

การพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนฐาน

ปริญญานิพนธ์
ของ
ณรังสรรค์ สุธะเจริญชัยกุล

เสนอต่อบัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ เพื่อเป็นส่วนหนึ่งของการศึกษาตาม
หลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาการจัดการทางวิศวกรรม
พฤษภาคม 2553

การพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานี่ฐาน

ปริญญานิพนธ์
ของ
ณรังสรรค์ สุรเจริญชัยกุล

เสนอต่อบัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ เพื่อเป็นส่วนหนึ่งของการศึกษาตาม
หลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาการจัดการทางวิศวกรรม

พฤษภาคม 2553

ลิขสิทธิ์เป็นของมหาวิทยาลัยศรีนครินทรวิโรฒ

การพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานี่ฐาน

บทคัดย่อ

ของ

ณรังสรรค์ สุรเจริญชัยกุล

เสนอต่อบัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ เพื่อเป็นส่วนหนึ่งของการศึกษาตาม

หลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาการจัดการทางวิศวกรรม

พฤษภาคม 2553

ณรังสรรค์ สรเจริญชัยกุล. (2553). การพัฒนาประสิทธิภาพระบบการจัดการประสิทธิภาพส่วนของ
สถานี่ฐาน. ปริญญาโท วิศวกรรม. (การจัดการทางวิศวกรรม). กรุงเทพฯ:บัณฑิตวิทยาลัย
มหาวิทยาลัยศรีนครินทรวิโรฒ. คณะกรรมการควบคุม: ผู้ช่วยศาสตราจารย์ ดร.นำคุณ
ศรีสนิท, ดร.พรศักดิ์ เรียบร้อยเจริญ.

งานวิจัยนี้เป็นการพัฒนาประสิทธิภาพระบบการจัดการประสิทธิภาพส่วนของสถานี่ฐาน
ขึ้น เพื่อให้สามารถทราบปัญหา และเป็นเครื่องมือในการแก้ไขปัญหา ทำให้ข้อมูลมีความถูกต้อง ลด
ระยะเวลาในการตรวจสอบและแก้ไขปัญหาของวิศวกร จากการสำรวจการเกิดปัญหาความผิดปกติ
ของค่าตัววัดต่างๆ (Measurement) ของผู้ให้บริการโทรศัพท์เคลื่อนที่ระหว่างเดือน มกราคม ถึง
เดือนกันยายน พ.ศ. 2552 พบปัญหาเกิดจากระบบการจัดการประสิทธิภาพส่วนของสถานี่ฐาน คิด
เป็นจำนวน 35% โดยการทำงานของระบบจะใช้ Shell script ร่วมกับ Oracle database โดย
หลักการทำงานของโปรแกรมจะประกอบด้วย 4 ส่วน ได้แก่ การตรวจสอบความครบถ้วนของข้อมูล
ตามเกณฑ์ที่กำหนด การตรวจสอบสาเหตุของข้อมูลที่มีความผิดปกติตั้งแต่การดึงข้อมูลจากชุมสาย
ต้นทางจนจบกระบวนการของระบบจัดการประสิทธิภาพส่วนของสถานี่ฐาน การดึงข้อมูลที่เป็นผล
จากความผิดพลาดของระบบการจัดการประสิทธิภาพส่วนของสถานี่ฐาน และ การเตือนและส่ง
รายงานแก่วิศวกร

ผลที่ได้จากการทำงานของโปรแกรมทำให้สามารถแก้ไขความผิดพลาดที่เกิดจากระบบ
จัดการประสิทธิภาพส่วนของสถานี่ฐานในส่วนของการดึงข้อมูลที่เกิดขึ้นได้ และทำให้การผิดพลาด
ของข้อมูลที่เกิดจากระบบการจัดการประสิทธิภาพลดลง อีกทั้งลดระยะเวลาในการตรวจสอบและ
แก้ไขปัญหาของวิศวกร

EFFICIENCY DEVELOPMENT OF PERFORMANCE MANAGEMENT SYSTEM (PMS)
FOR BASE STATIONS

AN ABSTRACT
BY
NARANGSAN SURACHAROENCHAIKUL

Presented in Partial Fulfillment of the Requirements for the
Master of Engineering Degree in Engineering Management
at Srinakharinwirot University

May 2010

Narangsan Suracharoenchaikul. (2010). *Efficiency development of performance management system for base stations*. Master thesis, M.Eng. (Engineering Management). Bangkok: Graduate School, Srinakharinwirot University.
Advisor Committee: Asst. Prof. Dr. Namkhun Srisanit, Dr.Pronsak Raibroycharoen.

The objective of this research is to automatically check percentage of complete data, recover data and send the result alert and report the processing to staff. The result of the research enables fast and correct data as well as reducing check time for staff. The researcher discovers the irregularity of the traffic measurement from one mobile network operator in Thailand between January and September 2009, which is deviant from the normal process of the performance management system for more than 35%. The software is created from the shell script and oracle database. The process is consisted of four parts as follows: 1. Checking the completion of final data (Pcal) threshold percentage, 2. Checking the cause of incomplete data from source data until the final process of the performance management system is reached, 3. Reprocessing from the problem of the performance management system and 4. Sending the alert or report to staff.

The results of this research show that it can automatically fix the problem, which is an incomplete part of receiving data source, reduce data percentage error from the processes of the performance management system and decrease staff's time to check and fix this problem.

ปริญญานิพนธ์
เรื่อง
การพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานี่ฐาน
ของ
ณรังสรรค์ สุระเจริญชัยกุล

ได้รับอนุมัติจากบัณฑิตวิทยาลัยให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาการจัดการทางวิศวกรรม
ของมหาวิทยาลัยศรีนครินทรวิโรฒ

..... คณบดีบัณฑิตวิทยาลัย
(รองศาสตราจารย์ ดร.สมชาย สันติวัฒนกุล)
วันที่ เดือน พฤษภาคม พ.ศ. 2553

คณะกรรมการควบคุมปริญญานิพนธ์

คณะกรรมการสอบปากเปล่า

..... ประธาน
(ผู้ช่วยศาสตราจารย์ ดร.นำคุณ ศรีสนิท)

..... ประธาน
(รองศาสตราจารย์ ธนรัตน์ แต้ววัฒนา)

..... กรรมการ
(ดร.พรศักดิ์ เรียบร้อยเจริญ)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.วชิระ จงบุรี)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.นำคุณ ศรีสนิท)

..... กรรมการ
(ดร.พรศักดิ์ เรียบร้อยเจริญ)

ประกาศคุณูปการ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความสามารถช่วยเหลือ และความเอาใจใส่อย่างดียิ่งตลอดจนการให้คำแนะนำ และข้อคิดเห็นที่เป็นประโยชน์อย่างยิ่งสำหรับการปรับแก้ไขข้อบกพร่อง จากคณะกรรมการผู้ควบคุมปริญญานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.นำคุณ ศรีสนิท และ ดร.พรศักดิ์ เรียบร้อยเจริญ กรรมการควบคุมปริญญานิพนธ์ที่ได้ให้ความเมตตากรุณาเป็นที่ปรึกษา และให้ความช่วยเหลือชี้แนะแนวทางในสิ่งที่เป็ประโยชน์ต่อการศึกษาและการทำปริญญานิพนธ์นี้ด้วยความเอาใจใส่ตลอดมา รวมทั้ง รองศาสตราจารย์ ธนรัตน์ แต้วัฒนา และ ผู้ช่วยศาสตราจารย์ ดร.วชิระ จงบุรี กรรมการสอบปริญญานิพนธ์ ที่กรุณาให้ข้อเสนอแนะต่างๆ เพิ่มเติมแก่ผู้วิจัย ทำให้ปริญญานิพนธ์ฉบับนี้มีความสมบูรณ์ยิ่งขึ้น ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอกราบขอบพระคุณคณาจารย์และกรรมการบริหารหลักสูตรสาขาการจัดการวิศวกรรม คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒทุกท่าน ที่ได้กรุณาประสิทธิ์ประสาทความรู้ต่างๆ ให้แก่ผู้วิจัย และอบรมสั่งสอนเป็นอย่างดี

ขอขอบคุณ บริษัท แอดวานซ์ อินโฟร์ เซอร์วิส จำกัด (มหาชน) สำหรับข้อมูลตัวอย่างที่ใช้ในงานวิจัยนี้และขอขอบคุณผู้เชี่ยวชาญทุกท่าน ซึ่งได้ให้ความอนุเคราะห์ข้อมูลและคำแนะนำที่เป็นประโยชน์ในการวิจัย

ขอขอบคุณพี่ๆและเพื่อนๆ รวมถึงบุคคลอีกหลายท่านที่ไม่ได้กล่าวนามไว้ ณ ที่นี้ที่ได้ให้ความช่วยเหลือและเป็นกำลังใจให้กับผู้วิจัยมาโดยตลอด

สุดท้ายนี้ ผู้วิจัยขอโน้มราลึกถึงคุณของบิดามารดาและครูอาจารย์ ที่อบรมสั่งสอนให้ความรู้เป็นกำลังใจและให้การสนับสนุนผู้วิจัยด้วยดีตลอดมา

ณรังสรรค์ สุระเจริญชัยกุล

สารบัญ

บทที่	หน้า
1 บทนำ	1
ความเป็นมาและความสำคัญของปัญหา	1
วัตถุประสงค์การวิจัย	3
ขอบเขตการวิจัย	3
ขั้นตอนการดำเนินการวิจัย	3
ประโยชน์ที่คาดว่าจะได้รับ	4
กรอบแนวคิดในการวิจัย	4
สมมติฐานการวิจัย	4
2 ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง	5
หลักการทำงานของโทรศัพท์เคลื่อนที่ในส่วน of สถานีฐาน	5
ระบบโทรศัพท์เคลื่อนที่	5
Traffic measurement	7
ระบบการจัดการประสิทธิภาพส่วน of สถานีฐาน	7
หลักการของเครื่องมือโปรแกรมที่ใช้ในการวิจัย	8
Oracle Database	8
ภาษา Standard relational database Query Language (SQL)	12
โปรแกรม Shell Script	19
งานวิจัยที่เกี่ยวข้อง	38
3 วิธีการดำเนินการวิจัย	40
ประชากรและกลุ่มตัวอย่าง	40
เครื่องมือที่ใช้ในการวิจัย	41
วิเคราะห์และออกแบบ	41
ทดสอบเครื่องมือในการวิจัย	44
ประเมินผลเครื่องมือในการวิจัย	44
การวิเคราะห์ค่าทางสถิติ	45

