที่เคยกำหนด	67
ภาพที่ 4. 54 ตัวอย่างข้อความบอกสถานที่ที่ไม่เคยมีการกำหนดสถานที่.....	67
ภาพที่ 4. 55 ตัวอย่างการออกจากการใช้ฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล.....	67
ภาพที่ 4. 56 ตัวอย่างหากไม่มีการกำหนดสถานที่เก็บไฟล์ข้อมูลที่ Backup บน Server.....	68
ภาพที่ 4. 57 ตัวอย่างการออกจากการใช้งานหัวข้อที่ 4 โดยผู้ใช้ Enter	68
ภาพที่ 4. 58 ตัวอย่างการออกจากการใช้งานหัวข้อที่ 4 โดยผู้ใช้พิมพ์ “q” จากนั้น Enter.....	68
ภาพที่ 4. 59 ตัวอย่างการแปลงไฟล์ข้อมูลดิบเป็นไฟล์สกุล txt	69
ภาพที่ 4. 60 ตัวอย่างการออกจากการใช้งานหัวข้อที่ 5 โดยผู้ใช้ Enter	69
ภาพที่ 4. 61 ตัวอย่างการออกจากการใช้งานหัวข้อที่ 5 โดยผู้ใช้พิมพ์ “q” จากนั้น Enter.....	69
ภาพที่ 4. 62 ตัวอย่างการใช้งานหัวข้อที่ 5.....	69
ภาพที่ 4. 63 ตัวอย่างฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล (Set File Location for Export)	70
ภาพที่ 4. 64 ตัวอย่างฟังก์ชันในการ Monitor Hardware (Server Information) หัวข้อ CPU Info.....	71
ภาพที่ 4. 65 ตัวอย่างฟังก์ชันในการ Monitor Hardware (Server Information) หัวข้อ Memory Info	72

ภาพที่ 4. 66 ตัวอย่างฟังก์ชันในการ Monitor Hardware (Server Information) หัวข้อ Memory Info ส่วนที่ 1	73
ภาพที่ 4. 67 ตัวอย่างฟังก์ชันในการ Monitor Hardware (Server Information) หัวข้อ Memory Info ส่วนที่ 2	74
ภาพที่ 4. 68 ตัวอย่างฟังก์ชันในการ Monitor Hardware (Server Information) หัวข้อ IP Address ..	75
ภาพที่ 4. 69 ตัวอย่างฟังก์ชันในการ Monitor Hardware (Server Information) หัวข้อ Manipulate Disk Partition Table	76
ภาพที่ 4. 70 ตัวอย่างฟังก์ชันในการ Monitor System (System Information) หัวข้อ Free and Use Memory	77
ภาพที่ 4. 71 ตัวอย่างฟังก์ชันในการ Monitor System (System Information)	78
ภาพที่ 4. 72 ตัวอย่างฟังก์ชันในการ Monitor System (System Information)	79
ภาพที่ 4. 73 ตัวอย่างฟังก์ชันในการ Monitor System (System Information) หัวข้อ System Uptime	80
ภาพที่ 4. 74 ตัวอย่างฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup (Backup Log File) หัวข้อ Set Backup Log File Location.....	81
ภาพที่ 4. 75 ตัวอย่างฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup (Backup Log File) หัวข้อ View Backup Log File (Today).....	82
ภาพที่ 4. 76 ตัวอย่างฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup (Backup Log File) หัวข้อ View Backup Log File (Yesterday).....	82
ภาพที่ 4. 77 ตัวอย่างฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup (Backup Log File) หัวข้อ View Backup Log File (History)	82
ภาพที่ 4. 78 ตัวอย่างฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup (Backup Log File) หัวข้อ View Backup Log File (Difference Time).....	83
ภาพที่ 4. 79 ตัวอย่างฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป (Generate Monthly Report) หัวข้อ Set Data File Location	84
ภาพที่ 4. 80 ตัวอย่างฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป (Generate Monthly Report) หัวข้อ Set File sar.txt Location.....	85
ภาพที่ 4. 81 ตัวอย่างฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป (Generate Monthly Report) หัวข้อ Get Data from “sar”	86

ภาพที่ 4. 82 ตัวอย่างฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป (Generate Monthly Report) หัวข้อ Change Data to Text File.....	86
ภาพที่ 4. 83 ตัวอย่างฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป	87
ภาพที่ 4. 84 ตัวอย่างโปรแกรม FileZilla	88
ภาพที่ 4. 85 ตัวอย่างการโอนย้ายข้อมูล	88
ภาพที่ 4. 86 ตัวอย่างที่เพิ่มลงไปไฟล์ข้อมูล main.cf	89
ภาพที่ 4. 87 ตัวอย่างที่เพิ่มลงไปไฟล์ข้อมูล sasl_passwd	90
ภาพที่ 4. 88 ตัวอย่างที่เพิ่มลงไปไฟล์ข้อมูล sasl_passwd	91
ภาพที่ 4. 89 ตัวอย่างการกำหนดค่า crontab -e.....	91
ภาพที่ 4. 90 ตัวอย่างฟังก์ชันตรวจสอบ CPU และส่ง Email ด้วย HTML	92
ภาพที่ 4. 91 ตัวอย่างฟังก์ชันตรวจสอบสวอปและส่ง Email ด้วย HTML	92
ภาพที่ 4. 92 ตัวอย่างฟังก์ชันตรวจสอบ Memory และส่ง Email ด้วย HTML	93
ภาพที่ 4. 93 ตัวอย่างฟังก์ชันตรวจสอบ Hard Disk และส่ง Email พร้อมแนบไฟล์.....	94
ภาพที่ ก. 1 การติดตั้ง Oracle VM VirtualBox	101
ภาพที่ ก. 2 การติดตั้ง Oracle VM VirtualBox (ต่อภาพที่ ก.1)	101
ภาพที่ ก. 3 การติดตั้ง Oracle VM VirtualBox (ต่อภาพที่ ก.2)	102
ภาพที่ ก. 4 การติดตั้ง Oracle VM VirtualBox (ต่อภาพที่ ก.3)	102
ภาพที่ ก. 5 การติดตั้ง Oracle VM VirtualBox (ต่อภาพที่ ก.4)	103
ภาพที่ ก. 6 การติดตั้ง Oracle VM VirtualBox (ต่อภาพที่ ก.5)	103
ภาพที่ ข. 1 การติดตั้ง FileZilla	105
ภาพที่ ข. 2 การติดตั้ง FileZilla (ต่อภาพที่ ข.1).....	105
ภาพที่ ข. 3 การติดตั้ง FileZilla (ต่อภาพที่ ข.2).....	106
ภาพที่ ข. 4 การติดตั้ง FileZilla (ต่อภาพที่ ข.3).....	106
ภาพที่ ข. 5 การติดตั้ง FileZilla (ต่อภาพที่ ข.4).....	107
ภาพที่ ข. 6 การติดตั้ง FileZilla (ต่อภาพที่ ข.5).....	107
ภาพที่ ค. 1 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox.....	109

[illegible]

ภาพที่ ค. 23 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox.....	120
ภาพที่ ค. 24 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox.....	121
ภาพที่ ค. 25 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox.....	121
ภาพที่ ค. 26 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox.....	122
ภาพที่ ค. 27 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox.....	122
ภาพที่ ค. 28 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox.....	123



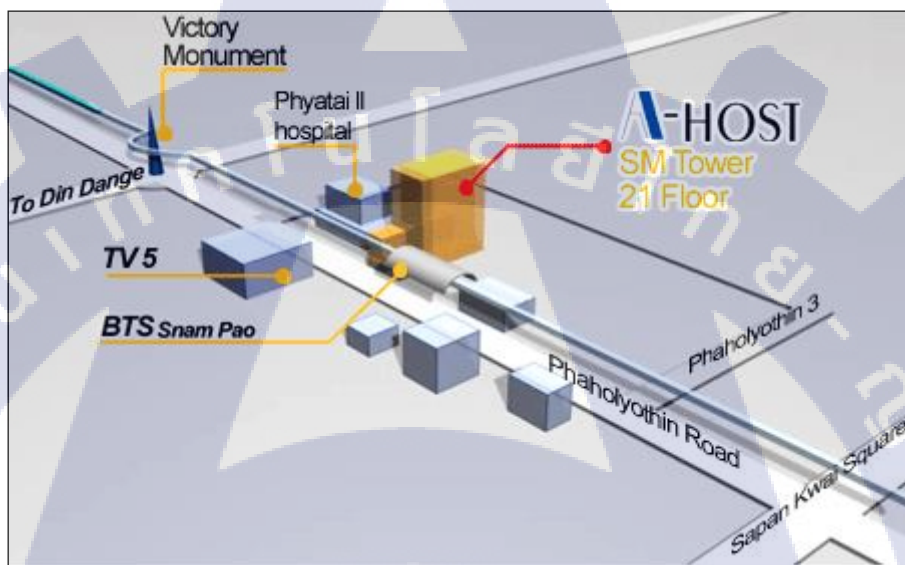
สารบัญตาราง

ตารางที่ 1. 1 หน้าที่งานที่นักศึกษาได้รับมอบหมาย	7
ตารางที่ 2. 1 Unix – Linux Command.....	14
ตารางที่ 2. 2 Unix – Linux Command (ต่อตารางที่ 2.1).....	15
ตารางที่ 2. 3 Unix – Linux Command (ต่อตารางที่ 2.2).....	16
ตารางที่ 2. 4 Unix – Linux Command (ต่อตารางที่ 2.3).....	17
ตารางที่ 2. 5 Unix – Linux Command (ต่อตารางที่ 2.4).....	18
ตารางที่ 2. 6 Unix – Linux Command (ต่อตารางที่ 2.5).....	19
ตารางที่ 2. 7 ตัวแปรแบบพิเศษ.....	27
ตารางที่ 2. 8 เปรียบเทียบจำนวนเต็ม	28
ตารางที่ 2. 9 เปรียบเทียบ String	29
ตารางที่ 2. 10 ตรวจสอบ File	29
ตารางที่ 2. 11 เงื่อนไขอื่นๆ.....	29
ตารางที่ 3. 1 แผนงานปฏิบัติงาน	37
ตารางที่ 4. 1 เมนูหลักของ Script สำหรับการ Monitor	48
ตารางที่ 4. 2 เมนูสำหรับ Monitor Hardware ของ Shell Script สำหรับการ Monitor.....	49
ตารางที่ 4. 3 เมนูสำหรับ Monitor System ของ Shell Script สำหรับการ Monitor.....	51
ตารางที่ 4. 4 เมนูสำหรับเข้าดูไฟล์ Backup ของ Shell Script สำหรับการ Monitor.....	53
ตารางที่ 4. 5 เมนูสำหรับเข้าไปสร้างไฟล์ข้อมูลเก็บค่าสถานะระบบแบบสรุปของ Shell Script สำหรับการ Monitor	54

บทที่ 1

บทนำ

1.1 ชื่อและสถานที่ตั้งของสถานประกอบการ



ภาพที่ 1.1 สถานประกอบการ A-HOST Company Limited

ชื่อสถานประกอบการ: A-HOST Company Limited.

ที่ตั้งสถานประกอบการ: 979/53 SM Tower 21st Fl., Phaholyothin Rd.,
Samsennai, Phayatai, Bangkok 10400

Tel: 02-298-0625-32 Ext. 5309 | Fax: 02-298-0053

1.2 ลักษณะธุรกิจของสถานประกอบการหรือการให้บริการขององค์กร

บริษัทเอ-โฮสต์จำกัดได้ก่อตั้งขึ้นเมื่อปีพ.ศ.2542 ในฐานะหนึ่งบริษัทในเครือของบริษัทชิสเต็มส์คอร์ปอเรชั่น (มหาชน) จำกัดและเป็นผู้เชี่ยวชาญด้านการบริหารจัดการระบบสารสนเทศ (Information Technology : IT) และบริการเสริมต่างๆสำหรับลูกค้าตั้งแต่เริ่มธุรกิจขนาดย่อมไปจนถึงขนาดกลางธุรกิจหลักของบริษัทเอ-โฮสต์คือการให้บริการโฮสติ้ง (Hosting) ซึ่งเป็นการเปิดเครื่องคอมพิวเตอร์แม่ข่ายในการให้เข้าพื้นที่เพื่อวางระบบและบริการระบบสารสนเทศด้วยผลิตภัณฑ์ของออราเคิล (Oracle) เป็นซอฟต์แวร์สำหรับการวางแผนบริหารทรัพยากรของระดับแนวหน้าของโลก

เอโฮสต์ถือกำเนิดขึ้นจากกลุ่มผู้เชี่ยวชาญด้านสารสนเทศท่ามกลางภาวะเศรษฐกิจตกต่ำทั่วภูมิภาคเอเชียแต่เอ-โฮสต์ก็สามารถเติบโตอย่างรวดเร็วและมั่นคงตั้งแต่แรกก่อตั้งด้วยจุดแข็งในฐานะผู้บุกเบิกธุรกิจโฮสติ้งเซอร์วิสพร้อมทั้งนำธุรกิจแนวใหม่อย่างการให้บริการระบบโปรแกรมประยุกต์หรือ ASP (Application Services Provider) เข้ามาให้บริการเป็นรายแรกในเมืองไทยอีกทั้งยังเป็นผู้ให้บริการรายแรกนอกประเทศสหรัฐอเมริกา

ในฐานะผู้นำในอุตสาหกรรมนี้เป็นเวลามากกว่า 10 ปีเอ-โฮสต์ได้เสริมสร้างความแข็งแกร่งทางธุรกิจด้วยบริการที่มีความโดดเด่นและรวบรวมเอาทรัพยากรบุคคลซึ่งได้สั่งสมประสบการณ์และความชำนาญเอาไว้อย่างพร้อมเพรียงส่งผลให้ศูนย์ข้อมูลของเอ-โฮสต์ในปัจจุบันมีความสมบูรณ์ด้วยกลุ่มเซิร์ฟเวอร์ (Server) ที่เชื่อมต่อกันในลักษณะการจัดกลุ่ม (Clustering) ซึ่งเปี่ยมสมรรถนะสามารถให้บริการแก่ผู้ใช้งานจำนวนมากได้ในเวลาเดียวกัน

นอกจากนี้เอ-โฮสต์ยังคิดค้นระบบรักษาความปลอดภัยระบบสำรองข้อมูลและระบบบริหารจัดการรวมถึงอุปกรณ์ต่างๆอย่างครบครันเพื่อให้เอ-โฮสต์สามารถตอบสนองต่อระดับความต้องการในระดับสูงสุดที่ลูกค้าคาดหวังได้ตลอดจนเป็นการสร้างความมั่นใจให้แก่ลูกค้าที่ใช้บริการโฮสติ้งและแอปพลิเคชันต่างๆของเอ-โฮสต์ว่าจะได้รับทั้งประสิทธิภาพและความปลอดภัยอย่างครบครันธุรกิจการให้บริการระบบโปรแกรมประยุกต์ในรูปแบบ ASP เอโฮสต์ไม่เพียงแต่ให้บริการด้านโปรแกรมประยุกต์ด้านการดำเนินธุรกิจทางอิเล็กทรอนิกส์ระดับโลกของออราเคิลพร้อมโครงสร้างพื้นฐานทางเทคโนโลยีสารสนเทศเท่านั้นแต่ยังมีบริการที่ครอบคลุมตั้งแต่การให้คำปรึกษาการสนับสนุนและการให้บริการทั่วไปอย่างพร้อมพร้อมครบครัน

นอกจากธุรกิจโฮสติ้งและธุรกิจการให้บริการโปรแกรมประยุกต์ในรูปแบบ ASP ซึ่งถือเป็นธุรกิจหลักเอ-โฮสต์ยังดำเนินธุรกิจอย่างต่อเนื่องโดยการขยายหน่วยงานใหม่เพิ่มขึ้นนั่นก็คือ Core Technology Division หน่วยงานเทคโนโลยีหลักที่ให้คำตอบเบ็ดเสร็จแก่ลูกค้าด้วยระบบฐานข้อมูลและเครื่องมือต่างๆของออราเคิลซึ่งช่วยเพิ่มความสามารถของลูกค้าในการออกแบบพัฒนา

ปรับเปลี่ยนระบบแอปพลิเคชันให้เหมาะสมกับธุรกิจนั้นๆภายใต้คำปรึกษาแนะนำและการวางระบบของเอ-โฮสต์ลูกค้าสามารถบริหารระบบฐานข้อมูลของตนเองและดูแลระบบดังกล่าวได้อย่างมีประสิทธิภาพ

ตลอดระยะเวลามากกว่า 10 ปีในการดำเนินธุรกิจเอ-โฮสต์ไม่เพียงแต่ในฐานะผู้บุกเบิกธุรกิจโฮสติ้งและธุรกิจการให้บริการโปรแกรมประยุกต์ในรูปแบบ ASP เท่านั้นแต่เอ-โฮสต์ยังได้ทำการติดตั้งระบบสารสนเทศรวมทั้งผลิตภัณฑ์ของออราเคิลให้กับลูกค้าจนประสบความสำเร็จเป็นจำนวนมากซึ่งหลายรายเป็นหนึ่งในร้อยบริษัทชั้นนำของประเทศไทยแต่ที่สำคัญกว่านั้นก็คือการที่เอ-โฮสต์ได้สารสัมพันธ์กับลูกค้าและพันธมิตรทางธุรกิจอย่างแนบแน่นจนกลายเป็นหุ้นส่วนทางกลยุทธ์และได้รับแต่งตั้งให้เป็น OCAP (Oracle Certified Advantage Partner) รายแรกในประเทศไทย



ภาพที่ 1.2 รางวัลที่ A-HOST Company Limited ได้รับ

1.2.1 Hosting & Outsource Service

เอ-โฮสต์ได้ปรับปรุงและขยายการบริการ Hosting และการให้บริการภายนอก (Outsource) จนสามารถครอบคลุมความต้องการของลูกค้าได้อย่างหลากหลายโดยยึดหลักในการให้บริการที่เรียกว่า “Peace of Mind for the customer” ซึ่งหมายถึงการที่จะทำงานให้กับลูกค้าแบบครบวงจรเพื่อให้ลูกค้าจะสามารถใช้งานระบบสารสนเทศได้อย่างมีประสิทธิภาพได้อย่างสบายใจไร้ความกังวลต่อความเสี่ยงต่างๆไม่ว่าจะเป็นเรื่องของปัญหาทางเทคนิคการจัดระบบและข้อมูลสำรองการ

ปรับแต่งระบบให้ได้ประสิทธิภาพสูงสุด (Performance Tuning) และที่สำคัญที่สุดคือการที่เข้ามาบริการในด้านการบริหารจัดการบุคลากรทางด้านสารสนเทศทั้งหมดแทนลูกค้า

การใช้บริการ Hosting และ Outsource จะทำให้ลูกค้าสามารถทุ่มเทกำลังสมองเวลาและทรัพยากรขององค์กรให้กับธุรกิจที่เป็นแกนหลัก (Core Business) ซึ่งเป็นสิ่งที่ลูกค้าถนัดกว่า โดยทั่วไปแล้วบริการ Hosting และ Outsource จะประกอบด้วยส่วนประกอบและบริการย่อยๆ ดังต่อไปนี้

1. High Availability and High Performance IT Infrastructure
2. Dedicated or Co-Location Service
3. Disaster Site
4. Oracle E-Business Application (ERP, CRM, SCM)
5. ERP Implementation Service
6. System and Database Administration
7. Help Desk
8. On-Request Service i.e. On-Site Support, Software Customization

ทั้งนี้การบริการ Hosting และ Outsource สามารถครอบคลุมได้ทั้งระบบที่ใช้เทคโนโลยีของอราเคิลและระบบที่ใช้เทคโนโลยีอื่นๆ

1.2.2 Oracle Core Technology Products and Advanced Services

เอ-โฮสต์เป็นผู้ดำเนินการดำเนินธุรกิจในฐานะผู้แทนจำหน่ายเพิ่มมูลค่าให้กับอราเคิลโดยไม่เพียงแต่ทำหน้าที่ในการจัดจำหน่ายสินค้าในกลุ่มแกนหลักของเทคโนโลยี (Core Technology) ของอราเคิลทุกประเภทแต่ยังมีทีมผู้เชี่ยวชาญที่จะให้การสนับสนุนและบริการเสริมอย่างครบวงจรแก่บริษัทลูกค้าและลูกค้าไม่ว่าจะเป็นการร่วมจัดกิจกรรมทางการตลาดการฝึกอบรมการติดตั้งระบบและการให้คำปรึกษาเพื่อแก้ไขปัญหาต่างๆ สินค้าและบริการที่อยู่ในกลุ่ม Oracle Core Technology Products and Advanced Services ได้แก่

1. Oracle Database and Database Options
2. Oracle Business Intelligence Suite
3. Business Partner Development
4. System Installation, Integration and Optimization

5. Oracle Fusion Middleware (รวมถึง BEA)
6. สิ้นค้าอื่นๆทุกประเภทของออราเคิล
7. Marketing and Lead Generation Activities
8. SOA-Based Development and Implementation

ผลสำเร็จในการดำเนินธุรกิจประเภทนี้ทั้งในด้านการตลาดและบริการทำให้เอ-โฮสต์ได้รับรางวัล ASEAN Partner of the Year ในปี 2005

1.2.3 Oracle Enterprise Performance Management (EPM) และ Hyperion Business

Intelligence Products and Services

ความต้องการสูงสุดประการหนึ่งของผู้บริหารในการนำเอาระบบสารสนเทศมาใช้ในการองค์กรไม่ว่าจะเป็นภาครัฐหรือเอกชนคือการทำให้ผู้บริหารสามารถได้ข้อมูลที่แสดงให้เห็นถึงสถานะในการดำเนินธุรกิจได้อย่างแม่นยำรวดเร็วและการนำเอาข้อมูลมาวิเคราะห์และวางทั้งในระดับปฏิบัติการและในระดับกลยุทธ์เพื่อให้ธุรกิจสามารถได้เปรียบในการแข่งขันปรับตัวตามเศรษฐกิจได้ในทุกสถานการณ์ Oracle Enterprise Performance Management (EPM) และ Hyperion Business Intelligence จัดเป็นซอฟต์แวร์ชั้นนำของโลกที่สามารถสนองต่อความต้องการในลักษณะดังกล่าวได้เป็นอย่างดี

เอ-โฮสต์มีทีมงานที่ปรึกษาที่มีประสบการณ์ทั้งทางด้านธุรกิจและทางด้านเทคนิค รวมถึงความเข้าใจในระบบของ ERP ของออราเคิลอย่างลึกซึ้งจึงทำให้สามารถให้บริการที่ปรึกษาเพื่อออกแบบติดตั้งเชื่อมโยงและปรับใช้ระบบให้กับลูกค้าจนเกิดประสิทธิภาพสูงสุด อีกทั้งยังมีความยืดหยุ่นและให้การตอบสนองที่เร็วกว่าเมื่อเทียบกับการว่าจ้างที่ปรึกษาจากต่างประเทศ

1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร



ภาพที่ 1.3 คณะผู้บริหาร A-HOST Company Limited

1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย

ตำแหน่ง: System Engineer

หน่วยงาน: System & Application Managed Services Division

KPI Description	Expected Results
1. ศึกษาค้นคว้าข้อมูล Shell Script	- มีความเข้าใจเกี่ยวกับ Shell - สามารถออกแบบ Script และเขียน Script ได้
2. เขียน Shell Script	- สามารถเขียน Script ให้สามารถ Monitor, View Backup Log File และ Generate Monthly Report - สามารถเขียน Shell Script ให้สามารถแจ้งเตือน Email เมื่อเกิดเหตุการณ์
3. ช่วยสนับสนุนงานด้านต่างๆ และ แก้ปัญหาภายในบริษัท	- ช่วยสนับสนุนการแก้ไขปัญหาด้านการใช้งาน Email, Internet และ Printer - ช่วยเตรียมเครื่อง Computer และ Notebook - ช่วย Configure Switch และ Router - ช่วยลงโปรแกรมพื้นฐานต่างๆ ให้ Computer และ Notebook ของบริษัท - สร้าง Layout แผนผังบริษัท A-HOST Company Limited. - สร้าง Layout แผนผังข้อมูล Printer ภายในบริษัท - สร้าง Layout แผนผังข้อมูล Network ในบริษัท

ตารางที่ 1.1 หน้าที่งานที่นักศึกษาได้รับมอบหมาย

1.5 พนักงานที่ปรึกษาและตำแหน่งของพนักงานที่ปรึกษา

นายภานุพงษ์ วงษ์ประดิษฐ์

ตำแหน่ง: System Engineer

1.6 ระยะเวลาที่ปฏิบัติงาน

ระยะเวลาปฏิบัติงานสหกิจศึกษาเป็นเวลา 4 เดือน

- เริ่มต้นปฏิบัติงานสหกิจศึกษา วันที่ 4 มิถุนายน 2561
- สิ้นสุดงานสหกิจศึกษา วันที่ 28 กันยายน 2561

1.7 วัตถุประสงค์ของการปฏิบัติงานและโครงการที่ได้รับมอบหมาย

- เพื่อศึกษาการและหาประสบการณ์การทำงานจริงภายในองค์กร
- เพื่อนำความรู้ที่ได้รับไปใช้ในการพัฒนาโครงการและการทำงานในอนาคต

1.8 ผลที่คาดว่าจะได้รับจากการปฏิบัติงานและโครงการที่ได้รับมอบหมาย

- ได้ประสบการณ์ในการทำงานจริง
- ได้นำความรู้มาประยุกต์ใช้ในการทำงาน

บทที่ 2

ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน

2.1 ทฤษฎีที่เกี่ยวข้องในการทำงาน

2.1.1 Unix

Unix จัดอยู่ในกลุ่มระบบปฏิบัติการ (OS) แบบ Multitasking หรือ Multiuser ซึ่งถือกำเนิดที่สถาบัน Bell Labs วัตถุประสงค์หลักที่พัฒนาขึ้นมาเพื่อเป็น Platform สำหรับการเขียน Software เพื่อใช้ใน Run ระบบอื่นๆ แต่ก็มีขยายขอบเขตออกไปจนในที่สุดกลายเป็นระบบปฏิบัติการ ซึ่งลักษณะของ Unix คือการใช้งานด้วย Text และเก็บข้อมูลเป็นลำดับชั้น มีเครื่องมือ Command ให้ใช้งานมากมาย และทำงานรวมกันโดยใช้ Pipe (|) เป็นตัวเชื่อม

ระบบของ Unix ถูกบริหารจัดการภายใต้ Program หลักคือ Kernel เพื่อใช้ในการ Start/Stop Program อื่นๆและใช้ในการจัดการ File System ในระดับล่าง อีกทั้งยังคอยจัดการ Resource ที่มีอยู่ให้ Program อื่นๆใช้งานได้โดยไม่ชนกัน

ประวัติของ Unix

หลังจากถือกำเนิดขึ้นของ “Multics” ซึ่งเป็น Operating System ในช่วง 1960 ที่ Bell Labs ประสบปัญหามากมายทำให้ทาง Bell Labs ถอด Project นี้ออก และตัดสินใจทำใหม่โดยวางขอบเขตให้เล็กลงตั้งชื่อใหม่ว่า “Unics” (Uniplexed Information and Computing System) แต่เนื่องจากเสียงพ้องกับ “Multics” ภายหลังจึงเรียกว่า “Unix”

ในปี 1972 ได้มีการเขียน Unix ใหม่ด้วยภาษา C ซึ่งแต่เดิมเป็น Assembly ให้เป็นภาษาขั้นสูงแทน (ภาษา C) ผลคือมันทำให้ Software ทำงานได้ไวขึ้น และก็เขียน Code ได้สั้นลง หลังจากนั้นช่วง 1980 ก็เกิดการขยายตัวในตลาดอย่างมากทำให้เกิด Unix ขึ้นหลายชนิด เช่น HP-UX, Solaris, AIX และในปี 2000 ทาง Apple เองก็นำมาใช้เป็นพื้นฐานของระบบปฏิบัติการ Mac OS X ตัว Unix จึงถูกออกแบบให้ใช้งานกับระบบ Server, Workstations และ อุปกรณ์ Mobile

การทำงานบน Unix

Unix จะมี Program ที่เรียกว่า Unix Shell สำหรับคำสั่งผ่าน Command Line ถ้าเทียบกับ Windows ก็คือ cmd.exe โดยบน Unix แต่ละ User ที่เข้ามาใช้งานสามารถเลือก Shell หลายแบบเพื่อใช้งาน ตามความต้องการที่ต่างกัน

- Login Shell ทำหน้าที่หลังจาก Login สำเร็จ ทั้งต่อเข้าเครื่องโดยตรง (Console) และต่อผ่าน Telnet
- Interactive Shell สามารถรับคำสั่งแต่ไม่ได้เริ่มตั้งแต่ Login เช่น การเปิด Terminal
- Non-Interactive Shell เป็นการทำงานเพื่อ Run Script หรือ ชุดคำสั่ง

ประเภทของ Shell Unix

- Bourne Shell

Bourne Shell (sh) ผู้สร้างคือ “Stephen Bourne” โดยปกติแล้วชื่อ Program จะเป็น “sh” อยู่ที่ Path “/bin/sh” ซึ่งมีการพัฒนาต่อออกมามากหลายชนิด ดังนี้

1. Almquist Shell (ash)
2. Bourne-Again Shell (bash)
3. Debian Almquist Shell (dash)
4. Korn Shell (ksh)
5. MirBSD Korn Shell (mksh)
6. Z Shell (zsh)

- C Shell

C Shell (csh) เขียนขึ้นโดย “Bill Joy” อาศัยพื้นฐานจากรูปแบบของภาษา C โดยสามารถรองรับ Feature แบบ Interactive มากมาย ในหลายๆระบบปัจจุบันอาจจะมีการ Link ไปที่ TENEX C Shell (tcsh) แทน ซึ่งมีการพัฒนาจาก csh ให้ดีขึ้นและ Feature หลายอย่างก็ถูกนำไปใช้ใน Shell ประเภทอื่นๆด้วย

2.1.2 Linux

Linux คือ Unix ชนิดหนึ่งซึ่งเป็นระบบปฏิบัติการที่เป็น Open Source Software โดยมีการพัฒนาแจกจ่ายให้ผู้ใช้งานใช้งานได้ฟรี ตามความหมายของ Linux แล้วจริงๆหมายถึง Linux Kernel หรือ Operating System Kernel ซึ่งทำหน้าที่เป็นตัวกลางเชื่อมต่อระหว่าง Hardware และ Application เพื่อบริหารจัดการ Resource ที่มีอยู่ให้เหมาะสม

เริ่มแรก Linux เกิดขึ้นจากการพัฒนาบน Computer (PC) ที่ใช้ Chipset Intelx86 (32bit) แต่แล้วก็มีการพัฒนาให้รองรับ Platform อื่นๆทั่วไป เพราะการเข้าครอบครองตลาดของ Android บน Smartphone ทำให้ Linux กลายเป็นระบบปฏิบัติการที่แพร่หลายมาก อีกทั้งยังสามารถใช้งานในด้าน Server หรือ Mainframe

Desktop Linux เริ่มมีการเพิ่มให้รองรับการทำงานควบคู่ไปกับระบบ Windows เช่น X11, Mir และ Wayland บางชนิดออกแบบมาสำหรับ Server ซึ่งจะตัดส่วนที่เป็น Graphical ออก แล้วเพิ่ม Software ที่ไว้ทำงานเป็น Application Platform อย่างเช่น LAMP เพราะ Linux ค่อนข้างเสรีในการติดตั้งหรือปรับแต่งระบบเพื่อตอบสนองต่อความต้องการของ User

ประวัติของ Linux

แรกเริ่มจากการคิดค้น Unix ขึ้นมาไว้สำหรับเป็นระบบปฏิบัติการสำหรับ Server และระบบขนาดเล็ก ต่อมาได้มีคนพัฒนา Unix สำหรับ Personal Computer ขึ้นมาในชื่อ “MINIX” ในปี 1987 แต่เนื่องจากอยู่ภายใต้ License สำหรับด้านการศึกษาเท่านั้น ทำให้นาย Linus Torvalds เริ่มพัฒนา Operating System Kernel ของตนเองขึ้นมาเพื่อใช้งาน ซึ่งสุดท้ายแล้วก็กลายมาเป็น Linux Kernel

ส่วนประกอบของ Linux Operation System

- The Bootloader เป็น Software ที่ทำหน้าที่จัดการเรื่องการ Boot ของ Computer สำหรับ User มันก็คือหน้าจอที่ใช้แสดงขึ้นในช่วงที่กำลังเริ่มเข้าสู่ระบบปฏิบัติการ
- The Kernel ส่วนนี้เรียกได้ว่าเปรียบเสมือนคำเรียกของ “Linux” เพราะมันคือระบบส่วนกลางที่ทำหน้าที่จัดการทรัพยากรต่างๆเช่น CPU, Memory และอุปกรณ์ต่อเสริมต่างๆ เป็น Layer ต่ำสุดที่อยู่ใกล้กับ OS
- Daemons เป็นส่วนที่ทำงานอยู่เบื้องหลัง (Background Service) เริ่มทำงานตั้งแต่ระหว่างที่ Boot และเริ่ม Login เข้าสู่ระบบ
- The Shell เป็นคำที่มักจะคุ้นเคยกันสำหรับ Linux เพราะว่า Shell คือการทำงานของคำสั่งที่ทำให้คุณสามารถควบคุมและสั่งการผ่าน การพิมพ์ตัวอักษรเข้าไป ซึ่งเป็นส่วนหนึ่งที่ทำให้

User หลายคนค่อนข้างกลัวในการใช้งาน แต่ใน Linux Desktop รุ่นใหม่ไม่จำเป็นต้องใช้ Command Line แล้ว

- Graphical Server เป็นระบบที่ช่วยเสริมการแสดงผลบนจอ Monitor
- Desktop Environment คือส่วนที่ User ใช้งานจริง ซึ่งมีให้เลือกได้หลายที่โดยซึ่งก็คือชุดของ Application ต่างๆที่ถูกจำมารวมกัน เช่น Managers, Configuration Tools, Web Browsers, Games
- Applications เนื่องจาก Desktop Environment นั้นไม่ได้จัด Application มาครบเหมือน Windows หรือ Mac เนื่องจาก Linux มี Software ที่มีคุณภาพที่ง่ายต่อการค้นหาและติดตั้ง Linux ที่ได้รับความนิยมส่วนใหม่มักจะมีเครื่องมือที่ใช้สำหรับการค้นหาและติดตั้ง Application ติดมาให้ เช่น Ubuntu Linux ก็จะมี Software Center คือ apt ที่ใช้ในการ Download และติดตั้ง Application จากศูนย์กลาง

2.1.3 Kernel

Kernel คือ Program ที่เป็นศูนย์กลางในระบบ Computer ทำหน้าที่ควบคุมการทำงานตั้งแต่เริ่ม Boot Server รวมถึงการ Start/Stop Program และ Input/Output จาก Software ทั้งหมด คอยจัดการทรัพยากรต่างๆ หรือ Hardware เช่น Keyboard, Monitor, Printer, Speaker โดย Kernel ทำหน้าที่เชื่อมการทำงานระหว่าง Software และ Hardware เข้าด้วยกันนั่นเอง

ส่วนที่เป็น Code สำคัญของ Kernel จะถูก Load เข้าส่วน Memory ที่จองไว้ให้เฉพาะ เพื่อป้องกันการเขียนทับจาก Application อื่น การเชื่อมต่อของ Kernel เป็นระดับต่ำสุดที่เกี่ยวข้องกับ Hardware เมื่อมีการ Request ไปยัง Kernel เราจะเรียกว่า System Call และส่วนที่ใช้งานจะเรียกว่า Resource

หน้าที่ของ Kernel

หลักคือทำงานเป็นสื่อกลางในการเข้าถึงทรัพยากรของระบบดังนี้

1. หน้าที่ของ Central Processing Unit

ทำหน้าที่ควบคุมจัดการ Program ที่กำลังใช้งานโดย Kernel จะรับผิดชอบในการตัดสินใจว่า Program แต่ละตัวจะจองหน่วยประมวลผล Core ไหนและกี่ Core ในการทำงาน

2. หน้าที่ของ Random-Access Memory

ใช้ในการเก็บข้อมูลของ Program ที่ใช้งาน ซึ่งโดยปกติจะมี Program จำนวนมากเข้ามาใช้งานตลอดเวลาตามความต้องการของแต่ละ Application ซึ่ง Kernel มีหน้าที่ตัดสินใจว่า Memory ส่วนไหนที่ Process แต่ละอันสามารถใช้งานได้และควรทำอะไรเมื่อ Memory ไม่เพียงพอ

3. Input/Output (I/O) Devices

I/O ของแต่ละอุปกรณ์ เช่น Keyboard, Mouse, Disk, Printer, Network Adapter หรือ จอ Monitor ทั้งหมดนี้ Kernel จะควบคุมการสื่อสารระหว่าง Application และ Hardware

ประเภทของ Kernel

1. Monolithic Kernels

เกิดขึ้นในยุคเริ่มแรกของ Kernel โดยระบบพื้นฐานทั้งหมดเช่น Process และการจัดการ Memory จะถูกรวมอยู่ใน Module เดียวกันภายใน Kernel ซึ่งเป็นผลทำให้ Kernel มีขนาดใหญ่และยากต่อการดูแล ภายหลังจึงได้มีการแยก Module ออกมาและทำการเลือก Load ใช้งานตามความเหมาะสม เป็นเสมือน Extension ให้ OS เลือกใช้ทำให้ไม่ต้องทำการปิดและ Compile ใหม่ทั้งหมด เมื่อมีที่แก้ Bug

ปัจจุบัน Linux ออกแบบตาม Monolithic ตัวอย่างเช่น

- Linux
- MS-DOS
- Microsoft Windows 9x Series
- Agnix

2. Microkernels

จากปัญหาในเรื่องขนาดของ Kernel ที่โตขึ้นเรื่อยๆ ของ Monolithic ทำให้มีการแยกส่วนของระบบพื้นฐานเช่น Driver, Protocol Stack, File System ออกมา Run ข้างนอก ทำให้ลดขนาดของ Kernel ลง และยังเพิ่ม Security และ Stability ให้กับ OS อีกด้วย โดยทั้งหมดจะทำงานในส่วนของผู้ใช้ Space และทำงานบนระบบตามการเรียกใช้ของ Program ระบบปฏิบัติการ QNX ออกแบบตาม Microkernel ตัวอย่างเช่น