สารบัญ (ต่อ)

บทที่	หน้า
4 ผลการดำเนินการวิจัย.....	49
ลักษณะการทำงานและผลลัพธ์จากโปรแกรมพัฒนาประสิทธิภาพของ ระบบการจัดการสมรรถนะส่วนสถานี่ฐาน.....	49
ผลของการประเมินผลเพื่อทดสอบความเชื่อมั่นของเครื่องมือใน การวิจัย.....	53
ผลของการประเมินผลเกี่ยวกับความสามารถของระบบและ ทดสอบสมมติฐานการวิจัย.....	55
5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ.....	63
สรุปผลการวิจัย.....	63
อภิปรายผลการวิจัย.....	65
ข้อเสนอแนะ.....	65
บรรณานุกรม.....	66
ภาคผนวก.....	69
ประวัติย่อผู้วิจัย.....	74

บัญชีตาราง

ตาราง	หน้า
1 คำสั่งเกี่ยวกับโครงสร้างตาราง.....	14
2 คำสั่งเกี่ยวกับการจัดการข้อมูล.....	15
3 ตัวอย่างผลลัพธ์การใช้คำสั่ง GROUP BY.....	17
4 ตัวอย่างผลลัพธ์การใช้คำสั่ง ORDER BY.....	18
5 แสดงรูปแบบการเขียน และความหมายของ Shell script.....	24
6 แสดงการใช้งานคำสั่งTest.....	27
7 แสดงความหมายและทางเลือกของพารามิเตอร์.....	28
8 แสดงความหมายและทางเลือกของพารามิเตอร์ (ต่อ).....	28
9 ความคลาดเคลื่อนในการตัดสินใจ.....	48
10 ผลการประเมินความพึงพอใจในการใช้งานระบบจากผู้เชี่ยวชาญ.....	53
11 ผลการประเมินสมรรถนะ (Performance) ของระบบจากผู้เชี่ยวชาญ.....	54
12 การหาค่าความแปรปรวนของแบบสอบถามเพื่อใช้คำนวณสัมประสิทธิ์แอลฟา.....	55
13 จำนวนและค่าเฉลี่ยของ BSC ที่ผิดปกติและต้องทำการแก้ไข.....	56
14 สรุปจำนวนและค่าเฉลี่ยของ BSC ที่ผิดปกติและต้องทำการแก้ไข.....	57
15 แสดงเวลาการประมวลผลในการตรวจสอบและแก้ไขปัญหา.....	58
16 แสดงสถิติเปรียบเทียบระยะเวลาการทำงานเฉลี่ย.....	60

บัญชีภาพประกอบ

ภาพประกอบ	หน้า
1 สถิติการพบปัญหาความผิดปกติของค่าตัววัด.....	2
2 GSM Network Elements.....	6
3 GSM Interfaces.....	6
4 Oracle Architecture.....	10
5 แผนผังการทำงานของโปรแกรม.....	43
6 แสดงการตั้งเวลาทำงานของโปรแกรม.....	49
7 แสดงผลจากการทำงานของโปรแกรมที่ช่วยหาค่าข้อมูลที่มีค่าน้อยกว่า 95 %.....	50
8 แสดงผลจากการทำงานหาค่า BSC ที่มีค่าน้อยกว่า 95 %.....	50
9 แสดงผลการตรวจสอบข้อมูล BSC ที่ต้นทาง.....	51
10 แสดง Website การแสดงผล Query ข้อมูลในฐานข้อมูล.....	51
11 แสดงผลการส่งผลลัพธ์ในรูปแบบข้อความสั้น (SMS).....	52
12 แสดงผลการประมวลผลดึงข้อมูล BSC ที่ขาดหายไป.....	52
13 ระยะเวลาเฉลี่ยในการประมวลผลด้วยโปรแกรมและด้วยผู้ดูแลระบบ.....	60

บทที่ 1

บทนำ

ความเป็นมาและความสำคัญของปัญหา

เนื่องด้วยในยุคปัจจุบัน การใช้งานเทคโนโลยีมีจำนวนผู้ใช้งานเป็นจำนวนมาก ไม่ว่าจะเป็น อินเทอร์เน็ต ไปจนถึงโทรศัพท์เคลื่อนที่ ซึ่งในปัจจุบัน ถือเป็นปัจจัยที่ 5 ในการดำรงชีวิต โดยจากสถิติในแถลงการณ์ของสหภาพโทรคมนาคมระหว่างประเทศ (ITU) ประเมินการสิ้นปี 2009 จะมีผู้ใช้บริการโทรศัพท์เคลื่อนที่ถึง 4.6 ล้านล้านคนและจำนวนโทรศัพท์เคลื่อนที่ 67 เครื่องต่อ 100 ครวเรือน โดยเติบโตจากปี 1998 ที่มีโทรศัพท์เคลื่อนที่เพียง 5 เครื่องต่อ 100 ครวเรือน ทำให้ปริมาณการใช้งานเครือข่าย อีกทั้งค่าตัววัด (Measurement) มีจำนวนมากขึ้นเป็นจำนวนมาก ไม่ว่าจะเป็นเกิดจากการใช้งาน หรือ จำนวนชุมสายต่างๆ ที่เพิ่มมากขึ้น ทำให้วิธีการจัดการ ไม่ว่าจะเป็น การตรวจสอบปริมาณ ข้อมูลต่างๆ มีความยากลำบากยิ่งขึ้น (International Telecommunication Union. 2009: 1)

โดยค่าตัววัดต่างๆ (Measurement) สามารถนำข้อมูลที่ได้นั้นจะต้องมีความถูกต้องครบถ้วนมาเก็บในฐานข้อมูล เพื่อนำข้อมูลมาประเมินการใช้งานและแนวโน้มการใช้งาน ของผู้ใช้งานแต่ละพื้นที่ อีกทั้งสามารถทำให้ทราบถึงประสิทธิภาพหรือความผิดปกติของสถานีฐานแต่ละสถานี เพื่อที่จะนำค่าที่ได้มาปรับปรุงหรือปรับเปลี่ยนและประเมินการขยายอุปกรณ์ของสถานีฐาน

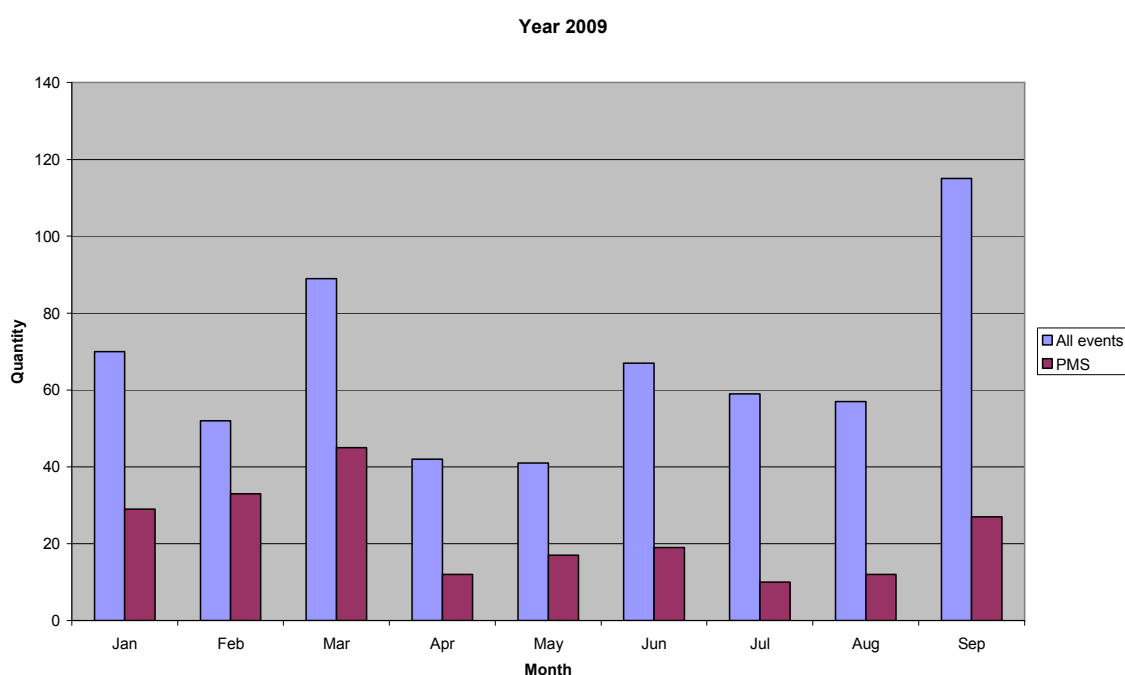
ระบบการจัดการประสิทธิภาพ หรือ Performance Management System (PMS) นั้นเป็นระบบที่เก็บข้อมูลค่าตัววัดต่างๆ (Measurement) เช่น จากสถานีฐาน (Base Station) มาทำการคำนวณ และเก็บลงฐานข้อมูล (Database) เพื่อนำข้อมูลมาแสดงในรูปแบบต่างๆ เช่น แผนภูมิเส้น แผนภูมิแท่ง ตาราง เพื่อสามารถเห็นถึงการใช้งาน ประสิทธิภาพ ของอุปกรณ์ต่างๆ เพื่อให้บุคคลที่ต้องการใช้งานนำไปใช้งานได้สะดวก เช่น ฝ่ายวางแผนและพัฒนา ผู้ดูแลระบบ สามารถข้อมูลส่วนนี้ไปวางแผนพัฒนาการใช้งานในอนาคต และสามารถแสดงได้ว่าอุปกรณ์ส่วนใดบ้าง มีปริมาณการใช้งานสูง ไม่สามารถรองรับการใช้งานได้ ต้องรีบปรับปรุงระบบโดยเร็ว โดยสามารถระบุข้อ(ชุมสาย)ภาค สถานีฐาน และระยะเวลาที่ต้องการให้แสดงข้อมูลได้