- AIX
- AmigaOS
- Amoeba

- Android OS
- Chorus Microkernel
- EROS
- Haiku

3. Hybrid Kernels

ถูกนำมาใช้งานกับ OS ระดับ Commercial มีลักษณะคล้าย Microkernel ยกเว้นแต่ว่ามันได้รวมเอา Code เสริมใน Kernel Space มาเพิ่มความสามารถโดยใช้เป็น Extension ให้กับ Microkernel ด้วยคุณสมบัติของ Monolithic Kernel ซึ่งต่างจาก Monolithic แท้ๆเพราะ Monolithic ไม่สามารถ Load Module ในขณะที่ทำงานได้ เพราะฉะนั้นจึงสรุปได้ว่า Hybrid Kernel เป็น Microkernel ที่มี Code เสริมบางอย่างบน Kernel Space ที่ช่วยทำงานได้ไวขึ้น

ตัวอย่างเช่น

- Microsoft Windows NT 3.1, NT 3.5, NT 3.51, NT 4.0, 2000, XP
- macOS

2.1.4 คำสั่ง Unix – Linux Command

คำสั่งพื้นฐานที่ทำงานบนระบบ Unix และ Linux ผ่าน Shell เพื่อใช้งานคำสั่ง Unix – Linux Command บางคำสั่ง Linux Command อาจต้องใช้ User Root เพราะเป็นคำสั่งที่อันตรายเนื่องจากเกี่ยวข้องกับ File System หรือ Security

ลำดับ	คำสั่ง Unix - Linux	รายละเอียด
1	adduser	ใช้เพิ่มหรือเปลี่ยนแปลง User บน Linux
2	arch	ใช้แสดงรุ่น Hardware ของเครื่อง Server
3	awk	ใช้ค้นหาข้อมูล Text ในรูปแบบที่ซับซ้อน
4	basename	คำสั่งใช้แสดงเฉพาะส่วนของชื่อ Filename
5	bc	คำสั่งคำนวณตัวเลข ตามสูตรทางคณิตศาสตร์
6	cal	คำสั่งแสดงปฏิทิน วันเดือนปี
7	cat	แสดงผลข้อมูลภายใน File ในรูปแบบ Text
8	chgrp	คำสั่งเปลี่ยนเจ้าของ Group ของ File

ตารางที่ 2.1 Unix – Linux Command

ลำดับ	คำสั่ง Unix - Linux	รายละเอียด
9	chmod	คำสั่งเปลี่ยนสิทธิ์ในการเข้าถึง File
10	chown	คำสั่งเปลี่ยนเจ้าของ File หรือ Directory
11	cksum	คำสั่งนับจำนวน Bytes ของ File
12	clear	คำสั่งล้างหน้าจอ Screen
13	cmp	คำสั่งวิเคราะห์เปรียบเทียบ Files ในระดับ Bytes
14	comm	คำสั่งวิเคราะห์เปรียบเทียบ File ที่ละบรรทัด
15	cp	คำสั่งทำสำเนาหรือ Copy ข้อมูล
16	cron	ควบคุมการเริ่มทำงานของ Job Schedule
17	crontab	ใช้ตั้งเวลาให้คำสั่งเริ่มทำงานตามที่ต้องการ
18	csplit	คำสั่งแตก File ตามจำนวนบรรทัด
19	cut	คำสั่งตัดข้อมูล File เป็น Field Column
20	date	คำสั่งแสดงเวลาวันเดือนปี
21	dc	คำสั่งเครื่องคิดเลขแบบตั้งโต๊ะ
22	dd	คำสั่ง Backup ข้อมูลใน Harddisk
23	df	คำสั่งแสดงข้อมูลพื้นที่ Disk ทั้งหมด
24	diff	คำสั่งวิเคราะห์เปรียบเทียบ File ที่ละบรรทัด
25	dir	คำสั่งแสดงข้อมูล Directory
26	dircolors	คำสั่งที่ใช้ในการปรับสีของผลลัพธ์ ls
27	dirname	คำสั่งแสดงชื่อ Directory ของ File
28	du	คำสั่งดูข้อมูลรายละเอียดขนาด File
29	echo	คำสั่งในการแสดงผลบนหน้าจอ Screen
30	ed	คำสั่ง Editor File ชนิดหนึ่ง
31	egrep	คำสั่งค้นหบบรรทัดใน File ที่ตรงเงื่อนไข
32	env	คำสั่งสร้าง Environment ในการ Run Program
33	expand	คำสั่งเปลี่ยนข้อมูล File จาก Tab เป็น Space
34	expr	คำสั่งที่ใช้ประมวลผลตรรกะคณิตศาสตร์

ตารางที่ 2.2 Unix – Linux Command (ต่อตารางที่ 2.1)

ลำดับ	คำสั่ง Unix - Linux	รายละเอียด
35	factor	คำสั่งแยกตัวประกอบทางคณิตศาสตร์
36	fdisk	คำสั่งบริหารจัดการ Disk Partition
37	find	คำสั่งใช้ในการค้นหา File หรือ Directory
38	fmt	คำสั่งจัดเรียงข้อมูลภายใน File ในรูป Format
39	fold	คำสั่งจัดเรียงความยาวตัวอักษรแต่ละบรรทัด
40	free	คำสั่งแสดงข้อมูลการใช้งาน Memory
41	fsck	คำสั่งตรวจสอบและซ่อมแซม File System
42	gawk	ใช้ค้นหาข้อมูล Text ในรูปแบบเดียวกับ awk
43	grep	คำสั่งค้นหาบรรทัดใน File ที่ตรงเงื่อนไข
44	groups	คำสั่งแสดงข้อมูล Group ของ System User
45	gunzip	คำสั่งยกเลิกการบีบอัดข้อมูล File
46	gzip	คำสั่งบีบอัดข้อมูล File หรือ การ Zip File
47	head	คำสั่งแสดงข้อมูลบางส่วนภายใน File
48	hostname	คำสั่งแสดงข้อมูลชื่อของเครื่อง Server
49	id	คำสั่งแสดงข้อมูล User, Group ในระบบ
50	ifconfig	คำสั่งแสดงข้อมูลและเปลี่ยนค่า Interface Server
51	info	คำสั่งข้อมูลโปรแกรมบนระบบทั้งหมดที่ใช้งาน
52	iptables	คำสั่งจัดการกรอง IP Port ที่เข้ามาใช้งาน
53	join	คำสั่งเชื่อมข้อมูล 2 File ด้วย Field ที่เหมือนกัน
54	kill	คำสั่งส่ง Signal หรือยกเลิกการทำงาน Process
55	less	คำสั่งอ่านข้อมูลและค้นหาข้อมูลใน File
56	ln	คำสั่งสร้าง Link เชื่อมโยงกันระหว่าง File
57	locate	คำสั่งใช้ในการค้นหา File หรือ Directory
58	logname	คำสั่งแสดงชื่อ User Login
59	ls	คำสั่งแสดงข้อมูลภายใน Directory
60	man	คำสั่งแสดงคู่มือการใช้งาน Program

ตารางที่ 2.3 Unix – Linux Command (ต่อตารางที่ 2.2)

ลำดับ	คำสั่ง Unix - Linux	รายละเอียด
61	mkdir	คำสั่งสร้าง Directory
62	more	คำสั่งอ่านข้อมูลและค้นหาข้อมูลใน File
63	mount	คำสั่งติดตั้งใช้งานอุปกรณ์ที่เชื่อมต่อ
64	mv	คำสั่งย้ายตำแหน่ง File หรือ Directory
65	netstat	คำสั่งแสดงสถานะ Network Connection ทั้งหมด
66	nice	คำสั่งจัดลำดับความสำคัญของ Process
67	nl	คำสั่งแสดงเลขที่บรรทัดของข้อมูลใน File
68	nohup	คำสั่งป้องกันการหยุดของ Background Process
69	passwd	คำสั่งเปลี่ยน Password ของ System User
70	paste	คำสั่งเชื่อมข้อมูลที่ละบรรทัดจากหลาย File
71	pathchk	คำสั่งเช็ค Path ในระบบว่ามีถูกต้อง
72	ping	คำสั่งตรวจสอบสถานะ Server ปลายทาง
73	pr	คำสั่งแสดงข้อมูลภายใน File ในรูปแบบสิ่งพิมพ์
74	printf	คำสั่งแสดงผลข้อมูลบนหน้าจอ Screen
75	ps	คำสั่งแสดง Process ที่ทำงานใน Server
76	pwd	คำสั่งแสดง Directory หรือ Path ที่อยู่ปัจจุบัน
77	rcp	คำสั่งคัดลอก File ข้ามเครื่อง Server
78	rm	คำสั่งลบ File หรือ Directory
79	rmdir	คำสั่งลบ Directory
80	rsync	คำสั่ง Sync ข้อมูล File ระหว่าง Server
81	screen	คำสั่งสร้าง Session Screen ขึ้นมาใหม่อีกจอ
82	sdiff	คำสั่งวิเคราะห์เปรียบเทียบข้อมูล File ที่ละบรรทัด
83	sed	คำสั่งเปลี่ยนแปลงข้อมูล Text ที่มีรูปแบบซับซ้อน
84	seq	คำสั่งแสดงเลข Sequence Number
85	shutdown	คำสั่งปิดการทำงานของระบบ
86	sleep	คำสั่งหน่วงเวลา

ตารางที่ 2.4 Unix – Linux Command (ต่อตารางที่ 2.3)

ลำดับ	คำสั่ง Unix - Linux	รายละเอียด
87	sort	คำสั่งในการจัดเรียงข้อมูล File ที่ละบรรทัด
88	split	คำสั่งแตก File ตามจำนวนบรรทัด
89	su	คำสั่ง Login ด้วย User ID อื่น
90	sum	คำสั่งการตรวจสอบ Checksum และขนาด Block
91	sync	คำสั่งเขียนข้อมูล Memory ลง Disk
92	tac	คำสั่งแสดงข้อมูลใน File แบบกลับหลัง
93	tail	คำสั่งแสดงข้อมูลบางส่วนภายใน File
94	tar	คำสั่งจัดเก็บรวบรวม File ข้อมูล
95	tee	คำสั่งอ่านข้อมูลพร้อมกับเขียนข้อมูลลง File
96	time	คำสั่งจับเวลาการทำงาน Process
97	top	คำสั่งจัดเรียงอันดับแสดงการทำงานของ Process
98	touch	คำสั่งเปลี่ยนแปลง File Timestamps
99	tr	คำสั่งค้นหาและเปลี่ยนแปลงข้อมูล Text
100	traceroute	คำสั่งแสดงเส้นทางการทำงาน Network
101	tsort	คำสั่งจัดเรียงข้อมูลแบบ Topological
102	tty	คำสั่งแสดงชนิดของ Terminal ที่ใช้งาน
103	uname	คำสั่งแสดงชื่อระบบของ Server
104	unexpand	คำสั่งเปลี่ยน Space เป็น Tab
105	uniq	คำสั่งในการจัดเรียงข้อมูลแบบไม่ซ้ำกัน
106	units	คำสั่งในการแปลงค่าหน่วยวัด
107	useradd	คำสั่งสร้าง User และจัดการ User บนระบบ
108	userdel	คำสั่งลบ User ออกจากระบบ
109	usermod	คำสั่งเปลี่ยนแปลงข้อมูลของ User
110	vdir	คำสั่งแสดงข้อมูล Directory
111	w	คำสั่งแสดง User ที่ Login รวมถึงคำสั่งที่ใช้งาน
112	watch	คำสั่ง Monitor Process ที่ทำงานอยู่

ตารางที่ 2.5 Unix – Linux Command (ต่อตารางที่ 2.4)

ลำดับ	คำสั่ง Unix - Linux	รายละเอียด
113	wc	คำสั่งนับจำนวนคำและบรรทัดจาก File
114	whereis	คำสั่งค้นหาตำแหน่ง File Program
115	which	คำสั่งตำแหน่ง File Program
116	who	คำสั่งแสดงข้อมูล User ที่ Login ขณะนั้น
117	whoami	คำสั่งแสดงชื่อ User ที่ใช้ Login
118	xargs	สร้างคำสั่งใหม่จาก Output ที่ได้ก่อนหน้านี้
119	yes	คำสั่งแสดงข้อมูล Text ที่ต้องการวนซ้ำไปเรื่อยๆ
120	nano	คำสั่งในการสร้างหรือแก้ไข File ข้อมูล Text
121	vi	คำสั่งในการสร้างหรือแก้ไข File ข้อมูล Text
122	telnet	คำสั่งโปรโตคอลเชื่อมต่อสื่อสารด้วยข้อมูลตัวอักษร
123	ssh	คำสั่งเชื่อมต่อ Shell Server แบบเข้ารหัสความปลอดภัย
124	scp	คำสั่งคัดลอก File ข้อมูลแบบเข้ารหัสความปลอดภัย

ตารางที่ 2. 6 Unix – Linux Command (ต่อตารางที่ 2.5)

คำสั่งพื้นฐาน Unix Command หรือ Linux Command จะถูกนำมาใช้เขียน Shell Script หรือชุดคำสั่ง Program เพื่อให้สะดวกและยังลดระยะเวลาในการทำงานได้อย่างมาก

2.1.5 Shell

Shell คือตัวแปลงคำสั่งที่ช่วยให้ผู้ใช้งานสามารถเข้าถึงระบบ Operating System มีด้วยกัน 2 แบบ คือ

- CLI – Command line interface รับคำสั่งโดยข้อมูล text และแสดงผลในรูปแบบ text เช่นกัน
- GUI – Graphical user interface รับคำสั่งโดยอาศัย mouse และ รูปบนจอ monitor

ขึ้นอยู่กับระบบ Computer ที่ใช้งานว่าออกแบบมาสำหรับงานประเภทไหน โดยคำว่า “Shell” มีที่มาจากการทำงานที่มันทำหน้าที่เป็น Layer ครอบคลุมการทำงานของ Kernel อีกที

ส่วนมากแล้วตัว Shell ไม่ได้มีการเชื่อมต่อโดยตรงภายใต้ Kernel ถึงแม้ว่าตัว Shell จะสื่อสารกับผู้ใช้งานผ่านทางอุปกรณ์ที่เชื่อมต่อกับ Computer แต่ Shell ก็ยังคงใช้ Kernel API

เหมือนกับ Application ที่ใช้งานกัน โดย Shell จะเข้ามาบริการจัดการระบบโดยอาศัยการแปลงคำสั่งที่ User ป้อนเข้ามา และนำผลลัพธ์กลับไปแสดง ซึ่งการที่ Shell ทำงานเหมือน Application ทำให้จำเป็นต้องการเปลี่ยนแปลงไปใช้ Application อื่นที่ทำงานเหมือนกัน นอกจากนี้สำหรับ Shell ที่ทำงานในระดับภายใน (Local System) ยังมีวิธีการที่ผู้ใช้งานภายนอกสามารถเข้ามาใช้งานได้โดยอาศัย Remote Access เช่น Secure Shell Protocol ที่นิยมใช้งานกันในรูปแบบ Text Command Line และก็ยังมีการ Remote Desktop ที่ใช้งานแบบ GUI ให้สามารถเข้ามา Remote จัดการได้จากภายนอก

2.1.6 Text Shell (CLI)

Command-Line Interface หรือ CLI เป็น Shell ชนิดแรกเริ่มแรกที่ยึดเพียง Keyboard ในการทำงานส่งข้อมูลให้กับ OS เช่น การ Load ข้อมูล Application, แสดงข้อมูล File หรือ เก็บข้อมูลของผู้ใช้งานเอง โดยระบบปฏิบัติการ (OS) ประเภท Unix มักจะมีจำนวน Shell Program ที่หลากหลายชนิดมาก ด้วยความหลากหลายของคำสั่ง รวมถึง ความสามารถต่างๆที่แตกต่างกัน แต่สำหรับบาง OS ก็อาจจะมี CLI แค่ชนิดเดียวเช่น MS-DOS

ลักษณะของ Shell Prompt คือ \$ หรือมักเรียกกันว่า Command Prompt เมื่อ Prompt นี้แสดงขึ้นมาแสดงว่าระบบเตรียมรอรับคำสั่ง โดย Shell จะอ่านจาก Input ข้อมูลที่ผู้ใช้ป้อนเข้าไปก่อน Enter และทำงานด้วยวิธีตรวจสอบข้อมูลของคำแรกที่ป้อนเข้าว่าคำนั้นไม่ใช่กลุ่มตัวอักษรที่ไม่มีความหมาย รวมถึง Space และ Tab

- Bourne shell (sh)
- Debian Almquist shell (dash)
- Bourne-Again shell (bash)
- C shell (csh)

Shell ทำการอ่านค่า Configure File ตามสภาพแวดล้อมที่ต่างกันของ Shell โดย File พวกนี้จะประกอบด้วยคำสั่งเฉพาะของ Shell แต่ละชนิด รวมถึงพวกตัวแปรต่างๆ และก็รูปสัญลักษณ์ที่แสดงออกมา

2.1.7 Graphical Shell

Graphical Shell หลักๆมีการใช้งานอยู่บน GUI ซึ่งเป็น Graphical User Interface โดยจะอนุญาตให้ผู้ใช้งานสามารถเข้าถึง เปิด-ปิด หรือโยกย้าย รวมถึงย่อขนาดของ Windows ได้ ส่วนมาก

แล้ว GUI จะมีลักษณะที่ประกอบด้วย Desktop Environment, Start Menu และ Task Bar เหมือนการใช้งานใน Window OS ตัวอย่างเช่น

- Window 3.x series
- Window 95, NT
- Window XP
- Linux Desktop

2.1.8 Shell Script

Shell Script คือ การทำงานในแบบ Command-Line Interpreter ผ่าน Shell ทำงานได้เปรียบเสมือนภาษา Program ชนิดหนึ่ง Shell Script ช่วยให้ผู้ใช้สามารถทำงานได้รวดเร็วมากยิ่งขึ้น โดยการรวมเอาคำสั่งและเงื่อนไขที่ผู้ใช้ต้องการใส่ลงใน File และทำการ Execute File นั้นเพื่อผลลัพธ์ที่ได้ในรูปแบบ Text ส่วนมากใช้งานเพื่อให้ทำงานบนระบบอัตโนมัติตามเวลาที่เราต้องการ สำหรับบน Windows หรือ DOS เราจะเรียกสิ่งนี้ว่า Batch File ซึ่งก็คือการเอาชุดคำสั่งมาทำเป็น File เช่นเดียวกับ Shell Script และ Run ผ่าน cmd.exe

โดยปกติแล้ว Unix หรือ Linux ตั้งแต่เริ่มติดตั้งมักจะรวมเอา Korn Shell (ksh) ติดมาให้ตั้งแต่แรก รวมถึง Shell รุ่นแรกๆ เช่น Bourne Shell (sh) , C Shell (csh) นอกจากนี้ยังมี Remote Shell (rsh) หรือ Secure Shell (ssh) ที่ใช้ในการ Remote ควบคุมจากภายนอกระบบ

ความสามารถของ Shell Script คือ

1. เป็นตัวย่อคำสั่ง

Shell Script ช่วยให้การใช้งานระบบมากขึ้น ด้วยการกำหนดค่า Environment รวมถึงคำสั่งต่างๆ ที่ต้องทำ รวมเข้าไปไว้ใน Script เดียวแต่ยังทำงานเหมือนคำสั่ง Unix ปกติ เช่น Shell Script สำหรับการแสดงข้อมูล File (ls) แต่ทำการล้างข้อมูลบน Screen ก่อน (clear)

```
#!/bin/sh
clear
ls -al
```


2. รวบรวมชุดคำสั่ง

งานที่ต้อง Run คำสั่งซ้ำๆ กันหลายครั้ง หรือ Batch Job สามารถรวบรวมเข้าเป็น Shell Script เพื่อให้ระบบค่อยๆ ทำงานทีละชุดคำสั่งไปเรื่อยๆ แทนที่จะต้องมาคอย Run คำสั่งเองทีละบรรทัด ตัวอย่างเช่น Directory ที่มีข้อมูล Source Code C ที่เราต้องทำการ Compile ทีละ File เราสามารถใช้ Shell Script ของ C Shell ในการช่วย Compile File ให้อัตโนมัติ

```
#!/bin/csh
echo compiling...
cc -c foo.c
cc -c bar.c
cc -o myprog foo.o bar.o qux.o
```

3. การใช้งานทั่วไป

นอกจาก batch job ที่ทำงานแบบซ้ำๆ แล้ว เราสามารถนำเอา Shell Loop มาใช้ทำงานให้มีความสะดวกยิ่งขึ้น

```
#!/bin/bash
for jpg; do
    png="{jpg%.jpg}.png"
    echo converting "$jpg" ...
    if convert "$jpg" jpg.to.png ; then
        mv jpg.to.png "$png"
    else
        echo 'jpg2png: error: failed output saved in "jpg.to.png".' >&2
        exit 1
    fi
done; echo all conversions successful
exit 0
```

2.1.9 การเขียน Shell Script

Shell ที่นิยมใช้เขียน Shell script

1. Bourne shell (/bin/sh)

Bourne shell (/bin/sh) เป็น shell ในยุคแรกๆ ที่มีใช้กันอย่างแพร่หลาย มีการกำหนดโครงสร้างภาษาคัดลอกกับภาษาอัลกอ (Algo) สามารถเขียน Shell Script ได้ และยังเป็น Standard Shell ที่มีใน UNIX ทุกตัว และยังสามารถย้าย Shell Script ไปยัง UNIX ระบบอื่นโดยไม่ต้องแก้ไขอะไรได้อีกด้วย จะมี Default Prompt เป็นเครื่องหมาย “\$”

2. C shell (/bin/csh)

C shell (/bin/csh) เป็น Shell ที่พัฒนาขึ้นมาหลังจาก Bourne Shell มีรูปแบบคำสั่งและไวยากรณ์เหมือนกับภาษา C มี Function การทำงานหลายที่ดีและอย่างสะดวก อีกทั้งยังสามารถควบคุมการไหลของข้อมูลได้ดีกว่า Bourne Shell และยังมีความสามารถในการเรียกใช้คำสั่งที่ใช้ไปแล้ว จะมี Default Prompt เป็นเครื่องหมาย “%”

3. Korn shell (/bin/ksh)

Korn shell (/bin/ksh) เป็น shell ที่พัฒนามาจากต้นแบบของ Bourne shell และ C shell สามารถทำงานใน Function ของ Bourne Shell ได้ทุกอย่าง การเขียน Shell Script ทำได้ง่ายและรัดกุมขึ้น สามารถนำคำสั่งที่ใช้ไปแล้วกลับมา Execute ไปใหม่ได้ ถือได้ว่า Korn shell เป็นการรวมเอาข้อดีของ Bourne shell และ C Shell มาไว้ด้วยกัน แต่ไม่ได้มีใน UNIX ทุกตัว จะมี Default Prompt เป็นเครื่องหมาย “\$”

4. Bourne Again Shell

Bourne Again Shell (/bin/bash หรือ /usr/local/bin/bash) เป็นการเอา Bourne Shell นำกลับมาพัฒนาใหม่ สามารถทำงานแบบ Line Editing ได้และยังได้เพิ่มประสิทธิภาพในการทำงานอีกหลายอย่าง Bash Shell นี้ไม่ใช่ Standard UNIX Shell แต่เป็น Default Shell ของ Linux ในปัจจุบันจะมี Default Prompt เป็นเครื่องหมาย “\$”

ขั้นตอนการเขียน Shell Script

1. สร้าง File Shell Script

Shell Script เป็น File Text ธรรมดาสามารถสร้างด้วย editor ชนิดไหนก็ได้ เช่น vi, nano, gedit, emacs เป็นต้น

```
$ vi hello.sh
```

2. กำหนด Interpreter หรือ Shebang

บรรทัดแรกของ Shell Script เราต้องกำหนด Shebang เพื่อให้ Program รู้ว่าควรใช้ Interpreter ใดในการ Load ข้อมูล File ซึ่งสัญลักษณ์ที่ใช้คือ `#!/<shell program>`

```
#!/bin/bash
```

3. ใส่ชุดคำสั่ง command line

ส่วนนี้คือ Code Program ที่เราต้องการให้ Shell Script ทำงาน ซึ่งสามารถศึกษาชุดคำสั่ง Shell จำลองให้ Shell Script แสดงคำว่า “Hello World” ด้วยคำสั่ง echo และลอง Save File

```
#!/bin/bash
echo "Hello World"
```

4. กำหนดสิทธิ์ Execute ให้กับ Shell Script

หลังจากเราสร้าง Shell Script ขึ้นมาได้แล้วลองสังเกตจะเห็นว่า File ที่ได้มีสิทธิ์หรือ Permission ของ User Owner (-rw-rw-r-) เพียงแค่ Write กับ Read เท่านั้น ไม่สามารถ Execute ได้ ซึ่งก็ไม่ต่างกับ Text File ธรรมดา

```
$ ls -ltr test.sh
-rw-rw-r-- 1 staff staff 32 May 13 15:28 test.sh
```

5. การเรียกใช้งาน Shell Script (Execute)

พิมพ์ Full Path ของ Shell Script หรือ ถ้าอยู่ใน Directory ก็ใช้ ./<shell script> ลงที่ Prompt ของ Shell แล้ว Enter จากนั้น Program จะทำงานเป็นการ Execute Shell Script ตั้งแต่บรรทัดแรก จนถึงบรรทัดสุดท้ายพร้อมกับแสดงผลลัพธ์ที่ได้ออกมาที่หน้าจอ Screen จนจบและออกจาก Program ให้อีก

```
$ ./test.sh
Hello World
```

6. การสร้างตัวแปรใน Shell Script (Variable)

Shell Script เหมือนกับโปรแกรมทั่วไปที่มีตัวแปรไว้เก็บค่าต่างๆ สำหรับใช้งาน ตัวแปรที่ใช้ใน Shell นั้นไม่จำเป็นต้องประกาศชนิดตัวแปรเหมือนกับภาษา C สามารถตั้งค่าแล้วนำไปใช้ได้ทันที

```
#!/bin/bash
var1=shell
var2=script
var3='shell script'
var4="shell script"
```

7. การเรียกใช้งานตัวแปรใน Shell Script (Variable)

เวลาที่จะต้องการแสดงค่าให้ใช้เครื่องหมาย "\$" นำหน้าตัวแปร

```
#!/bin/bash
var1=shell
var2=script
var3='shell script'
var5='$var1 $var2'
echo $var1
echo $var2
echo $var3
echo $var5
```

ซึ่งเมื่อทำการ Run Shell Script จะเห็นความแตกต่างระหว่างการใช้ Single Quote และ Double Quote คือสิ่งที่พิมพ์ในเครื่องหมาย Quote (') จะมีค่าตามสิ่งที่พิมพ์ ส่วนในเครื่องหมาย Double Quote (") จะเป็นการอ้างอิงนำผลลัพธ์ที่ตัวแปรเก็บไว้หรือค่าที่ตัวแปรเก็บไว้มาใช้

```
$ ./test.sh
shell
script
shell script
shell script
$var1 $var2
shell script
```

8. การใช้งานตัวแปร Environment (Environment Variable)

ตัวแปรที่ใช้ใน Shell จะมีสองชนิดด้วยกัน คือ

- ตัวแปรธรรมดา (Variable)
- ตัวแปรสภาพแวดล้อม (Environment Variable)

ตัวแปรสภาพแวดล้อมนั้นคล้ายกับตัวแปรธรรมดาแต่แตกต่างกันที่ เมื่อ Program Run ใน Shell นั้น จะสืบทอดตัวแปรสภาพแวดล้อมและค่าที่อยู่ในตัวแปรสภาพแวดล้อมไปด้วย เราสามารถสร้างตัวแปรธรรมดาให้เป็นตัวแปรสภาพแวดล้อมได้ โดยการใช้คำสั่ง Export

```
#!/bin/bash
echo $NAME
```

เนื่องจากตัวแปร NAME ยังไม่มีการกำหนดค่า และ ไม่ได้เป็น Environment Variable จึงมีค่าเป็น Null แต่ถ้าเรามีการ Export ประกาศให้ NAME เป็น Environment Variable และ Assign ค่าให้แล้ว จะทำการสืบทอดต่อไปยัง Shell Script ด้วย

```
$ ./test.sh
$ export NAME=saixiii
```

```
$ ./test.sh
saixiii
```

9. ตัวแปรแบบพิเศษ

เป็นตัวแปรที่ระบบ Assign ค่าให้เอง ไม่ต้องประกาศค่าให้ ซึ่งส่วนใหญ่เป็นค่าที่เกี่ยวข้องกับระบบ Environment ของ Shell Script เองที่เราสามารถเอาไปใช้งานได้

\$0 \$1 \$2	ใช้สำหรับอ้างอิงชื่อ Shell Script และ Argument ของ Shell Script เรียกว่า Position Parameter
\$#	ใช้บอกจำนวน Argument ที่อยู่ใน Shell Script นั้นๆ
\$*	แทน Argument ของ Script เรียงกันทั้งหมด
\$@	คล้ายกับ \$* แต่จะใช้ช่องว่างคั่นระหว่าง Position Parameter
\$?	ใช้แสดงสถานการณ์จบการทำงานครั้งสุดท้าย ถ้าสั่งได้ถูกต้องไม่มี Error จะแสดงค่า 0 ออกมา แต่ถ้าผิดพลาดจะแสดงค่าที่ไม่ใช่ 0 ออกมา
\$!	ใช้แสดง Process ID ของ Shell ที่ทำงานอยู่

ตารางที่ 2.7 ตัวแปรแบบพิเศษ

10. การ Comment

เครื่องหมาย “#” ใช้แสดงหมายเหตุ เขียนไว้ตรงไหนใน Shell Script ก็ได้ (บรรทัดแรกใช้สำหรับ Shebang) ใช้สำหรับการเขียนบันทึกใน Script โดย Shell จะถือว่าข้อความที่อยู่หลังเครื่องหมาย “#” เป็นหมายเหตุไม่มีผลต่อตัว Program

#!/bin/bash # echo “Test Comment” var=hello echo \$var
\$./test.sh hello

จะเห็นได้ว่าข้อความที่อยู่หลังเครื่องหมาย “#” จะเห็นว่าไม่มี Test Comment โชว์มาทางหน้าจอ

11. ตัวแปรแถวลำดับ (Array Variable)

ตัวแปร Array นั้นจะมี Index เป็นตัวเลขเริ่มตั้งแต่ 0 เหมือนกับตัวแปรในภาษา C หรือภาษาอื่นๆ แต่เวลาที่ต้องการเรียกใช้ตัวแปรต้องใช้เครื่องหมาย “{ }” ช่วยจับกลุ่มด้วย เพราะไม่เช่นนั้นค่าที่ออกมาจะออกมาเฉพาะค่าแรกเท่านั้น ดูได้จากตัวอย่าง

```
#!/bin/bash
txt[0]="a"
txt[1]="b"
txt[2]="c"
echo $txt[0] $txt[1] $txt[2]
echo ${txt[0]} ${txt[1]} ${txt[2]}
```

```
$ ./test.sh
a[0] a[1] a[2]
a b c
```

12. รูปแบบเงื่อนไข (Expression)

เงื่อนไข	ความหมาย
int1 -eq int2	เป็นจริงเมื่อ int1 เท่ากับ int2
int1 -ne int2	เป็นจริงเมื่อ int1 ไม่เท่ากับ int2
int1 -gt int2	เป็นจริงเมื่อ int1 มากกว่า int2
int1 -ge int2	เป็นจริงเมื่อ int1 มากกว่า หรือ เท่ากับ int2
int1 -le int2	เป็นจริงเมื่อ int1 น้อยกว่า int2
int1 -lt int2	เป็นจริงเมื่อ int1 น้อยกว่า หรือ เท่ากับ int2

ตารางที่ 2.8 เปรียบเทียบจำนวนเต็ม

เงื่อนไข	ความหมาย
<code>str1 = str2</code>	เป็นจริงเมื่อ <code>str1</code> เหมือนกับ <code>str2</code>
<code>str1 != str2</code>	เป็นจริงเมื่อ <code>str1</code> ไม่เหมือนกับ <code>str2</code>
<code>Str</code>	เป็นจริงเมื่อ <code>str</code> ไม่เป็นค่าว่าง
<code>-n str</code>	เป็นจริงเมื่อ <code>str</code> มีความยาวมากกว่า 0
<code>-z str</code>	เป็นจริงเมื่อ <code>str</code> มีความยาวเป็น 0

ตารางที่ 2.9 เปรียบเทียบ String

<code>-d filename</code>	เป็นจริงเมื่อ <code>filename</code> เป็น directory
<code>-f filename</code>	เป็นจริงเมื่อ <code>filename</code> เป็น file
<code>-r filename</code>	เป็นจริงเมื่อ <code>filename</code> อ่านได้โดยโปรแกรม
<code>-w filename</code>	เป็นจริงเมื่อ <code>filename</code> เขียนได้โดยโปรแกรม
<code>-x filename</code>	เป็นจริงเมื่อ <code>filename</code> run ได้โดยโปรแกรม
<code>-s filename</code>	เป็นจริงเมื่อ <code>filename</code> มีขนาดไม่เป็น 0

ตารางที่ 2.10 ตรวจสอบ File

เงื่อนไข	ความหมาย
<code>!expr</code>	เป็นจริงเมื่อ <code>exp</code> เป็นเท็จ
<code>exp1 -a exp2</code>	เป็นจริงเมื่อ <code>exp1</code> และ <code>exp2</code> เป็นจริง
<code>exp1 -o exp2</code>	เป็นจริงเมื่อ <code>exp1</code> หรือ <code>exp2</code> เป็นจริง

ตารางที่ 2.11 เงื่อนไขอื่นๆ

13. เงื่อนไข If Expression

```
if [ expression ];then
    commands
elif [ expression ];then
    commands
else
    commands
fi
```

จากตัวอย่างต่อไปเป็นการเปรียบเทียบ ตัวแปร String

```
#!/bin/bash
var1="cat"
var2="cat"
if [ $var1 = $var2 ];then
    echo "Yes, it is equal"
else
    echo "No, it is not equal"
fi
```

```
$ ./test.sh
Yes, it is equal
```

ในตัวอย่างแสดงการเปรียบเทียบตัวอักษรของตัวแปร text1 กับ text2 ว่าเหมือนกันหรือไม่ ถ้าเหมือนกันให้แสดงข้อความว่า “Yes, it is equal” หรือถ้าต่างกันให้แสดงข้อความว่า “No, it is not equal” ทั้งนี้เงื่อนไข if จะมี หรือ ไม่มี elif กับ else ก็ได้

14. เงื่อนไข Case Switch

คำสั่ง Case เป็นคำสั่งที่ใช้ในการเปรียบเทียบเช่นเดียวกับคำสั่ง if แต่มีรูปแบบการทำงานที่ต่างกัน

```
case string in
    str1) commands;;
    str2) commands;;
    str3) commands;;
    *) commands;;
esac
```

จากตัวอย่างใช้ Switch มาเปรียบเทียบตัวแปร String

```
#!/bin/bash
var1="cat"
case $var1 in
    "ant") echo "It is an ant";;
    "bee") echo "It is an bee";;
    "cat") echo "It is an cat";;
    *) echo "I don't know";;
esac
```

```
$ ./test.sh
It is an cat
```

ในตัวอย่างแสดงการเปรียบเทียบตัวอักษรของตัวแปร Text ว่าตรงกับ String ที่เป็นเงื่อนไขของ Case หรือไม่ ถ้าเหมือนกันก็ให้แสดงข้อความที่ตั้งไว้ในเงื่อนไข

15. การใช้ Loop While

คำสั่ง while เป็นคำสั่งที่ใช้ในการวนรอบการทำงานของ Program โดยจะวนรอบจนกว่าเงื่อนไขที่กำหนดไว้จะเป็นค่าเท็จ

```
while [ expression ]
do
    commands
done
```

การสร้าง Loop While ด้วยเงื่อนไขตัวเลข ที่มีการ Incremental

```
#!/bin/bash
num=1
while [ $num -le 10 ]
do
    echo "Loop $num"
    ((num = $num+1))
done
```

```
$ ./test.sh
Loop 1
Loop 2
```

```
Loop 3
Loop 4
Loop 5
Loop 6
Loop 7
Loop 8
Loop 9
Loop 10
```

ในตัวอย่างจะเป็นการแสดงข้อความ “Loop” 10 บรรทัด โดยให้ตัวแปร num มีค่าเป็น 1 ต่อจากนั้นก็เข้าสู่ Loop While โดยมีเงื่อนไขว่า \$num น้อยกว่าหรือเท่ากับ 10 Loop While จะทำการแสดงข้อความ “Loop” โดยวนรอบการทำงานไปเรื่อยๆ และจะทำการเพิ่มค่า num ที่ละ 1 ก่อน

จะวนรอบใหม่ เมื่อ Program วน Loop ครบตามเงื่อนไขแล้ว จะได้ผลลัพธ์เป็นการแสดงข้อความ “Loop” 10 บรรทัด

16. การใช้ Loop For

คำสั่ง for เป็นคำสั่งที่มีการทำงานคล้ายคลึงกับคำสั่ง while แต่จะแตกต่างกันตรงที่การเขียนเงื่อนไขที่ใช้ในการวน Loop

```
for var in list
do
    commands
done
```

ทำการสร้าง Loop for โดยการนับจำนวนตาม list in

```
#!/bin/bash
for i in 1 2 3 4 5
```

```
do
    echo "Loop $i"
done
$ ./test.sh
Loop 1
Loop 2
Loop 3
Loop 4
Loop 5
```

ในตัวอย่างเป็นการแสดงข้อความ “Loop” ตามด้วยตัวเลข ในเงื่อนไข for นั้น “i” จะรับค่าตัวแปรที่อยู่หลัง “in” เข้ามาเก็บไว้ในตัวเอง ซึ่งตั้งไว้เป็น “1 2 3 4 5” มาทีละตัว (จะใช้ช่องว่างในการแบ่งว่ามีกี่ตัว หรือที่ Argument ในเงื่อนไข) Loop for จะทำการแสดงข้อความตามจำนวน Argument ที่อยู่หลัง “in” ซึ่งใน Program กำหนดเป็น “1 2 3 4 5” ซึ่งมีอยู่ 5 ตัว โดยถูกแบ่งด้วยช่องว่าง เมื่อ Run Program แล้วจะเห็นได้ว่า ข้อความ “Loop” ตามด้วยจำนวนตัวเลข ถูกแสดงออกมา 5 บรรทัด โดยที่ตัวเลขนั้นมาจากตัวแปร “\$i” ที่อยู่ในบรรทัด echo “Loop” \$i นั่นเอง


```
for ((i=1;i<5;i++))  
do  
    echo "Loop $i"  
done
```

17. การส่งค่า Input Argument

จากหัวข้อตัวแปรแบบพิเศษ เราสามารถนำเอาตัวแปรพวกนี้มาใช้แทนการรับค่า Input Argument ได้ เพื่อให้ Shell Script สามารถทำงานได้ตามสถานการณ์ที่เปลี่ยนแปลงไป

```
#!/bin/bash  
var1=$1  
var2=$2  
echo "$1 & $2"  
$ ./test.sh a b  
a & b  
$ ./test.sh "a c d" b  
a c d & b
```

ผลลัพธ์จากตัวอย่างจะเห็นได้ว่า ค่า \$1 แทนค่าได้เท่ากับ Input Argument แรก และ \$2 แทนค่าได้เท่ากับ Input Argument ถัดไปซึ่งแบ่งด้วย Space และถ้าเราอยากให้ค่า Space ติดเข้าไปกับตัวแปรด้วยให้ใช้ Double Quote ในการประกาศรวมใน Argument

2.2 เทคโนโลยีที่ใช้ในการปฏิบัติงาน

2.2.1 โปรแกรม Oracle VM VirtualBox



ภาพที่ 2. 1 Logo Oracle VM VirtualBox

โปรแกรม Oracle VM VirtualBox เป็นโปรแกรมประเภท Virtual Machine คือการใช้พื้นที่ส่วนหนึ่งในฮาร์ดดิสก์ (Hard Disk) จำลองเป็นเครื่องคอมพิวเตอร์ขึ้นมาซึ่งมีการใช้ทรัพยากรต่างๆ เช่น หน่วยความจำ (RAM) การ์ดจอ การ์ดเน็ตเวิร์ค (NIC) ร่วมกันกับเครื่องคอมพิวเตอร์ ประโยชน์ของโปรแกรมคือ เป็นการช่วยให้สามารถทดสอบระบบปฏิบัติการหรือโปรแกรมต่างๆ ที่ต้องติดตั้ง โดยที่ไม่ต้องหาเครื่องคอมพิวเตอร์มาติดตั้งเพิ่ม

2.2.2 โปรแกรม FileZilla



ภาพที่ 2. 2 Logo FileZilla

โปรแกรม FileZilla เป็นโปรแกรม Open source ใช้สำหรับส่งไฟล์จากเครื่องส่วนตัวไปยังเครื่อง Server และดึงไฟล์จากเครื่อง Server มายังเครื่องส่วนตัวโดยใช้โปรโตคอล FTP port: 21

บทที่ 3

แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

3.1 แผนงานปฏิบัติงาน

หัวข้องาน	ม.ย.				ก.ค.				ส.ค.				ก.ย.			
1. ศึกษาการใช้งานโปรแกรมที่ใช้ในการพัฒนา Shell Script																
2. ทำความเข้าใจความต้องการของผู้ใช้และศึกษาข้อมูลที่มีความจำเป็น																
3. ทดลองเขียน Shell Script เบื้องต้นและทดสอบใช้งาน Shell Script																
4. เขียน Shell Script																
4.1 เขียน Shell Script สำหรับ Monitor โดยมี Function ดู Backup Log File และ Generate Monthly Report เป็น Function เสริม																
4.2 เขียน Shell Script สำหรับแจ้งเตือน Email																
5. ตรวจสอบและแก้ไข Shell Script เมื่อพบข้อผิดพลาด																
6. ทำคู่มือการติดตั้งโปรแกรมและทำคู่มือการใช้งาน Shell Script																
7. ทำรูปเล่มโครงงานและเตรียมการนำเสนอโครงงาน																
8. ช่วยสนับสนุนงานด้านต่างๆ และแก้ปัญหาภายในบริษัท																

ตารางที่ 3.1 แผนงานปฏิบัติงาน

3.2 รายละเอียดงานที่นักศึกษาปฏิบัติในงานสหกิจศึกษา หรือรายละเอียดโครงการที่ได้รับมอบหมาย

ในการปฏิบัติการสหกิจศึกษาในครั้งนี้ ได้มีโอกาสเข้าร่วมงานในแผนก System Engineer (SE) โดยได้ทำการพัฒนา Shell Script ที่ใช้สำหรับ Monitor ระบบบน Linux

โดยเริ่มแรกของโครงการนี้ ได้ทำการศึกษาการใช้งาน Linux จากนั้นจึงเริ่มศึกษา Command ต่างๆที่ใช้ในการ Monitor ระบบบน Linux และดำเนินการออกแบบ Shell Script ตามความต้องการของผู้ใช้ให้มีความเหมาะสม เมื่อทำการออกแบบแล้วก็เริ่มดำเนินการพัฒนา Shell Script เมื่อพัฒนาเสร็จแล้วจะต้องตรวจสอบความถูกต้อง และดำเนินการแก้ไขเมื่อพบปัญหาหรือข้อผิดพลาด

3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงานหรือโครงการ

3.3.1 ศึกษาการใช้งานโปรแกรมที่จำเป็นต่อการพัฒนา Shell Script

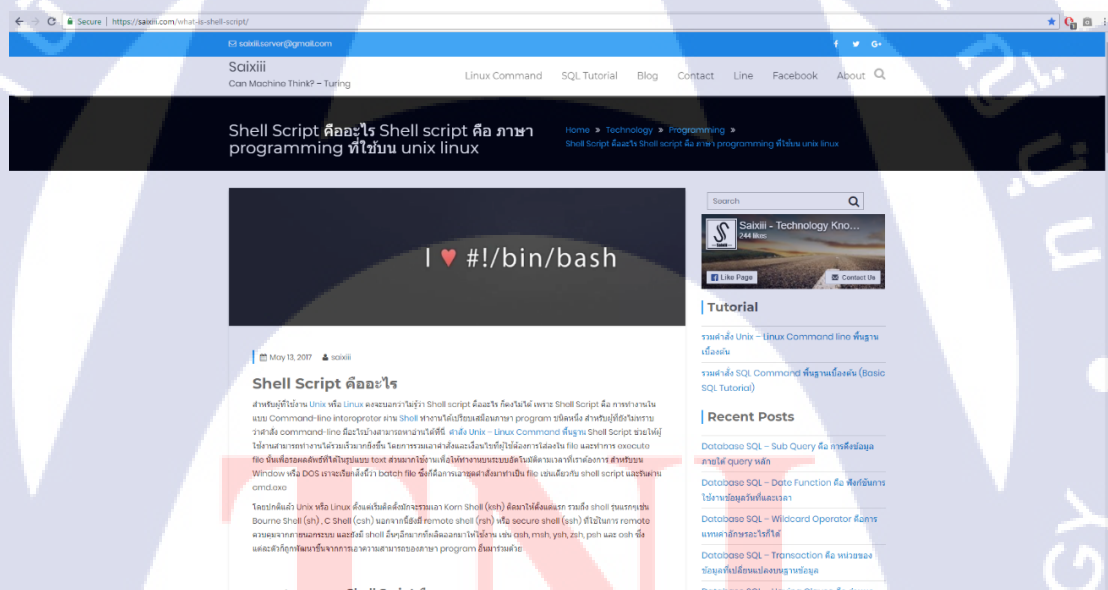
โปรแกรมที่ทำการศึกษาการใช้งานคือ โปรแกรม Oracle VM VirtualBox และ โปรแกรม FileZilla ซึ่งโปรแกรม Oracle VM VirtualBox เป็นโปรแกรมที่ช่วยในการจำลองเครื่องคอมพิวเตอร์เพื่อให้สามารถติดตั้ง OS Linux สำหรับการพัฒนา Shell Script และ โปรแกรม FileZilla ซึ่งเป็นโปรแกรมที่ช่วยให้มีความสะดวกในการส่งไฟล์ข้อมูลผ่านเครื่องจริงไปยังโปรแกรมจำลองอย่าง Oracle VM VirtualBox

3.3.2 ทำความเข้าใจความต้องการของผู้ใช้และศึกษาข้อมูลที่มีความจำเป็น

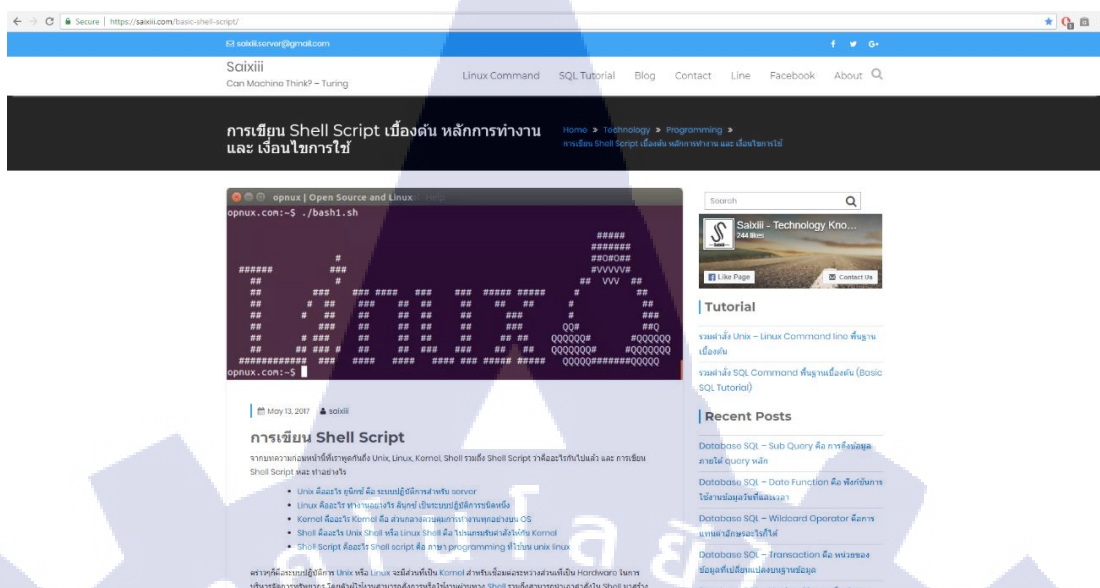
หาข้อมูลต่างๆ ที่เกี่ยวข้องกับความต้องการของผู้ใช้จากแหล่งข้อมูลต่างๆ เช่น Website หรือ หนังสือต่างๆ และนำข้อมูลที่ได้จากการค้นคว้ามาศึกษาหาข้อมูลที่จะเป็นสำหรับการเขียน Shell Script



ภาพที่ 3.1 ตัวอย่าง Website ที่ทำการหาข้อมูลเกี่ยวกับ Shell



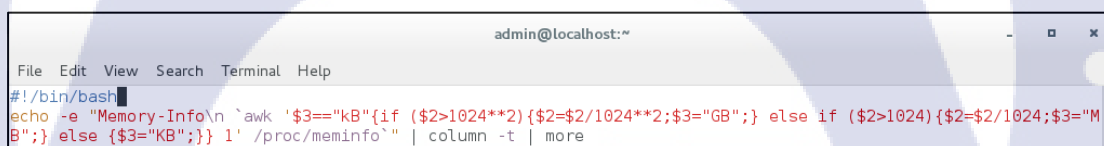
ภาพที่ 3.2 ตัวอย่าง Website ที่ทำการหาข้อมูลเกี่ยวกับ Shell Script



ภาพที่ 3.3 ตัวอย่าง Website ที่ทำการหาข้อมูลเกี่ยวกับการ Shell Script เบื้องต้น

3.3.3 ทดลองเขียน Shell Script เบื้องต้นและ ทดสอบใช้งาน Shell Script

เขียน Shell Script ตัวอย่างและทำการทดสอบใช้งาน Shell Script ให้ตรงตามผลลัพธ์ที่ต้องการ เช่น เขียน Shell Script ให้สามารถคำนวณวันที่หรือเขียนให้แสดงรายละเอียดต่างๆของระบบปฏิบัติการ OS โดยจะทำการทดสอบ Shell Script เป็นส่วนๆ



ภาพที่ 3.4 ตัวอย่าง Shell Script ที่ใช้ในการแสดงรายละเอียดข้อมูล

3.3.4 เขียน Shell Script สำหรับ Monitor

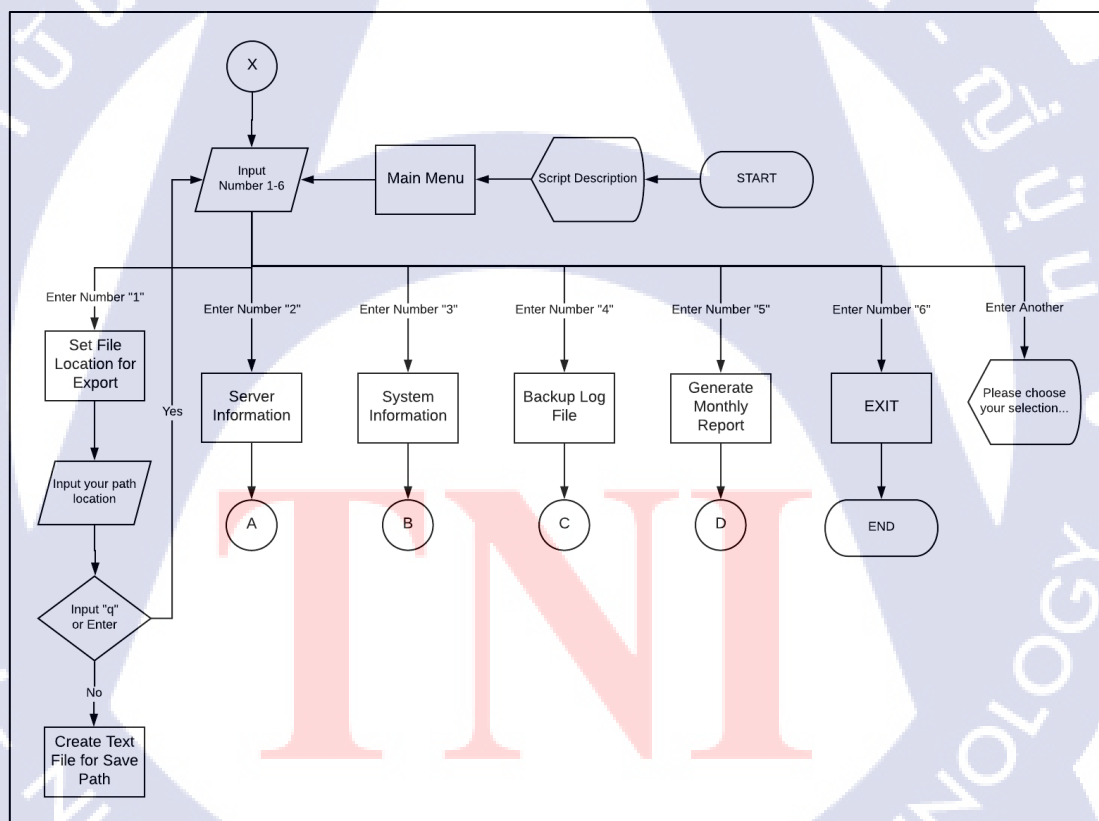
การเขียน Shell Script สำหรับ Monitor นั้นจะนำ Shell Script ตัวอย่างที่ทำการทดสอบสำเร็จมาปรับปรุงแก้ไขและนำมารวบรวมเป็น Shell Script เดียวกัน โดยจะคัดกรอง Shell Script ที่มีประสิทธิภาพและตรงตามความต้องการของผู้ใช้

Shell Script สำหรับการ Monitor นั้นสามารถช่วยให้ผู้ใช้งาน Shell Script มีความสะดวกและรวดเร็วในการ Monitor โดยที่ผู้ใช้งานไม่จำเป็นต้องเรียกใช้คำสั่งในการ Monitor หลายคำสั่ง

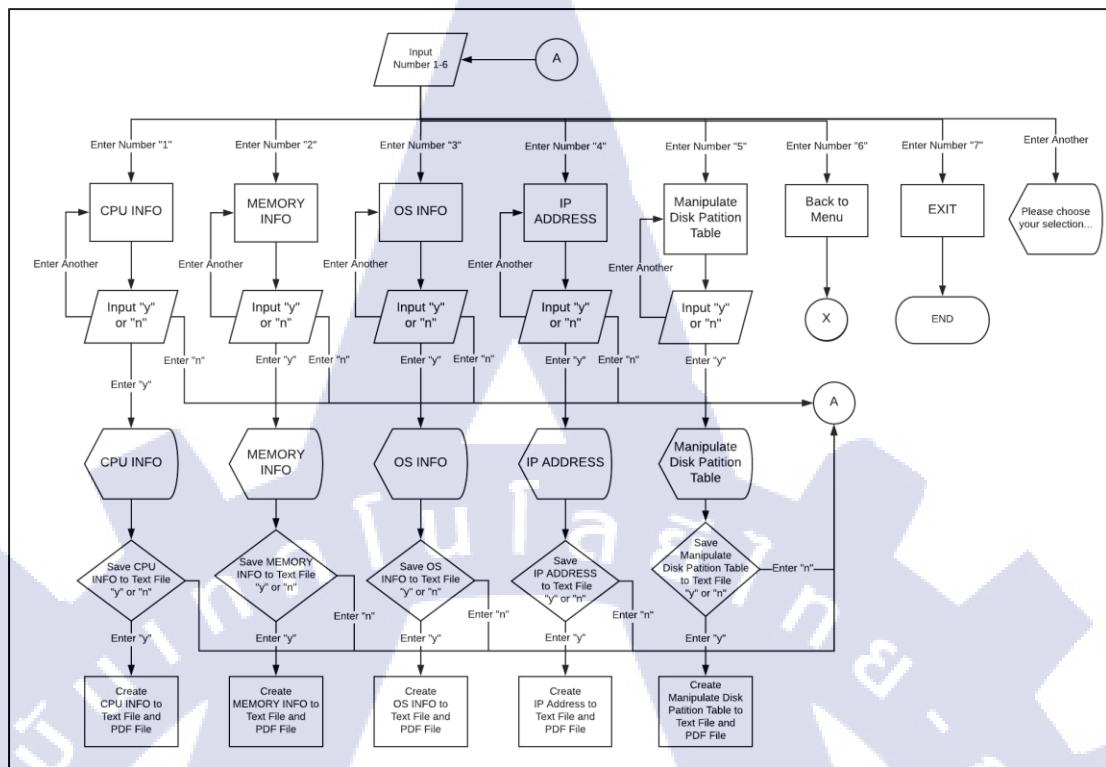
ทั้งนี้ภายในการทำงานของ Shell Script สำหรับการ Monitor ผู้ใช้งานสามารถที่จะ Monitor ได้ใน 2 ส่วนหลักคือ System และ Hardware

Shell Script สำหรับการ Monitor ยังมีส่วนเสริมที่ช่วยให้ผู้ใช้ Shell Script สามารถที่จะสรุปผลการ Monitor และทำการบันทึกผลการ Monitor ออกมาเป็นไฟล์ข้อมูลและตัว Shell Script สำหรับการ Monitor ยังสามารถเข้าสู่ไฟล์ข้อมูลที่ทำกร Backup เอาไว้บนเครื่อง Server โดย Shell Script สำหรับการ Monitor จะสามารถตรวจสอบระยะเวลาที่ทำการ Backup ข้อมูลและสามารถบอกได้ว่าระหว่างทำการ Backup ข้อมูล ข้อมูลที่ได้จากการทำ Backup นั้นเกิดข้อผิดพลาดหรือไม่ เกิดข้อผิดพลาด

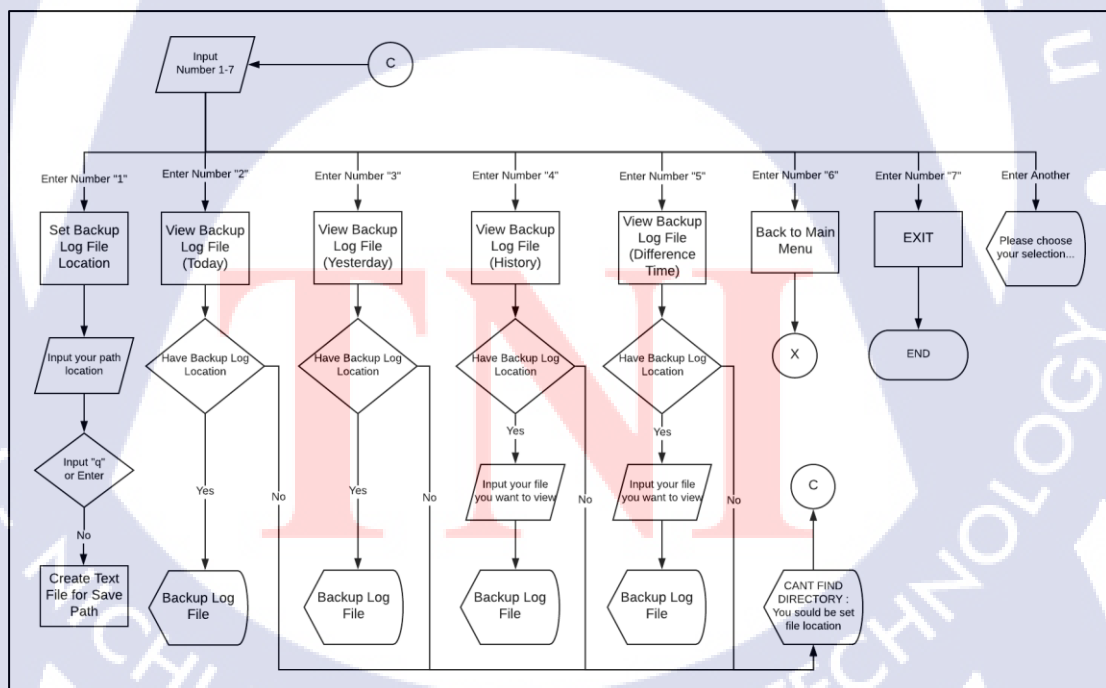
ในการใช้ Shell Script สำหรับ Monitor System และ Monitor Hardware แต่ละครั้งผู้ใช้งาน Shell Script สำหรับ Monitor สามารถที่จะเก็บข้อมูลที่ได้จากการ Monitor ออกมาในรูปแบบของไฟล์ข้อมูล โดยไฟล์มูลที่ได้จะออกมาในรูปแบบไฟล์สกุล txt และสกุล pdf



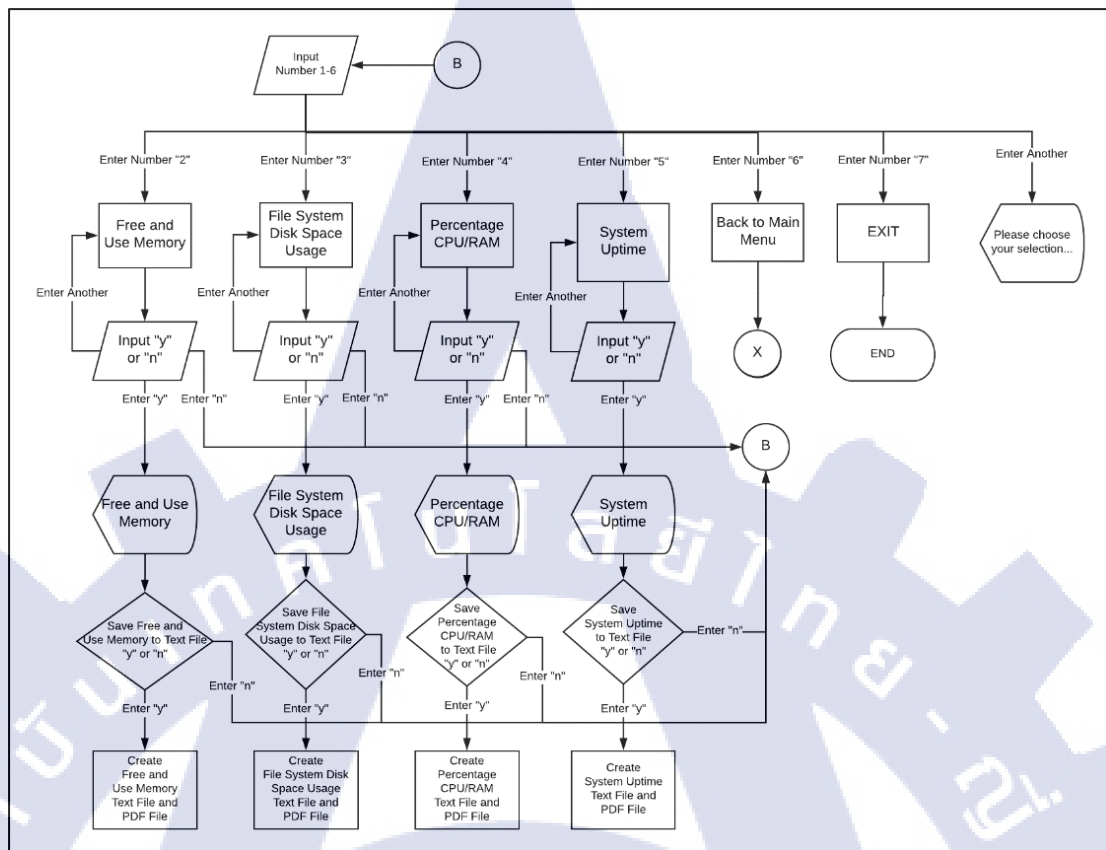
ภาพที่ 3.5 ตัวอย่างกระบวนการทำงานของ Script



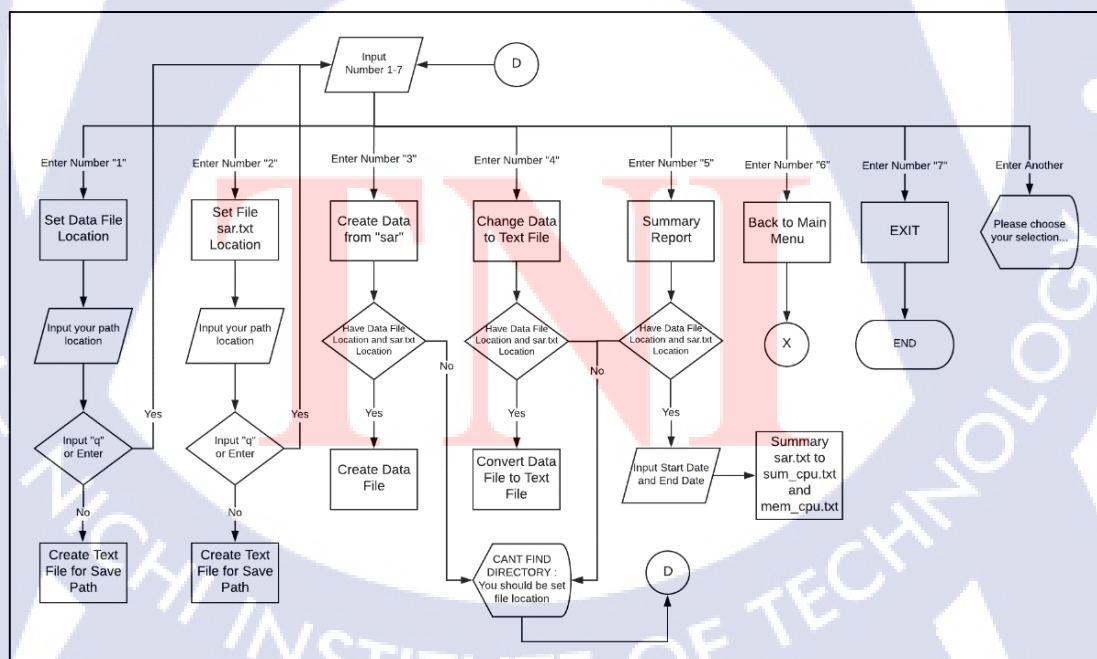
ภาพที่ 3. 6 ตัวอย่างกระบวนการทำงานของ Script ในส่วน Server Information



ภาพที่ 3. 7 ตัวอย่างกระบวนการทำงานของ Script ในส่วน Backup Log File



ภาพที่ 3.8 ตัวอย่างกระบวนการทำงานของ Script ในส่วน System Information



ภาพที่ 3.9 ตัวอย่างกระบวนการทำงานของ Script ในส่วน Generate Monthly Report

3.3.5 เขียน Shell Script สำหรับแจ้งเตือน Email

การเขียน Shell Script สำหรับแจ้งเตือน Email นั้นจะนำ Shell Script ตัวอย่างที่ทำการทดสอบสำเร็จมาปรับปรุงแก้ไข โดยจะทำให้ Shell Script มีความสามารถตามความต้องการของผู้ใช้

Shell Script สำหรับการแจ้งเตือน Email นั้นสามารถช่วยให้ผู้ใช้งาน Shell Script สำหรับการแจ้งเตือน Email สามารถป้องกันความเสี่ยงที่จะเกิดปัญหาล้นบนเครื่อง Server โดยที่ผู้ใช้งาน Shell Script สำหรับการแจ้งเตือน Email จะได้รับ Email แจ้งเตือนเมื่อ Server มีเหตุการณ์ต่างๆ เช่น Hard- Disk ของเครื่อง Server หรือ Memory เครื่อง Server ใกล้จะเต็ม ทั้งนี้ Shell Script สำหรับการแจ้งเตือน Email สามารถแก้ไขผู้รับส่ง Email ได้ โดยผู้ใช้งานจะต้องใช้คำสั่งในการแก้ไขเพื่อที่จะเข้าไปแก้ไข Shell Script สำหรับการแจ้งเตือน Email รวมทั้งผู้ใช้งานยังสามารถกำหนดค่าสูงสุดที่ใช้สำหรับการแจ้งเตือน Email ได้ เช่น เมื่อการทำงานของ CPU มีค่าเกินกว่า 80 เปอร์เซ็นต์เครื่อง Server จะทำการแจ้งเตือน Email มายังผู้ใช้ (Email ที่กำหนดเอาไว้สำหรับผู้รับ Email) โดยผู้ใช้งานสามารถกำหนดค่าจาก 80 เปอร์เซ็นต์เป็นจำนวน 90 เปอร์เซ็นต์หรือเป็นจำนวนที่ผู้ใช้งานต้องการได้

Shell Script สำหรับการแจ้งเตือน Email มีคุณสมบัติในการตรวจสอบข้อมูล Memory ตรวจสอบการใช้งาน Hard Disk และการทำงานของ CPU บนเครื่อง Server เพื่อให้ Shell Script สำหรับการแจ้งเตือน Email สามารถส่ง Email เพื่อแจ้งเตือนผู้ใช้งาน Shell Script สำหรับการแจ้งเตือน Email เมื่อเกิดเหตุการณ์ตามคุณสมบัติของ Shell Script สำหรับการแจ้งเตือน Email

ในการใช้ Shell Script สำหรับการแจ้งเตือน Email ผู้ใช้สามารถกำหนดค่าในคำสั่ง crontab -e เพื่อที่จะให้ Shell Script สำหรับการแจ้งเตือน Email ตรวจสอบข้อมูล Memory ตรวจสอบการใช้งาน Hard Disk และการทำงานของ CPU บนเครื่อง Server ตามเวลาที่เรากำหนด

3.3.6 แก้ไข Shell Script เมื่อพบข้อผิดพลาด

นำ Shell Script ที่พัฒนามาตรวจสอบหาข้อผิดพลาดและทำการแก้ไขข้อผิดพลาด โดยจะ
มอง
ความต้องการของผู้ใช้งานและข้อผิดพลาดจากการใช้งาน
Shell Script

3.3.7 ทำคู่มือการติดตั้งและใช้งาน Script

ทำเอกสารการติดตั้งและการใช้งาน Shell Script สำหรับ Monitor และ Shell Script สำหรับการแจ้งเตือน Email เพื่อให้ผู้ใช้งานเข้าใจการใช้งานเบื้องต้นและสามารถใช้ Shell Script ได้อย่างมีประสิทธิภาพ

3.3.8 ทำรูปเล่มโครงการและเตรียมการนำเสนอโครงการ

นำข้อมูลที่ได้จากการพัฒนา Shell Script มาทำเป็นรูปเล่มโครงการและทำสไลด์สำหรับนำเสนอโครงการ

3.3.9 ช่วยสนับสนุนงานด้านต่างๆ และแก้ปัญหาภายในบริษัท

ช่วยสนับสนุนการแก้ไขปัญหาเบื้องต้น เช่น แก้ไขเครื่องปริ้นเตอร์มีปัญหา แก้ไขอีเมลไม่สามารถส่งเข้าออกได้ ติดตั้งโทรศัพท์ ติดตั้ง OS และ โปรแกรมพื้นฐาน เช่น Adobe, Microsoft Office, Antivirus เป็นต้น



บทที่ 4

ผลการดำเนินงาน การวิเคราะห์และสรุปผลต่างๆ

4.1 ขั้นตอนและผลการดำเนินงาน

ในช่วงระยะเวลา 4 เดือนของการปฏิบัติงานสหกิจศึกษา ได้มีการศึกษาหาข้อมูลเพื่อนำมาใช้ทำโปรเจกต์โดยได้รับการแนะนำจากพนักงานที่ปรึกษา และเนื่องจากได้รับมอบหมายงานให้ศึกษาเกี่ยวกับการเขียน Shell Script ทำให้มีความรู้เกี่ยวกับ Shell และ Command ที่ใช้ในการทำงานบนระบบปฏิบัติการ Linux ซึ่งขั้นตอนและผลการดำเนินงานต่างๆ มีรายละเอียดดังนี้

4.1.1 ศึกษาและทำความเข้าใจความต้องการของผู้ใช้งาน

ในการเขียน Shell Script จำเป็นต้องมีความเข้าใจเกี่ยวกับการทำงานของระบบปฏิบัติการ Linux และจำเป็นต้องมีความเข้าใจ Command ที่ใช้ในการเขียน Shell Script รวมทั้งทำความเข้าใจความต้องการของผู้ใช้งานเพื่อให้ตรงตามความต้องการ

4.1.2 เขียน Shell Script และติดตั้งใช้งาน Shell Script ที่พัฒนา

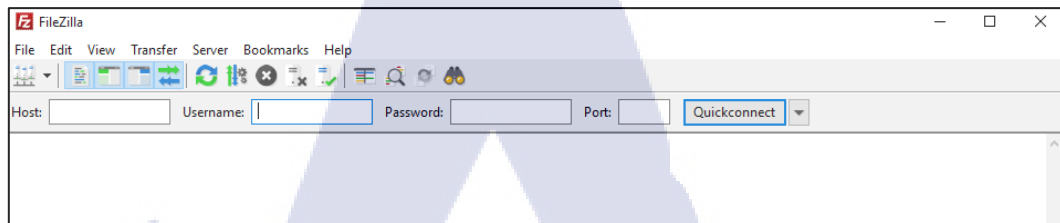
เนื่องจากผู้ใช้งาน Script มีความต้องการให้ Script มีความสามารถในการ Monitor และแจ้งเตือน Email ทำให้ต้องมีการเขียน Script ขึ้นมา 2 ตัวคือ Shell Script สำหรับการ Monitor และ Shell Script สำหรับการแจ้งเตือน Email โดยภายใน Script สำหรับการ Monitor นั้นจะมี 2 Function เสริมอย่างการดู Log File และ Generate Monthly Report ทำให้ Script นั้นมีความสามารถมากขึ้น นอกเหนือจากการ Monitor เพียงอย่างเดียว และ Script สำหรับการแจ้งเตือน Email จะเป็นการแจ้งเตือนเมื่อเกิดเหตุการณ์ภายใน Server

โดยในการเขียน Shell Script สำหรับการ Monitor และ Shell Script สำหรับการแจ้งเตือน Email นั้นได้มีการเลือกใช้ Shell Script สกุล Bash (Bash Shell Script) เนื่องจาก Bash Shell Script มีประสิทธิภาพมากกว่า Shell สกุลอื่นๆ และยังเป็น Default Shell ของ Linux ในปัจจุบัน

การติดตั้ง Shell Script สำหรับการ Monitor

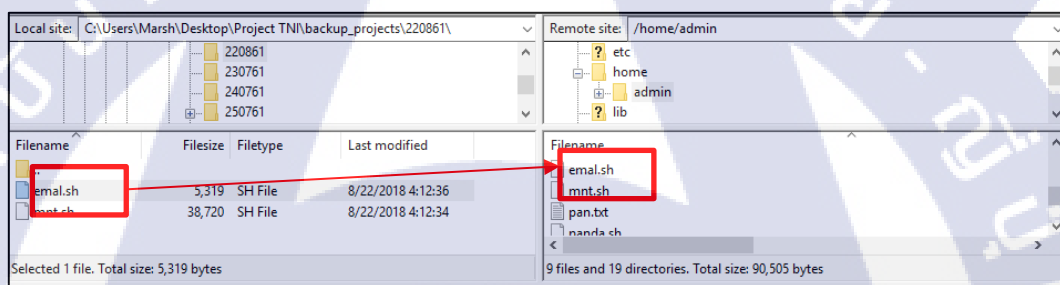
1. ให้ผู้ใช้เปิดโปรแกรม FileZilla สำหรับการนำ Shell Script สำหรับการ Monitor ไปไว้บนเครื่อง Server

- หลังจากที่ผู้ใช้เปิดโปรแกรม FileZilla ให้ผู้ใช้เชื่อมต่อเครื่อง Server โดยการกำหนดโฮสต์ (Host) โดยใส่ IP Address และ Username/Password และทำการกำหนด Port เป็น 22 (Port 22 คือ Port ที่ใช้ในการรับส่งข้อมูลระหว่างเครื่องผู้ใช้งานกับเครื่อง Server)