ปัญหาความผิดปกติของค่าตัววัดต่างๆ ที่ไม่สามารถจัดเก็บเข้าสู่ระบบ และประมวลผลได้นั้น สามารถแบ่งได้เป็น 2 ประเภทหลัก

1. ปัญหาที่เกิดจากระบบชุมสาย หรือ สถานีฐาน เกิดความผิดปกติไม่สามารถส่งข้อมูลดังกล่าวมายังระบบการจัดการประสิทธิภาพได้ เช่น เกิดความขัดข้องบนระบบการสร้างข้อมูลและส่งข้อมูลค่าตัววัดต่างๆ, มีการทำกิจกรรมซ่อมบำรุงกับชุมสาย หรือ เครื่อง Server ที่ทำการประมวลผลข้อมูล, ระบบเครือข่าย (Network) ระหว่างชุมสายไม่สามารถติดต่อกันได้ เป็นต้น

2. ปัญหาที่เกิดขึ้นกับระบบการจัดการประสิทธิภาพ ไม่สามารถทำงานได้ตามปกติ เช่น โปรแกรมที่ใช้ในการแปลงข้อมูลดิบ (Raw data) ไม่ทำงาน, เกิดความผิดพลาดข้อมูลไม่สามารถนำข้อมูลลงฐานข้อมูลได้ เป็นต้น

จากการสำรวจการเกิดปัญหาความผิดปกติของค่าตัววัดต่างๆ (Measurement) ของผู้ให้บริการโทรศัพท์เคลื่อนที่ระบบ GSM (Global System for Mobile communications) ระหว่างเดือน มกราคม ถึง เดือนกันยายน พ.ศ. 2552 พบปัญหาเกิดขึ้น จำนวน 592 ครั้ง และเป็นปัญหาที่เกิดจากระบบการจัดการสมรรถนะส่วนฐานจำนวน 204 ครั้ง สามารถแจกแจงรายละเอียดในแต่ละเดือนดังนี้



ภาพประกอบ 1 สถิติการพบปัญหาความผิดปกติของค่าตัววัด

จากภาพประกอบ 1 จะเห็นได้ว่าโดยเฉลี่ยแล้ว จะเกิดปัญหาความผิดปกติที่ทำให้ไม่สามารถนำข้อมูลค่าวัดต่างๆ เข้าสู่ระบบได้ตามปกติคิดเฉลี่ย ในแต่ละเดือนจะเกิดขึ้นจำนวน 66 ครั้ง และสามารถระบุว่าเป็นปัญหาจากระบบการจัดการสมรรถนะเฉลี่ยเดือนละ 23 ครั้ง คิดเป็นจำนวน 35% ของปัญหาทั้งหมด

โดยวิธีการตรวจสอบข้อมูล เพื่อจำแนกปัญหาและทำการแก้ไขปัญหาก็ต้องใช้เวลาตรวจสอบระบบ ในการตรวจสอบข้อมูลและแก้ไขปัญหาระบบดังกล่าว และจากการเก็บข้อมูลระหว่างเดือน มกราคม ถึง เดือนกันยายน พ.ศ. 2552 เวลาที่ใช้ในการทำการตรวจสอบและแก้ไขปัญหามูลบนระบบการจัดการสมรรถนะส่วนฐาน เพื่อให้ข้อมูลส่วนดังกล่าวมีความถูกต้องและครบถ้วนเฉลี่ยวันละ 3 ชั่วโมง

ดังนั้นทางผู้วิจัยได้เล็งเห็นถึงปัญหาและความสำคัญดังกล่าว จึงได้จัดทำการวิจัยพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานีฐานขึ้น เพื่อให้สามารถทราบปัญหา อีกทั้งเป็นเครื่องมือในการแก้ไขปัญหา ทำให้ข้อมูลมีความถูกต้อง ลดระยะเวลาในการตรวจสอบและแก้ไขปัญหาของวิศวกร และผู้ใช้งาน เช่น ผู้ที่มีหน้าที่ในการวางแผนระบบ สามารถเข้าใช้งานข้อมูลดังกล่าวได้ถูกต้องและรวดเร็ว ทำให้สามารถนำข้อมูลที่ได้ไปพิจารณาประเมินปรับปรุงระบบเครือข่ายโทรศัพท์เคลื่อนที่ให้เกิดความสอดคล้องกับปริมาณการใช้งานหรือ ตอบสนองการใช้งานของผู้ใช้บริการได้ดียิ่งขึ้น หรือผู้ที่มีหน้าที่ทางการตลาด สามารถระบุแนวโน้มของการใช้งานในแต่ละพื้นที่และคิดโปรโมชั่น ตอบสนองการใช้งานของผู้ใช้งานโทรศัพท์เคลื่อนที่ได้อย่างถูกต้องรวดเร็ว ทำให้เกิดข้อได้เปรียบกว่าคู่แข่งรายอื่น ที่ไม่มีการพัฒนาประสิทธิภาพระบบการจัดการประสิทธิภาพ

วัตถุประสงค์ของการวิจัย

1. ศึกษาและพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานีฐาน เพื่อช่วยตรวจสอบและแก้ปัญหาค่าตัววัด (Measurement) เบื้องต้นของเครือข่ายโทรศัพท์เคลื่อนที่
2. เพื่อให้ได้ระบบพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานีฐาน ที่ช่วยเพิ่มประสิทธิภาพในการตรวจสอบความถูกต้องและสามารถแก้ไขปัญหาเบื้องต้นได้รวดเร็ว

ขอบเขตของการวิจัย

ขอบเขตของการวิจัยแบ่งออกเป็นหัวข้อ ดังนี้

1. ศึกษาและพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานีฐานของชุมสายยี่ห้อ Huawei ในส่วนของ Radio (ระดับ Base Station Controller (BSC) – Base Transceiver Station (BTS)) รุ่น B32 เท่านั้น
2. การพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานีฐานของชุมสายที่ศึกษาสามารถใช้ได้กับทุกภาคในประเทศไทย
3. โดยระบบที่พัฒนาขึ้นเพื่อนำไปใช้ประยุกต์ใช้กับระบบการจัดการสมรรถนะส่วนสถานีฐานที่มีการใช้งานอยู่

ขั้นตอนการดำเนินงานวิจัย

ในการวิจัยนี้สามารถแบ่งขั้นตอนในการวิจัยออกเป็น 5 ขั้นตอน ดังนี้

1. ศึกษาที่มาและสาเหตุของปัญหา ระบบการจัดการสมรรถนะส่วนสถานีฐาน
2. เก็บรวบรวมข้อมูลที่เกี่ยวข้องของระบบการจัดการสมรรถนะส่วนสถานีฐาน

3. วิเคราะห์และออกแบบระบบพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนฐาน
4. ทดสอบระบบพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนฐาน โดยนำไปประยุกต์ใช้กับระบบเดิมที่มีอยู่
5. สรุปและประเมินผลการดำเนินการวิจัย จากการทดลองโดยเปรียบเทียบประสิทธิภาพการทำงานที่ดีขึ้น

ประโยชน์ที่คาดว่าจะได้รับ

1. ระบบพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนฐานสามารถตรวจสอบและแก้ไขปัญหาค่าตัววัด (Measurement) เบื้องต้นของเครือข่ายโทรศัพท์เคลื่อนที่
2. ระบบพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนฐาน สามารถเพิ่มประสิทธิภาพในการตรวจสอบความถูกต้องและสามารถแก้ไขปัญหาเบื้องต้นได้รวดเร็ว

กรอบแนวคิดในการวิจัย

ในการวิจัยครั้งนี้ ผู้วิจัยมุ่งเน้นพัฒนาโปรแกรมเพื่อสามารถใช้เป็นเครื่องมือในการช่วยตรวจสอบและแก้ไขปัญหาค่าตัววัด (Measurement) ของเครือข่ายโทรศัพท์เคลื่อนที่ ซึ่งสามารถช่วยลดระยะเวลาในการตรวจสอบ และแก้ไขปัญหาเบื้องต้น อีกทั้งมีความถูกต้อง เมื่อเปรียบเทียบกับ การตรวจสอบและแก้ไขปัญหาด้วยวิธีปกติของผู้ดูแลระบบเอง โดยอาศัยกรอบแนวคิด ดังนี้

1. กระบวนการพัฒนาโปรแกรมอาศัยหลักการวิเคราะห์การทำงานและการแก้ไขปัญหาจากการทำงานจริงของผู้ดูแลระบบ
2. การทดสอบโปรแกรมด้วยการตั้งการทำงานเปรียบเทียบกับการทำงานจริงของผู้ดูแลระบบ

สมมุติฐานการวิจัย

1. ผลลัพธ์จากการประมวลผลของระบบที่สร้างขึ้น มีความถูกต้องสามารถตรวจสอบและแก้ไขปัญหาเบื้องต้นได้
2. ระบบที่สร้างขึ้นใช้ระยะเวลาในการประมวลผล น้อยกว่าระยะเวลาในการพิจารณาด้วยวิธีปกติของผู้ดูแลระบบเอง

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องในบทนี้ผู้วิจัยได้ศึกษาจากเอกสาร แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง ประกอบด้วย

1. หลักการทำงานของโทรศัพท์เคลื่อนที่ ในส่วนของสถานีฐาน Traffic Measurement และ ระบบการจัดการประสิทธิภาพส่วนของสถานีฐาน
2. หลักการของเครื่องมือโปรแกรมที่ใช้ในการวิจัย Oracle Database, ภาษา SQL และ Shell Script
3. งานวิจัยที่เกี่ยวข้อง

หลักการทำงานของโทรศัพท์เคลื่อนที่ในส่วนของสถานีฐาน Traffic Measurement และ ระบบการจัดการประสิทธิภาพส่วนของสถานีฐาน

ระบบโทรศัพท์เคลื่อนที่ (MOBILE STATION)