ภาพที่ 4.1 ตัวอย่างโปรแกรม FileZilla

- นำ Shell Script ชื่อ mnt.sh ไปวางบนเครื่อง Server โดยการลาก Shell Script สำหรับการ Monitor ไปบนเครื่อง Server



ภาพที่ 4.2 ตัวอย่างการโอนย้ายข้อมูล (ต่อภาพที่ 4.1)

- หลังจากที่ผู้ใช้นำ Shell Script สำหรับการ Monitor ไปวางบนเครื่อง Server ให้ผู้ใช้ทำการเปลี่ยนแปลงการเข้าถึงของ Script (Permission) โดยใช้ Command `chmod 755 mnt.sh` หรือ `chmod +x mnt.sh` ในส่วนที่ Shell Script สำหรับการ Monitor ถูกวางเอาไว้
- จากนั้นผู้ใช้งานจะสามารถเข้าใช้งาน Shell Script สำหรับการ Monitor ได้โดยใช้ Command `./mnt.sh`

การใช้งาน Shell Script สำหรับการ Monitor

หลังจากผู้ใช้งานเข้าใช้งาน Shell Script สำหรับการ Monitor ในหน้าแรกของ Shell Script สำหรับการ Monitor จะปรากฏรายละเอียดของ Shell Script สำหรับการ Monitor ขึ้นมา โดยจะโชว์รายละเอียดตามหัวข้อ คือ ชื่อของ Shell Script (mnt.sh), รายละเอียดของ Shell Script, ผู้เขียนและพัฒนา Shell Script, วันที่ Shell Script พัฒนาเสร็จ, เวอร์ชันของ Shell Script และสกุล Shell ที่ใช้

โดยในหน้ารายละเอียดของ Shell Script สำหรับการ Monitor ผู้ใช้สามารถ Enter เพื่อเข้าไปยังการทำงานของ Shell Script สำหรับการ Monitor

```

admin@localhost:/home/admin
File Edit View Search Terminal Help
title       : mnt.sh
description : This script will help you for monitor and generate monthly report,
              This script should be root to use
author      : Kunathip Suntornpatcharakun
date        : Jul-26-2018
version     : 1
usage       : bash mnt.sh
  
```

ภาพที่ 4.3 ตัวอย่างหน้ารายละเอียด Shell Script สำหรับการ Monitor

หลังจากที่ผู้ใช้เข้ามาหน้าการทำงานของ Shell Script สำหรับการ Monitor จะปรากฏเมนูหลัก (Main Menu) ขึ้นมาซึ่งในหัวข้อที่อยู่ในเมนูหลักมีอยู่ด้วยกัน 6 หัวข้อตามตารางที่ 4.1

หัวข้อ	รายละเอียด
1. Set File Location for Export	การกำหนดสถานที่เก็บไฟล์ข้อมูลที่ได้จากการ Monitor
2. Server Information	การ Monitor Hardware
3. System Information	การ Monitor System
4. Backup Log File	การเข้าไปดูไฟล์ข้อมูลที่ Backup ไว้บนเครื่อง Server และตรวจสอบระยะเวลาในการ Backup ข้อมูล
5. Generate Monthly Report	การเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป
6. Exit	ออกจาก Shell Script สำหรับการ Monitor

ตารางที่ 4.1 เมนูหลักของ Script สำหรับการ Monitor

ผู้ใช้งาน Shell Script สำหรับการ Monitor สามารถเลือกหัวข้อในเมนูหลักได้โดยการใส่ค่าตัวเลข 1-6 ตามหัวข้อ เช่น หากผู้ใช้ต้องการกำหนดสถานที่เก็บไฟล์ข้อมูล (Set File Location for Export) ให้ผู้ใช้พิมพ์ 1 และกด Enter เพื่อเป็นการเข้าใช้งานฟังก์ชันสำหรับการกำหนดสถานที่เก็บไฟล์ข้อมูล เป็นต้น

```

admin@localhost:/home/admin
File Edit View Search Terminal Help
Main Menu
1)Set File Location for Export
2)Server Information
3)System Information
4)Backup Log File
5)Generate Monthly Report
6)Exit
Enter your selection:
  
```

ภาพที่ 4.4 ตัวอย่างหน้าเมนูหลักของ Shell Script สำหรับการ Monitor (Main Menu)

การใช้งานฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล (Set File Location for Export)

ฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูลเป็นการกำหนดสถานที่เก็บไฟล์ข้อมูลที่เกิดจาก Monitor System (System Information) และ Monitor Hardware (Server Information)

การใช้งานฟังก์ชันในการ Monitor Hardware (Server Information)

โดยภายในฟังก์ชันในการ Monitor Hardware จะมีหัวข้อแยกออกเป็น 7 หัวข้อตามตารางที่ 4.2

หัวข้อ	รายละเอียด
1. CPU Info	บอกรายละเอียดข้อมูลของ CPU
2. Memory Info	บอกรายละเอียดข้อมูลของ Memory ในส่วนของ Hardware
3. OS Info	บอกรายละเอียดข้อมูลของระบบปฏิบัติการ
4. IP Address	บอกไอพีแอดเดรสทุกตัวที่มีการเชื่อมต่อภายในเครื่อง Server
5. Manipulate Disk Partition Table	บอกรายละเอียดของ Disk เช่น Disk มีกี่ตัว, ขนาดของ Disk เป็นต้น
6. Back to Main Menu	กลับไปหน้าเมนูหลัก (Main Menu)
7. Exit	ออกจาก Shell Script สำหรับการ Monitor

ตารางที่ 4.2 เมนูสำหรับ Monitor Hardware ของ Shell Script สำหรับการ Monitor

```

Server Information
1)CPU Info
2)Memory Info
3)OS Info
4)IP Address
5)Manipulate Disk Partition Table
6)Back to Main Menu
7)Exit
Enter your selection :
  
```

ภาพที่ 4.5 ตัวอย่างเมนูสำหรับการ Monitor Hardware (Server Information)

```

CPU Info
Architecture:      x86_64
CPU op-mode(s):    32-bit, 64-bit
Byte Order:        Little Endian
CPU(s):            4
On-line CPU(s) list: 0-3
Thread(s) per core: 1
Core(s) per socket: 4
Socket(s):         1
NUMA node(s):      1
Vendor ID:         GenuineIntel
CPU family:        6
Model:            158
Model name:        Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz
Stepping:          9
CPU MHz:           2808.002
BogoMIPS:          5616.00
Hypervisor vendor: KVM
Virtualization type: full
L1d cache:         32K
L1i cache:         32K
L2 cache:          256K
L3 cache:          6144K
--More--

```

ภาพที่ 4. 6 ตัวอย่างการ Monitor Hardware ในหัวข้อ CPU Info

```

Memory-Info
MemTotal:          5.57518 GB
MemFree:           4.19065 GB
MemAvailable:      4.85213 GB
Buffers:           968 KB
Cached:            650.371 MB
SwapCached:        0 KB
Active:            851.371 MB
Inactive:          336.398 MB
Active(anon):      537.09 MB
Inactive(anon):    9.5625 MB
Active(file):      314.281 MB
Inactive(file):    326.836 MB
Unevictable:       0 KB
Mlocked:           0 KB
SwapTotal:         1.19921 GB
SwapFree:          1.19921 GB
Dirty:             0 KB
Writeback:         0 KB
AnonPages:         536.469 MB
Mapped:            160.039 MB
Shmem:             10.1992 MB
Slab:              120.348 MB
--More--

```

ภาพที่ 4. 7 ตัวอย่างการ Monitor Hardware ในหัวข้อ Memory Info

```

Hostname: localhost.localdomain
OS Version: 7.3
Architecture: 64-bit
Kernel Name: 4.1.12-61.1.18.el7uek.x86_64
Release: Red Hat Enterprise Linux Server release 7.3 (Maipo)
Do you want to export file [y/n]? █

```

ภาพที่ 4. 8 ตัวอย่างการ Monitor Hardware ในหัวข้อ OS Info

```

All IP Address
1: lo: 127.0.0.1/8
2: enp0s3: 10.0.2.15/24
3: enp0s8: 192.168.56.101/24
4: virbr0: 192.168.122.1/24
Gateway: 10.0.2.2
Do you want to export file [y/n]?: █

```

ภาพที่ 4.9 ตัวอย่างการ Monitor Hardware ในหัวข้อ IP Address

```

Disk /dev/sda: 12.9 GB, 12884901888 bytes, 25165824 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 label type: dos
Disk identifier: 0x000a5fa3

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *        2048       2099199       1048576    83   Linux
/dev/sda2                2099200       25165823       11533312    8e  Linux LVM

Disk /dev/mapper/ol-root: 10.5 GB, 10515120128 bytes, 20537344 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 /dev/mapper/ol-swap: 1287 MB, 1287651328 bytes, 2514944 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
Do you want to export file [y/n]?:

```

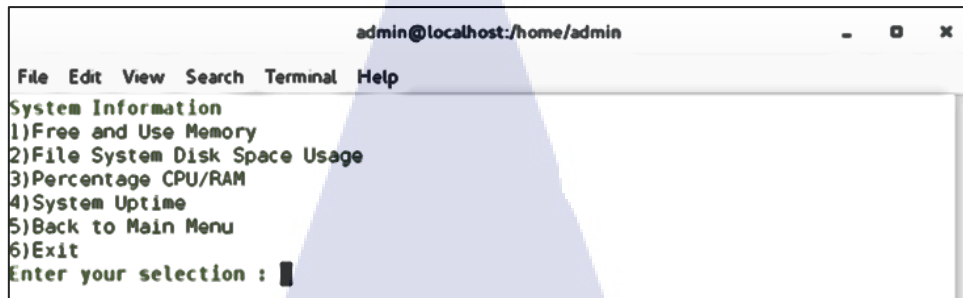
ภาพที่ 4.10 ตัวอย่างการ Monitor Hardware ในหัวข้อ Manipulate Disk Partition Table

การใช้งานฟังก์ชันในการ Monitor System (System Information)

โดยภายในฟังก์ชันในการ Monitor System จะมีหัวข้อแยกออกเป็น 7 หัวข้อตามตารางที่ 4.3

หัวข้อ	รายละเอียด
1. Free and Use Memory	บอกรายละเอียดของ Memory ในส่วนของ System
2. File System Disk Space Usage	บอกรายละเอียดของ File System
3. Percentage CPU/RAM	บอกเปอร์เซ็นต์การทำงานของ CPU/RAM
4. System Uptime	บอกเวลาที่เริ่มใช้งานระบบ
6. Back to Main Menu	กลับไปหน้าเมนูหลัก (Main Menu)
7. Exit	ออกจาก Shell Script สำหรับการ Monitor

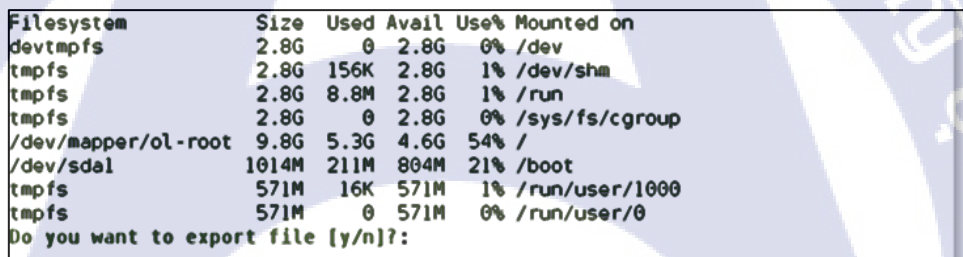
ตารางที่ 4.3 เมนูสำหรับ Monitor System ของ Shell Script สำหรับการ Monitor



ภาพที่ 4. 11 ตัวอย่างเมนูสำหรับ Monitor System (System Information)



ภาพที่ 4. 12 ตัวอย่างการ Monitor System ในหัวข้อ Free and Use Memory



ภาพที่ 4. 13 ตัวอย่างการ Monitor System ในหัวข้อ File System Disk Space Usage



ภาพที่ 4. 14 ตัวอย่างการ Monitor System ในหัวข้อ Percentage CPU/RAM



ภาพที่ 4. 15 ตัวอย่างการ Monitor System ในหัวข้อ System Uptime

การใช้งานฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup (Backup Log File)

โดยภายในฟังก์ชันดูไฟล์ข้อมูลที่ Backup ไว้บนเครื่อง Server จะมีหัวข้อแยกออกเป็น 7 หัวข้อตามตารางที่ 4.4

หัวข้อ	รายละเอียด
1. Set Backup Log File Location	กำหนดสถานที่ที่ผู้ใช้ทำการเก็บไฟล์ที่ Backup ไว้
2. View Backup Log File (Today)	เข้าไปดูไฟล์ Backup ของวันที่ปัจจุบัน
3. View Backup Log File (Yesterday)	เข้าไปดูไฟล์ Backup ของวันเมื่อวาน
4. View Backup Log File (History)	กำหนดชื่อไฟล์ Backup ที่ต้องการเพื่อเข้าไปดูไฟล์ Backup โดยจะมีรายชื่อไฟล์ทั้งหมดที่กำหนดสถานที่ที่ทำการเก็บไฟล์ Backup ในหัวข้อ Set Backup Log File Location
5. View Backup Log File (Difference Time)	กำหนดชื่อไฟล์ Backup ที่ต้องการเพื่อแสดงระยะเวลาในการ Backup โดยจะบอกเวลาเริ่ม เวลาที่หยุด ความต่างของเวลาและความผิดพลาดที่เกิดระหว่างการ Backup
6. Back to Main Menu	กลับไปหน้าเมนูหลัก (Main Menu)
7. Exit	ออกจาก Shell Script สำหรับการ Monitor

ตารางที่ 4. 4 เมนูสำหรับเข้าดูไฟล์ Backup ของ Shell Script สำหรับการ Monitor

```
Backup Log File
1)Set Backup Log File Location
2)View Backup Log File (Today)
3)View Backup Log File (Yesterday)
4)View Backup Log File (History)
5)View Backup Log File (Difference Time)
6)Back to Main Menu
7)Exit
Enter your selection:
```

ภาพที่ 4. 16 ตัวอย่างเมนูสำหรับการเข้าดูไฟล์ Backup (Backup Log File Menu)

การใช้งานฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป (Generate Monthly Report)

โดยภายในฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป จะมีหัวข้อแยกออกเป็น 7 หัวข้อตามตารางที่ 4.5

หัวข้อ	รายละเอียด
1. Set Data File Location	กำหนดสถานที่ที่จะนำไฟล์ข้อมูลไปเก็บ ที่ได้จากการสร้างไฟล์ข้อมูลที่เก็บค่าสถานะของระบบ
2. Set File sar.txt Location	กำหนดสถานที่ที่จะทำการเก็บไฟล์สกุล txt เอาไว้เพื่อนำไปใช้ในหัวข้อที่ 5 (Summary Report)
3. Get Data from “sar”	สร้างไฟล์ข้อมูลที่เก็บค่าสถานะของระบบ
4. Change Data to Text File	กำหนดวันเวลาของไฟล์ข้อมูลโดยจะมีวันที่เริ่มและวันล่าสุดที่ต้องการและนำไปสร้างไฟล์ข้อมูลที่ได้จากหัวข้อที่ 3 (Get Data from “sar”) แปลงเป็นไฟล์สกุล txt เพื่อนำไปใช้ในหัวข้อที่ 5 (Summary Report)
5. Summary Report	กำหนดเดือนเพื่อนำไฟล์ข้อมูลในหัวข้อที่ 4 (Change Data to Text File) มารวมกันเป็นไฟล์สรุปผล 2 ไฟล์
6. Back to Main Menu	กลับไปหน้าจอหลัก หรือ Main Menu
7. Exit	ออกจาก Shell Script สำหรับ Monitor

ตารางที่ 4.5 เมนูสำหรับเข้าไปสร้างไฟล์ข้อมูลเก็บค่าสถานะระบบแบบสรุปของ Shell Script
สำหรับการ Monitor

```
Generate Monthly Report
1)Set Data File Location
2)Set File sar.txt Location
3)Create Data from "sar"
4)Change Data to Text File
5)Summary Report
6)Back to Main Menu
7)Exit
Enter your selection:
```

ภาพที่ 4. 17 ตัวอย่างเมนูสำหรับเข้าไปสร้างไฟล์ข้อมูลเก็บค่าสถานะระบบแบบสรุป

เงื่อนไขและกรณีต่างๆใน Shell Script สำหรับการ Monitor ในหัวข้อต่างๆ.

ฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล (Set File Location for Export)

- การแสดงผลทางหน้าจอหากเข้าใช้ฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล

กรณีที่ 1 : หากฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูลเคยกำหนดสถานที่เก็บไฟล์ข้อมูล Script จะมีข้อความบอกสถานที่ที่เคยกำหนด

```
Enter your selection: 1
Your file location is /home/admin/marsh/
Enter your Location (Ex./home/user/Files_location)
If you want to go back enter [q]:
```

ภาพที่ 4. 18 ตัวอย่างข้อความบอกสถานที่ที่เคยกำหนด

กรณีที่ 2 : หากฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูลไม่เคยกำหนดสถานที่เก็บไฟล์ข้อมูล Script จะมีข้อความบอกว่าไม่เคยมีการกำหนดสถานที่

```
Enter your selection: 1
You dont have file location
Enter your Location (Ex./home/user/Files_location)
If you enter space your path will be [/])
If you want to go back enter [q]:
```

ภาพที่ 4. 19 ตัวอย่างข้อความบอกสถานที่ที่ไม่เคยมีการกำหนดสถานที่

- การออกจากการใช้ฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล

เงื่อนไขที่ 1 : ผู้ใช้สามารถ Enter เพื่อทำการออกจากการใช้งานฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล

เงื่อนไขที่ 2 : ผู้ใช้สามารถพิมพ์ “q” จากนั้น Enter เพื่อทำการออกจากการใช้งานฟังก์ชันกำหนดที่เก็บไฟล์ข้อมูล

```
Enter your Location (Ex./home/user/Files_location)
If you enter space your path will be [/])
If you want to go back enter [q]: q
Back to Main Menu
Enter return to continue...
```

ภาพที่ 4. 20 ตัวอย่างการออกจากการใช้ฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล

ฟังก์ชันในการ Monitor Hardware (Server Information)

- การแสดงผลหน้าจอหากเกิดข้อผิดพลาดภายในฟังก์ชันในการ Monitor Hardware

เงื่อนไข 1 : หากผู้ใช้ไม่ได้พิมพ์ตัวเลข 1-7 หรือเว้นช่องว่างแล้วจากนั้น Enter เพื่อเข้าใช้งาน หัวข้อของฟังก์ชันในการ Monitor Hardware ตัว Script จะมีข้อความบอกแจ้งเตือนผู้ใช้

```
Server Information
1)CPU Info
2)Memory Info
3)OS Info
4)IP Address
5)Manipulate Disk Partition Table
6)Back to Main Menu
7)Exit
Enter your selection :
Please choose your selection...
Enter return to continue...
```

ภาพที่ 4. 21 ตัวอย่างผู้ใช้ไม่ได้พิมพ์ตัวเลขหรือเว้นช่องว่าง

เงื่อนไข 2 : หากผู้ใช้พิมพ์ตัวอักษรหรือตัวเลขนอกเหนือจากตัวเลข 1-7 Script จะมีข้อความบอกแจ้งเตือนผู้ใช้

```
Server Information
1)CPU Info
2)Memory Info
3)OS Info
4)IP Address
5)Manipulate Disk Partition Table
6)Back to Main Menu
7)Exit
Enter your selection : test
Please choose your selection...
Enter return to continue...
```

ภาพที่ 4. 22 ตัวอย่างผู้ใช้พิมพ์ตัวอักษรหรือนอกเหนือจากตัวเลขที่กำหนด

- การบันทึกไฟล์ข้อมูลในหัวข้อต่างๆของฟังก์ชันในการ Monitor Hardware

เงื่อนไขที่ 1 : หากผู้ใช้ต้องการบันทึกไฟล์ข้อมูลหลังจากทำการ Monitor Hardware ผู้ใช้สามารถพิมพ์ “y” จากนั้น Enter เพื่อทำการบันทึกไฟล์ข้อมูลไปยังสถานที่ที่ผู้ใช้กำหนดซึ่งในการกำหนดสถานที่ที่ผู้ใช้สามารถกำหนดได้ในฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล (Set File Location for Export)

```

Architecture:      x86_64
CPU op-mode(s):    32-bit, 64-bit
Byte Order:        Little Endian
CPU(s):            1
On-line CPU(s) list: 0
Thread(s) per core: 1
Core(s) per socket: 1
Socket(s):         1
NUMA node(s):      1
Vendor ID:         GenuineIntel
CPU family:        6
Model:             158
Model name:        Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz
Stepping:          9
CPU MHz:           2800.000
BogoMIPS:          5616.00
Hypervisor vendor: KVM
Virtualization type: full
L1d cache:         32K
L1i cache:         32K
L2 cache:          256K
L3 cache:          6144K
NUMA node0 CPU(s): 0
Do you want to export file [y/n]?:

```

ภาพที่ 4. 23 ตัวอย่างคำถามบันทึกไฟล์ข้อมูลหลังจาก Monitor Hardware
ในหัวข้อที่ 1 (CPU Info)

เงื่อนไขที่ 1 กรณีที่ 1 : หากผู้ใช้มีการกำหนดสถานที่เก็บไฟล์ข้อมูล Script จะทำการบันทึกไฟล์ข้อมูลไว้ใน Directory ที่กำหนดไว้ไฟล์ข้อมูลที่ได้จะมี 2 ไฟล์ ที่แตกต่างกันตรงสกุลไฟล์ โดยจะเป็นสกุลไฟล์ txt และไฟล์ pdf

```

Do you want to export file [y/n]?: y
[ 1 page * 1 copy ] left in /home/admin/marsh/Aug-22-2018_cpuinfo.ps
convert /home/admin/marsh/Aug-22-2018_cpuinfo.ps to /home/admin/marsh/Aug-22-2018_cpuinfo.pdf
Enter return to continue...

```

ภาพที่ 4. 24 ตัวอย่างหากมีการกำหนดสถานที่เก็บไฟล์ข้อมูลแล้วบันทึกไฟล์ข้อมูล

เงื่อนไขที่ 1 กรณีที่ 2 : หากผู้ใช้ไม่กำหนดสถานที่เก็บไฟล์ข้อมูล Script จะแสดงข้อความแจ้งเตือนให้ผู้ใช้ไปกำหนดสถานที่เก็บไฟล์ข้อมูลก่อนที่จะทำการบันทึกไฟล์ข้อมูล

```

Do you want to export file [y/n]?: y
DONT HAVE FILE LOCATION : You should be set file location
Enter return to continue...

```

ภาพที่ 4. 25 ตัวอย่างหากไม่มีการกำหนดสถานที่เก็บไฟล์ข้อมูลแล้วบันทึกไฟล์ข้อมูล

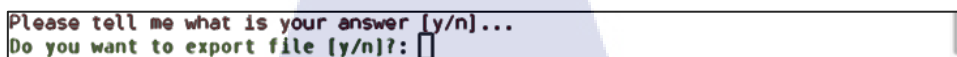
เงื่อนไขที่ 2 : หากผู้ใช้ไม่ต้องการบันทึกไฟล์ข้อมูลหลังจากการ Monitor Hardware ผู้ใช้สามารถพิมพ์ “n” จากนั้น Enter เพื่อออกไปยังหน้าเมนูฟังก์ชันในการ Monitor Hardware



```
Do you want to export file (y/n)?: n
Enter return to continue...
```

ภาพที่ 4. 26 ตัวอย่างหากผู้ใช้พิมพ์ “n” หลังจากนั้น Enter

เงื่อนไขที่ 3 : หากผู้ใช้พิมพ์ตัวอักษรนอกเหนือจาก “y” หรือ “n” Script จะมีข้อความบอกแจ้งเตือนผู้ใช้ และให้ผู้ใช้พิมพ์คำตอบอีกครั้ง

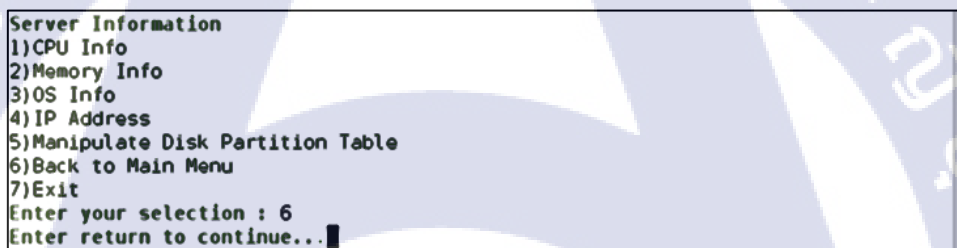


```
Please tell me what is your answer (y/n)...
Do you want to export file (y/n)?:
```

ภาพที่ 4. 27 ตัวอย่างหากผู้ใช้พิมพ์นอกเหนือ “n” หรือ “y” หลังจากนั้น Enter

- การออกจากการใช้ฟังก์ชันในการ Monitor Hardware

เงื่อนไขที่ 1 : ผู้ใช้สามารถพิมพ์เลข 6 จากนั้น Enter จะเป็นการทำให้ผู้ใช้ย้อนกลับไปหน้าจอหลัก โดยการพิมพ์เลข 6 หมายถึงการใช้งานฟังก์ชันหัวข้อย้อนกลับไปเมนูหลัก (Back to Main Menu) ที่มีอยู่ในเมนูการ Monitor Hardware



```
Server Information
1)CPU Info
2)Memory Info
3)OS Info
4)IP Address
5)Manipulate Disk Partition Table
6)Back to Main Menu
7)Exit
Enter your selection : 6
Enter return to continue...
```

ภาพที่ 4. 28 ตัวอย่างหากผู้ใช้เลือกพิมพ์เลข 6 จะเป็นการออกจากหน้าจอการ Monitor Hardware

เงื่อนไขที่ 2 : ผู้ใช้สามารถพิมพ์เลข 7 จากนั้น Enter จะเป็นการทำให้ผู้ใช้ออกจากการใช้งาน Script

ฟังก์ชันในการ Monitor System (System Information)

- การแสดงผลหน้าจอหากเกิดข้อผิดพลาดภายในฟังก์ชันในการ Monitor System

กรณีที่ 1 : หากผู้ใช้ไม่ได้พิมพ์ตัวเลข 1-6 หรือเว้นช่องว่างเพื่อเข้าใช้งานหัวข้อของฟังก์ชันในการ Monitor System ตัว Script จะมีข้อความบอกแจ้งเตือนผู้ใช้


```

System Information
1)Free and Use Memory
2)File System Disk Space Usage
3)Percentage CPU/RAM
4)System Uptime
5)Back to Main Menu
6)Exit
Enter your selection :
Please choose your selection...
Enter return to continue...

```

ภาพที่ 4. 29 ตัวอย่างผู้ใช้ไม่ได้พิมพ์ตัวเลขหรือเว้นช่องว่าง

กรณีที่ 2 : หากผู้ใช้พิมพ์ตัวอักษรหรือตัวเลขนอกเหนือจากตัวเลข 1-6 Script จะมีข้อความบอกแจ้งเตือนผู้ใช้

```

System Information
1)Free and Use Memory
2)File System Disk Space Usage
3)Percentage CPU/RAM
4)System Uptime
5)Back to Main Menu
6)Exit
Enter your selection : test
Please choose your selection...
Enter return to continue...

```

ภาพที่ 4. 30 ตัวอย่างผู้ใช้พิมพ์ตัวอักษรหรือนอกเหนือจากตัวเลขที่กำหนด

- การบันทึกไฟล์ข้อมูลในหัวข้อต่างๆของฟังก์ชันในการ Monitor System

เงื่อนไขที่ 1 : หากผู้ใช้ต้องการบันทึกไฟล์ข้อมูลหลังจากทำการ Monitor System ผู้ใช้สามารถพิมพ์ “y” จากนั้น Enter เพื่อทำการบันทึกไฟล์ข้อมูลไปยังสถานที่ที่ผู้ใช้กำหนดซึ่งในการกำหนดสถานที่ที่ผู้ใช้สามารถกำหนดได้ในฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล (Set File Location for Export)

```

Display Memory In Megabyte
total      used      free      shared  buff/cache  available
Mem:       998      442      226       11       321       486
Swap:      1227      276      951
Percent Memory Use: 44.7%
Percent Swap Use: 22.5%
Do you want to export file [y/n]? :

```

ภาพที่ 4. 31 ตัวอย่างคำถามบันทึกไฟล์ข้อมูลหลังจาก Monitor System
ในหัวข้อที่ 1 (Free and Use Memory)

เงื่อนไขที่ 1 กรณีที่ 1 : หากผู้ใช้มีการกำหนดสถานที่เก็บไฟล์ข้อมูล Script จะทำการบันทึกไฟล์ข้อมูลไว้ใน Directory ที่กำหนดไว้ไฟล์ข้อมูลที่ได้จะมี 2 ไฟล์ ที่แตกต่างกันตรงสกุลไฟล์ โดยจะเป็นสกุลไฟล์ txt และไฟล์ pdf

```
Do you want to export file [y/n]?: y
[ 1 page * 1 copy ] left in /home/admin/marsh/Aug-22-2018_freeusememory.ps
convert /home/admin/marsh/Aug-22-2018_freeusememory.ps to /home/admin/marsh/Aug-
22-2018_freeusememory.pdf
Enter return to continue...
```

ภาพที่ 4. 32 ตัวอย่างหากมีการกำหนดสถานที่เก็บไฟล์ข้อมูลแล้วบันทึกไฟล์ข้อมูล

เงื่อนไขที่ 1 กรณีที่ 2 : หากผู้ใช้ไม่กำหนดสถานที่เก็บไฟล์ข้อมูล Script จะแสดงข้อความแจ้งเตือนให้ผู้ใช้เข้าไปกำหนดสถานที่เก็บไฟล์ข้อมูลก่อนที่จะทำการบันทึกไฟล์ข้อมูล

```
Do you want to export file [y/n]?: y
DONT HAVE FILE LOCATION : You should be set file location
Enter return to continue...
```

ภาพที่ 4. 33 ตัวอย่างหากไม่มีการกำหนดสถานที่เก็บไฟล์ข้อมูลแล้วบันทึกไฟล์ข้อมูล

เงื่อนไขที่ 2 : หากผู้ใช้ไม่ต้องการบันทึกไฟล์ข้อมูลหลังจากการ Monitor System ผู้ใช้สามารถพิมพ์ “n” จากนั้น Enter เพื่อออกไปยังหน้าเมนูฟังก์ชันในการ Monitor System

```
Do you want to export file [y/n]?: n
Enter return to continue...
```

ภาพที่ 4. 34 ตัวอย่างหากผู้ใช้พิมพ์ “n” หลังจากนั้น Enter

เงื่อนไขที่ 3 : หากผู้ใช้พิมพ์ตัวอักษรนอกเหนือจาก “y” หรือ “n” Script จะมีข้อความบอกแจ้งเตือนผู้ใช้ และให้ผู้ใช้พิมพ์คำตอบอีกครั้ง

```
Please tell me what is your answer [y/n]...
Do you want to export file [y/n]?:
```

ภาพที่ 4. 35 ตัวอย่างหากผู้ใช้พิมพ์นอกเหนือ “n” หรือ “y” หลังจากนั้น Enter

- การออกจากการใช้ฟังก์ชันในการ Monitor System

เงื่อนไขที่ 1 : ผู้ใช้สามารถพิมพ์เลข 5 จากนั้น Enter จะเป็นการทำให้ผู้ใช้อย้อนกลับไปหน้าเมนูหลัก โดยการพิมพ์เลข 5 หมายถึงการใช้งานฟังก์ชันย้อนกลับไปเมนูหลัก (Back to Main Menu) ที่มีอยู่ในเมนูการ Monitor System

```
System Information
1)Free and Use Memory
2)File System Disk Space Usage
3)Percentage CPU/RAM
4)System Uptime
5)Back to Main Menu
6)Exit
Enter your selection : 5
Enter return to continue...
```

ภาพที่ 4. 36 ตัวอย่างหากผู้ใช้เลือกพิมพ์เลข 5 จะเป็นการออกจากหน้าเมนูการ Monitor System

เงื่อนไขที่ 2 : ผู้ใช้สามารถพิมพ์เลข 6 จากนั้น Enter จะเป็นการทำให้ผู้ใช้ออกจากการใช้งาน Script

ฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup (Backup Log File)

- การแสดงผลทางหน้าจอหากเข้าใช้ฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup

กรณีที่ 1 : หากผู้ใช้ไม่ได้พิมพ์ตัวเลข 1-7 หรือเว้นช่องว่างเพื่อเข้าใช้งานหัวข้อของฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup ตัว Script จะมีข้อความบอกแจ้งเตือนผู้ใช้

```
Backup Log File
1)Set Backup Log File Location
2)View Backup Log File (Today)
3)View Backup Log File (Yesterday)
4)View Backup Log File (History)
5)View Backup Log File (Difference Time)
6)Back to Main Menu
7)Exit
Enter your selection:
Please choose your selection...
Enter return to continue...
```

ภาพที่ 4. 37 ตัวอย่างผู้ใช้ไม่ได้พิมพ์ตัวเลขหรือเว้นช่องว่าง

กรณีที่ 2 : หากผู้ใช้พิมพ์ตัวอักษรหรือตัวเลขนอกเหนือจากตัวเลข 1-7 Script จะมีข้อความบอกแจ้งเตือนผู้ใช้

```
Backup Log File
1)Set Backup Log File Location
2)View Backup Log File (Today)
3)View Backup Log File (Yesterday)
4)View Backup Log File (History)
5)View Backup Log File (Difference Time)
6)Back to Main Menu
7)Exit
Enter your selection: test
Please choose your selection...
Enter return to continue...
```

ภาพที่ 4. 38 ตัวอย่างผู้ใช้ได้พิมพ์ตัวอักษรหรือนอกเหนือจากตัวเลขที่กำหนด

- การใช้งานหัวข้อที่ 1 ของฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup

การแสดงผลทางหน้าจอหากใช้งานหัวข้อที่ 1 ของฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup

กรณีที่ 1 : หากหัวข้อที่ 1 ของฟังก์ชันกำหนดสถานที่ที่ Server เก็บไฟล์ข้อมูลที่ Backup เคยกำหนดสถานที่ Script จะมีข้อความบอกสถานที่ที่เคยกำหนด

```
Enter your selection: 1
Your file location is /home/admin/marsh/
Enter your Location (Ex./home/user/Files_location)
If you want to go back enter [q]:
```

ภาพที่ 4. 39 ตัวอย่างข้อความบอกสถานที่ที่เคยกำหนด

กรณีที่ 2 : หากหัวข้อที่ 1 ของฟังก์ชันกำหนดสถานที่ที่ Server เก็บไฟล์ข้อมูลที่ Backup ไม่เคยกำหนดสถานที่ Script จะมีข้อความบอกว่าไม่เคยมีการกำหนดสถานที่

```
Enter your selection: 1
You dont have file location
Enter your Location (Ex./home/user/Files_location)
If you enter space your path will be [/]
If you want to go back enter [q]:
```

ภาพที่ 4. 40 ตัวอย่างข้อความบอกสถานที่ที่ไม่เคยมีการกำหนดสถานที่

การออกจากการใช้ฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล

เงื่อนไขที่ 1 : ผู้ใช้สามารถ Enter เพื่อทำการออกจากการใช้งานหัวข้อที่ 1 และกลับไปยังเมนูในการเข้าไปดูไฟล์ข้อมูลที่ Backup

เงื่อนไขที่ 2 : ผู้ใช้สามารถพิมพ์ “q” จากนั้น Enter เพื่อทำการออกจากการใช้งานหัวข้อที่ 1 และกลับไปยังเมนูในการเข้าไปดูไฟล์ข้อมูลที่ Backup

```
Enter your Location (Ex./home/user/Files location)
If you enter space your path will be [/]
If you want to go back enter [q]: q
Back to Main Menu
Enter return to continue...
```

ภาพที่ 4. 41 ตัวอย่างการออกจากการใช้ฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล

การแสดงผลหลังกำหนดสถานที่ในหัวข้อที่ 1 และเข้าใช้งานหัวข้อที่ 2-5 ของฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup

เงื่อนไขที่ 1 : หากมีการกำหนดสถานที่ที่ Server เก็บไฟล์ข้อมูลที่ Backup เอาไว้ ผู้ใช้จะสามารถใช้ฟังก์ชันในหัวข้อที่ 2-5 โดยการกำหนดสถานที่ที่ Server เก็บไฟล์ข้อมูลที่ Backup เอาไว้ผู้ใช้สามารถเข้าใช้ฟังก์ชันในหัวข้อที่ 1

เงื่อนไขที่ 2 : หากไม่มีการกำหนดสถานที่ที่ Server เก็บไฟล์ข้อมูลที่ Backup เอาไว้ ผู้ใช้จะไม่สามารถใช้ฟังก์ชันในหัวข้อที่ 2-5

```
Enter your selection: 2
CANT FIND DIRECTORY : You should be set file location
Enter return to continue...
```

ภาพที่ 4. 42 ตัวอย่างหากไม่มีการกำหนดสถานที่ที่เก็บไฟล์ข้อมูลที่ Backup บน Server

- การเข้าใช้งานหัวข้อที่ 4 ของฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup

เงื่อนไขที่ 1 : หากผู้ใช้พิมพ์ “q” จากนั้น Enter หรือ Enter จะเป็นการออกจากการใช้งานหัวข้อที่ 4 และกลับไปยังเมนูในการเข้าไปดูไฟล์ข้อมูลที่ Backup

```
Enter your selection: 4
Apr-28-2018 Apr-29-2018 Apr-30-2018 Aug-09-2018 Aug-10-2018
Enter Backup Log File Name
If you want to go back enter [q]:
Back to Backup Log File Menu
Enter return to continue...
```