ระบบสื่อสารวิทยุเซลลูลาร์เป็นการสื่อสารของเน็ตเวิร์ค โทรศัพท์ที่มีความสำคัญมากที่สุด แรกเริ่มของเน็ตเวิร์ค (per-cellular) ไม่ค่อยได้รับการยอมรับเท่าที่ควร หลักการของระบบสื่อสารเซลลูลาร์ พื้นฐานเกิดขึ้นเมื่อปี 1949 โดย Bell Laboratories จนกระทั่งปี 1980 เทคโนโลยีด้านต่าง ๆ ถูกนำมาใช้กับเน็ตเวิร์คเพื่อสร้างและให้บริการกับลูกค้าในลักษณะเน็ตเวิร์คสาธารณะ ระบบสื่อสารเซลลูลาร์เคลื่อนที่ที่ได้รับการพัฒนาที่เวลาต่างกัน ในเมืองที่แตกต่างกันด้วย และทำให้เกิดความแตกต่างกันในด้านต่าง ๆ กันด้วย ในแต่ละระบบที่มีโครงสร้างต่างกัน เช่น ความถี่ และระยะห่างของแต่ละช่องสัญญาณ ฯลฯ ดังนั้นจึงมีระบบจำนวนที่แตกต่างกันที่เป็นมาตรฐานการสื่อสารเซลลูลาร์เคลื่อนที่ที่ใช้กันอยู่ทั่วโลกในปัจจุบันนี้ (จักรี จันทสร. 2552: Online)

1. หลักการเบื้องต้นในการทำงานของระบบโทรศัพท์เคลื่อนที่ ลักษณะรูปร่างของเน็ตเวิร์ค ระบบการสื่อสารเซลลูลาร์เคลื่อนที่จะมีพื้นที่การครอบครองที่แบ่งออกเป็นจำนวนพื้นที่ขนาดเล็กมาก ๆ เราเรียกพื้นที่ขนาดเล็กนี้ว่า เซลล์ (cell) ซึ่งในแต่ละเซลล์จะมีสถานีหลัก (Base station system) ที่มีสัญญาณการสื่อสารครอบคลุมในแต่ละที่ของแต่ละเซลล์ แต่ละสถานีหลักถูกต่อโดยเส้นทางการต่อที่แน่นอน (Fixed link) ไปยังศูนย์กลางการให้บริการการสวิตชิงเคลื่อนที่ (MSC); mobile services switching centre ซึ่งโดยทั่วไปชุมสายโทรศัพท์ดิจิทัลจะมีซอฟต์แวร์เป็นตัวจัดการให้บริการทั้งหมดแก่ผู้ใช้บริการ เน็ตเวิร์คโทรศัพท์เคลื่อนที่ส่วนมากประกอบด้วยจำนวนของ MSCs ซึ่งแต่ละ MSCs จะมีสถานีหลัก(BSS) เป็นของตัวเองจำนวนมากที่ต่ออยู่โดยเส้นทางการต่อ

2. สถานีฐาน (Base Station System) เป็นอุปกรณ์ที่ต่อเชื่อมโดยตรงกับ MSC กับ MS ด้วยระบบ Transmission หลาย ๆ แบบ เช่นอาจจะเป็น Microwave, Cables หรือ Fiber Optic เป็นต้น BSS สามารถติดตั้งได้ทั้งในอาคารหรือ Container และสามารถกำหนดให้ใช้งานได้เป็นแบบ Omnidirectional หรือ Sector cells ได้ BASE STATION SYSTEM (BSS) เป็นระบบที่ประกอบด้วย BASE STATION CONTROLLER (BSC) และ BASE TRANSCIEVER STATION (BTS) ซึ่งอาจมีได้หลายตัว รวมทั้งส่วนของ TRANSCODER หลัก ๆ ของ BSS มีดังนี้

- จัดการเกี่ยวกับ RF (Radio frequency) link
- กำหนดช่องสัญญาณสำหรับ MS ในการติดต่อกัน และ บอกกำลังงานที่จะใช้ในการ

ติดต่อ

- รอรับคำสั่งจาก MSC ในการที่จะยกเลิกการใช้ช่องสัญญาณ กับ MS ซึ่งอาจเกิดได้จาก 2 กรณี คือ มีการทำ Handover หรือ เลิกติดต่อกัน

- ควบคุมประสิทธิภาพการทำงานของแต่ละ Cell
- กำหนดรูปแบบของช่องสัญญาณทั้งแบบ Traffic และ Signaling
- สะสมข้อมูลต่าง ๆ เช่น ความแรงของสัญญาณจาก Cell ติดกัน และทำการของ Cell ต่าง ๆ ที่เหมาะสมที่จะใช้ในการทำ Handover รวมทั้งส่งข้อมูลเหล่านี้ไปยัง MSC (Mobile Service Switching Center)

Traffic measurement

การวัดอัตรา Traffic ภายในเครือข่ายระบบเครือข่ายช่วยให้ผู้จัดการและนักวิเคราะห์ตัดสินใจเกี่ยวกับการดำเนินงานและการวางแผนสำหรับการพัฒนาระยะยาว การวัด Traffic จะมีการใช้งานในหลายๆหน้าที่พื้นฐานเช่น

- ระบุรูปแบบของ Traffic และแนวโน้ม
- คำนวณปริมาณ Traffic และความหนาแน่นในแต่ละวงจรหรือกลุ่ม
- การตรวจสอบการให้บริการ
- ประเมินและจัดการขนาดเครือข่าย
- คาดคะเนประสิทธิภาพการทำงาน
- ประเมิน และการจัดการด้านการเครือข่าย SS7
- การตรวจสอบประสิทธิภาพของช่องทางทั่วไปที่ส่งสัญญาณเครือข่าย

ระบบการจัดการประสิทธิภาพส่วนของสถานีฐาน

ระบบการจัดการประสิทธิภาพ หรือ Performance Management System (PMS) นั้นเป็นระบบที่เก็บข้อมูลค่าวัดต่างๆ (Measurement) เช่น จากสถานีฐาน (Base Station) มาทำการคำนวณ และเก็บลงฐานข้อมูล (Database) เพื่อนำข้อมูลมาแสดงในรูปแบบต่างๆ เช่น แผนภูมิเส้น แผนภูมิแท่ง ตาราง เพื่อสามารถเห็นถึงการใช้งาน ประสิทธิภาพ ของอุปกรณ์ต่างๆ เพื่อให้บุคคลที่ต้องการใช้งานนำไปใช้งานได้สะดวก เช่น ฝ่ายวางแผนและพัฒนา ผู้ดูแลระบบ สามารถข้อมูลส่วนนี้

ไปวางแผนพัฒนาการใช้งานในอนาคต และ สามารถแสดงได้ว่าอุปกรณ์ส่วนใดบ้าง มีปริมาณการใช้งานสูง ไม่สามารถรองรับการใช้งานได้ ต้องรีบปรับปรุงระบบโดยเร็ว โดยสามารถระบุห้อย(ซุ่มสาย)ภาค สถานีฐาน และระยะเวลาที่ต้องการให้แสดงข้อมูลได้

ระบบการจัดการประสิทธิภาพส่วนของสถานีฐาน จะเป็นการเก็บข้อมูลเฉพาะในส่วน of สถานีฐานเท่านั้น โดยขั้นตอนการทำงานจะเป็นดังนี้

1. ข้อมูลค่าวัด (Measurement) ของสถานีฐานแต่ละสถานีจะทำการส่งมายัง ส่วนควบคุมของสถานีฐาน (BSC)

2. ระบบการจัดการประสิทธิภาพส่วนของสถานีฐาน จะทำการเข้าไปเก็บข้อมูลจากส่วนควบคุมของสถานีฐาน (BSC) และนำข้อมูลดังกล่าวแต่ละภาค มาทำการแปลงข้อมูลลงตารางฐานข้อมูลดิบ (Raw Database) เรียกรวมว่าเป็นส่วนของข้อมูลดิบ (Raw data)

3. หลังจากส่วนของข้อมูลดิบเสร็จสิ้น ระบบการจัดการประสิทธิภาพส่วนของสถานีฐาน จะมีโปรแกรมที่จะนำข้อมูลในฐานข้อมูลดิบ มาทำการคำนวณและนำผลลัพธ์ที่ได้มาลงตารางฐานข้อมูลหลังจากคำนวณ (Per-Calculate Database) โดยแยกเป็นแต่ละประเภทของข้อมูล เพื่อให้คนที่ต้องการจะใช้ข้อมูลสามารถนำข้อมูลไปแสดงผลได้ทันที

ปัจจุบันระบบการจัดการประสิทธิภาพส่วนของสถานีฐาน มีการกำหนดพันธะสัญญาการให้บริการ หรือ Service Level Agreement (SLA) ว่า ข้อมูลบนฐานข้อมูลหลังจากคำนวณ (Per-Calculate Database) จะต้องสมบูรณ์มากกว่า 95 % ก่อนเวลา 10 นาฬิกาของวันถัดไป

หลักการของเครื่องมือโปรแกรมที่ใช้ในการวิจัย Oracle Database, ภาษา SQL และ Shell Script

Oracle Database

Oracle Database Server จัดว่าเป็น Object-Relational Database Management System (ORDBMS) ซึ่งหมายความว่า สามารถทำงานได้ ทั้งในรูปแบบ Rational และบางคุณสมบัติของ Object Oriented ได้ โดย Oracle Database Server นั้น มีความสามารถทำงานร่วมกันกับ Software หลายๆ ตัวได้ จากผู้ผลิตหลายราย และสนับสนุนมาตรฐานระบบเปิดต่างๆ (กรมที่ดิน. 2552: Online)

Oracle Database Server จะประกอบด้วย 2 ส่วนหลัก คือ

- Oracle Database จะเป็นส่วนของการจัดเก็บข้อมูล
- Oracle Server Instance จะประกอบด้วย Memory และ Background Process

ในการติดต่อใช้งานกับ Oracle Database นั้น เราต้องใช้ภาษา SQL (ซึ่งบางท่านเรียกว่า SEQUEL) ซึ่งเป็นภาษาที่ใช้ในการกำหนด และจัดการกับ Database (DDL, DML) การทำงานกับ Database ในแบบ Relational Database นั้นหมายความว่า Database Engine จะมีการจัดเก็บ

ข้อมูล ในลักษณะที่เป็นกลุ่มของข้อมูลที่มีความสัมพันธ์กัน ใน 1 Database สามารถที่จะมี Table ตั้งแต่ 1 table เป็นต้นไป และในแต่ละ Table ก็สามารถมีได้หลาย Column หลาย Row