ภาพที่ 4. 43 ตัวอย่างการออกจากการใช้งานหัวข้อที่ 4 โดยผู้ใช้ Enter

```
Enter your selection: 4
Apr-28-2018 Apr-29-2018 Apr-30-2018 Aug-09-2018 Aug-10-2018
Enter Backup Log File Name
If you want to go back enter [q]: q
Back to Backup Log File Menu
Enter return to continue...
```

ภาพที่ 4. 44 ตัวอย่างการออกจากการใช้งานหัวข้อที่ 4 โดยผู้ใช้พิมพ์ “q” จากนั้น Enter

เงื่อนไขที่ 2 : หากผู้ใช้พิมพ์ชื่อไฟล์ข้อมูลที่ Backup อยู่บนเครื่อง Server จากนั้น Enter Script จะแสดงข้อมูลภายในไฟล์ข้อมูลที่ Backup เอาไว้

```
Backup Log File
1)Set Backup Log File Location
2)View Backup Log File (Today)
3)View Backup Log File (Yesterday)
4)View Backup Log File (History)
5)View Backup Log File (Difference Time)
6)Back to Main Menu
7)Exit
Enter your selection: 4
Apr-28-2018 Apr-29-2018 Apr-30-2018 Aug-09-2018 Aug-10-2018
Enter Backup Log File Name
If you want to go back enter [q]: Apr-28-2018
```

ภาพที่ 4. 45 ตัวอย่างการใช้งานฟังก์ชันในหัวข้อที่ 4 ของฟังก์ชันกำหนดสถานที่ที่ Server เก็บไฟล์ข้อมูลที่ Backup

```
Sat Apr 28 01:30:00 THAIST 2018
Loading tape slot 6 into drive ...
Loading tape completed ! ....
Starting backup RPC database.....
Sat Apr 28 01:30:07 THAIST 2018
Starting backup PROD database.....

SQL*Plus: Release 10.2.0.4.0 - Production on Sat Apr 28 01:30:07 2018

Copyright (c) 1982, 2007, Oracle. All Rights Reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

ภาพที่ 4. 46 ตัวอย่างผลลัพธ์การใช้งานหัวข้อที่ 2-4 ของฟังก์ชันกำหนดสถานที่ที่ Server เก็บไฟล์ข้อมูลที่ Backup

- การใช้งานหัวข้อที่ 5 ของฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup

เงื่อนไขที่ 1 : หากผู้ใช้พิมพ์ “q” จากนั้น Enter หรือ Enter จะเป็นการออกจากการใช้งานหัวข้อที่ 5 และกลับไปยังเมนูในการเข้าไปดูไฟล์ข้อมูลที่ Backup

```
Enter your selection: 5
Apr-28-2018 Apr-29-2018 Apr-30-2018 Aug-09-2018 Aug-10-2018
Enter Backup Log File Name
If you want to go back enter [q]:
Back to Backup Log File Menu
Enter return to continue...
```

ภาพที่ 4. 47 ตัวอย่างหากไม่มีการกำหนดสถานที่ที่เก็บไฟล์ข้อมูลที่ Backup บน Server


```

Enter your selection: 5
Apr-28-2018 Apr-29-2018 Apr-30-2018 Aug-09-2018 Aug-10-2018
Enter Backup Log File Name
If you want to go back enter [q]: q
Back to Backup Log File Menu
Enter return to continue...

```

ภาพที่ 4. 48 ตัวอย่างหากไม่มีการกำหนดสถานที่เก็บไฟล์ข้อมูลที่ Backup บน Server

เงื่อนไขที่ 2 : หากผู้ใช้พิมพ์ชื่อไฟล์ข้อมูลที่ Backup อยู่บนเครื่อง Server จากนั้น Enter Script จะแสดงระยะเวลาที่ใช้ในการบันทึกไฟล์ข้อมูลที่ Backup และจะโชว์ข้อผิดพลาดระหว่างการ Backup

```

Backup Log File
1)Set Backup Log File Location
2)View Backup Log File (Today)
3)View Backup Log File (Yesterday)
4)View Backup Log File (History)
5)View Backup Log File (Difference Time)
6)Back to Main Menu
7)Exit
Enter your selection: 5
Apr-28-2018 Apr-29-2018 Apr-30-2018 Aug-09-2018 Aug-10-2018
Enter Backup Log File Name
If you want to go back enter [q]: Apr-28-2018
Start: 01:30:00
Stop: 01:30:36
Different Time: 00:00:36

ERROR: The archive content is not fully backed up.
Enter return to continue...

```

ภาพที่ 4. 49 ตัวอย่างผลลัพธ์การใช้งานหัวข้อที่ 5 โดยมีข้อผิดพลาดของฟังก์ชันกำหนดสถานที่ที่ Server เก็บไฟล์ข้อมูลที่ Backup

```

Backup Log File
1)Set Backup Log File Location
2)View Backup Log File (Today)
3)View Backup Log File (Yesterday)
4)View Backup Log File (History)
5)View Backup Log File (Difference Time)
6)Back to Main Menu
7)Exit
Enter your selection: 5
Apr-28-2018 Apr-29-2018 Apr-30-2018 Aug-09-2018 Aug-10-2018
Enter Backup Log File Name
If you want to go back enter [q]: Apr-30-2018
Start: 01:30:00
Stop: 05:17:37
Different Time: 03:47:37

Enter return to continue...

```

ภาพที่ 4. 50 ตัวอย่างผลลัพธ์การใช้งานหัวข้อที่ 5 โดยไม่มีข้อผิดพลาดของฟังก์ชันกำหนดสถานที่ที่ Server เก็บไฟล์ข้อมูลที่ Backup

ฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป (Generate Monthly Report)

- การแสดงผลทางหน้าจอหากผู้ใช้ฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป

กรณีที่ 1 : หากผู้ใช้ไม่ได้พิมพ์ตัวเลข 1-7 หรือเว้นช่องว่างเพื่อเข้าใช้งานหัวข้อของฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป Script จะมีข้อความบอกแจ้งเตือนผู้ใช้

```
Generate Monthly Report
1)Set Data File Location
2)Set File sar.txt Location
3)Create Data from "sar"
4)Change Data to Text File
5)Summary Report
6)Back to Main Menu
7)Exit
Enter your selection:
Please choose your selection...
Enter return to continue...
```

ภาพที่ 4. 51 ตัวอย่างผู้ใช้ไม่ได้พิมพ์ตัวเลขหรือเว้นช่องว่าง

กรณีที่ 2 : หากผู้ใช้พิมพ์ตัวอักษรหรือตัวเลขนอกเหนือจากตัวเลข 1-7 Script จะมีข้อความบอกแจ้งเตือนผู้ใช้

```
Generate Monthly Report
1)Set Data File Location
2)Set File sar.txt Location
3)Create Data from "sar"
4)Change Data to Text File
5)Summary Report
6)Back to Main Menu
7)Exit
Enter your selection: test
Please choose your selection...
Enter return to continue...
```

ภาพที่ 4. 52 ตัวอย่างผู้ใช้ได้พิมพ์ตัวอักษรหรือนอกเหนือจากตัวเลขที่กำหนด

- การเข้าใช้งานหัวข้อที่ 1 และ 2 ของฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป

การแสดงผลทางหน้าจอหากเข้าใช้งานหัวข้อที่ 1 และ 2 ของฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป

กรณีที่ 1 : หากหัวข้อที่ 1 และ 2 ของฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุปเคยกำหนดสถานที่ที่ใช้ในการเก็บไฟล์ดิบหรือไฟล์สกุล txt Script จะมีข้อความบอกสถานที่ที่เคยกำหนด

```
Now your file location is /home/admin/marsh/
Enter your location (Ex./home/user/location)
If you want to go back enter [q]:
```

ภาพที่ 4. 53 ตัวอย่างข้อความบอกสถานที่ที่เคยกำหนด

กรณีที่ 2 : หากหัวข้อที่ 1 และ 2 ของฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุปไม่เคยกำหนดสถานที่ที่ใช้ในการเก็บไฟล์ดิบหรือไฟล์สกุล txt Script จะมีข้อความบอกว่าไม่เคยมีการกำหนดสถานที่

```
Enter your selection: 1
You dont have file location
Enter your Location (Ex./home/user/Files_location)
If you enter space your path will be [/])
If you want to go back enter [q]:
```

ภาพที่ 4. 54 ตัวอย่างข้อความบอกสถานที่ที่ไม่เคยมีการกำหนดสถานที่

การออกจากการใช้ฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป

เงื่อนไขที่ 1 : ผู้ใช้สามารถ Enter เพื่อทำการออกจากการใช้งานหัวข้อที่ 1 และ 2 เพื่อกลับไปยังเมนูในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป

เงื่อนไขที่ 2 : ผู้ใช้สามารถพิมพ์ “q” จากนั้น Enter เพื่อทำการออกจากการใช้งานหัวข้อที่ 1 และ 2 เพื่อกลับไปยังเมนูในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป

```
Enter your Location (Ex./home/user/Files_location)
If you enter space your path will be [/])
If you want to go back enter [q]: q
Back to Main Menu
Enter return to continue...
```

ภาพที่ 4. 55 ตัวอย่างการออกจากการใช้ฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล

การแสดงผลหลังกำหนดสถานที่ในหัวข้อที่ 1 และ 2 เพื่อเข้าใช้งานหัวข้อที่ 3-5 ของฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป

เงื่อนไขที่ 1 : หากมีการกำหนดสถานที่ที่ใช้ในการเก็บไฟล์ดิบและไฟล์สรุป txt ผู้ใช้จะสามารถใช้ฟังก์ชันในหัวข้อที่ 2-5 โดยการกำหนดสถานที่ที่ใช้ในการเก็บไฟล์ดิบและไฟล์สรุป txt ผู้ใช้สามารถเข้าใช้ฟังก์ชันในหัวข้อที่ 1 และ 2 เพื่อเข้าไปกำหนดสถานที่

เงื่อนไขที่ 2 : หากไม่มีการกำหนดสถานที่ที่ใช้ในการเก็บไฟล์ดิบและไฟล์สรุป txt ผู้ใช้จะไม่สามารถใช้ฟังก์ชันในหัวข้อที่ 2-5

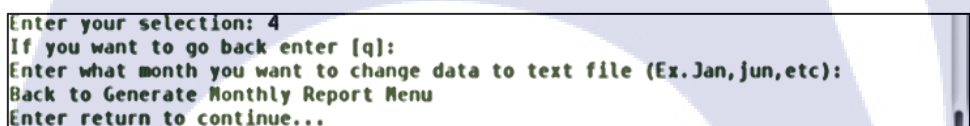


```
Enter your selection: 3
CANT FIND DIRECTORY : You should be set file location
Enter return to continue...
```

ภาพที่ 4. 56 ตัวอย่างหากไม่มีการกำหนดสถานที่ที่เก็บไฟล์ข้อมูลที่ Backup บน Server

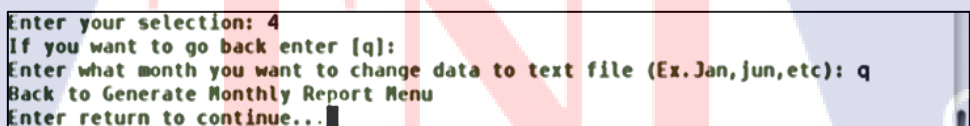
การเข้าใช้งานหัวข้อที่ 4 ของฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป

เงื่อนไขที่ 1 : หากผู้ใช้พิมพ์ “q” จากนั้น Enter หรือ Enter จะเป็นการออกจากการใช้งานหัวข้อที่ 4 และกลับไปยังเมนูในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป



```
Enter your selection: 4
If you want to go back enter [q]:
Enter what month you want to change data to text file (Ex. Jan, jun, etc):
Back to Generate Monthly Report Menu
Enter return to continue...
```

ภาพที่ 4. 57 ตัวอย่างการออกจากการใช้งานหัวข้อที่ 4 โดยผู้ใช้ Enter



```
Enter your selection: 4
If you want to go back enter [q]:
Enter what month you want to change data to text file (Ex. Jan, jun, etc): q
Back to Generate Monthly Report Menu
Enter return to continue...
```

ภาพที่ 4. 58 ตัวอย่างการออกจากการใช้งานหัวข้อที่ 4 โดยผู้ใช้พิมพ์ “q” จากนั้น Enter

เงื่อนไขที่ 2 : หากผู้ใช้พิมพ์เดือนจากนั้น Enter Script จะทำการดึงไฟล์ข้อมูลดิบที่ได้ทำการกำหนดสถานที่ไว้ในหัวข้อที่ 1 ของฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุปแปลงเป็นไฟล์สรุป txt และนำไปเก็บไว้ในสถานที่ที่เรากำหนดไว้ในหัวข้อที่ 2 ของฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป โดยหลังจากการ

ทำงานเสร็จสิ้น Script จะไม่กลับไปหน้าจอของฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป แต่จะให้เลือกเดือนอีกครั้ง

```
Cannot open /home/admin/marsh/Jan-28-2018: No such file or directory
Cannot open /home/admin/marsh/Jan-29-2018: No such file or directory
Cannot open /home/admin/marsh/Jan-30-2018: No such file or directory
Cannot open /home/admin/marsh/Jan-31-2018: No such file or directory
If you want to go back enter [q]:
Enter what month you want to change data to text file (Ex. Jan, jun, etc):
```

ภาพที่ 4. 59 ตัวอย่างการแปลงไฟล์ข้อมูลดิบเป็นไฟล์สกุล txt

- การเข้าใช้งานหัวข้อที่ 5 ของฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป

เงื่อนไขที่ 1 : หากผู้ใช้พิมพ์ “q” จากนั้น Enter หรือ Enter จะเป็นการออกจากการใช้งานหัวข้อที่ 5 และกลับไปยังเมนูในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป

```
Enter your selection: 5
If you want to go back enter [q]:
Start (Ex. Jun-25-2018)

Back to Generate Monthly Report Menu
Enter return to continue...
```

ภาพที่ 4. 60 ตัวอย่างการออกจากการใช้งานหัวข้อที่ 5 โดยผู้ใช้ Enter

```
If you want to go back enter [q]:
Start (Ex. Jun-25-2018)
q
Enter return to continue...
```

ภาพที่ 4. 61 ตัวอย่างการออกจากการใช้งานหัวข้อที่ 5 โดยผู้ใช้พิมพ์ “q” จากนั้น Enter

เงื่อนไขที่ 2 : หากผู้ใช้พิมพ์วันเดือนปี (เริ่มต้น) – วันเดือนปี (สิ้นสุด) ที่ต้องการสรุปค่าสถานะระบบ Script จะทำผลสรุปของค่าสถานะ

```
Generate Monthly Report
1)Set Data File Location
2)Set File sar.txt Location
3)Create Data from "sar"
4)Change Data to Text File
5)Summary Report
6)Back to Main Menu
7)Exit
Enter your selection: 5
If you want to go back enter [q]:
Start (Ex. Jun-25-2018)
Jul-1-2018
End (Ex. Jul-30-2018)
Jul-30-2018
```

ภาพที่ 4. 62 ตัวอย่างการใช้งานหัวข้อที่ 5

ตัวอย่าง Script สำหรับการ Monitor

```

if [ -f "/tmp/lofpath.txt" ]
then
    lofpath=$(cat /tmp/lofpath.txt)
    echo -e "\e[1;32mYour file location is $lofpath"
    echo -e "\e[1;32mEnter your Location (Ex./home/user/Files_location)"
    echo -e "\e[1;32mIf you want to go back enter [q]: \e[0m\c"
    read setloc
    if [[ $setloc == "q" ]]
    then
        echo -e "\e[1;32mBack to Main Menu"
    elif [[ $setloc = "" ]];then
        echo -e "\e[1;32mBack to Main Menu"
    else
        if [[ -d "$setloc" ]]
        then
            echo -e "\e[1;32mYour location file: $setloc"
            echo $setloc/" > /tmp/lofpath.txt
        else
            echo -e "\e[0;31mDONT HAVE DIRECTORY : Please try agian
later\e[0m"
        fi
    fi
else
    echo -e "\e[1;32mYou dont have file location"
    echo -e "\e[1;32mEnter your Location (Ex./home/user/Files_location)"
    echo -e "\e[1;32mIf you enter space your path will be [/]"
    echo -e "\e[1;32mIf you want to go back enter [q]: \c"
    read setloc
    if [[ $setloc == "q" ]]
    then
        echo -e "\e[1;32mBack to Main Menu"
    elif [[ $setloc = "" ]];then
        echo -e "\e[1;32mBack to Main Menu"
    else
        if [[ -d "$setloc" ]]
        then
            echo -e "\e[1;32mYour location file: $setloc"
            echo $setloc/" > /tmp/lofpath.txt
        else
            echo -e "\e[0;31mDONT HAVE DIRECTORY : Please try agian later\e[0m"
        fi
    fi
fi;

```

ภาพที่ 4. 63 ตัวอย่างฟังก์ชันกำหนดสถานที่เก็บไฟล์ข้อมูล (Set File Location for Export)

จากภาพที่ 4.63 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการกำหนดสถานที่เก็บไฟล์ข้อมูลหรือฟังก์ชันสำหรับการกำหนดสถานที่เก็บไฟล์ข้อมูลหัวข้อ Set File Location for Export จาก Script จะเห็นได้ว่าหลังจากทำการกำหนดสถานที่ในการเก็บไฟล์ข้อมูล

ตัว Script จะทำการสร้างไฟล์สกุล txt ที่ชื่อ lofpath.txt (สถานที่เก็บไฟล์ lofpath.txt อยู่ที่ /tmp/lofpath.txt) เอาไว้เพื่อนำไว้ใช้เก็บตำแหน่งสถานที่ที่ใช้ในการเก็บไฟล์ที่เกิดจากการตรวจสอบระบบหรือ Monitor Hardware

```
echo -e "\e[0;32mCPU Info\n`lscpu`" | more
while true
do
echo -e "\e[1;32mDo you want to export file [y/n]? \e[0m: \c"
read yesno
if [[ $yesno == "y" ]]
then
if [ -f "/tmp/lofpath.txt" ]
then
lofpath=$(cat /tmp/lofpath.txt)
if [ -f "$lofpath`date +%b-%d-%Y`_cpuinfo.txt" ];then
rm $lofpath`date +%b-%d-%Y`_cpuinfo.txt
fi
echo -e "CPU-Info\n`lscpu`" | more > $lofpath`date +%b-%d-%Y`_cpuinfo.txt
export cpuinfo=$lofpath`date +%b-%d-%Y`_cpuinfo.txt
echo -e "\e[1;34m\c"
enscript -p $lofpath`date +%b-%d-%Y`_cpuinfo.ps $cpuinfo
echo "convert \"$lofpath`date +%b-%d-%Y`_cpuinfo.ps to
"$lofpath`date +%b-%d-%Y`_cpuinfo.pdf"
ps2pdf $lofpath`date +%b-%d-%Y`_cpuinfo.ps $lofpath`date +%b-
%d-%Y`_cpuinfo.pdf
rm $lofpath`date +%b-%d-%Y`_cpuinfo.ps
break;
else
echo -e "\e[0;31mDONT HAVE FILE LOCATION : You should be set file
location\e[0m"
break;
fi
elif [[ $yesno == "n" ]]
then
break;
else
clear;
echo -e "\e[0;32mCPU-Info\n`lscpu`" | more
echo -e "\e[0;31mPlease tell me what is your answer [y/n]..."
fi
done
;;
```

ภาพที่ 4. 64 ตัวอย่างฟังก์ชันในการ Monitor Hardware (Server Information) หัวข้อ CPU Info

จากภาพที่ 4.64 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการ Monitor Hardware หัวข้อ CPU Info จาก Script จะเห็นได้ว่าการใช้ Command lscpu เพื่อแสดงรายละเอียดข้อมูล CPU หลังจากที่มีการแสดงรายละเอียดข้อมูลของ CPU ตัว Script จะให้ผู้ตัดสินใจว่าผู้ใช้ต้องการที่จะทำการบันทึกข้อมูลหรือไม่ต้องการบันทึกข้อมูล หากผู้ใช้ทำการบันทึกข้อมูล ข้อมูลที่

ได้จะไปอยู่ในสถานที่ที่ผู้ใช้งานกำหนดเอาไว้โดยชื่อไฟล์ที่ได้คือ วันที่_cpuinfo.pdf และ วันที่_cpuinfo.txt

```
echo -e "\e[0;32mMemory-Info\n`awk 'S3=="KB"{if ($2>1024**2){S2=$2/1024**2;S3="GB";} else if ($2>1024){S2=$2/1024;S3="MB";} else {S3="KB";}} 1' /proc/meminfo`" | column -t | more

while true
do
echo -e "\e[97;42mDo you want to export file [y/n]? \e[0m: \c"
read yesno
if [[ $yesno == "y" ]]
then
if [ -f "/tmp/lofpath.txt" ]
then
lofpath=$(cat /tmp/lofpath.txt)
if [ -f "$lofpath`date +%b-%d-%Y`_meminfo.txt" ];then
rm $lofpath`date +%b-%d-%Y`_meminfo.txt
fi

echo -e "Memory-Info\n`awk 'S3=="KB"{if ($2>1024**2){S2=$2/1024**2;S3="GB";} else if ($2>1024){S2=$2/1024;S3="MB";} else {S3="KB";}} 1' /proc/meminfo`" | column -t | more > $lofpath`date +%b-%d-%Y`_meminfo.txt
echo -e "\e[1;34m\c"
export meminfo=$lofpath`date +%b-%d-%Y`_meminfo.txt
enscript -p $lofpath`date +%b-%d-%Y`_meminfo.ps $meminfo
echo "convert \"$lofpath`date +%b-%d-%Y`_meminfo.ps to \"$lofpath`date +%b-%d-%Y`_meminfo.pdf"
ps2pdf $lofpath`date +%b-%d-%Y`_meminfo.ps $lofpath`date +%b-%d-%Y`_meminfo.pdf
rm $lofpath`date +%b-%d-%Y`_meminfo.ps
break;
else
echo -e "\e[0;31mDONT HAVE FILE LOCATION : You should be set file location\e[0m"
break;
fi
elif [[ $yesno == "n" ]]
then
break;
else
clear;
echo -e "\e[0;32mMemory-Info\n`awk 'S3=="KB"{if ($2>1024**2){S2=$2/1024**2;S3="GB";} else if ($2>1024){S2=$2/1024;S3="MB";} else {S3="KB";}} 1' /proc/meminfo`" | column -t | more
echo -e "\e[0;31mPlease tell me what is your answer [y/n]..."
fi
done
;;
```

ภาพที่ 4.65 ตัวอย่างฟังก์ชันในการ Monitor Hardware (Server Information) หัวข้อ Memory Info

จากภาพที่ 4.65 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการ Monitor Hardware หัวข้อ Memory Info จาก Script จะเห็นได้ว่าการใช้ Command awk '\$3=="KB"'{if (\$2>1024**2){\$2=\$2/1024**2;\$3="GB";} else if (\$2>1024){\$2=\$2/1024;\$3="MB";} else {\$3="KB";}} 1' /proc/meminfo เพื่อแสดงรายละเอียดข้อมูล Memory ของ Hardware หลังจากที่มีการแสดงรายละเอียดข้อมูลของ Memory ของ Hardware แล้วตัว Script จะให้ผู้ใช้ตัดสินใจว่าผู้ใช้ต้องการที่จะทำการบันทึกข้อมูลหรือไม่ต้องการบันทึกข้อมูล หากผู้ใช้ทำการบันทึกข้อมูล ข้อมูลที่ได้จะไปอยู่ในสถานที่ที่ผู้ใช้กำหนดเอาไว้โดยชื่อไฟล์ที่ได้คือ วันที่_meminfo.pdf และ วันที่_meminfo.txt

```
clear;
echo -e "\e[0;32mHostname: `hostname`"
cut /etc/os-release -d '=' -f2 | sed 's// /g' | awk 'NR==2{print "\033[32mOS Version: \"$1\"\033[0m"}'
mc_type=`uname -m`
if [ ${mc_type} == 'x86_64' ]; then
    echo -e "\e[0;32mArchitecture: 64-bit"
else
    echo -e "\e[0;32mArchitecture: 32-bit"
fi
echo -e "\e[0;32mKernel Name: `uname -r`"
cat /etc/redhat-release | awk '{print "\033[32mRelease: \"$0\"\033[0m"}'
while true
do
echo -e "\e[1;32mDo you want to export file [y/n]? \e[0m: \c"
read yesno
if [[ $yesno == "y" ]]
then
    if [ -f "/tmp/lofpath.txt" ]
    then
        lofpath=$(cat /tmp/lofpath.txt)
        if [ -f "$lofpath`date +%b-%d-%Y`_osinfo.txt" ];then
            rm $lofpath`date +%b-%d-%Y`_osinfo.txt
        fi
        echo -e "Hostname: `hostname`" >> $lofpath`date +%b-%d-%Y`_osinfo.txt
        cut /etc/os-release -d '=' -f2 | sed 's// /g' | awk 'NR==2{print "OS Version: \"$1`" >> `date +%b-%d-%Y`_osinfo.txt
        mc_type=`uname -m`
        if [ ${mc_type} == 'x86_64' ]; then
            echo "Architecture: 64-bit" >> $lofpath`date +%b-%d-%Y`_osinfo.txt
        else
            echo "Architecture: 32-bit" >> $lofpath`date +%b-%d-%Y`_osinfo.txt
        fi
        echo -e "Kernel Name: `uname -r`" >> $lofpath`date +%b-%d-%Y`_osinfo.txt
        cat /etc/redhat-release | awk '{print "Release: \"$0`" >> $lofpath`date +%b-%d-%Y`_osinfo.txt
        echo -e "\e[1;34m\c"
        export osinfo=$lofpath`date +%b-%d-%Y`_osinfo.txt
        enscrip -p $lofpath`date +%b-%d-%Y`_osinfo.ps $osinfo
        echo "convert \"$lofpath`date +%b-%d-%Y`_osinfo.ps to \"$lofpath`date +%b-%d-%Y`_osinfo.pdf"
        ps2pdf $lofpath`date +%b-%d-%Y`_osinfo.ps $lofpath`date +%b-%d-%Y`_osinfo.pdf
        rm $lofpath`date +%b-%d-%Y`_osinfo.ps
        break;
    else
        echo -e "\e[0;31mDONT HAVE FILE LOCATION : You should be set file location\e[0m"
        break;
    fi
fi
```

ภาพที่ 4. 66 ตัวอย่างฟังก์ชันในการ Monitor Hardware (Server Information) หัวข้อ Memory Info

ส่วนที่ 1

```

elif [[ $yesno == "n" ]]
then
    break;
else
    clear;
    echo -e "\e[0;32m\nHostname: `hostname`"
    cut /etc/os-release -d '=' -f2 | sed 's// /g' | awk 'NR==2{print "\033[32mOS Version: "$1"\033[0m"}'
    mc_type=`uname -m`
    if [ ${mc_type} == 'x86_64' ]; then
        echo -e "\e[0;32mArchitecture: 64-bit"
    else
        echo -e "\e[0;32mArchitecture: 32-bit"
    fi
    echo -e "\e[0;32mKernel Name: `uname -r`"
    cat /etc/redhat-release | awk '{print "\033[32mRelease: "$0"\033[0m"}'
    echo -e "\e[0;31mPlease tell me what is your answer [y/n]...\e[0m"

fi
done
;;

```

ภาพที่ 4.67 ตัวอย่างฟังก์ชันในการ Monitor Hardware (Server Information) หัวข้อ Memory Info ส่วนที่ 2

จากภาพที่ 4.66 และภาพที่ 4.67 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการ Monitor Hardware หัวข้อ OS Info จาก Script จะเห็นได้ว่าการใช้ Command ต่างๆเพื่อใช้แสดงรายละเอียดข้อมูลของระบบปฏิบัติการ หลังจากที่มีการแสดงรายละเอียดข้อมูลของระบบปฏิบัติการแล้วตัว Script จะให้ผู้ใช้ตัดสินใจว่าผู้ใช้งานต้องการที่จะทำการบันทึกข้อมูลหรือไม่ ต้องการบันทึกข้อมูล หากผู้ใช้งานทำการบันทึกข้อมูล ข้อมูลที่ได้จะไปอยู่ในสถานที่ที่ผู้ใช้งานกำหนดเอาไว้โดยชื่อไฟล์ที่ได้คือ วันที่_osinfo.pdf และ วันที่_osinfo.txt

```

clear;

echo -e "\e[0;32mAll IP Address\n`ip -o -4 addr show | awk '{print $1 " " $2": " $4}'"
echo -e "\e[0;32mgateway: `ip r | grep default | awk '{print $3}'"
while true

do
echo -e "\e[1;32mDo you want to export file [y/n]? \e[0m: \c"
read yesnno
if [[ $yesnno == "y" ]]
then
    if [ -f "/tmp/lofpath.txt" ]
    then
        lofpath=$(cat /tmp/lofpath.txt)
        if [ -f "$lofpath`date +%b-%d-%Y`_ipaddress.txt" ];then
            rm $lofpath`date +%b-%d-%Y`_ipaddress.txt
        fi

        echo -e "All IP Address\n`ip -o -4 addr show | awk '{print $1 " " $2": " $4}'" >> $lofpath`date +%b-
%d-%Y`_ipaddress.txt
        echo -e "gateway: `ip r | grep default | awk '{print $3}'" >> $lofpath`date +%b-%d-
%Y`_ipaddress.txt
        echo -e "\e[1;34m\c"
        export ipaddress=$lofpath`date +%b-%d-%Y`_ipaddress.txt
        enscrip -p $lofpath`date +%b-%d-%Y`_ipaddress.ps $ipaddress
        echo "convert \"$lofpath`date +%b-%d-%Y`_ipaddress.ps to \"$lofpath`date +%b-%d-%Y`_ipaddress.pdf"
        ps2pdf $lofpath`date +%b-%d-%Y`_ipaddress.ps $lofpath`date +%b-%d-%Y`_ipaddress.pdf
        rm $lofpath`date +%b-%d-%Y`_ipaddress.ps

        break;
    else
        echo -e "\e[0;31mDONT HAVE FILE LOCATION : You should be set file location\e[0m"
        break;
    fi
elif [[ $yesnno == "n" ]]
then
    break;
else
    clear;
    echo -e "\e[0;32mAll IP Address\n`ip -o -4 addr show | awk '{print $1 " " $2": " $4}'"
    echo -e "\e[0;32mgateway: `ip r | grep default | awk '{print $3}'"
    echo -e "\e[0;31mPlease tell me what is your answer [y/n]..."

fi
done
;;

```

ภาพที่ 4. 68 ตัวอย่างฟังก์ชันในการ Monitor Hardware (Server Information) หัวข้อ IP Address

จากภาพที่ 4.68 คือ Script บางส่วนใน Script สำหรับการตรวจสอบระบบที่ใช้ในการ Monitor Hardware หัวข้อ IP Address จาก Script จะเห็นได้ว่าการใช้ Command ต่างๆเพื่อใช้แสดงรายละเอียด IP Address ที่เครื่อง Server ทำการเชื่อมต่อหรือใช้งานอยู่ หลังจากที่มีการแสดงรายละเอียด IP Address แล้วตัว Script จะให้ผู้ใช้ตัดสินใจว่าผู้ใช้งานต้องการที่จะทำการบันทึกข้อมูลหรือไม่ต้องการบันทึกข้อมูล หากผู้ใช้งานทำการบันทึกข้อมูล ข้อมูลที่ได้จะไปอยู่ในสถานที่ที่ผู้ใช้งานกำหนดเอาไว้โดยชื่อไฟล์ที่ได้คือวันที่_ipaddress.pdf และ วันที่_ipaddress.txt

```

clear;
    echo -e "\e[0;32m'fdisk -l'" | more
    while true
    do
    echo -e "\e[1;32mDo you want to export file [y/n]?\e[0m: \c"
    read yesnno
    if [[ $yesnno == "y" ]]
    then
        if [ -f "/tmp/lofpath.txt" ]
        then
            lofpath=$(cat /tmp/lofpath.txt)
            if [ -f "$lofpath'date +%b-%d-%Y'"_fdisk.txt" ];then
                rm $lofpath'date +%b-%d-%Y'"_fdisk.txt
            fi
            fdisk -l >> $lofpath'date +%b-%d-%Y'"_fdisk.txt
            echo -e "\e[1;34m\c"
            export fdiskz=$lofpath'date +%b-%d-%Y'"_fdisk.txt
            enscrip -p $lofpath'date +%b-%d-%Y'"_fdisk.ps $fdiskz
            echo "convert \"$lofpath'date +%b-%d-%Y'"_fdisk.ps to \"$lofpath'date +%b-%d-%Y'"_fdisk.pdf"
            ps2pdf $lofpath'date +%b-%d-%Y'"_fdisk.ps $lofpath'date +%b-%d-%Y'"_fdisk.pdf
            rm $lofpath'date +%b-%d-%Y'"_fdisk.ps
            break;
        else
            echo -e "\e[0;31mDONT HAVE FILE LOCATION : You should be set file location\e[0m"
            break;
        fi
    elif [[ $yesnno == "n" ]]
    then
        break;
    else
        clear;
        echo -e "\e[0;32m'fdisk -l'" | more
        echo -e "\e[0;31mPlease tell me what is your answer [y/n]..."
    fi
    done
    ;;

```

ภาพที่ 4. 69 ตัวอย่างฟังก์ชันในการ Monitor Hardware (Server Information) หัวข้อ Manipulate Disk Partition Table

จากภาพที่ 4.69 คือ Script บางส่วนใน Script สำหรับการตรวจสอบระบบที่ใช้ในการ Monitor Hardware หัวข้อ Manipulate Disk Partition Table จาก Script จะเห็นได้ว่าการใช้ Command fdisk เพื่อใช้แสดงรายละเอียด Disk หลังจากที่มีการแสดงรายละเอียด Disk แล้วตัว Script จะให้ผู้ใช้ตัดสินใจว่าผู้ต้องการที่จะทำการบันทึกข้อมูลหรือไม่ต้องการบันทึกข้อมูล หากผู้ใช้ทำการบันทึกข้อมูล ข้อมูลที่ได้จะไปอยู่ในสถานที่ที่ผู้ใช้กำหนดเอาไว้โดยชื่อไฟล์ที่ได้คือ วันที่_ipaddress.pdf และ วันที่_ipaddress.txt


```

clear;

echo -e "\e[0;32mDisplay Memory In Megabyte"
echo -e "\e[0;32mfree -m"
free -m | grep "Mem:" | awk '{printf "\033[32mPercent Memory Use: %3.1f%%\n\033[0m", $3*100/$2}'
free -m | grep "Swap:" | awk '{printf "\033[32mPercent Swap Use: %3.1f%%\n\033[0m", $3*100/$2}'
while true

do
echo -e "\e[1;32mDo you want to export file [y/n]? \e[0m: \c"
read yesnno
if [[ $yesnno == "y" ]]
then
    if [ -f "/tmp/lofpath.txt" ]
    then
        lofpath=$(cat /tmp/lofpath.txt)
        if [ -f "$lofpath`date +%b-%d-%Y`_freeusememory.txt" ];then
            rm $lofpath`date +%b-%d-%Y`_freeusememory.txt
        fi

        echo "Display Memory In Megabyte" >> $lofpath`date +%b-%d-%Y`_freeusememory.txt
        free -m >> $lofpath`date +%b-%d-%Y`_freeusememory.txt
        free -m | grep "Mem:" | awk '{printf "Percent Memory Use: %3.1f%%\n", $3*100/$2}' >>
$lofpath`date +%b-%d-%Y`_freeusememory.txt
        free -m | grep "Swap:" | awk 'NR==3{printf "Percent Swap Use: %3.1f%%\n", $3*100/$2}' >> $lofpath`date
+%b-%d-%Y`_freeusememory.txt
        echo -e "\e[1;34m\c"
        export freeusememory=$lofpath`date +%b-%d-%Y`_freeusememory.txt
        enscript -p $lofpath`date +%b-%d-%Y`_freeusememory.ps $freeusememory
        echo "convert \"$lofpath`date +%b-%d-%Y`_freeusememory.ps to \"$lofpath`date +%b-%d-
%Y`_freeusememory.pdf"
        ps2pdf $lofpath`date +%b-%d-%Y`_freeusememory.ps $lofpath`date +%b-%d-%Y`_freeusememory.pdf
        rm $lofpath`date +%b-%d-%Y`_freeusememory.ps
        break;
    else
        echo -e "\e[0;31mDONT HAVE FILE LOCATION : You should be set file location\e[0m"
        break;
    fi
elif [[ $yesnno == "n" ]]
then
    break;
else
    clear;
    echo -e "\e[0;32mDisplay Memory In Megabyte"
    echo -e "\e[0;32mfree -m"
    free -m | grep "Mem:" | awk '{printf "\033[32mPercent Memory Use: %3.1f%%\n\033[0m", $3*100/$2}'
    free -m | grep "Swap:" | awk '{printf "\033[32mPercent Swap Use: %3.1f%%\n\033[0m", $3*100/$2}'
    echo -e "\e[0;31mPlease tell me what is your answer [y/n]..."
fi
done
;;

```

ภาพที่ 4. 70 ตัวอย่างฟังก์ชันในการ Monitor System (System Information) หัวข้อ Free and Use Memory

จากภาพที่ 4.70 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการ Monitor System หัวข้อ Free and Use Memory จาก Script จะเห็นได้ว่าการใช้ Command free -m เพื่อใช้แสดงรายละเอียด Memory หลังจากที่มีการแสดงรายละเอียด Memory แล้วตัว Script จะให้ผู้ใช้ตัดสินใจว่าผู้ต้องการที่จะทำการบันทึกข้อมูลหรือไม่ต้องการบันทึกข้อมูล หากผู้ใช้ทำการบันทึกข้อมูล ข้อมูลที่ได้จะไปอยู่ในสถานที่ที่ผู้ใช้กำหนดเอาไว้โดยชื่อไฟล์ที่ได้คือ วันที่_freeusememory.pdf และ วันที่_freeusememory.txt

```

clear;
    echo -e "\e[0;32m`df -h | more`"
    while true
do
echo -e "\e[1;32mDo you want to export file [y/n]? \e[0m: \c"
read yesnno
if [[ $yesnno == "y" ]]
then
    if [ -f "/tmp/lofpath.txt" ]
    then
    if [ -f "$lofpath`date +%b-%d-%Y`_filesystem.txt" ];then
        rm $lofpath`date +%b-%d-%Y`_filesystem.txt
    fi
    lofpath=$(cat /tmp/lofpath.txt)
    df -h >> $lofpath`date +%b-%d-%Y`_filesystem.txt
    echo -e "\e[1;34m\c"
    export filesystem=$lofpath`date +%b-%d-%Y`_filesystem.txt
    enscript -p $lofpath`date +%b-%d-%Y`_filesystem.ps $filesystem
    echo "convert \"$lofpath`date +%b-%d-%Y`_filesystem.ps to \"$lofpath`date +%b-%d-%Y`_filesystem.pdf"
    ps2pdf $lofpath`date +%b-%d-%Y`_filesystem.ps $lofpath`date +%b-%d-%Y`_filesystem.pdf
    rm $lofpath`date +%b-%d-%Y`_filesystem.ps
    break;
    else
        echo -e "\e[0;31mDONT HAVE FILE LOCATION : You should be set file location\e[0m"
        break;
    fi
elif [[ $yesnno == "n" ]]
then
    break;
else
    clear;
    echo -e "\e[0;32m`df -h | more`"
    echo -e "\e[0;31mPlease tell me what is your answer [y/n]..."
fi
done
;;

```

ภาพที่ 4. 71 ตัวอย่างฟังก์ชันในการ Monitor System (System Information)

จากภาพที่ 4.71 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการ Monitor System หัวข้อ File System Disk Space Usage จาก Script จะเห็นได้ว่าการใช้ Command `df -h` เพื่อใช้แสดงรายละเอียดของ File System หลังจากที่มีการแสดงรายละเอียด File System แล้วตัว Script จะให้ผู้ใช้ตัดสินใจว่าผู้ใช้ต้องการที่จะทำการบันทึกข้อมูลหรือไม่ต้องการบันทึกข้อมูล หากผู้ใช้ทำการบันทึกข้อมูล ข้อมูลที่ได้จะไปอยู่ในสถานที่ที่ผู้ใช้กำหนดเอาไว้โดยชื่อไฟล์ที่ได้คือ `วันที่_filesystem.pdf` และ `วันที่_filesystem.txt`

```

clear;
echo -e "\e[0;32mPercentage CPU\nCPU:" `top -bn1 | grep "Cpu(s)" | awk '{printf
"\033[32m%3.1f%%\033[0m", $2+$4}`
echo -e "\e[0;32mPercentage RAM\nRAM:" `free -m | awk 'NR==2{printf "\033[32m%3.1f%%\033[0m",
$3*100/$2}`
while true
do
echo -e "\e[1;32mDo you want to export file [y/n]? \e[0m: \c"
read yesnno
if [[ $yesnno == "y" ]]
then
if [ -f "/tmp/lofpath.txt" ]
then
lofpath=$(cat /tmp/lofpath.txt)
if [ -f "$lofpath'date +%b-%d-%Y'_cpunram.txt" ];then
rm $lofpath'date +%b-%d-%Y'_cpunram.txt
fi
echo -e "Percentage CPU\nCPU:" `top -bn1 | grep "Cpu(s)" | awk '{printf "%3.1f%%",
$2+$4}` `>> $lofpath'date +%b-%d-%Y'_cpunram.txt
echo -e "Percentage RAM\nRAM:" `free -m | awk 'NR==2{printf "%3.1f%%", $3*100/$2}` `>>
$lofpath'date +%b-%d-%Y'_cpunram.txt
echo -e "\e[1;34m\c"
export cpunram=$lofpath'date +%b-%d-%Y'_cpunram.txt
enscript -p $lofpath'date +%b-%d-%Y'_cpunram.ps $cpunram
echo "convert \"$lofpath'date +%b-%d-%Y'_cpunram.ps to \"$lofpath'date +%b-%d-%Y'_cpunram.pdf"
ps2pdf $lofpath'date +%b-%d-%Y'_cpunram.ps $lofpath'date +%b-%d-%Y'_cpunram.pdf
rm $lofpath'date +%b-%d-%Y'_cpunram.ps
break;
else
echo -e "\e[0;31mDONT HAVE FILE LOCATION : You should be set file location\e[0m"
break;
fi
elif [[ $yesnno == "n" ]]
then
break;
else
clear;
echo -e "\e[0;32mPercentage CPU\nCPU:" `top -bn1 | grep "Cpu(s)" | awk '{printf
"\033[32m%3.1f%%\033[0m", $2+$4}`
echo -e "\e[0;32mPercentage RAM\nRAM:" `free -m | awk 'NR==2{printf "\033[32m%3.1f%%\033[0m",
$3*100/$2}`
echo -e "\e[0;31mPlease tell me what is your answer [y/n]..."
fi
done
;;

```

ภาพที่ 4. 72 ตัวอย่างฟังก์ชันในการ Monitor System (System Information)

จากภาพที่ 4.72 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการ Monitor System หัวข้อ Percentage CPU/RAM จาก Script จะเห็นได้ว่าการใช้ Command top และ free โดยมีการเพิ่มสมการที่ใช้สำหรับคำนวณเป็นเปอร์เซ็นต์ เพื่อใช้แสดงข้อมูลการทำงานของ CPU และ RAM หลังจากที่มีการแสดงข้อมูลการทำงานแล้วตัว Script จะให้ผู้ใช้ตัดสินใจว่าผู้ใช้ต้องการที่จะทำการบันทึกข้อมูลหรือไม่ต้องการบันทึกข้อมูล หากผู้ใช้ทำการบันทึกข้อมูล ข้อมูลที่ได้จะไปอยู่ในสถานที่ที่ผู้ใช้กำหนดเอาไว้โดยชื่อไฟล์ที่ได้คือวันที่_cpunram.pdf และ วันที่_cpunram.txt

```

clear;
echo -e "\e[0;32mSystem Uptime\nStart System: \c"
uptime | awk '{print $1}'
echo -e "\e[0;32mSystem Uptime: \c"
uptime | awk -F( |,|:|+ ' ' '{if ($7=="min") m=$6; else {if ($7~/^day/) {d=$6;h=$8;m=$9} else {h=$6;m=$7}}}'
{print d+0,"days,",h+0,"hours,",m+0,"minutes."}
while true
do
echo -e "\e[1;32mDo you want to export file [y/n]? \c"
read yesnno
if [[ $yesnno == "y" ]]
then
if [ -f "/tmp/lofpath.txt" ]
then
lofpath=$(cat /tmp/lofpath.txt)
if [ -f "$lofpath`date +%b-%d-%Y`_uptime.txt" ];then
rm $lofpath`date +%b-%d-%Y`_uptime.txt
fi

echo -e "System Uptime\nStart System: \c" >> $lofpath`date +%b-%d-%Y`_uptime.txt
uptime | awk '{print $1}' >> $lofpath`date +%b-%d-%Y`_uptime.txt
echo -e "System Uptime: \c" >> $lofpath`date +%b-%d-%Y`_uptime.txt
uptime | awk -F( |,|:|+ ' ' '{if ($7=="min") m=$6; else {if ($7~/^day/) {d=$6;h=$8;m=$9} else
{h=$6;m=$7}}}' {print d+0,"days,",h+0,"hours,",m+0,"minutes."} >> $lofpath`date +%b-%d-%Y`_uptime.txt
echo -e "\e[1;34m\c"
export uptime=$lofpath`date +%b-%d-%Y`_uptime.txt
enscript -p $lofpath`date +%b-%d-%Y`_uptime.ps $uptime
echo "convert \"$lofpath`date +%b-%d-%Y`_uptime.ps to \"$lofpath`date +%b-%d-%Y`_uptime.pdf"
ps2pdf $lofpath`date +%b-%d-%Y`_uptime.ps $lofpath`date +%b-%d-%Y`_uptime.pdf
rm $lofpath`date +%b-%d-%Y`_uptime.ps
break;
else
echo -e "\e[0;31mDONT HAVE FILE LOCATION : You should be set file location \c"
break;
fi
elif [[ $yesnno == "n" ]]
then
break;
else
clear;
echo -e "\e[0;32mSystem Uptime\nStart System: \c"
uptime | awk '{print $1}'
echo -e "\e[0;32mSystem Uptime: \c"
uptime | awk -F( |,|:|+ ' ' '{if ($7=="min") m=$6; else {if ($7~/^day/) {d=$6;h=$8;m=$9} else {h=$6;m=$7}}}'
{print d+0,"days,",h+0,"hours,",m+0,"minutes."}
echo -e "\e[0;31mPlease tell me what is your answer [y/n]..."
fi
done
;;

```

ภาพที่ 4. 73 ตัวอย่างฟังก์ชันในการ Monitor System (System Information) หัวข้อ System Uptime

จากภาพที่ 4.73 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการ Monitor System หัวข้อ System Uptime จาก Script จะเห็นได้ว่าการใช้ Command ต่างๆเพื่อใช้ในการแสดงข้อมูลระยะเวลาที่เครื่อง Server เปิดเครื่องและระยะเวลาระหว่างเครื่อง Server ทำงาน หลังจากที่มีการแสดงข้อมูลแล้วตัว Script จะให้ผู้ใช้ตัดสินใจว่าผู้ใช้ต้องการที่จะทำการบันทึกข้อมูลหรือไม่ ต้องการบันทึกข้อมูล หากผู้ใช้ทำการบันทึกข้อมูล ข้อมูลที่ได้จะไปอยู่ในสถานที่ที่ผู้ใช้กำหนดเอาไว้โดยชื่อไฟล์ที่ได้คือวันที่_uptime.pdf และ วันที่_uptime.txt

```

if [ -f "/tmp/lobckpath.txt" ]
then
    lobckpath=$(cat /tmp/lobckpath.txt)
    echo -e "\e[1;32mNow your file location is $lobckpath"
    echo -e "\e[1;32mEnter your location (Ex./home/user/Backup_location)"
    echo -e "\e[1;32mIf you want to go back enter [q]: \e[0m\c"
    read setbckpath
    if [[ $setbckpath == "q" ]]
    then
        echo -e "\e[1;32mBack to Backup Log File Menu"
        elif [[ $setbckpath = "" ]];then
        echo -e "\e[1;32mBack to Backup Log File Menu"
        else
            if [[ -d "$setbckpath" ]]
            then
                echo -e "\e[1;32mYour location backup log file: $setbckpath"
                echo "$setbckpath/" > /tmp/lobckpath.txt
            else
                echo -e "\e[0;31mDONT HAVE DIRECTORY : Please try agian later\e[0m"
            fi
        fi
    else
        echo -e "\e[1;32mYou dont have file location"
        echo -e "\e[1;32mEnter your location (Ex./home/user/Backup_location)"
        echo -e "\e[1;32mIf you want to go back enter [q]: \e[0m\c"
        read setbckpath
        if [[ $setbckpath == "q" ]]
        then
            echo -e "\e[1;32mBack to Backup Log File Menu"
            elif [[ $setbckpath = "" ]];then
            echo -e "\e[1;32mBack to Backup Log File Menu"
            else
                if [[ -d "$setbckpath" ]]
                then
                    echo -e "\e[1;32mYour location backup log file: $setbckpath"
                    echo "$setbckpath/" > /tmp/lobckpath.txt
                else
                    echo -e "\e[0;31mDONT HAVE DIRECTORY : Please try agian later\e[0m"
                fi
            fi
        fi
    fi
;;

```

ภาพที่ 4. 74 ตัวอย่างฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup (Backup Log File) หัวข้อ Set Backup Log File Location

จากภาพที่ 4.74 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการเข้าไปดูไฟล์ข้อมูลที่ Backup ในหัวข้อ Set Backup Log File Location จาก Script จะเห็นได้ว่าเป็นฟังก์ชันสำหรับการกำหนดสถานที่ที่มีไฟล์ Backup อยู่ หลังจากทำการกำหนดสถานที่ที่มีไฟล์ Backup อยู่ ตัว Script จะทำการสร้างไฟล์สกุล txt ที่ชื่อ lobckpath.txt (สถานที่เก็บไฟล์ lobckpath.txt อยู่ที่ /tmp/lobckpath.txt) เอาไว้เพื่อนำไว้ใช้เก็บตำแหน่งสถานที่ที่มีไฟล์ Backup อยู่

```

if [ -f "/tmp/lobckpath.txt" ]
then
    bckpath=$(cat /tmp/lobckpath.txt)
    bcktoday=$(cat $bckpath | date +%b-%d-%Y)*
    view $bcktoday
else
    echo -e "\e[0;31mCANT FIND DIRECTORY : You should be set file location"
fi
;;

```

ภาพที่ 4. 75 ตัวอย่างฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup (Backup Log File) หัวข้อ View Backup Log File (Today)

```

if [ -f "/tmp/lobckpath.txt" ]
then
    bckpath=$(cat /tmp/lobckpath.txt)
    dateyes=$(date -d "yesterday" +%b-%d-%Y)*
    view $bckpath$dateyes
else
    echo -e "\e[0;31mCANT FIND DIRECTORY : You should be set file location"
fi

```

ภาพที่ 4. 76 ตัวอย่างฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup (Backup Log File) หัวข้อ View Backup Log File (Yesterday)

```

if [ -f "/tmp/lobckpath.txt" ]
then
    bckpath=$(cat /tmp/lobckpath.txt)
    ls "$bckpath"
    echo -e "\e[1;32mEnter Backup Log File Name"
    echo -e "\e[1;32mIf you want to go back enter [q]: \e[0m\c"
    read bckname
    if [[ $bckname == "q" ]]
    then
        echo -e "\e[1;32mBack to Backup Log File Menu"
        elif [[ $bckname = "" ]];then
        echo -e "\e[1;32mBack to Backup Log File Menu"
        else
            view $bckpath$bckname
        fi
    else
        echo -e "\e[0;31mCANT FIND DIRECTORY : You should be set file location"
    fi
fi
;;

```

ภาพที่ 4. 77 ตัวอย่างฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup (Backup Log File) หัวข้อ View Backup Log File (History)

จากภาพที่ 4.75 ภาพที่ 4.76 และ ภาพที่ 4.77 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการเข้าไปดูไฟล์ข้อมูลที่ Backup ในหัวข้อ View Backup Log File (Today), View Backup Log File (Yesterday) และ View Backup Log File (History) ตามลำดับ โดยจะเห็นได้ว่าจาก

Script ของภาพทั้ง 3 ภาพที่ 4.75 เป็นการดูไฟล์ Backup ของวันที่ปัจจุบัน ภาพที่ 4.76 ดูไฟล์ Backup ของวันเมื่อวาน และภาพที่ 4.77 ดูไฟล์ Backup ตามชื่อที่ผู้ใช้ต้องการกำหนด

```
if [ -f "/tmp/lobckpath.txt" ]
then
    bckpath=$(cat /tmp/lobckpath.txt)
    ls "$bckpath"
    echo -e "\e[1;32mEnter Backup Log File Name"
    echo -e "\e[1;32mIf you want to go back enter [q]: \e[0m\c"
    read bckname
    if [[ $bckname == "q" ]]
    then
        echo -e "\e[1;32mBack to Backup Log File Menu"
    elif [[ $bckname = "" ]];then
        echo -e "\e[1;32mBack to Backup Log File Menu"
    else
        DateGrep=$(head $bckpath$bckname | head -n1 | awk '{print $2}')
        strtDate=$(head $bckpath$bckname | grep $DateGrep | head -n1 | awk '{print $4}')
        stpDate=$(tail $bckpath$bckname | grep $DateGrep | tail -n1 | awk '{print $4}')
        StartDate=$(date -u -d "$strtDate" +"%s")
        FinalDate=$(date -u -d "$stpDate" +"%s")
        echo -e "\e[0;32mStart: $strtDate"
        echo -e "\e[0;32mStop: $stpDate"
        echo -e "\e[0;32mDifferent Time: \c" && date -u -d "0 $FinalDate sec - $StartDate sec"
        + "%H:%M:%S"
        echo -e "\e[1;33m"
        cat $bckpath$bckname | grep 'ERROR|error'
    fi
else
    echo -e "\e[0;31mCANT FIND DIRECTORY : You should be set directory file"
fi
;
```

ภาพที่ 4. 78 ตัวอย่างฟังก์ชันในการเข้าไปดูไฟล์ข้อมูลที่ Backup (Backup Log File) หัวข้อ View Backup Log File (Difference Time)

จากภาพที่ 4.78 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการเข้าไปดูไฟล์ข้อมูลที่ Backup ในหัวข้อ View Backup Log File (Difference Time) จาก Script จะเห็นได้ว่าเป็นฟังก์ชันสำหรับการคำนวณระยะเวลาที่ทำการ Backup ไฟล์ข้อมูลและจะบอกข้อผิดพลาดที่เกิดขึ้นระหว่างการ Backup ไฟล์ข้อมูล

```

if [ -f "/tmp/losarpathdata.txt" ]
then
    losarpath=$(cat /tmp/losarpathdata.txt)
    echo -e "\e[1;32mNow your file location is $losarpath"
    echo -e "\e[1;32mEnter your location (Ex./home/user/location)"
    echo -e "\e[1;32mIf you want to go back enter [q]: \e[0m\c"
    read setlosarpath
    if [[ $setlosarpath == "q" ]]
    then
        echo -e "\e[1;32mBack to Generate Monthly Report Menu"
        elif [[ $setlosarpath = "" ]];then
        echo -e "\e[1;32mBack to Generate Monthly Report Menu"
    else
        if [[ -d "$setlosarpath" ]]
        then
            echo -e "\e[1;32mYour file location: $setlosarpath"
            echo "$setlosarpath/" > /tmp/losarpathdata.txt
            else
            echo -e "\e[0;31mDONT HAVE DIRECTORY : Please try agian later\e[0m"
        fi
    fi
else
    echo -e "\e[1;32mYou dont have file location"
    echo -e "\e[1;32mEnter your location (Ex./home/user/location)"
    echo -e "\e[1;32mIf you want to go back enter [q]: \e[0m\c"
    read setlosarpath
    if [[ $setlosarpath == "q" ]]
    then
        echo -e "\e[1;32mBack to Generate Monthly Report Menu"
        elif [[ $setlosarpath = "" ]];then
        echo -e "\e[1;32mBack to Generate Monthly Report Menu"
    else
        if [[ -d "$setlosarpath" ]]
        then
            echo -e "\e[1;32mYour file location: $setlosarpath"
            echo "$setlosarpath/" > /tmp/losarpathdata.txt
            else
            echo -e "\e[0;31mDONT HAVE DIRECTORY : Please try agian later\e[0m"
        fi
    fi
fi
;;

```

ภาพที่ 4. 79 ตัวอย่างฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป
(Generate Monthly Report) หัวข้อ Set Data File Location

จากภาพที่ 4.79 คือ Script บางส่วนใน Script สำหรับ Monitor ที่ใช้ในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุปในหัวข้อ Set Data File Location จาก Script จะเห็นได้ว่าเป็นฟังก์ชันสำหรับการกำหนดสถานที่ที่จะนำไฟล์ข้อมูลไปเก็บ ที่ได้จากการสร้างไฟล์ข้อมูลที่เก็บค่าสถานะของระบบ จาก Script จะเห็นได้ว่าหลังจากทำการกำหนดสถานที่ Script จะทำการสร้างไฟล์สกุล txt ที่ชื่อ losarpathdata.txt (สถานที่เก็บไฟล์ losarpathdata.txt อยู่ที่ /tmp/losarpathdata.txt) เอาไว้เพื่อนำไว้ใช้เก็บตำแหน่งสถานที่ที่จะนำไฟล์ข้อมูลไปเก็บ

```

if [ -f "/tmp/losarpath.txt" ]
then
    losarpath=$(cat /tmp/losarpath.txt)
    echo -e "\e[1;32mNow your file location is $losarpath"
    echo -e "\e[1;32mEnter your location (Ex./home/user/location)"
    echo -e "\e[1;32mIf you want to go back enter [q]: \e[0m\c"
    read setlosarpath
    if [[ $setlosarpath == "q" ]]
    then
        echo -e "\e[1;32mBack to Generate Monthly Report Menu"
        elif [[ $setlosarpath = "" ]];then
            echo -e "\e[1;32mBack to Generate Monthly Report Menu"
        else
            if [[ -d "$setlosarpath" ]]
            then
                echo -e "\e[1;32mYour file location: $setlosarpath"
                echo "$setlosarpath/" > /tmp/losarpath.txt
            else
                echo -e "\e[0;31mDONT HAVE DIRECTORY : Please try agian later\e[0m"
            fi
        fi
    fi
else
    echo -e "\e[1;32mYou dont have file location"
    echo -e "\e[1;32mEnter your location (Ex./home/user/location)"
    echo -e "\e[1;32mIf you want to go back enter [q]: \e[0m\c"
    read setlosarpath
    if [[ $setlosarpath == "q" ]]
    then
        echo -e "\e[1;32mBack to Generate Monthly Report Menu"
        elif [[ $setlosarpath = "" ]];then
            echo -e "\e[1;32mBack to Generate Monthly Report Menu"
        else
            if [[ -d "$setlosarpath" ]]
            then
                echo -e "\e[1;32mYour file location: $setlosarpath"
                echo "$setlosarpath/" > /tmp/losarpath.txt
            else
                echo -e "\e[0;31mDONT HAVE DIRECTORY : Please try agian later\e[0m"
            fi
        fi
    fi
fi
;;

```

ภาพที่ 4.80 ตัวอย่างฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป
(Generate Monthly Report) หัวข้อ Set File sar.txt Location

จากภาพที่ 4.80 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุปในหัวข้อ Set File sar.txt Location จาก Script จะเห็นได้ว่าเป็นฟังก์ชันสำหรับการกำหนดสถานที่ที่ทำการเก็บไฟล์สกุล txt เอาไว้เพื่อนำไปใช้ในหัวข้ออื่นๆ จาก Script จะเห็นได้ว่าหลังจากทำการกำหนดสถานที่ Script จะทำการสร้างไฟล์สกุล txt ที่ชื่อ losarpathdata.txt (สถานที่เก็บไฟล์ losarpathdata.txt อยู่ที่ /tmp/losarpathdata.txt) เอาไว้เพื่อนำไว้ใช้เก็บตำแหน่งสถานที่ที่จะนำไฟล์ข้อมูลไปเก็บ

```

if [ -f "/tmp/losarpathdata.txt" ]
then
    losarpathdata=$(cat /tmp/losarpathdata.txt)
    stat_date=`date | awk '{print $2 "-" $3 "-" $6}'`
    stat_loc=$losarpathdata
    export stat_loc stat_date
    /usr/bin/sar -o $stat_loc/$stat_date 60 605 >/dev/null 2>&1 &
else
    echo -e "\e[0;31mCANT FIND DIRECTORY : You should be set file location"
fi
;;

```

ภาพที่ 4. 81 ตัวอย่างฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป
(Generate Monthly Report) หัวข้อ Get Data from “sar”

จากภาพที่ 4.81 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุปในหัวข้อ Get Data from “sar” จาก Script จะเห็นได้ว่าเป็นฟังก์ชันสำหรับการสร้างไฟล์ข้อมูลที่เก็บค่าสถานะของระบบ โดยข้อมูลที่ได้อาจจะไปเก็บในสถานที่ที่ผู้ใช้กำหนดในหัวข้อ Set Data File Location ซึ่งข้อมูลที่ได้อาจจะเป็นข้อมูลดิบที่มีการเข้ารหัสเอาไว้

```

while true
do
    if [ -f "/tmp/losarpath.txt" ] || [ -f "/tmp/losarpathdata.txt" ]
    then
        losarpath=$(cat /tmp/losarpath.txt)
        losarpathdata=$(cat /tmp/losarpathdata.txt)
        dformat=`date +%Y`
        echo -e "\e[1;32mIf you want to go back enter [q]: \e[0m"
        echo -e "\e[1;32mEnter what month you want to change data to text file (Ex.Jan,jun,etc): \e[0m\c"
        read mth
        if [[ $mth == "q" ]]
        then
            echo -e "\e[1;32mBack to Generate Monthly Report Menu"
            break;
        elif [[ $mth == "" ]];then
            echo -e "\e[1;32mBack to Generate Monthly Report Menu"
            break;
        fi
        clear
        for i in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 23 24 25 26 27 28 29 30 31; do sar -r -u -f $losarpathdata$mth-${i}-2018 > $losarpath$mth-${i}.txt;done
        else
            echo -e "\e[0;31mCANT FIND DIRECTORY : You should be set file location"
            break;
        fi
    done
;;

```

ภาพที่ 4. 82 ตัวอย่างฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป
(Generate Monthly Report) หัวข้อ Change Data to Text File

จากภาพที่ 4.82 คือ Script บางส่วนใน Script สำหรับการ Monitor ที่ใช้ในการสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุปในหัวข้อ Change Data to Text File จาก Script จะเห็นได้ว่าเป็นฟังก์ชันสำหรับการแปลงไฟล์ดิบเป็นไฟล์สกุล txt โดยต้องกำหนดวันเวลาของไฟล์ข้อมูล โดยจะมีวันที่เริ่มและวันล่าสุดที่ต้องการและนำไปสร้างไฟล์ข้อมูลที่ได้จากหัวข้อ Get Data from “sar”

```

if [ -f "/tmp/losarpath.txt" ]
then
while true
do
echo -e "\e[1;32mIf you want to go back enter [q]: \e[0m"
echo -e "\e[1;32mStart (Ex.Jun-25-2018)\e[0m"
read strdte
if [[ $strdte == "q" ]]
then
echo -e "\e[1;32mBack to Generate Monthly Report Menu"
break;
elif [[ $strdte == "" ]];then
echo -e "\e[1;32mBack to Generate Monthly Report Menu"
break;
fi
echo -e "\e[1;32mEnd (Ex.Jul-30-2018)\e[0m"
read enddte
if [[ $enddte == "q" ]]
then
echo -e "\e[1;32mBack to Generate Monthly Report Menu"
break;
elif [[ $enddte == "" ]];then
echo -e "\e[1;32mBack to Generate Monthly Report Menu"
break;
fi
startdate=$(date -l -d "$strdte" ) || exit -1
enddate=$(date -l -d "$enddte + 1 day") || exit -1
losarpath=$(cat /tmp/losarpath.txt)
d=$startdate
if [ `date -d $strdte +%s` -ge `date -d $enddte +%s` ]
then
echo -e "\e[1;32m\"End\" should be greater than \"Start\"\\nPlease enter \"Start\" and \"End\" date again"
echo -e "\e[1;32mEnter return to continue...\e[0m\c"
read inp
clear
elif [ `date -d $strdte +%s` -le `date -d $enddte +%s` ];then
if [ -f $losarpath"sum_cpu.txt" ] || [ -f $losarpath"sum_mem.txt" ];then
rm $losarpath"sum_cpu.txt"
rm $losarpath"sum_mem.txt"
fi
while [ "$d" != "$enddate" ]; do
dformat=`date -d "$d" +%b-%d-%Y`
d=$(date -l -d "$dformat + 1 day")
dformat_use=`date -d "$dformat" +%b-%d`
dformat_use1=`date -d "$dformat" +%b`
echo $dformat
grep Average $losarpath$dformat_use1*.txt | egrep -v '[0-9]{4}' | sed 's/:Average/' | grep $dformat_use >>
$losarpath"sum_cpu.txt
grep Average $losarpath$dformat_use1*.txt | egrep '[0-9]{4}' | sed 's/:Average/' | grep $dformat_use >>
$losarpath"sum_mem.txt"
done
break
fi
done
else
echo -e "\e[0;31mCANT FIND DIRECTORY : You should be set file location"
fi
;;

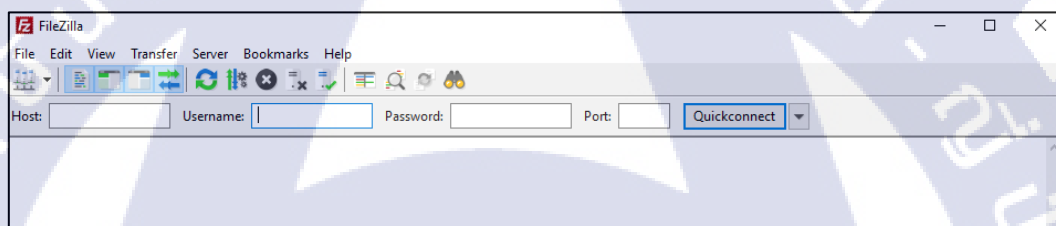
```

ภาพที่ 4. 83 ตัวอย่างฟังก์ชันในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป

จากภาพที่ 4.83 คือ Script บางส่วนใน Script สำหรับการตรวจสอบระบบที่ใช้ในการเข้าไปสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุปในหัวข้อ Summary Report จาก Script จะเห็นได้ว่าเป็นฟังก์ชันสำหรับการสร้างไฟล์ข้อมูลเพื่อเก็บค่าสถานะระบบแบบสรุป โดยผู้ใช้งานจะต้องกำหนดเดือนเพื่อนำไฟล์ข้อมูลในหัวข้อ Change Data to Text File มารวมกันเป็นไฟล์สรุปผล 2 ไฟล์ ชื่อ sum_cpu (การใช้งาน CPU) และ sum_mem (การใช้งาน Memory)

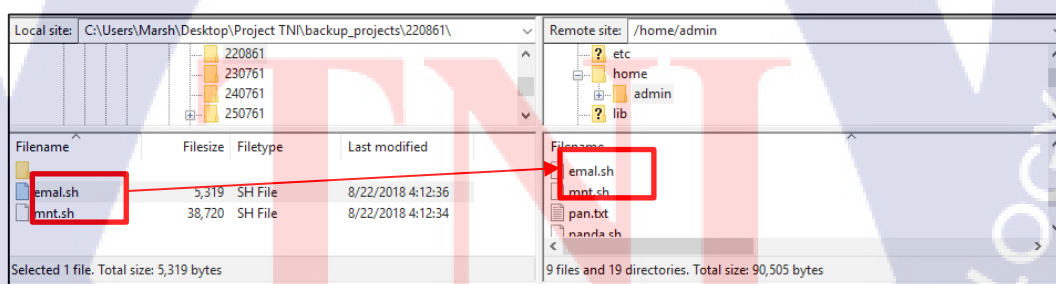
การติดตั้ง Script สำหรับการแจ้งเตือน Email

- ให้ผู้ใช้เปิดโปรแกรม FileZilla สำหรับการนำ Script ไปไว้บนเครื่อง Server
- หลังจากที่ใช้เปิดโปรแกรม FileZilla ให้ผู้ใช้เชื่อมต่อเข้ากับเครื่อง Server โดยการกำหนด Host โดย IP Address และ Username/Password และทำการกำหนด Port เป็น 22 (Port 22 คือ Port ที่ใช้ในการรับส่งข้อมูลระหว่างเครื่องผู้ใช้งานกับเครื่อง Server)



ภาพที่ 4.84 ตัวอย่างโปรแกรม FileZilla

- นำ Script ชื่อ emal.sh ไปวางบนเครื่อง Server โดยการลาก Script ไปฝั่ง Server



ภาพที่ 4.85 ตัวอย่างการโอนย้ายข้อมูล

- หลังจากที่ใช้ผู้นำ Script ไปวางบนเครื่อง Server ให้ผู้ใช้ทำการเปลี่ยนแปลงการเข้าถึงของ Script (Permission) โดยใช้ Command chmod 755 mnt.sh หรือ chmod +x emal.sh ในส่วนที่ Script ถูกวางเอาไว้

- หลังจากนั้นให้ผู้ใช้ทำการติดตั้งแพ็คเกจ postfix, cyrus-sasl-plain และ mailx โดยใช้ Command `yum -y install postfix cyrus-sasl-plain mailx`
- หลังจากทำการติดตั้งแพ็คเกจให้ผู้ใช้ทำการ restart service postfix โดยใช้ Command `systemctl restart postfix` และใช้ Command เพื่อเปิด postfix ทุกครั้งที่ระบบหรือเครื่อง Server ทำงาน โดยใช้ Command `system enable postfix`
- หลังจากทำการ restart ให้ผู้ใช้ทำการเข้าไปแก้ไขไฟล์ข้อมูลโดยสามารถแก้ไขได้ที่ `/etc/postfix/main.cf` โดยผู้ใช้สามารถใช้ Command `vi /etc/postfix/main.cf` หลังจากผู้ใช้เข้ามาภายในไฟล์ข้อมูลให้ผู้ใช้พิมพ์ “i” เพื่อเป็นการเปิดให้ผู้ใช้แก้ไขไฟล์ข้อมูลได้ โดยหลังจากที่ผู้ใช้สามารถแก้ไขไฟล์ข้อมูลได้แล้วให้ผู้ใช้พิมพ์ตามในภาพที่ 3 หลังจากแก้ไขเสร็จแล้วให้ผู้ใช้กด Esc และกดปุ่มเครื่องหมาย : แล้วตามด้วย `wq` จะเป็นการบันทึกและออกจากการแก้ไข

```
myhostname = hostname.example.com

relayhost = [smtp.gmail.com]:587
smtp_use_tls = yes
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_tls_CAfile = /etc/ssl/certs/ca-bundle.crt
smtp_sasl_security_options = noanonymous
smtp_sasl_tls_security_options = noanonymous
```

ภาพที่ 4.86 ตัวอย่างที่เพิ่มลงไปไฟล์ข้อมูล main.cf

- หลังจากทำการแก้ไขไฟล์ข้อมูลเสร็จให้ผู้ใช้สร้างไฟล์และแก้ไข `sasl_passwd` ที่ `/etc/postfix/` โดยใช้ Command `vi /etc/postfix/sasl_passwd` หลังจากเข้าไปในไฟล์ให้ผู้ใช้พิมพ์ตามภาพที่ 4 โดยในภาพที่ 4 คือการกำหนด Email ผู้ส่งโดยในภาพจะเป็นการกำหนด Port SMTP สำหรับ GMAIL โดยจะเป็น Port 587 ผู้ใช้สามารถเปลี่ยนจากการใช้ GMAIL เป็น HOTMAIL ได้โดยสามารถหา Port ได้ตามเว็บไซต์ HOTMAIL โดยตรง

```
[smtp.gmail.com]:587 username:password
```

ภาพที่ 4. 87 ตัวอย่างที่เพิ่มลงไปไฟล์ข้อมูล sasl_passwd

- หลังจากการกำหนด Email ผู้ส่งเรียบร้อยแล้วให้ผู้ใช้ทำการใช้ Command postmap และ

```
postmap /etc/postfix/sasl_passwd
chown root:postfix /etc/postfix/sasl_passwd*
chmod 640 /etc/postfix/sasl_passwd*
systemctl reload postfix
```

เปลี่ยนสิทธิการเข้าถึงไฟล์ sasl_passwd หลังจากนั้นทำการรีโหลด โดย Command ที่ใช้ในการทำคือ

- จากนั้นผู้ใช้จะสามารถใช้งาน Script ได้โดยใช้ Command crontab -e

การใช้งาน Shell Script สำหรับการแจ้งเตือน Email

ในการใช้ Script สำหรับแจ้งเตือน Email ผู้ใช้สามารถแก้ไขค่าที่กำหนดไว้ในตอนแรกคือในส่วนของการตรวจสอบข้อมูล Memory ตรวจสอบการใช้งาน Hard Disk และการทำงานของ CPU รวมถึงผู้รับการแจ้งเตือน Email โดยค่าที่กำหนดเอาไว้ในตอนแรกของการแจ้งเตือนคือ 0 และ Email ที่กำหนดได้ทำการคอมเมนต์เอาไว้ ผู้ใช้สามารถแก้ไขได้โดยใช้ Command vi เพื่อไปแก้ไขข้อมูลใน Script ดังนี้

- sentto="" คือการกำหนดค่าผู้รับ Email
- CPUTH=0 คือเปอร์เซ็นต์ที่กำหนดไว้ว่า CPU สามารถทำงานหนักได้กี่เปอร์เซ็นต์ก่อนจะทำการแจ้งเตือน
- SWAPTH=0 คือเปอร์เซ็นต์ที่กำหนดไว้ว่าสวอป (SWAP) สามารถใช้งานได้ถึงกี่เปอร์เซ็นต์ก่อนจะทำการแจ้งเตือน
- MEMTH=0 คือเปอร์เซ็นต์ที่กำหนดไว้ว่า Memory สามารถใช้งานได้ถึงกี่เปอร์เซ็นต์ก่อนจะทำการแจ้งเตือน
- DISKU=0 คือเปอร์เซ็นต์ที่กำหนดไว้ว่า Hard Disk สามารถใช้งานได้ถึงกี่เปอร์เซ็นต์ก่อนจะทำการแจ้งเตือน

```
#email to ..@examplemail.com
sentto=""
#threshold for cpu
export CPUTH=0
#threshold for swap
export SWAPTH=0
#threshold for memory
export MEMTH=0
#threshold for disk(use)
export DISKU=0
```

ภาพที่ 4. 88 ตัวอย่างที่เพิ่มลงไปไฟล์ข้อมูล sasl_passwd

โดยข้างต้นหากผู้ใช้ไม่สามารถหาเจอสามารถใช้ Command หลังจากการเข้าไปแก้ไขเพื่อค้นหาได้ โดยใช้ Command / ตามด้วยค่าที่ผู้ใช้ต้องการค้นหา เช่น /sentto หรือ /CPUTH เป็นต้น

หลังจากทำการกำหนดค่าภายใน Script สำหรับการแจ้งเตือน Email แล้วให้ผู้ใช้ทำการใช้ Command crontab -e โดยหลังจากนั้นผู้ใช้จะเข้าสู่หน้าต่างที่สามารถแก้ไขข้อมูลภายใน Command crontab -e โดยให้ผู้ใช้แก้ไขเป็น * * * * * /สถานที่ที่ผู้ใช้เก็บ Script สำหรับการแจ้งเตือน Email เอาไว้ เช่น */30 * * * * /admin/home/ema.sh ในที่นี้หมายถึงให้ Script ทำงานทุกๆ 30 นาที

- * ตัวแรกหมายถึง นาที (0-59)
- * ตัวที่ 2 หมายถึง ชั่วโมง (0-23)
- * ตัวที่ 3 หมายถึง วันในหนึ่งเดือน (1-31)
- * ตัวที่ 4 หมายถึง เดือนในหนึ่งปี (1-12)
- * ตัวที่ 5 หมายถึง วันในหนึ่งสัปดาห์ (0-6) (วันอาทิตย์ = 0)



ภาพที่ 4. 89 ตัวอย่างการกำหนดค่า crontab -e

ตัวอย่าง Script สำหรับการแจ้งเตือน Email