Memory Structure

- SGA
 - Shared pool
 - Library Cache
 - Data Dictionary cache
 - Database Buffer cache
 - Redo buffer cache
- PGA

SGA (System Global Area) เป็นโครงสร้างหน่วยความจำส่วนที่สำคัญมากของ Oracle Server ซึ่งประกอบด้วย buffer cache หลายประเภทด้วยกันตามประเภทการใช้งาน ซึ่งจุดประสงค์โดยรวมก็เพื่อลดเวลาในการทำ I/O ให้มากที่สุดเพื่อประสิทธิภาพที่ดีในการทำงานกับฐานข้อมูล ภายใน SGA นั้น ประกอบด้วย buffer cache ประเภทต่าง ๆ ซึ่งในบทความนี้จะกล่าวถึงเฉพาะส่วนประกอบสำคัญ ๆ ดังนี้

Redo Log Buffers Cache จะทำการบันทึก การเปลี่ยนแปลงที่เกิดขึ้นในฐานข้อมูล โดยการเปลี่ยนแปลงเหล่านี้ก็จะถูกเขียนลงไป Online redo log files ซึ่งการบันทึกการเปลี่ยนแปลงที่เกิดขึ้นนี้สามารถที่จะนำไปใช้ในการ roll-forward ในระหว่างที่ฐานข้อมูลกระทำการกู้ข้อมูล โดยการกระทำนี้จะเกิดขึ้นก่อนที่จะเขียนหรือบันทึกข้อมูลลงไป Online redo log files ขนาดของ Redo Log Buffers Cache จะสามารถตั้งค่าผ่าน LOG_BUFFER parameter ในไฟล์ init.ora ได้

Database Buffer Cache การทำสำเนาข้อมูลที่ถูกเรียกใช้จาก data files โดยจะช่วยให้การทำงานในการเรียกใช้ข้อมูลจากฐานข้อมูลมีความรวดเร็วขึ้นโดยการเก็บข้อมูลที่ถูกเรียกใช้งานเป็นประจำเอาไว้ใน Database Buffer Cache โดยสามารถใช้คำสั่ง DB_BLOCK_SIZE ในการกำหนดเนื้อที่ของ Database Buffer Cache

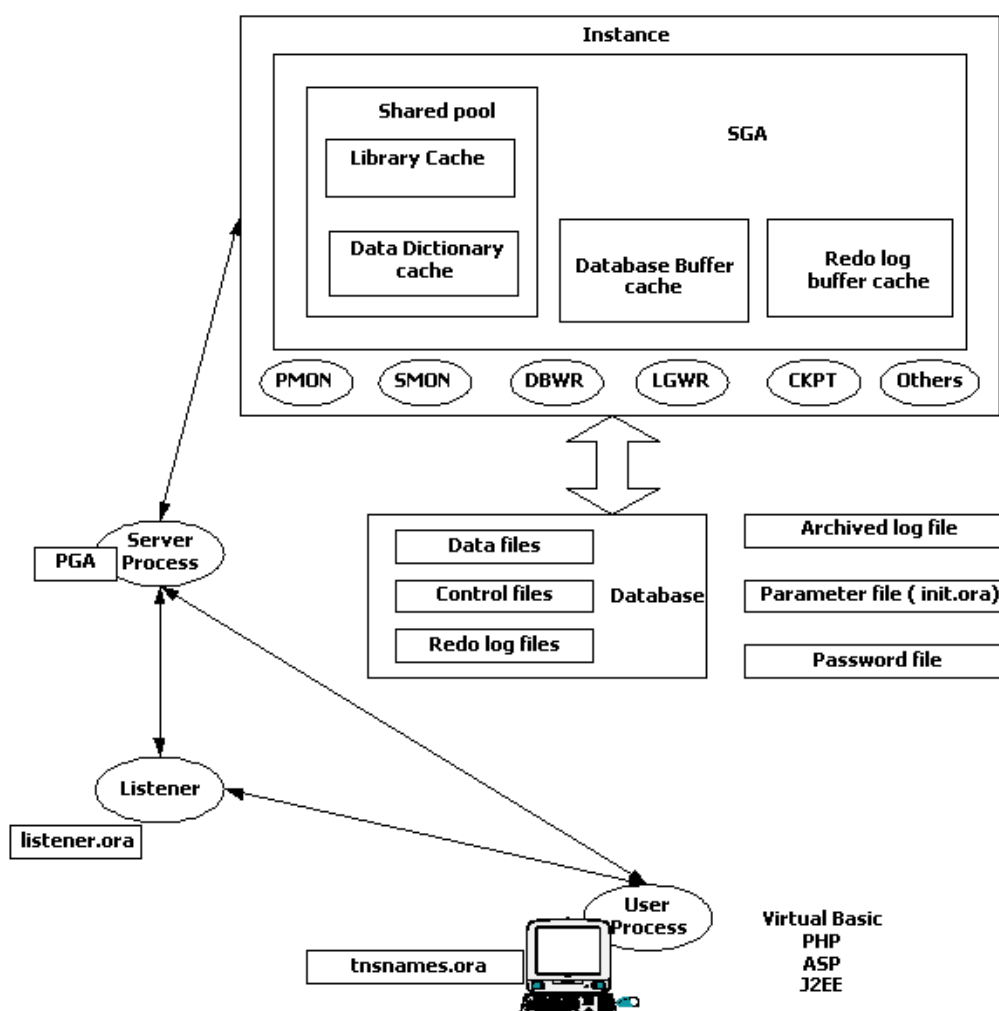
Shared pool ถูกใช้สำหรับเก็บการกระทำของ SQL statement และ รายละเอียดต่างๆของข้อมูล ในการกำหนดขนาดของ Shared pool ใช้คำสั่ง SHARED_POOL_SIZE โดย Shared pool มีส่วนประกอบด้วยกัน 2 ส่วนคือ

Library Buffer Cache จะใช้สำหรับเก็บประโยคคำสั่ง SQL และ PL/SQL ที่ใช้ในการทำงานโดย Library Buffer Cache ประกอบด้วย 2 โครงสร้าง คือ Shared SQL area และ Shared PL/SQL area ขนาดของ Library Buffer Cache จะขึ้นอยู่กับกำหนดจาก Share pool

Data Dictionary Cache จะเก็บรวบรวมรายละเอียดต่าง ๆ ที่ใช้ฐานข้อมูลรวมทั้งข้อมูลที่เกี่ยวข้อง tables, indexes, columns, users, privileges และ object ฐานข้อมูลอื่น ๆ ขนาดของ Data Dictionary Cache จะขึ้นอยู่กับกำหนดจาก Share pool

PGA เป็นพื้นที่ (พื้นที่ส่วนนี้จะถูกจองเมื่อมีการ start ของ server process เกิดขึ้น) ที่อยู่ในหน่วยความจำซึ่งจะช่วย user process ในการกระทำการดังตัวอย่าง เช่น bind variable information, sort areas, cursor information และ session information ซึ่งโดยทั่วไปแล้วในการกระทำการของ SQL statements ใน shared area ถูกเรียกใช้จาก Library cache แล้วคุณสงสัยไหมว่าทำไม user จึงต้องการพื้นที่ในการที่จะกระทำการเป็นของตนเอง เหตุผลก็คือ user ต้องการที่จะกระทำการ select หรือ update โดยไม่มีการถูกใช้ข้อมูลร่วมกันได้ (Shared) เพราะฉะนั้น PGA จึงถูกใช้ในการเก็บข้อมูลจริง ๆ ในที่ซึ่งไม่มีผู้ใดสามารถมองเห็นได้ สำหรับที่จะกระทำ SQL statement

Main components of Oracle Server



ภาพประกอบ 4 Oracle Architecture

ที่มา: กรมที่ดิน. (2008). *Oracle Database Administration*. (Online).

Oracle Server

Oracle Database

- Data files
- Control files
- Redo log files
- Archived log files
- Parameter file
- Password file

An Oracle Database คือส่วนของ Oracle Server ที่จะจัดเก็บข้อมูลต่าง ๆ ไว้ที่ storage ที่เป็นลักษณะ Persistence หรือเรียกง่าย ๆ ว่าถ้าปิดไฟฟ้าแล้วข้อมูลต้องยังอยู่ ซึ่งแน่นอนว่า Disk เป็น Storage ที่เป็นที่ยอมรับที่สุด กล่าวโดยสรุปก็คือ Oracle Database จะถูกจัดเก็บบน Disk ในลักษณะเป็น files โดยแบ่งเป็นประเภท ดังนี้

Data Files คือ ส่วนในการจัดเก็บข้อมูลของระบบฐานข้อมูล ซึ่ง Data File นั้นเป็นชื่อเรียกทาง Physical ในขณะที่ทาง Logical จะเรียกว่า Tablespace โดยจะทำการตัดแบ่งไฟล์ไปจัดเก็บในหน่วยเล็กๆ ที่เรียกว่า Data blocks โดยส่วนหัวของบล็อกจะเรียกว่า Header ซึ่งจะจัดเก็บรายละเอียดของ Data Files เช่นขนาดของไฟล์ ขนาดของบล็อก พื้นที่จัดเก็บตาราง เวลาที่สร้าง เป็นต้น เมื่อมีการเปิดใช้งานข้อมูลในฐานข้อมูล ออราเคิลจะทำการตรวจสอบรายละเอียดของไฟล์ที่ส่วนหัวนี้ เพื่ออ่านข้อมูลใน Data Files มาเก็บในหน่วยความจำชั่วคราวแบบแคช ทำให้สามารถค้นหาข้อมูลได้รวดเร็วมากขึ้นซึ่งเราจะเรียกส่วนนี้ว่า Database Buffer cache

Tablespace อย่างน้อยที่สุดจะต้องมี System Tablespace ซึ่งจะทำหน้าที่เก็บ Data Dictionary ของ Database ทั้งหมด พูดให้ง่ายขึ้นก็คือเป็นข้อมูลอย่างเช่น มี tables, user หรือ objects อื่น ๆ อะไรบ้าง ใครเป็นเจ้าของ เป็นต้น เวลาที่เราทำงานกับฐานข้อมูลของออราเคิล เพื่อประสิทธิภาพในการทำงานที่ดี ขอแนะนำให้สร้าง Tablespace ใหม่ขึ้นมาเพื่อใช้เก็บข้อมูลของแต่ละงานแยกกัน โดยปกติถ้าเราใช้ DBCA เป็นตัวสร้าง Database ตัว DBCA จะสร้าง tablespace อื่นนอกเหนือจาก Systems Tablespace ให้เราด้วยแล้ว