```
CPU=$(vmstat 1 1 | tail -1 | awk '{print $13+$14}')
if [ $CPU -gt $CPUTH ]
then
    echo "<h3>CPU Threshold : $CPUTH percent on $SERVER</h3>
    <table frame='box' bgcolor='#E1E1E0' align='left' width='500px'>
    <tr bgcolor='#E1E1E0'><td><b>Status</b></td></tr>
    <tr bgcolor='white'><td>CPU exceeded $CPU us+sy percent</td></tr>
    </table><br><br><br><br><br><br><b>on $SERVER</b><br><br>`date`</br></b>" |
    mailx -s "$(echo -e "CPU alert for $SERVER(${CPU}%)\"Content-Type: text/html")" -r "user"
    $sentto
fi
```

ภาพที่ 4. 90 ตัวอย่างฟังก์ชันตรวจสอบ CPU และส่ง Email ด้วย HTML

จากภาพที่ 6 คือ Script บางส่วนของ Script สำหรับการแจ้งเตือน Email ในส่วนของฟังก์ชันที่ใช้ในการแจ้งเตือนเมื่อ CPU ทำงานหนักมากกว่าค่าที่กำหนดเอาไว้ โดยฟังก์ชันที่ใช้ในการแจ้งเตือน CPU จะมีความใกล้เคียงกับฟังก์ชันที่ใช้ในการแจ้งเตือนอื่นๆใน Script สำหรับ Script แจ้งเตือน CPU จะเป็นการร่วม Shell Script เข้ากับ HTML เพื่อให้ Email มีความน่าสนใจมากยิ่งขึ้น ซึ่ง HTML สามารถออกแบบหรือตกแต่ง Email ในแบบต่างๆ เหมือนกับฟังก์ชันอื่นๆใน Script

```
SWAP=$(free -m | grep "Swap:" | awk '{printf "%3.0f", $3*100/$2}')
if [ $SWAP -gt $SWAPTH ]
then
    echo -e "<h2>Swap Threshold : $SWAPTH percent</h2><br><br>
    <table frame='box' bgcolor='#E1E1E0' align='left' width='500px'>
    <tr bgcolor='#E1E1E0'><td><b>Status</b></td><td></td></tr>
    <tr bgcolor='white'><td>Swap exceeded $SWAP percent</td></tr>
    <tr bgcolor='white'><td>Free Swap : `free -m | grep 'Swap:' | awk '{print $4}'`
    megabytes</td></tr>
    </table><br><br><br><br><br><br><br><br><br><br><br>
    <br><b>on $SERVER</b><br><br>`date`</br></b>" | mailx -s "$(echo -e "Swap alert for
    $SERVER (${SWAP}%)\"Content-Type: text/html")" -r "user" $sentto
fi
```

ภาพที่ 4. 91 ตัวอย่างฟังก์ชันตรวจสอบสวอปและส่ง Email ด้วย HTML

จากภาพที่ 7 คือ Script บางส่วนของ Script สำหรับการแจ้งเตือน Email ในส่วนของฟังก์ชันที่ใช้ในการแจ้งเตือนเมื่อสวอป มีการใช้งานมากกว่าค่าที่กำหนดเอาไว้ โดยฟังก์ชันที่ใช้ในการแจ้งเตือนสวอปจะมีความใกล้เคียงกับฟังก์ชันที่ใช้ในการแจ้งเตือนอื่นๆใน Script สำหรับ Script แจ้งเตือนสวอปจะเป็นการร่วม Shell Script เข้ากับ HTML เพื่อให้ Email มีความน่าสนใจ

มากยิ่งขึ้น ซึ่ง HTML สามารถออกแบบหรือตกแต่ง Email ในแบบต่างๆ เหมือนกับฟังก์ชันอื่นใน Script

```
MEM=$(free -m | grep "Mem:" | awk '{printf "%.3f", $3*100/$2}')
if [ $MEM -gt $MEMTH ]
then
    echo -e "
        <h3>Memory Threshold : $MEMTH percent on $SERVER</h3><br></br>
        <table frame="box" bgcolor="#E1E1E0" align="left" width="500px">
        <tr bgcolor="#E1E1E0"><td><b>Status</b></td><td></td></tr>
        <tr bgcolor="white"><td>Memory exceeded $MEM percent</td></tr>
        <tr bgcolor="white"><td>Free Memory : `free -m | grep "Mem:" | awk '{print $4}'`
        megabytes</td></tr>
        </table><br></br><br></br><br></br><br></br><br><b>on
        $SERVER</br><br>`date`</br></b>" | mail -s "$(echo -e "Memory alert for
        $SERVER($MEM)%")\nContent-Type: text/html)" -r "user" $sentto
    fi
```

ภาพที่ 4. 92 ตัวอย่างฟังก์ชันตรวจสอบ Memory และส่ง Email ด้วย HTML

จากภาพที่ 8 คือ Script บางส่วนของ Script สำหรับการแจ้งเตือน Email ในส่วนของฟังก์ชันที่ใช้ในการแจ้งเตือนเมื่อ Memory มีการใช้งานมากกว่าค่าที่กำหนดเอาไว้ โดยฟังก์ชันที่ใช้ในการแจ้งเตือน Memory จะมีความใกล้เคียงกับฟังก์ชันที่ใช้ในการแจ้งเตือนอื่นๆ ใน Script สำหรับ Script แจ้งเตือน Memory จะเป็นการรวม Shell Script เข้ากับ HTML เพื่อให้ Email มีความน่าสนใจมากยิ่งขึ้น ซึ่ง HTML สามารถออกแบบหรือตกแต่ง Email ในแบบต่างๆ เหมือนกับฟังก์ชันอื่นใน Script