Control Files คือไบนารีไฟล์ซึ่งเก็บข้อมูลเกี่ยวกับระบบปฏิบัติการที่ใช้งานอยู่ชื่อฐานข้อมูล เวลาที่สร้างชื่อ Data Files และ Online Redo Log Files รวมถึง Archived Redo Log Files ด้วย ทุกครั้งที่มีการ mount ฐานข้อมูลก็จะเกิด Control File ขึ้นเพื่อระบุ Data Files และ Online Redo Log Files ที่ต้องใช้ในการทำงานของระบบฐานข้อมูล ถ้าหากมีการเปลี่ยนแปลง เช่นมีการสร้าง Data File หรือ Redo Log File ใหม่ขึ้นมาก็จะทำการบันทึกลงใน Control File ซึ่งโดยปกติแล้วควรจะแยกเก็บเป็น mirror ไว้บนฮาร์ดดิสก์คนละตัวกัน เพื่อป้องกันการ Fail ของฮาร์ดดิสก์ และเก็บเป็นไฟล์นามสกุล .con นอกจากนี้เรายังควรที่จะทำการสำเนาไฟล์นี้เอาไว้เพื่อกรณีฉุกเฉินด้วย

Online Redo Log Files ออราเคิลจะมี Redo Log File เพื่อใช้ในการจัดเก็บการเปลี่ยนแปลงทุกอย่างที่เกิดขึ้นกับ Database เพื่อนำไว้ใช้ในกรณีที่เหตุการณ์ไม่ปกติกับ Database อย่างเช่น มีคนชนตึง plug ไฟฟ้าของเครื่องออก โดยที่ยังไม่ได้ทำการ Shutdown Database ก่อน

Archived Redo Log Files ส่วนนี้จัดเก็บข้อมูลจาก Online Redo Log File ที่มีการจัดเก็บจนเต็มแล้ว โดยแยกเก็บในพื้นที่ภายนอก ที่สามารถขยายได้ (เนื่องจาก Online Redo Log File นั้นเมื่อบันทึกจนเต็มก็จะย้ายไปทำงานที่ตัวถัดไป จนกระทั่งทุกตัวบันทึกเต็มหมดแล้ว ก็จะกลับมาเริ่มที่ตัวแรกใหม่ ซึ่งจะเริ่มเขียนทับข้อมูลเดิม ดังนั้นออราเคิลจึงได้ทำการย้ายไปเก็บใน Archive Redo Log File แทน)ทำให้เราสามารถจัดเก็บข้อมูลย้อนหลังได้มากขึ้นเพื่อว่าเวลาที่เกิดปัญหาขึ้นจะได้สามารถนำ ข้อมูลกลับคืนมาได้ครบถ้วน Archived Redo log files จะเกิดขึ้นในกรณีที่ Database เรากำหนด mode เป็น ARCHIVELOG Mode ซึ่งจะทำให้เราสามารถกู้ข้อมูลกลับคืนมาได้ทั้งหมด ในกรณีที่ปัญหาเกิดขึ้น นอกจากนั้นจะมี file ประกอบการทำงานอีก 2 file คือ

Parameter File ทำหน้าที่ เก็บ Parameter ต่าง ๆ ของ Database ที่เราใช้งานซึ่งรวมไปถึงการกำหนดขนาดของ SGA และพฤติกรรมการทำงานต่าง ๆ ฉะนั้นเวลา DBA เค้าคิดจะ tuning เจ้าตัว Database เค้าก็จะมาปรับแต่งที่ file นี้

Password File ทำหน้าที่เก็บ User และ password ของคนที่มีสิทธิ startup และ shutdown Database ได้

ภาษา Standard relational database Query Language (SQL)

ภาษาที่กลายเป็นภาษามาตรฐานสำหรับระบบฐานข้อมูลคือ ภาษา Standard relational database Query Language หรือเอสคิวแอล (SQL) หรือซีคิวอล (SE-QUEL) ซึ่งเป็นภาษาที่พัฒนาขึ้นมาโดยบริษัทไอบีเอ็ม ภาษา SQL (Standard Query Language) เป็นส่วนหนึ่งของระบบฐานข้อมูลแบบรีเลชันเนล (Relational Database) ที่ได้รับความนิยมมากเพราะง่ายต่อความเข้าใจ และอยู่ในรูปภาษาอังกฤษ ภาษา SQL แบ่งออกเป็น 2 ส่วน คือ (สตีเวน โพรซีเอส 2552: Online)

- ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language-DDL)
- ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language: DML)
 - ภาษาควบคุม (Control Language)
 - ภาษาในการเลือกข้อมูล (Data Query Language)

รูปแบบการใช้คำสั่ง SQL สามารถใช้ได้เป็น 2 รูปแบบ ดังนี้ คือ

• คำสั่ง SQL ที่ใช้เรียกดูข้อมูลได้ทันที (Interactive SQL) เป็นการเรียกใช้คำสั่ง SQL สั่งงานบนจอภาพ เพื่อเรียกดูข้อมูลในขณะที่ทำงานได้ทันที เช่น

```
SELECT CITY
FROM SUPPLIER
```


WHERE SNO = 'SE';

- คำสั่ง SQL ที่ใช้เขียนร่วมกับโปรแกรมอื่น ๆ (Embedded SQL) เป็นคำสั่ง SQL ที่ใช้ร่วมกับคำสั่งของโปรแกรมภาษาต่างๆ เช่น PL/1 PASCAL ฯลฯ หรือแม้แต่ว่ากับคำสั่งในโปรแกรมที่ระบบจัดการฐานข้อมูลนั้นมีใช้เฉพาะ เช่น ORACLE มี PL/SQL (Procedural Language /SQL) ที่สามารถเขียนโปรแกรมและนำคำสั่ง SQL มาเขียนร่วมด้วย เป็นต้น

ตัวอย่างการใช้คำสั่ง SQL ในภาษา PL/1

EXEC SQL SELECT CITY

INTO :XCITY

FROM SUPPLIER

WHERE SNO = 'S4';

2.1 ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language-DDL) เป็นภาษาที่ใช้นิยามโครงสร้างข้อมูล เพื่อเปลี่ยนแปลงหรือยกเลิกโครงสร้างฐานข้อมูลตามที่ต้องการ โครงสร้างดังกล่าวคือ สคีมา (Schema) นั้นเอง ตัวอย่างเช่น การกำหนดให้ฐานข้อมูลประกอบด้วยตารางอะไรบ้าง ชื่ออะไร ประเภทใด มีอินเด็กซ์ (Index) ภาษา DDL ประกอบด้วย 3 คำสั่งคือ

- คำสั่งการสร้าง (Create) ได้แก่ การสร้างตารางและอินเด็กซ์

CREATE TABLE <Table name>

(Attribute 1 Type 1,

Attribute 2 Type 2,)

CREATE Unique Index on X<Table name>

เช่น

CREATE TABLE S11

(SNO CHAR(5) Not NULL,

SNAME CHAR(10) ,

STATUS integer)

CREATE Unique Index XS11 on S11(SNO)

- คำสั่งเปลี่ยนแปลงโครงสร้าง

ALTER TABLE < ชื่อตารางที่ตั้งขึ้น >

<คำสั่งการเปลี่ยนแปลง> (<ชื่อคอลัมน์ ประเภทข้อมูล>);

ตัวอย่างเช่น

ALTER TABLE SUPPLIER

ADD (LAST_SNAME Char(10));

- คำสั่งยกเลิก (Drop) ต่างๆ

การลบโครงสร้างตาราง

DROP TABLE < ชื่อตารางที่ตั้งขึ้น >

ภาษาดังกล่าวคือ ภาษาที่ใช้สร้างฐานข้อมูลลงในคอมพิวเตอร์ หลังจากที่เรารู้ได้ออกแบบแล้วว่า ฐานข้อมูลมีกี่เลชัน แต่ละเลชันมีความสัมพันธ์อย่างไร จากนั้นการใช้ภาษา DDL นี้แปลงเลชันต่างๆ ให้อยู่ในรูปภาษาสำหรับนิยามข้อมูล เพื่อป้อนเข้าสู่ระบบฐานข้อมูล เพื่อสร้างฐานข้อมูลที่แท้จริงให้เกิดขึ้นในคอมพิวเตอร์ ภาษา DDL สามารถสรุปคำสั่งต่างๆ ได้ดังต่อไปนี้

ตาราง 1 คำสั่งเกี่ยวกับโครงสร้างตาราง

คำสั่ง	ความหมาย
CREATE TABLE	นิยามโครงสร้างข้อมูลในรูปตารางบนฐานข้อมูล
DROP TABLE	ลบโครงสร้างตารางข้อมูลออกจากระบบ
ALTER TABLE	แก้ไขปรับปรุงโครงสร้างตาราง
CREATE INDEX	สร้างดัชนีของตาราง
DROP INDEX	ลบดัชนีของตารางออกจากระบบ
CREATE VIEW	กำหนดโครงสร้างวิวของผู้ใช้
DROP VIEW	ลบโครงสร้างวิวออกจากระบบ

คำสั่งนิยามโครงสร้างตาราง

การสร้างตารางในฐานข้อมูลแบบรีเลชันแนล โดยเฉพาะฐานข้อมูลขนาดใหญ่บนระบบ UNIX จะทำการป้อนคำสั่งในลักษณะเท็กซ์โหมด (Text Mode) เข้าไปในระบบฐานข้อมูลดังรูปแบบต่อไปนี้

CREATE TABLE <ชื่อตาราง>

(<ชื่อคอลัมน์ ประเภทของข้อมูล>[,<ชื่อคอลัมน์ ประเภทของข้อมูล>]....);

ประเภทของข้อมูล แบ่งเป็น 5 ประเภทใหญ่ๆ ขึ้นอยู่กับซอฟต์แวร์ระบบฐานข้อมูลที่ใช้ว่าคืออะไร ตัวอย่างเช่น CHAR, INTEGER, DATE ฯลฯ

คำสั่งการลบโครงสร้างตาราง

DROP TABLE <ชื่อตารางที่ต้องการลบ>

คำสั่งการเปลี่ยนแปลงโครงสร้างตาราง

ในกรณีที่ต้องการเปลี่ยนแปลงโครงสร้างตารางที่เคยนิยามไว้ สามารถใช้คำสั่งต่อไปนี้

ALTER TABLE <ชื่อตารางที่ต้องการเปลี่ยนแปลง>

<คำสั่งการเปลี่ยนแปลง>[,<ชื่อคอลัมน์ ประเภทของข้อมูล>]>

คำสั่งดัชนี

ดัชนี (INDEX) มีความสำคัญมากต่อฐานข้อมูลเชิงสัมพันธ์ เนื่องจากระบบฐานข้อมูลแบบรีเลชันแนล (RDBMS) จะใช้ดัชนีในการค้นหาซึ่งเป็นที่ต้องการได้อย่างรวดเร็ว โดยดัชนีที่ถูกสร้างขึ้น จะเก็บไว้แยกจากตารางในพื้นที่ต่างหากของคอมพิวเตอร์ โดยปกติ ถ้าไม่มีการประกาศดัชนีไว้ การค้นหาข้อมูลในตาราง นั้นจะต้องทำ แบบเรียงลำดับจากแถวที่หนึ่งจนถึงแถวสุดท้าย การสร้างดัชนีสำหรับตารางใดๆ จะทำได้โดยการเลือกคอลัมน์ใดคอลัมน์หนึ่งจากตารางมาเป็นดัชนี และตารางหนึ่งๆ สามารถมีได้หลายดัชนี

นอกจากเพิ่มความรวดเร็วในการดึงข้อมูลแล้ว ยังสามารถนำไปใช้ในการควบคุมคอลัมน์ที่นำมาสร้างเป็นดัชนีให้มีการเก็บข้อมูลที่ไม่ซ้ำกัน (Unique) อีกด้วย การสร้างดัชนีจะใช้คำสั่ง CREATE INDEX แล้วตามด้วยชื่อดัชนีที่เราตั้งขึ้น ดังรูปแบบต่อไปนี้

CREATE [UNIQUE] INDEX < ชื่อตารางที่ตั้งขึ้น >

ON (<ชื่อตารางที่สร้างดัชนี> (< ชื่อคอลัมน์_1> [, < ชื่อคอลัมน์_2>]...);

การลบดัชนี เมื่อต้องการลบดัชนีที่สร้างขึ้น ก็สามารถทำได้ด้วยคำสั่ง DROP INDEX แล้วตามด้วยชื่อดัชนีที่ต้องการลบ ดังรูปแบบดังนี้

DROP INDEX <ชื่อดัชนี>

2.2 ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language-DML) หลังจากที่เราสร้างโครงสร้างฐานข้อมูลขึ้นแล้ว คำสั่งต่อไปในการป้อนข้อมูลลงในฐานข้อมูลและเปลี่ยนแปลงข้อมูลในฐานข้อมูล โดยการใช้ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language-DML) ใช้จัดการข้อมูลภายในตารางภายในฐานข้อมูล และภาษาแก้ไขเปลี่ยนแปลงตารางแบ่งออกเป็น 4 Statement คือ

- Select Statement: การเรียกหา (Retrieve) ข้อมูลจาก ฐานข้อมูล
- Insert Statement: การเพิ่มเติมข้อมูลลงใน ตาราง (Table) จาก ฐานข้อมูล
- Delete Statement: การลบข้อมูลลงออกจาก ตาราง (Table) จาก ฐานข้อมูล
- Update Statement: การเปลี่ยนแปลงข้อมูลลงใน ตาราง (Table) จาก ฐานข้อมูล

ตาราง 2 คำสั่งเกี่ยวกับการจัดการข้อมูล

คำสั่ง	ความหมาย
SELECT	เรียกค้นข้อมูลในตาราง
INSERT	เพิ่มแถวข้อมูลลงในตาราง
DELETE	ลบแถวข้อมูล
UPDATE	ปรับปรุงแถวข้อมูลในตาราง

คำสั่งค้นหาข้อมูล (Query Statement)

คำสั่ง SELECT เป็นคำสั่งการเรียกดูข้อมูล หรือ ค้นหาข้อมูล ตามเงื่อนไขที่ระบุ เนื่องจากคำสั่ง SELECT เป็นคำสั่งที่มีรูปแบบการใช้งานที่ง่ายเพื่อช่วยในการค้นหาข้อมูลที่ซับซ้อน ดังมีรูปแบบดังนี้

SELECT <ชื่อคอลัมน์ที่ต้องการดูข้อมูล>

FROM <ชื่อตาราง>

WHERE <เงื่อนไขตามที่ระบุ>

SELECT --- เป็นคำสั่งให้ทำการเรียกดูข้อมูลในคอลัมน์ที่ระบุซึ่งอาจจะมากกว่า หนึ่งก็ได้ และถ้ามีมากกว่าหนึ่งคอลัมน์ต้องคั่นด้วย คอมม่า (,) และนอกจากนี้ยังสามารถใช้เครื่องหมายดอกจัน (*) เพื่อแสดงถึงการขอข้อมูลทั้งหมดได้อีกด้วย

FROM --- เป็นคำส่วนประกอบของคำสั่งที่บอกถึงตารางที่ต้องการดู ซึ่งอาจจะมีมากกว่าหนึ่งตารางก็ได้ ที่จะถูกเรียกใช้จากคำสั่ง SELECT

WHERE--- เป็นส่วนประกอบของคำสั่งที่ใช้บ่งบอกเงื่อนไขที่จะใช้ในการค้นหาข้อมูล ขึ้นมาจากตารางใดๆ ที่อยู่หลัง FROM นี้

การเรียกดูแบบซ้อนกัน (Nested SELECT Statement)

SELECT <ชื่อคอลัมน์>

FROM <ชื่อตาราง>

WHERE <ชื่อคอลัมน์> IN

(SELECT <ชื่อคอลัมน์>

FROM <ชื่อตาราง>

WHERE <ชื่อคอลัมน์>)

คำสั่งเติมข้อมูล (Insert Statement)

INSERT INTO < ชื่อตาราง >

VALUES (< ชื่อคอลัมน์_1> [, < ชื่อคอลัมน์_2>]...);

คำสั่งแก้ไขและลบแถว (Update Statement)

UPDATE < ชื่อตาราง >

SET <ค่าที่ต้องการ>

WHERE <เงื่อนไข>

ภาษาควบคุม (Control Language)

ใช้เป็นภาษาที่ใช้ควบคุมระบบรักษาความปลอดภัยของฐานข้อมูล ประกอบด้วยคำสั่ง 2 คำสั่งคือ

- คำสั่ง GRANT เป็นคำสั่งที่ใช้กำหนดสิทธิให้กับผู้ใช้แต่ละคนให้มีสิทธิกระทำการใดกับข้อมูล เช่น การเพิ่มข้อมูล การแก้ไข หรือ การลบข้อมูลในตารางใดบ้าง

- คำสั่ง REVOKE เป็นคำสั่งให้มีการยกเลิกสิทธิ์นั้นหลังจากที่ได้ GRANT แล้ว

ค่าบูลีน (Boolean-Type Data)

- AND
- OR
- NOT
- =
- > หรือ >=
- < หรือ <=
- <>
- Built-In Function
- COUNT
- SUM
- AVG
- MAX
- MIN
- DISTINCT

ชุดคำสั่ง GROUP BY, ORDER BY และ HAVING

เนื่องจากข้อมูลที่น่าสนใจมักจะเป็น ผลสรุป หรือ ข้อมูลที่ใช้ในการตัดสินใจ ภาษา SQL จึงได้ออกแบบให้มี ชุดคำสั่งพิเศษ ที่ทำ หน้าที่แบ่งออกเป็นกลุ่มๆ ได้แก่

ชุดคำสั่ง GROUP BY เป็นคำสั่งให้มีการจัดกลุ่มแถวข้อมูลตามคอลัมน์ โดยข้อมูลที่เหมือนกันจะถูกจัดให้อยู่ในกลุ่มเดียวกัน ดังนั้นจากตัวอย่างตาราง S

```
SELECT S#, MAX (QTY)
FROM SPJ
GROUP BY S#
```

ตาราง 3 ตัวอย่างผลลัพธ์การใช้คำสั่ง GROUP BY

S#	QTY
S1	700
S2	800
S3	500
S4	300
S5	800

ชุดคำสั่ง ORDER BY เป็นคำสั่งให้มีการเรียงลำดับข้อมูลในแถวข้อมูลตามคอลัมน์ โดยข้อมูลที่เหมือนกันจะถูกจัดเรียงจากน้อยไปมาก ถ้าไม่การระบุ แต่ถ้าระบุว่าเป็น DESC จะเรียงจากมากไปน้อย ดังนั้นจากตัวอย่างต่อไปนี้

```
SELECT S#, MAX (QTY)
FROM SPJ
GROUP BY S#
ORDER BY 2 DESC
```

ตาราง 4 ตัวอย่างผลลัพธ์การใช้คำสั่ง ORDER BY

S#	QTY
S5	800
S2	800
S1	700
S3	500
S4	300

หมายเหตุ ตัวเลข 2 ที่อยู่หลัง ORDER BY จะเป็น การบอกระบบให้ทำ การเรียงลำดับข้อมูลตามลำดับคอลัมน์ที่ 2 ของคำสั่ง SQL ซึ่งก็คือ MAX (QTY) นั่นเอง

ชุดคำสั่ง HAVING

HAVING เหมือนกับคำสั่ง WHERE ที่ต้องตามด้วยเงื่อนไข แต่ HAVING จะใช้ในกรณีที่มีการจัดกลุ่มหรือการใช้ GROUP BY