```

file=/tmp/disk.pdf
export i=`df -h | tail -n +2 | wc -l`
export x=0
export y=1
export z=false
while [[ $x -le $i ]]
do
    x=$((x+1))
    ch=`df -h | awk "NR==$x"{'print $5;exit'} | sed 's/%/'`
    if [[ $ch -gt $DISKU ]]
    then
        while [ $y = 1 ]
        do
            y=$((y+1))
            echo -e "Use Disk Threshold: $DISKU percent on $SERVER\n`df -h | awk
            "NR==1"{'print $0}'" > /tmp/disk.txt
            done
            echo "`df -h | awk "NR==$x"{'print $0}'" >> /tmp/disk.txt
            z=true
        fi
    done
    if [ $z == "true" ];then
        echo -e "\nAll of Filesystem" >> /tmp/disk.txt
        df -h >> /tmp/disk.txt
        export DISK=/tmp/disk.txt
        enscript -p /tmp/disk.ps $DISK
        ps2pdf /tmp/disk.ps /tmp/disk.pdf
        rm /tmp/disk.ps
        echo "remove /tmp/disk.ps"
        rm /tmp/disk.txt
        echo -e "Use Disk Threshold : $DISKU percent on $SERVER\n`date`" | mailx -a "$file" -s
        "Disk alert for $SERVER (Attachment File)" -r "user" $sentto
        rm /tmp/disk.pdf
    fi
done

```

ภาพที่ 4. 93 ตัวอย่างฟังก์ชันตรวจสอบ Hard Disk และส่ง Email พร้อมแนบไฟล์

จากภาพที่ 9 คือ Script บางส่วนของ Script สำหรับการแจ้งเตือน Email ในส่วนของฟังก์ชันที่ใช้ในการแจ้งเตือนเมื่อ Hard Disk มีการใช้งานมากกว่าค่าที่กำหนดเอาไว้ โดยฟังก์ชันตรวจสอบฮาร์ดดิสก์จะมีความซับซ้อนมากกว่าฟังก์ชันอื่นเนื่องจากก่อนที่จะได้ผลลัพธ์ที่แสดงออกมามีการตรวจสอบซ้ำหลายครั้ง สำหรับ Script ตรวจสอบ Hard Disk จะเป็นการส่ง Email พร้อมแนบไฟล์ pdf เนื่องจากไม่สามารถดึง HTML เข้ามาร่วมใช้งานได้จึงทำให้ Email ไม่มีความน่าสนใจและอาจทำให้ผู้ใช้คิดว่าเป็น Email แคม จึงใช้วิธีการแนบไฟล์ข้อมูลแทน

4.2 ผลการวิเคราะห์ข้อมูล

จากการที่ได้จัดทำโปรเจก โดยงานที่ได้รับในการฝึกงานเป็นงานด้านการเขียน Shell Script เพื่อใช้ในการตรวจสอบเชิงป้องกันระบบ จึงได้ออกมาเป็น Script Monitor และ Script Email Alert ซึ่งเป็น Script ที่ผู้จัดทำได้ทำขึ้นด้วยตนเอง จากการศึกษาและเขียน Shell Script ทำให้ออกมาเป็น Script

4.3 วิเคราะห์ข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์และจุดมุ่งหมายการปฏิบัติงานหรือการจัดทำโครงการ

เมื่อทำการวิเคราะห์และเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์ในการปฏิบัติงานแล้ว จะเห็นว่าความรู้ที่ได้จากการศึกษาหาข้อมูลสามารถนำมาใช้ได้ในการทำงาน ซึ่งผลลัพธ์ที่ออกมาเป็นไปตามความต้องการของผู้ใช้งาน Script แต่ว่าในบางขั้นตอนผู้ใช้งานมีการเปลี่ยนความต้องการทำให้งานเกิดการล่าช้า เช่น การเพิ่ม Function ให้ตัว Script ทำให้ต้องใช้ระยะเวลาในการศึกษาเพิ่มเติม

- ความสามารถในการใช้ Command บน OS Linux ได้เป็นอย่างดี ในการประยุกต์ใช้งานการเขียน Script สำหรับ Monitor และ Script สำหรับการแจ้งเตือน Email
- เข้าใจการทำงานของ Shell และ OS Linux เบื้องต้น
- การเขียน Shell Script เพื่อให้สามารถตรวจสอบและป้องกันการเกิดปัญหาขึ้นบน Server
- การใช้โปรแกรม Oracle VM VirtualBox เพื่อใช้ในการติดตั้ง OS สำหรับการเขียน Shell Script และทดสอบการใช้งาน Script บนระบบปฏิบัติการ Linux
- การใช้โปรแกรม FileZilla เพื่อให้เกิดความสะดวกสบายในการโอนย้ายข้อมูลจากเครื่องจำลองไปเครื่องจริง และจากเครื่องจริงไปเครื่องจำลอง
- การช่วยสนับสนุนการทำงานของพนักงานที่ปรึกษา

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการที่ได้ศึกษาและเขียน Shell Script สำหรับ Monitor และ เขียน Shell Script สำหรับแจ้งเตือน Email ได้ผลสรุปดังนี้

5.1.1 สรุปผลการศึกษา Shell Script

- มีความเข้าใจเกี่ยวกับ Shell และ Shell Script
- สามารถพลิกแพลงการใช้งาน Command ต่างๆ ของ Shell Script ได้

5.1.2 สรุปผลการเขียน Shell Script

- ได้มีการนำ Bash Shell Script มาใช้เขียน Shell Script สำหรับ Monitor และ Shell Script สำหรับการแจ้งเตือน Email
- สามารถเขียน Shell Script ได้ตรงตามความต้องการของผู้ใช้ โดยได้ Script ออกมา 2 ตัว คือ Script สำหรับการ Monitor และ Script สำหรับการแจ้งเตือน Email

5.1.3 สรุปผลโดยรวม

- สามารถทำการติดตั้งและใช้งาน Script สำหรับการ Monitor บนระบบปฏิบัติการ Linux และ Centos6.4-Centos7
- สามารถทำการติดตั้งและใช้งาน Script สำหรับการแจ้งเตือน Email โดยสามารถติดตั้งและใช้งานได้บนระบบปฏิบัติการ Linux และ Centos6.4-Centos7
- สามารถ Monitor ได้รวดเร็วมากขึ้นจากที่ต้องใช้ Command ในการ Monitor หลาย Command
- สามารถแก้ปัญหาได้ทันเวลาที่เมื่อ Server เกิดปัญหา เช่น Hard Disk ใกล้จะเต็มหรือการทำงานของ CPU มีการทำงานที่หนักเกินไปก็จะมีการส่ง Email ไปยัง Email ที่ผู้ใช้ Script กำหนดเอาไว้

จากผลลัพธ์ทั้งหมด ทำให้สามารถที่จะสรุปได้ว่า Script สำหรับ Monitor และ Script สำหรับการแจ้งเตือน Email สามารถช่วยให้ผู้ใช้สามารถดูแลและป้องกันได้ทันเวลาที่เมื่อเกิดปัญหขึ้นบนเครื่อง Server

5.2 แนวทางการแก้ไข้ปัญหา

ปัญหาที่พบในระหว่างการศึกษา Shell Script ส่วนแรกจะเป็นในส่วนเรื่องการทำความเข้าใจ Command ต่างๆที่ใช้ในการเขียน Bash Shell Script เนื่องจาก Command ที่ใช้ในการเขียน Bash Shell Script มีจำนวนที่เยอะ และสามารถพลิกแพลง Command ต่างๆที่ใช้ในการเขียน Script มาใช้งานได้อย่างหลากหลาย ซึ่งทำให้กระบวนการศึกษาเพื่อนำมาใช้งาน เบื้องต้นค่อนข้างไปได้ช้า ซึ่งแนวทางการแก้ไข้ปัญหานี้ คือการเขียน Shell Script ทดลองหลาย Command เพื่อทำให้เกิดความคุ้นชินในการใช้งาน Command ต่างๆและทำให้สามารถมองภาพรวมในการใช้ Command ต่างๆได้ ส่วนปัญหาที่ 2 คือ

ในส่วนของการกำหนดการตั้งต่างๆใน Script สำหรับการแจ้งเตือน Email เนื่องจากมีการกำหนดค่าต่างๆมากมายและในการกำหนดค่าต่างๆนั้นจะไม่มีแจ้งเตือนการเปลี่ยนแปลงในการกำหนดค่าทำให้ไม่ทราบว่าค่าที่แก้ไข้นั้นตรงตามความต้องการหรือไม่ตรง เช่น การกำหนดให้ Server สามารถส่ง Email ไปที่ผู้ใช้งาน Script ซึ่งแนวทางการแก้ไข้ปัญหาคือควรสมัคร Email สำหรับการทดสอบในการรับส่ง Email

5.3 ข้อเสนอแนะจากการดำเนินงาน

5.3.1 ควรมีพื้นฐานในการใช้งานระบบปฏิบัติการ Linux เบื้องต้นอย่างการใช้ Command และการตั้งค่าระบบพื้นฐานต่างๆ เช่น สามารถใช้ Command ในการแก้ไข้ไฟล์ข้อมูลได้หรือ Command ในการเข้า Directory ต่างๆ

5.3.2 ควรมีพื้นฐานในการเขียน Shell Script เบื้องต้นเนื่องจากการเขียน Shell Script บางส่วนค่อนข้างมีความซับซ้อนทำให้ผู้พัฒนาที่ไม่มีพื้นฐานอาจเกิดความล่าช้า

5.3.3 ในการลงมือเขียน Shell Script ต้องหาข้อมูลเพื่อเตรียมความพร้อมก่อนลงมือและต้องวางแผนให้การทำงานเสมอเพื่อให้เกิดความรวดเร็วและมองเห็นภาพรวมก่อนลงมือทำงานจริง

เอกสารอ้างอิง

- 1 saixiii, 2017, Shell คืออะไร Unix Shell หรือ Linux Shell คือ โปรแกรมรับคำสั่งให้กับ Kernel[Online], Available: <https://saixiii.com/what-is-shell-unix-linux/>
- 2 saixiii, 2017, Shell Script คืออะไร Shell script คือ ภาษา programming ที่ใช้บน unix linux[Online], Available: <https://saixiii.com/what-is-shell-script/>
- 3 saixiii, 2017, Unix คืออะไร ยูนิกซ์ คือระบบปฏิบัติการของ Server[Online], Available: <https://saixiii.com/what-is-unix/>
- 4 saixiii, 2017, Linux คืออะไร ทำงานอย่างไร ลินุกซ์ เป็นระบบปฏิบัติการชนิดหนึ่ง[Online], Available: <https://saixiii.com/what-is-linux/>
- 5 saixiii, 2017, Kernel คืออะไร Kernel คือ ส่วนกลางควบคุมการทำงานทุกอย่างบน OS[Online], Available: <https://saixiii.com/what-is-kernel/>
- 6 ethand, 2015, Configure a Postfix Relay through Gmail on CentOS7[Online], Available: <https://devops.profitbricks.com/tutorials/configure-a-postfix-relay-through-gmail-on-centos-7/>
- 7 W3schools, 1999-2018, HTML <table> Tag[Online], Available: https://www.w3schools.com/tags/tag_table.asp

ภาคผนวก

TNI

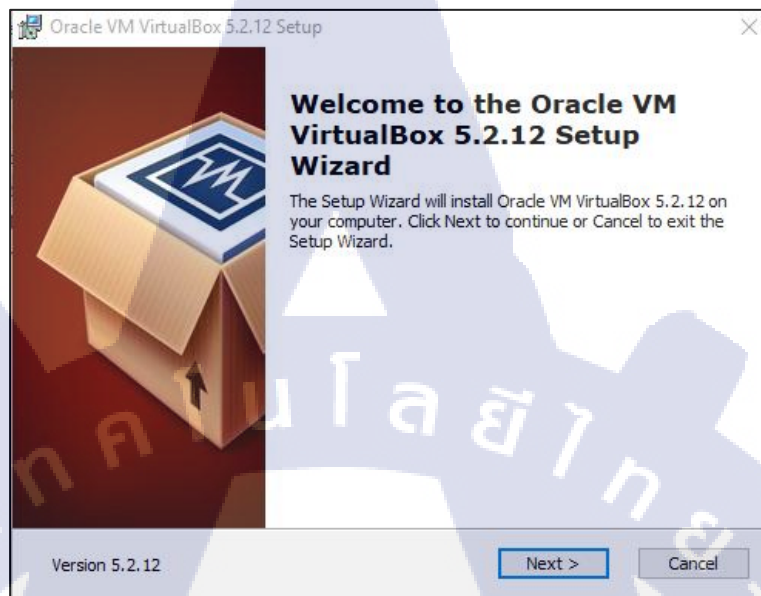
ภาคผนวก ก.

คู่มือการติดตั้งโปรแกรม Oracle VM Virtual Box

TNI

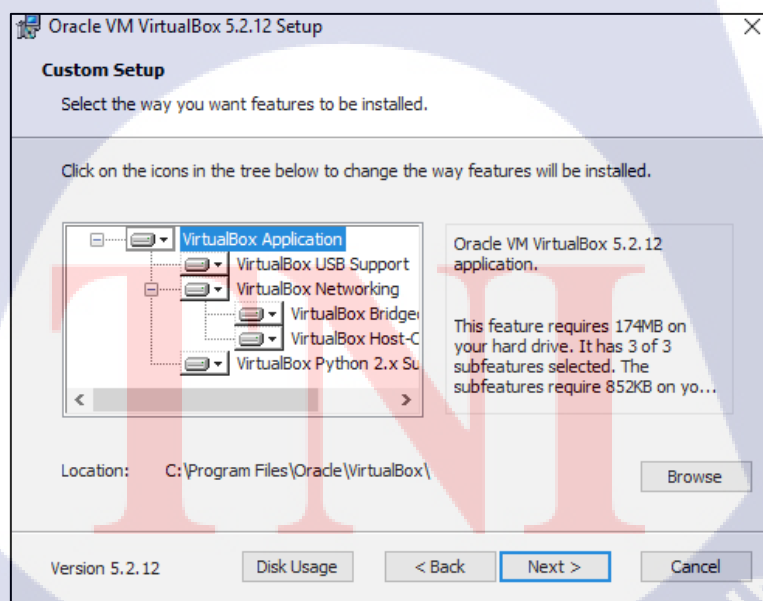
ขั้นตอนการติดตั้งโปรแกรม Oracle VM VirtualBox

- 1 กดติดตั้ง Oracle VM VirtualBox หลังจากเข้าตัวติดตั้งให้ทำการกด Next



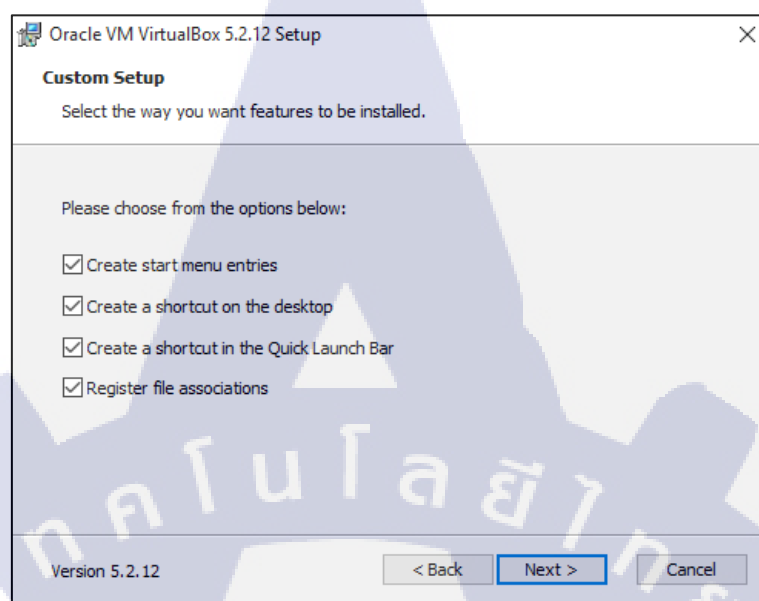
ภาพที่ ก. 1 การติดตั้ง Oracle VM VirtualBox

- 2 กด Browse เลือกสถานที่เก็บตัวโปรแกรม หลังจากเสร็จแล้วให้กด Next



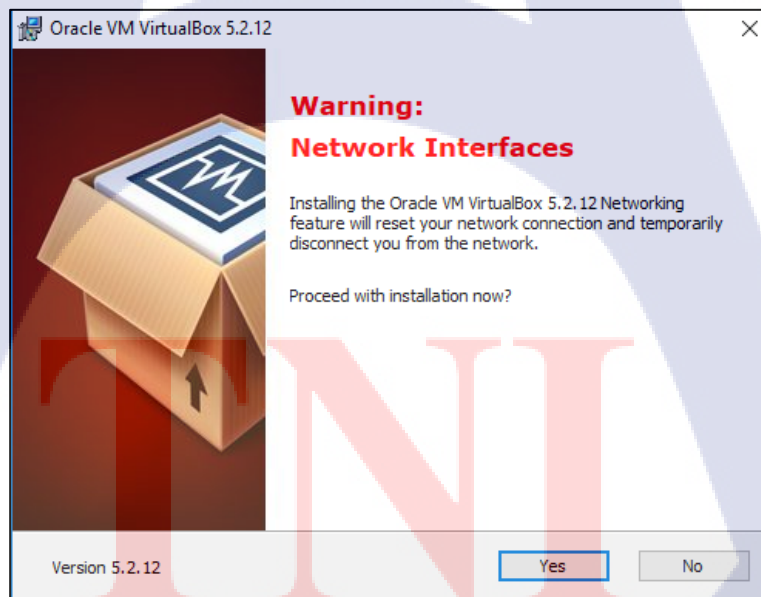
ภาพที่ ก. 2 การติดตั้ง Oracle VM VirtualBox (ต่อภาพที่ ก.1)

3 เลือกเป็น Default ของตัวติดตั้งและกด Next



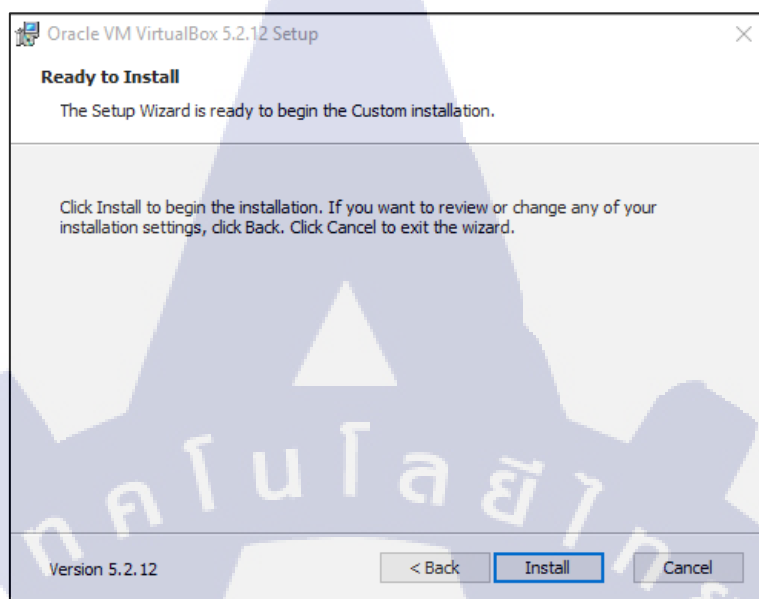
ภาพที่ ก. 3 การติดตั้ง Oracle VM VirtualBox (ต่อภาพที่ ก.2)

4 กด Yes



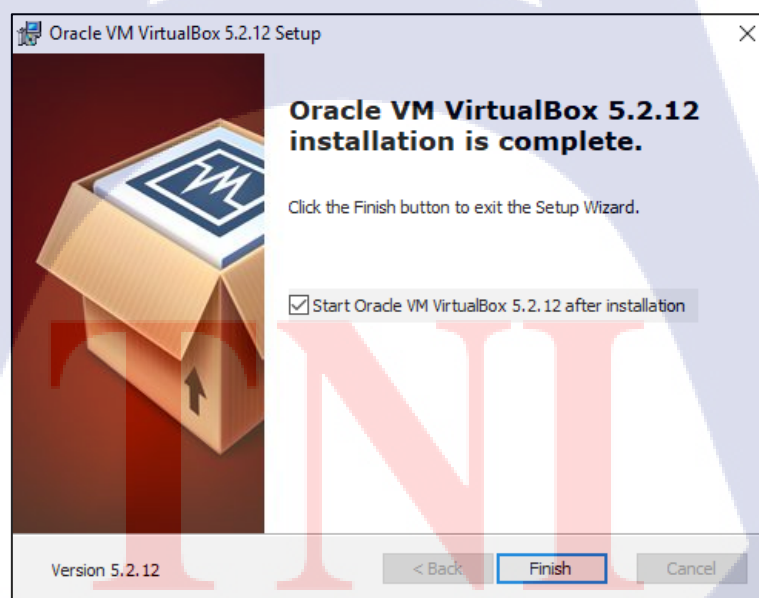
ภาพที่ ก. 4 การติดตั้ง Oracle VM VirtualBox (ต่อภาพที่ ก.3)

5 กด Install และรอจนโปรแกรมติดตั้งเสร็จ



ภาพที่ ก. 5 การติดตั้ง Oracle VM VirtualBox (ต่อภาพที่ ก.4)

6 กด Finish



ภาพที่ ก. 6 การติดตั้ง Oracle VM VirtualBox (ต่อภาพที่ ก.5)

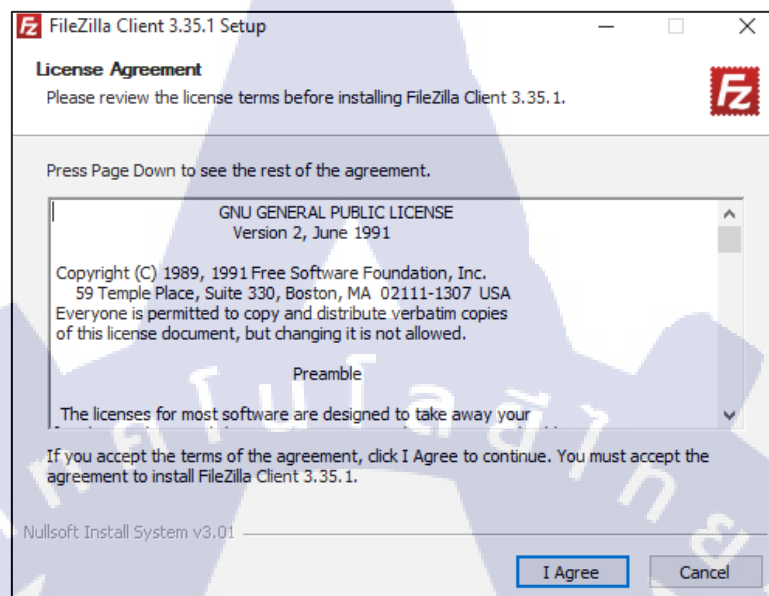
ภาคผนวก ข.

คู่มือการติดตั้งโปรแกรม FileZilla

TNI

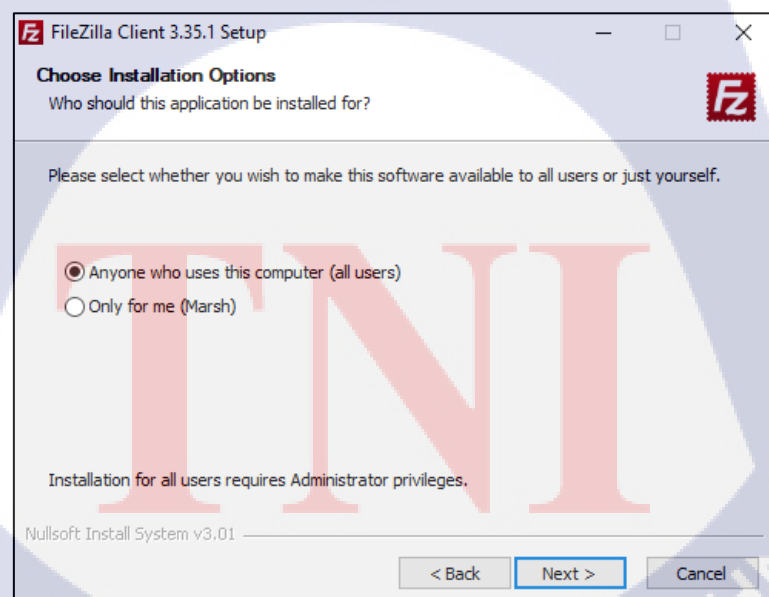
ขั้นตอนการติดตั้งโปรแกรม FileZilla

- 1 กดติดตั้ง FileZilla หลังจากเข้าตัวติดตั้งให้ทำการกด Next



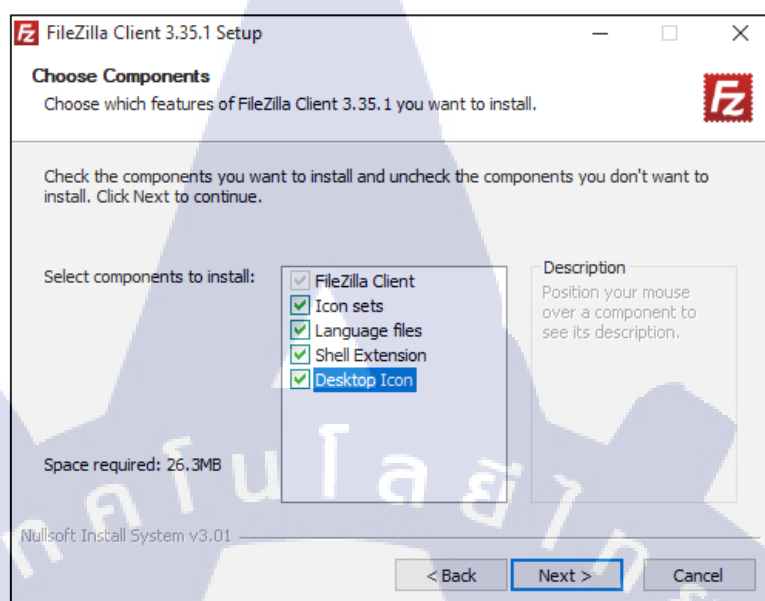
ภาพที่ ข. 1 การติดตั้ง FileZilla

- 2 กดเลือกว่าจะทำการติดตั้งให้ทุก User หรือเลือก User ที่กำลังใช้งาน



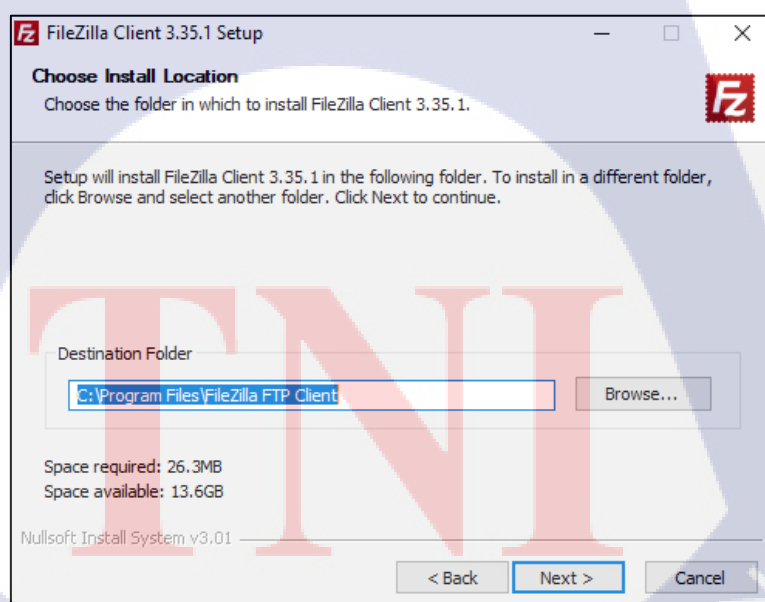
ภาพที่ ข. 2 การติดตั้ง FileZilla (ต่อภาพที่ ข.1)

3 กด Next



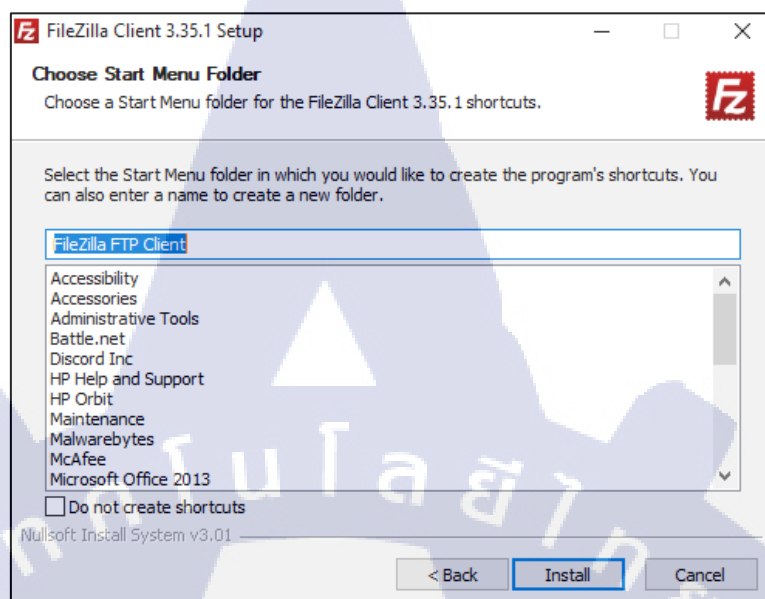
ภาพที่ ข. 3 การติดตั้ง FileZilla (ต่อภาพที่ ข.2)

4 กด Browse กำหนดสถานที่เก็บตัวโปรแกรม FileZilla และกด Next



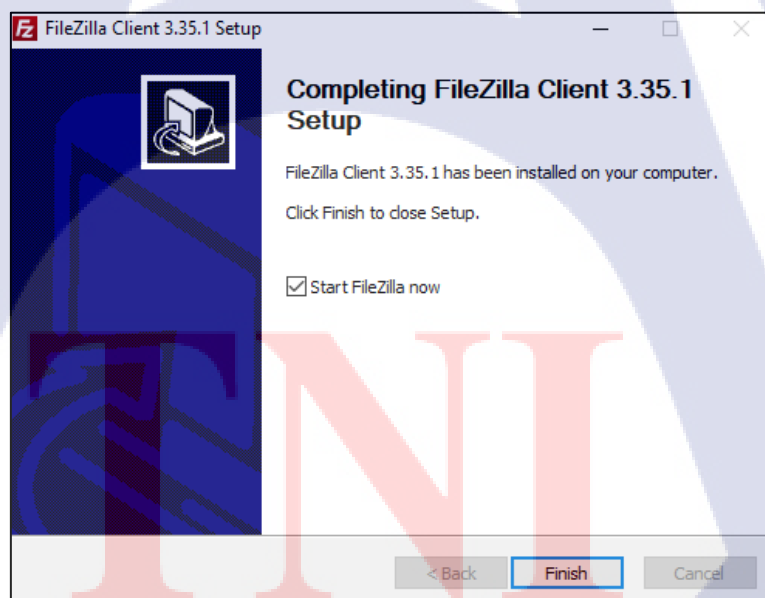
ภาพที่ ข. 4 การติดตั้ง FileZilla (ต่อภาพที่ ข.3)

5 กด Install และรอจนโปรแกรมติดตั้งเสร็จ



ภาพที่ ข. 5 การติดตั้ง FileZilla (ต่อภาพที่ ข.4)

6 กด Finish



ภาพที่ ข. 6 การติดตั้ง FileZilla (ต่อภาพที่ ข.5)



ภาคผนวก ค.

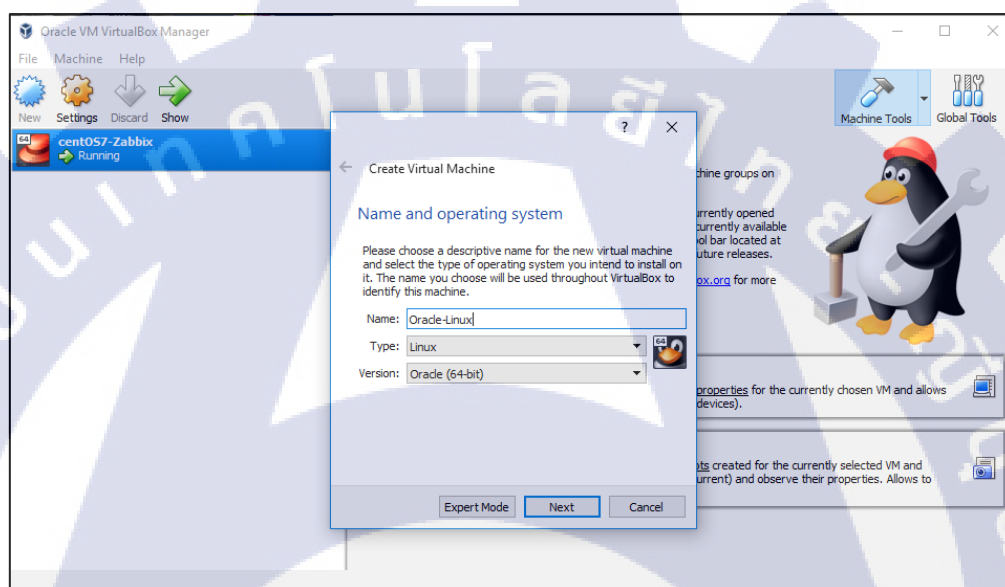
คู่มือการติดตั้งระบบปฏิบัติการ Linux

TNI

การติดตั้งระบบปฏิบัติการ Linux บนโปรแกรม Oracle VM VirtualBox

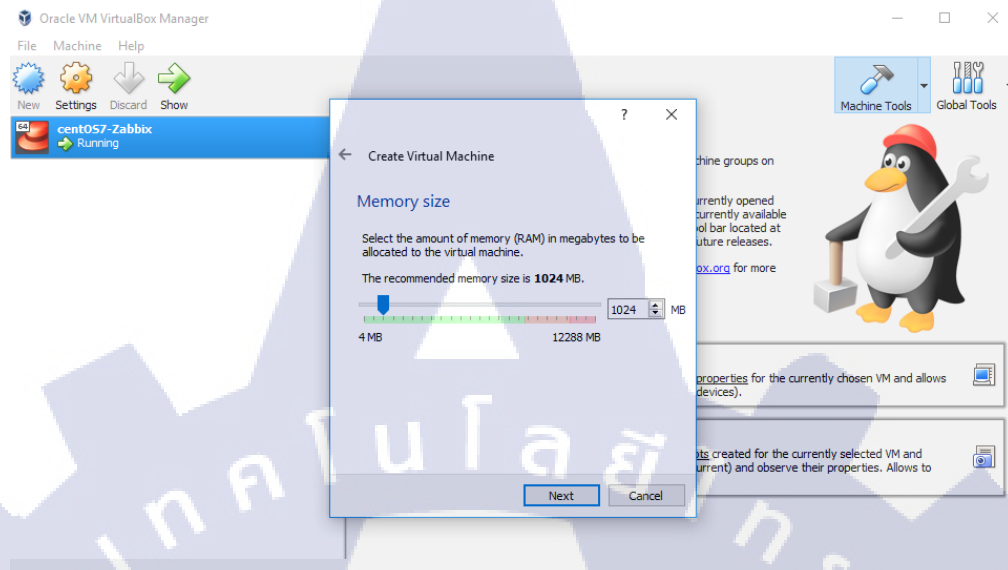
ก่อนที่จะติดตั้ง OS Linux จะต้องมีการเตรียมไฟล์ .iso ของระบบปฏิบัติการ Linux เพื่อที่จะนำมาใช้ในการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox ซึ่งตัว OS Linux นั้นสามารถหาได้จาก Website ต่างๆ ซึ่งการติดตั้งระบบปฏิบัติการ Linux (ในที่นี้ใช้ Oracle Linux ในการติดตั้ง) มีวิธีการดังนี้

- 1 กดเปิดโปรแกรม Oracle VM VirtualBox ขึ้นมา กดที่ New ทำการตั้งชื่อเป็น Oracle-Linux ตัวโปรแกรมจะทำการกำหนดค่า Type และ Version ให้อัตโนมัติ จากนั้นทำการกด Next



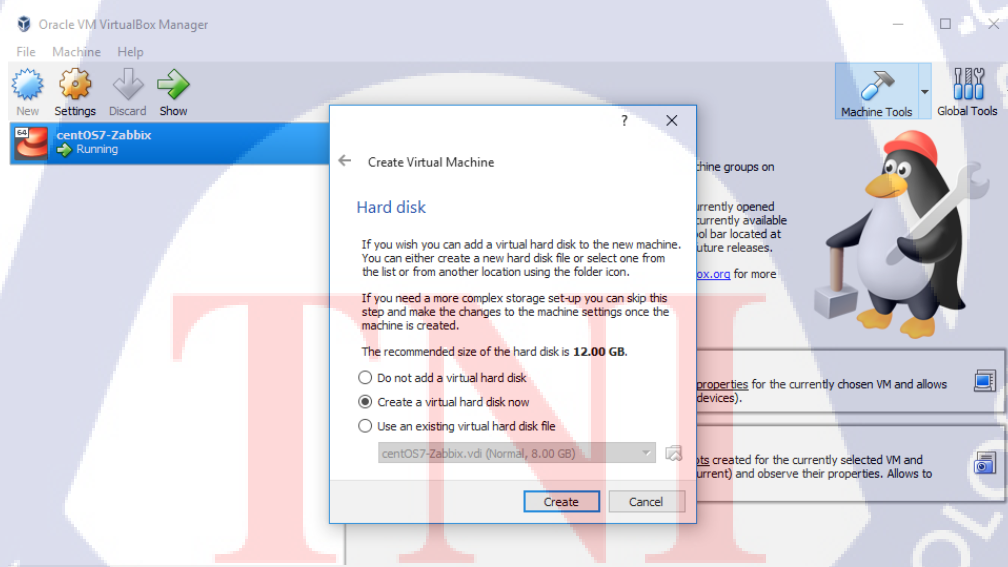
ภาพที่ ก. 1 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox

2 ในส่วนของรูปที่ 4.2 สามารถเลือก RAM ตรงตามความต้องการและทำการกด Next



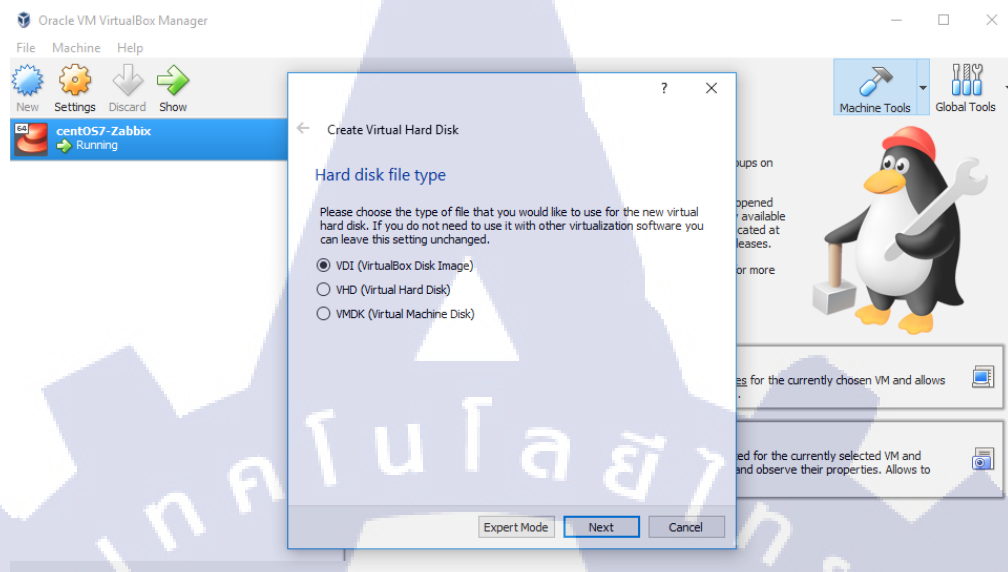
ภาพที่ ค. 2 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox (ต่อภาพที่ ค.1)

3 กดเลือก Create a virtual hard disk now และทำการกด Create



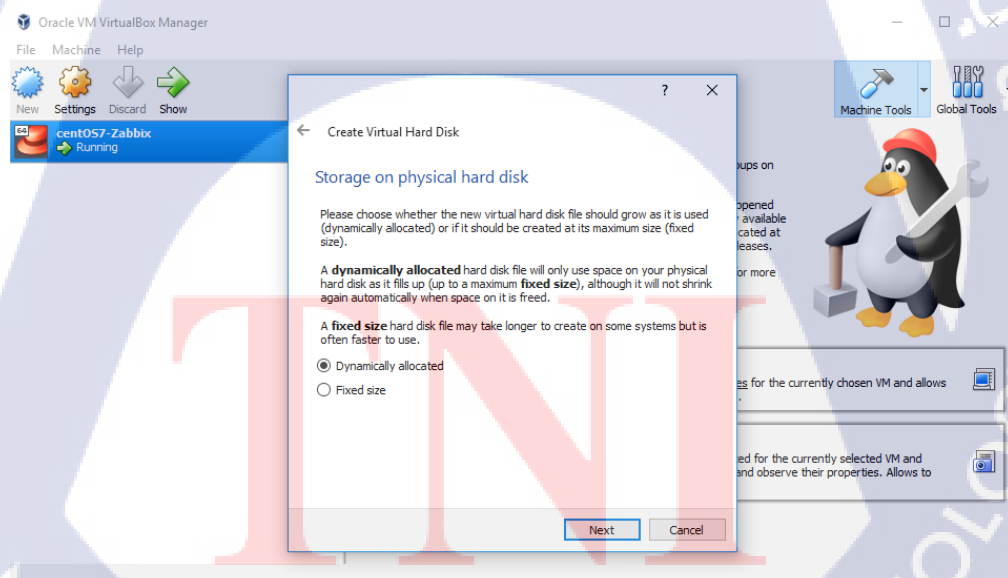
ภาพที่ ค. 3 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox (ต่อภาพที่ ค.2)

4 กดเลือก VDI และทำการกด Next



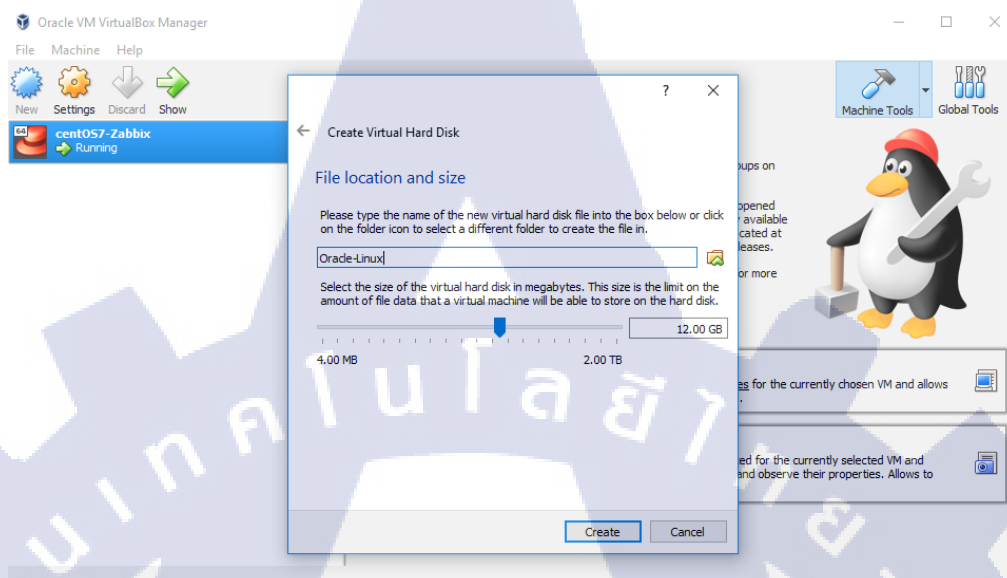
ภาพที่ ค. 4 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox (ต่อภาพที่ ค.3)

5 กดเลือก Dynamically allocated และทำการกด Next



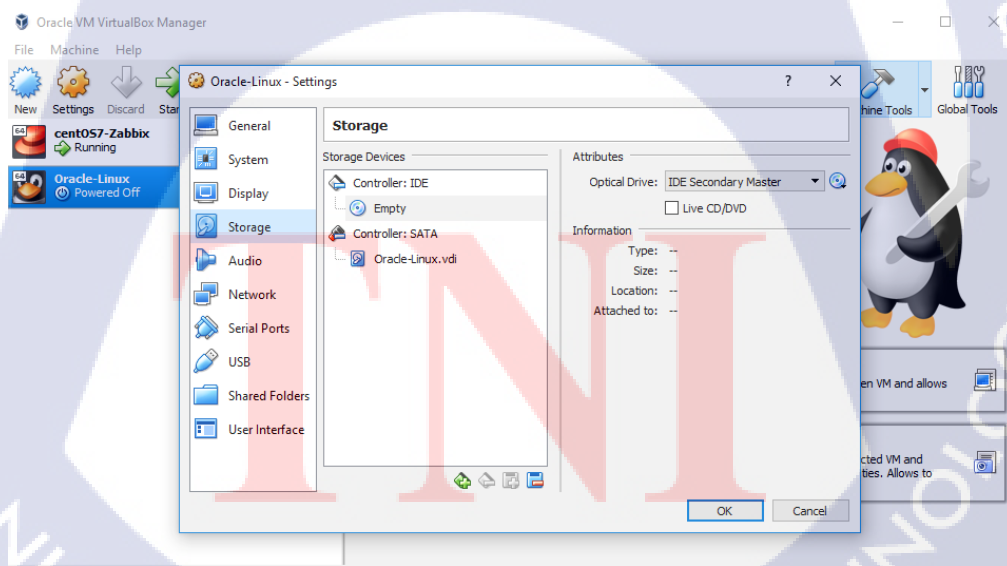
ภาพที่ ค. 5 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox (ต่อภาพที่ ค.4)

- 6 ตรงนี้จะเป็นการกำหนด Hard Disk ให้กับตัว OS Linux ให้กำหนดตามความต้องการหลังจากนั้นกด Create



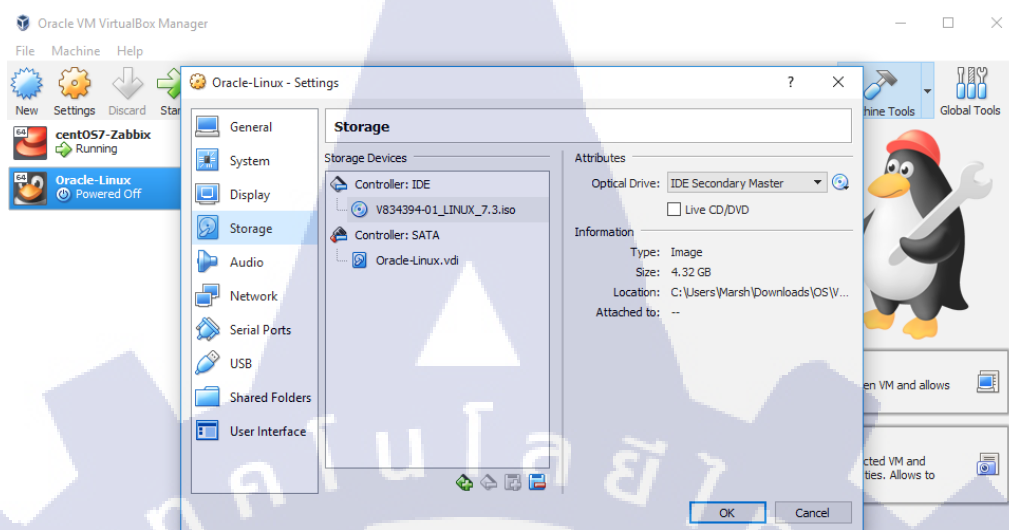
ภาพที่ ค. 6 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox (ต่อภาพที่ ค.5)

- 7 หลังจากนั้นให้คลิกขวาที่ Oracle-Linux และกดเข้า Setting เลือก Storage



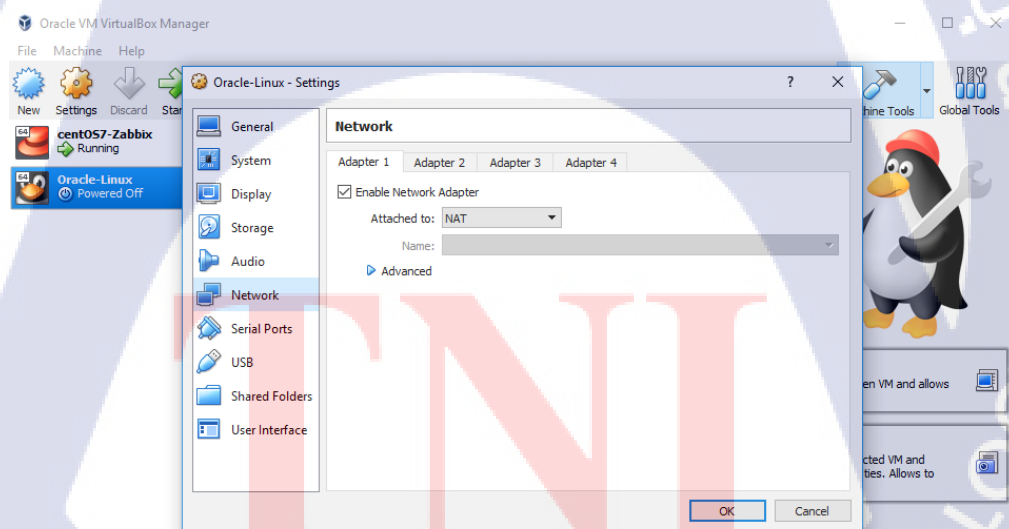
ภาพที่ ค. 7 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox (ต่อภาพที่ ค.6)

- 8 เลือกที่รูปแผ่น CD ในช่องของ Attributes และเลือกไฟล์ .iso ของ OS Linux ที่ต้องการติดตั้ง



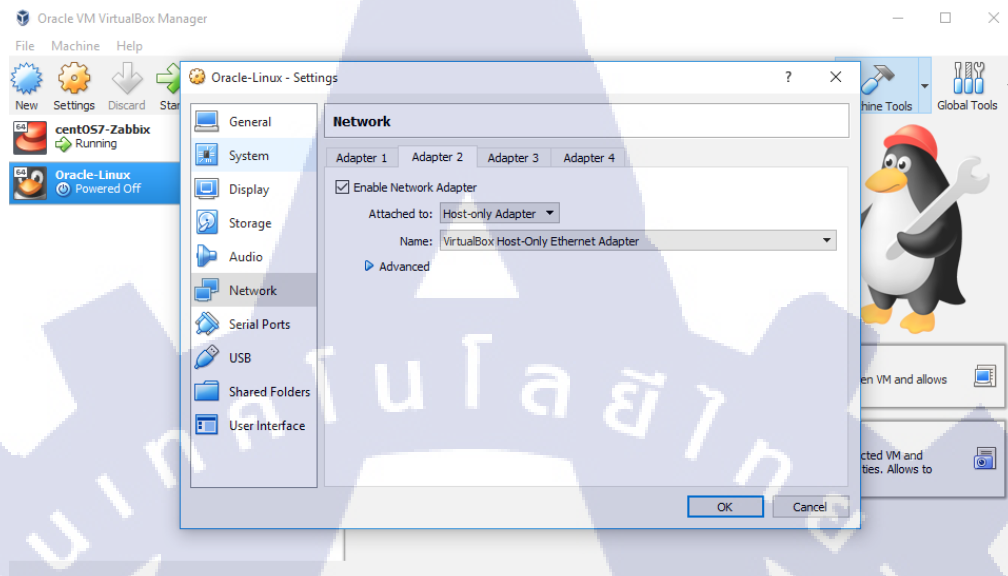
ภาพที่ ค. 8 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox (ต่อภาพที่ ค.7)

- 9 จากนั้นเข้ามาที่เมนู Network ให้ทำการกำหนดค่า Adapter 1 เป็น NAT



ภาพที่ ค. 9 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox (ต่อภาพที่ ค.8)

- 10 หลังจากนั้นทำการกำหนดค่าให้กับ Adapter 2 ให้เลือก Enable Network Adapter และทำการเลือก Attached to เป็น Host-only Adapter หลังจากนั้นทำการกด OK



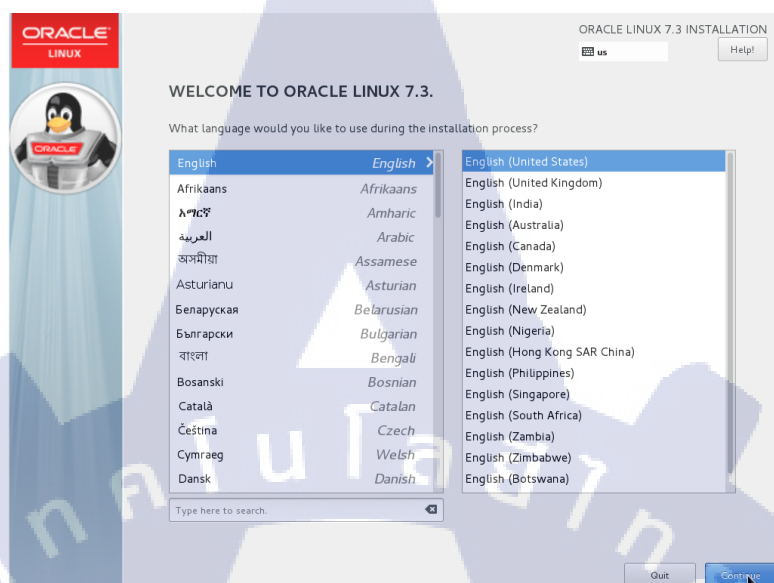
ภาพที่ ก. 10 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox (ต่อภาพที่ ก.9)

- 11 เมื่อมาอยู่หน้าเมนูหลักให้ทำการกดเลือก Oracle-Linux และกด Start หลังจากนั้นจะมาอยู่ที่หน้าจอการติดตั้ง OS Oracle Linux ให้ทำการเลือก Install Oracle Linux [Version]



ภาพที่ ก. 11 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox (ต่อภาพที่ ก.10)

12 หลังจากนั้นเลือกภาษาหลักให้กับตัว OS ที่ทำการติดตั้ง



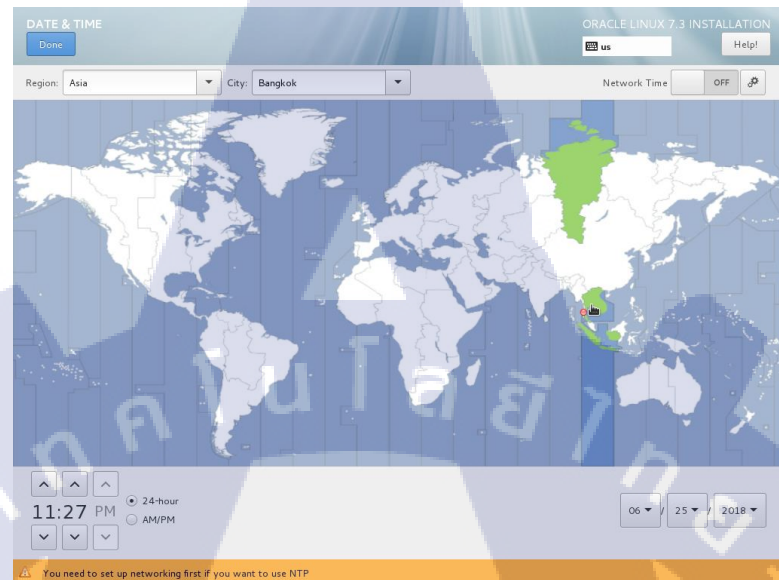
ภาพที่ ค. 12 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.11)

13 หลังจากนั้นทำการเลือก DATE & TIME



ภาพที่ ค. 13 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.12)

- 14 หลังจากนั้นให้เลือก Region และ City ที่ต้องการและทำการกำหนดเวลาปัจจุบันจากนั้นให้ทำการกด Done หลังจากเสร็จการกำหนดค่าต่างๆ



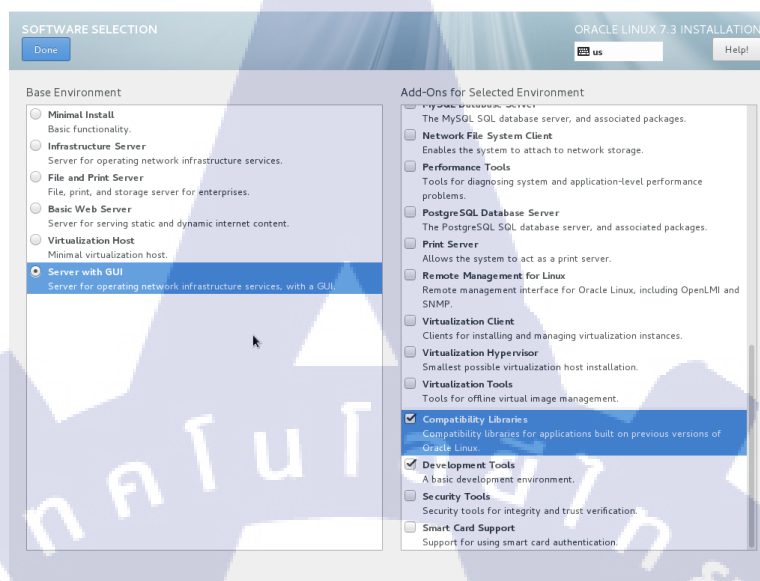
ภาพที่ ค. 14 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.13)

- 15 หลังจากกลับมาอยู่หน้าเมนู INSTALLATION SUMMARY ให้ทำการเลือก SOFTWARE SELECTION



ภาพที่ ค. 15 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.14)

16 หลังจากนั้นทำการเลือก Server with GUI และทำการกด Done



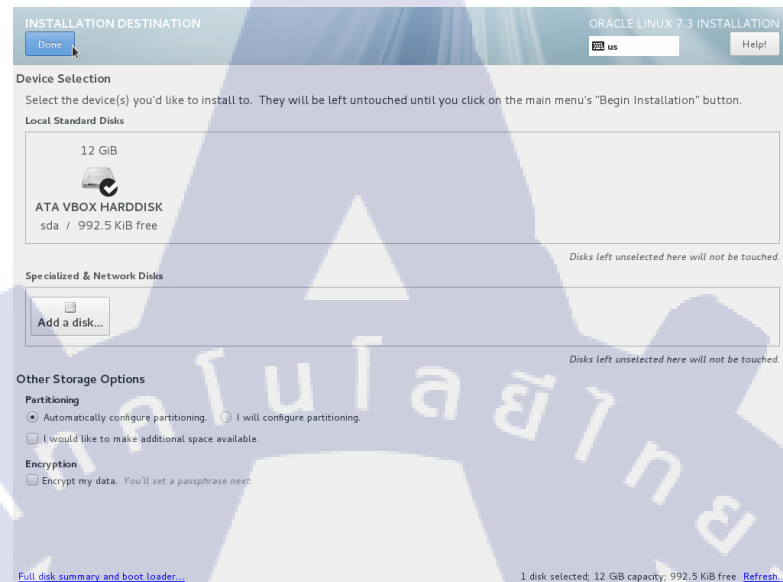
ภาพที่ ค. 16 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.15)

17 หลังจากนั้นให้เลือก INSTALATION DETINATION



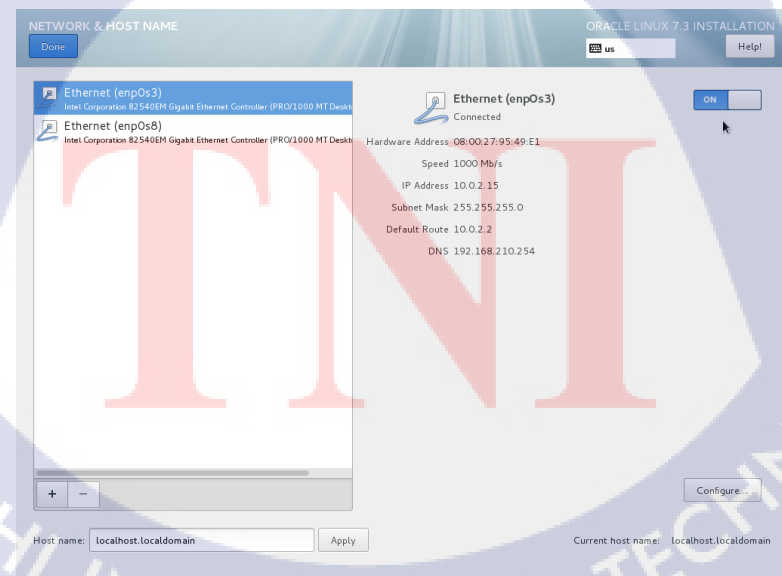
ภาพที่ ค. 17 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.16)

- 18 หลังจากเข้ามาใน INSTALLTION DESTINATION ให้กำหนดค่าขนาดของ Disk ที่ต้องการแต่ในที่นี้จะใช้ Default ให้ทำการกด Done



ภาพที่ ค. 18 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.17)

- 19 หลังจากนั้นจะกลับมาหน้า INSTALLATION SUMMARY ให้เลือก NETWORK & HOST NAME ทำการเลือกเปิด Port ทุกตัวและทำการกด Done



ภาพที่ ค. 19 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.18)

20 หลังจากกลับมาหน้า INSTALLATION SUMMARY ให้กด Begin Installation



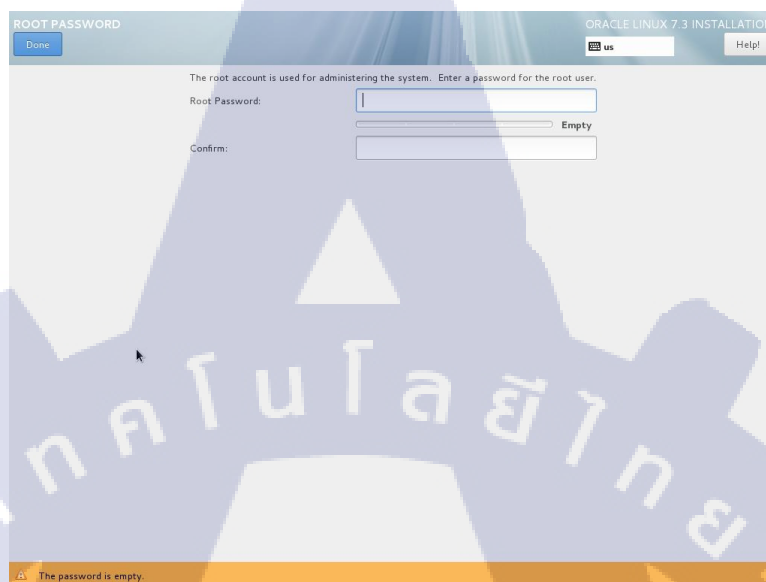
ภาพที่ ค. 20 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.19)

21 หลังจากนั้น ให้ทำการกำหนด ROOT PASSWORD และ USER CREATION



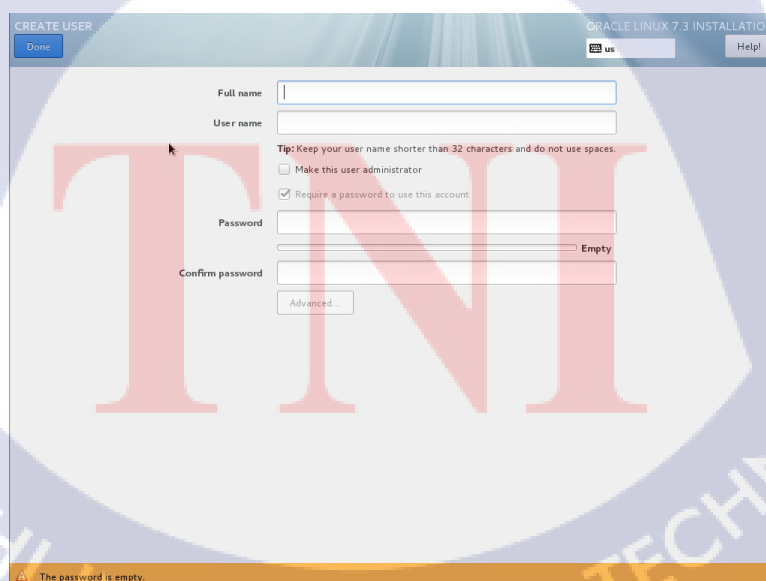
ภาพที่ ค. 21 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.20)

- 22 หลังจากเข้ามา ROOT PASSWORD ให้กำหนด Password ของ Root ตามความต้องการและทำการกด Done หลังกำหนดเสร็จสิ้น



ภาพที่ ค. 22 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.21)

- 23 หลังจากเข้ามา USER CREATION ให้กำหนดชื่อของ User และกำหนด Password ของ User ตามความต้องการและทำการกด Done หลังกำหนดเสร็จสิ้น



ภาพที่ ค. 23 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.22)

24 หลังจากนั้นให้รอจนกว่าจะ Install เสร็จสิ้นหลังจากนั้นให้กด Reboot



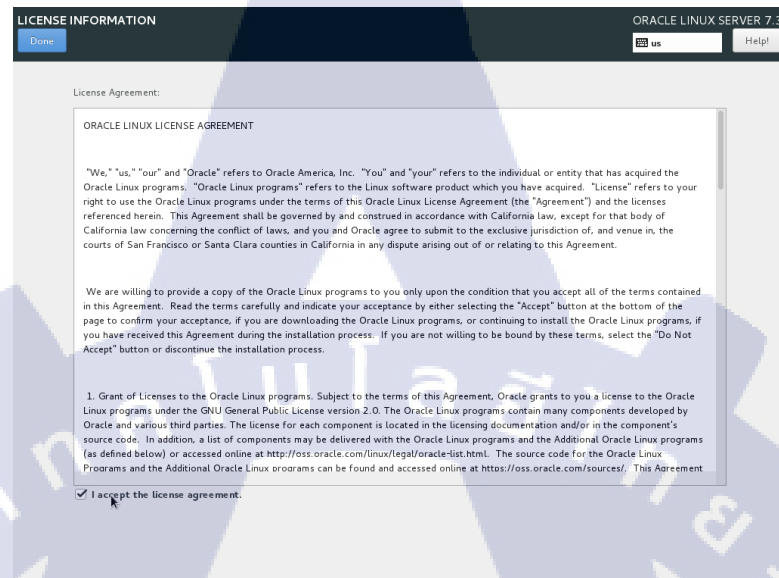
ภาพที่ ค. 24 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.23)

25 หลังจาก Reboot เสร็จสิ้น ให้ทำการกดเข้าไป LICENSE INFORMATION



ภาพที่ ค. 25 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.24)

- 26 หลังจากเข้าหน้า LICENSE INFORMATION ให้เลือก I accept the license agreement และทำการกด Done



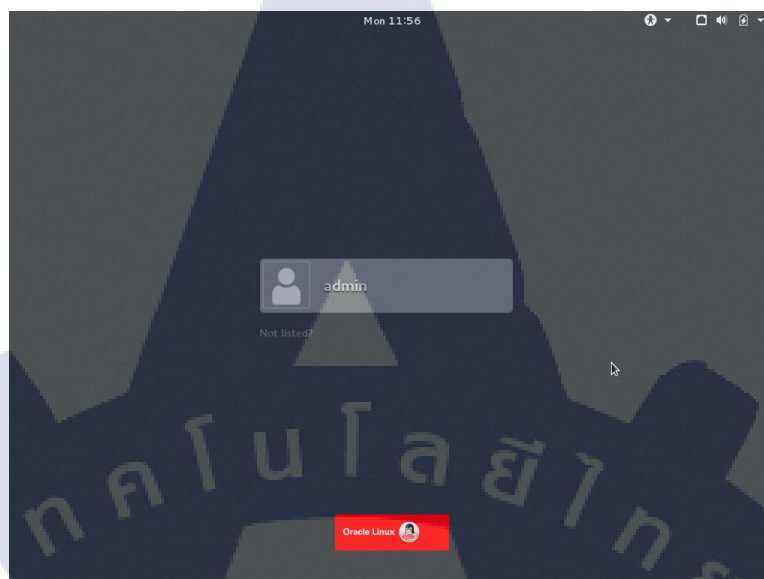
ภาพที่ ค. 26 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.25)

- 27 หลังจากนั้นให้กด FINISH CONFIGURATION



ภาพที่ ค. 27 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.26)

28 หลังจากเสร็จทุกอย่างเรียบร้อยแล้วก็จะสามารถเข้าใช้งาน Linux ได้ทันที



ภาพที่ ค. 28 ตัวอย่างการติดตั้งระบบปฏิบัติการ Linux บน Oracle VM VirtualBox
(ต่อภาพที่ ค.27)

ประวัติผู้จัดทำโครงการ

ชื่อ – สกุล

นายคุณาธิป สุนทรพัชรคุณ



วัน เดือน ปีเกิด

13 กันยายน พ.ศ. 2539

ประวัติการศึกษา

ระดับประถมศึกษา

โรงเรียนจินดาพงศ์

ระดับมัธยมศึกษาตอนต้น

โรงเรียนไทยคริสเตียน

ระดับมัธยมศึกษาตอนปลาย

โรงเรียนศรีวิกรม์

ระดับอุดมศึกษา

สถาบันเทคโนโลยีไทย – ญี่ปุ่น

ทุนการศึกษา

- ไม่มี -

ประวัติการฝึกอบรม

- Space Technology สถานที่ สถาบันเทคโนโลยีไทย-ญี่ปุ่น
- Microsoft DreamSpark Premium Training สถานที่ สถาบันเทคโนโลยีไทย-ญี่ปุ่น
- Data Center สถานที่ สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ผลงานที่ได้รับการตีพิมพ์

- ไม่มี -