สรุปชุดคำสั่งมาตรฐานของ SQL

1. CREATE TABLE [[database.]owner.]table_name
 (column_name datatype [not null | null] IDENTITY [(seed, increment)][constraint]
 [, column_name datatype [not null | null IDENTITY [(seed, increment)]]].
 [constraint] ...)
 [ON segment name]
2. DELETE [FROM] table_name
 WHERE column_name = "value"
3. ALTER TABLE [<database.>]<owner.>[<table_name>]

```
ADD <column_name><datatype>NULL [Canstsaint]
```

```
[WITH nocmeck]
```

```
{drop [ ]
```

```
4. UPDATE table_name
```

```
SET column_name= value
```

```
WHERE column_name =operator_value
```

```
5. CREATE VIEW View_name AS
```

```
SELECT column
```

```
FROM table_1 ....
```

```
WHERE table_key_1=table_key_2
```

โปรแกรม Shell Script เป็นได้ทั้งแปลคำสั่งและภาษาโปรแกรม เราจะศึกษาเชลล์ในฐานะที่มันเป็นภาษาโปรแกรมกัน ภาษาโปรแกรมทุกภาษาจะต้องมีประโยคคำสั่งและตัวแปร ประโยคคำสั่งที่สำคัญที่สุดก็คือ การอ่านข้อมูล การแสดงผลข้อมูล และการกำหนดค่าตัวแปร นอกจากนี้ ก็ควรมีประโยคเงื่อนไข ประโยคสำหรับการวนลูป (Loop) และควรจะสามารถสร้างโปรแกรมย่อยได้ รวมทั้งควรจะอนุญาตให้โปรแกรมย่อยเรียกตนเองได้ (recursion) แน่นอนว่าเชลล์คงไม่สามารถแสดงตัวว่าเป็นภาษาโปรแกรม ถ้าไม่มีคุณสมบัติครบถ้วนตามที่กล่าวมานั้น เราสามารถใช้เชลล์เขียนโปรแกรมโดยไม่ต้องพึ่งพาภาษาอื่น ๆ ได้หรือไม่? และถ้าได้ทำไมจึงมักมีผู้กล่าวว่ายูนิกซ์เป็นคู่หูกับ C เพราะถ้าเชลล์ดีจริงก็คงจะไม่ต้องใช้ภาษา C อีกต่อไป ที่จริงโปรแกรมเชลล์ในเชลล์สคริปต์ (เชลล์สคริปต์ก็คือโปรแกรมเชลล์) ได้แก่ การนำเอาคำสั่งเชลล์มาประกอบกันมากกว่า 1 คำสั่ง โดยการนำประโยคเงื่อนไข ลูป และอื่น ๆ รวมทั้งการใช้ตัวแปร มาประกอบกันเป็นโปรแกรมส่วนในแง่ของการเปรียบเทียบระหว่างโปรแกรมเชลล์ กับโปรแกรมทั่ว ๆ ไป สิ่งแรกที่ต้องทราบก็คือ เชลล์เป็นภาษาแบบ interpreter ในขณะที่ภาษาอื่น ๆ ส่วนมากจะเป็นคอมไพเลอร์นั่นก็หมายความว่า เชลล์จะต้องทำการแปล (interpret) โปรแกรมทุกครั้งที่เรารันโปรแกรม ทำให้โปรแกรมเชลล์นี้ทำงานได้ช้ากว่าโปรแกรมที่เขียนในภาษาอื่นทั่ว ๆ ไป แต่จุดเด่นของภาษาเชลล์ก็คือความง่ายในการเขียนโปรแกรม โดยเฉพาะกับลักษณะของงานที่มีคำสั่งเชลล์ใช้อยู่แล้ว นอกจากนี้ตัวแปรในโปรแกรมเชลล์จะเป็นในลักษณะที่ไม่ระบุชนิด (weak type) ทำให้เราไม่ต้องพะวงว่าจะเกิดการใช้ตัวแปรผิดชนิด เช่น การอ่านข้อมูลจากเทอร์มินัลไปเก็บในตัวแปร สามารถใช้ตัวแปรตัวเดียวกันได้ ไม่ว่าข้อมูลที่ใช้พิมพ์นั้นจะเป็นตัวเลข หรือข้อความ จะเห็นว่าในระบบยูนิกซ์เอง ก็มีคำสั่งหลาย ๆ คำสั่งที่เขียนขึ้นด้วยเชลล์ เช่น กระบวนการในการปิดระบบ (shut down) (วิทยาลัยชุมชนภูเก็ต. 2552: Online)

- ตัวแปรเชลล์

- เงื่อนไขอย่างง่าย ๆ

- ประโยคเงื่อนไข - if

- การวนลูปอย่างมีเงื่อนไข : ประโยค while และ until
- การใช้ here document
- ประโยค for
- ประโยค case
- การใช้ break และ continue
- การดีบั๊กโปรแกรมเชลล์
- คำสั่ง read

ตัวแปรเชลล์ เช่นเดียวกับภาษาโปรแกรมทั่ว ๆ ไปที่ใช้ตัวแปรเก็บค่าต่าง ๆ เชลล์ก็มีตัวแปรสำหรับใช้เก็บค่าเหมือนกัน ซึ่งก็มีกฎเกณฑ์คล้าย ๆ กับประโยคกำหนดค่า โดยบังคับไว้ว่าชื่อของตัวแปรจะต้องขึ้นต้นด้วยตัวอักษรเสมอ และถ้าต้องการกำหนดค่าให้เป็นข้อความที่มีเว้นวรรคก็ต้องไม่ลืมใส่เครื่องหมายคำพูด หรือคำพูดเขี้ยวเดียวครอบไว้ เช่น

```
$ author="Duangkaew Sawamiphadi"
```

```
$ U=Unix
```

ถ้าต้องการใช้ตัวแปรใดเฉพาะสำหรับอ่านก็สามารถกำหนดได้ด้วยคำสั่ง readonly ดังนี้

```
$ readonly author
```

```
$ author="other name"
```

```
author : is read only
```

```
$
```

หมายความว่าเราไม่สามารถเปลี่ยนแปลงค่าของ author ได้เพราะตัวแปรเชลล์ที่สร้างขึ้นนี้จะมีขอบเขตการใช้งานอยู่เฉพาะในเชลล์ที่กำลังใช้งานอยู่เท่านั้นเหล่าบรรดาตัวแปรเชลล์ที่สร้างขึ้นมานั้นจะเป็นที่ รู้จักเฉพาะเชลล์ตัวนี้เท่านั้น ที่นี้เมื่อออกคำสั่งอื่น ๆ ต่อไปก็เท่ากับได้สร้างชั้นของเชลล์ขึ้นมาใหม่ และถ้าต้องการให้เชลล์อื่นรู้จักตัวแปรเหล่านี้ก็จำเป็นต้อง export ตัวแปรออกไป ดังนี้

```
$ export author U
```

หลังจากที่ได้กำหนดค่า หรือ เปลี่ยนแปลงค่าให้กับตัวแปรเชลล์แล้วก็สามารถเรียกใช้ตัวแปรเหล่านั้นได้ โดยการใส่สัญลักษณ์ "\$" ไว้หน้าชื่อ เช่น

```
$ echu $author
```

```
Duangkaew Sawamiphakdi
```

```
$
```

ถ้าต้องการต่อค่าของตัวแปรที่เรียกใช้ ก็ต้องใส่วงเล็บปีกกาครอบชื่อตัวแปรเอาไว้ เช่น

```
$ echo ${U}+tm
```

```
Unixtm
```


ระบบนิพจน์แบ่งชนิดของตัวแปรเชลล์ออกเป็น 5 ชนิด ได้แก่

1. ตัวแปรที่ผู้ใช้กำหนดขึ้น (user - definable) ได้แก่ตัวแปรที่ผู้ใช้ตั้งชื่อให้ใหม่ และกำหนดค่าให้ด้วยประโยค

ชื่อ = ค่า

ในการกำหนดค่านี้ ต้องพิมพ์ชื่อตัวแปร เครื่องหมายเท่ากับ และค่าของตัวแปรติดกันไปเลย โดยห้ามมีช่องว่างเด็ดขาด

2. ตัวแปรตามตำแหน่งที่ (positional parameter) ตัวแปรที่เป็น argument ที่ส่งให้กับเชลล์สคริปต์ เช่น ถ้ามีเชลล์สคริปต์ชื่อ s1 และถ้าออกคำสั่ง

\$ s1 a1 a2 a3

จะถือว่า s1,a1,a2, และ a3 เป็น argument ที่ส่งค่าให้ตัวแปรตามตำแหน่งที่ อันได้แก่ \$0,\$1,\$2 และ \$3 ตามลำดับ

3. ตัวแปรที่กำหนดไว้แล้ว (predefined) คือบรรดาตัวแปรที่เชลล์กำหนดไว้เรียบร้อยแล้ว ซึ่งแบ่งออกเป็น 5 แบบ คือ

3.1 \$? หมายถึง สถานะการทำงานของคำสั่งครั้งสุดท้าย

3.2 \$\$ หมายถึง หมายเลขของโปรเซสของเชลล์

3.3 !\$ มีค่าเท่ากับหมายเลขของโปรเซสที่อยู่เบื้องหลัง

3.4 \$# คือ จำนวนของ argument ที่ส่งให้เชลล์สคริปต์ หรือคำสั่ง

3.5 \$- เก็บค่า flag ที่คำสั่งเชลล์เซตไว้

4. ตัวแปรคำสั่ง (command substitution) คือ ตัวแปรที่ถูกกำหนดค่าด้วยคำสั่ง และจะแสดงผลที่ได้จากการทำงานของคำสั่งนั้นเมื่อเรียกใช้

5. ตัวแปรพารามิเตอร์ (parameter substitution) คือลักษณะของตัวแปรที่ค่าของมันขึ้นอยู่กับค่าของพารามิเตอร์ที่ส่งให้แก่คำสั่งเชลล์สคริปต์นั้น ตัวอย่างที่จะแสดงการใช้ตัวแปรแต่ละชนิดกัน

ตัวอย่างที่ 1: การใช้ตัวแปรที่ผู้ใช้กำหนด ลองดูเชลล์สคริปต์ ที่บรรจุอยู่ในไฟล์ loginfo ต่อไปนี้

\$ cat loginfo

TIME="current date and time : \C"

NOUSERS="The number of users : \C"

ME="My accout is : \C"

echu "\$TIME"

date

echo "\$NOUSERS"

who | wc -1

```
echo "$ME"
```

```
who am I
```

```
exit 0
```

```
$
```

ในตัวอย่างนี้เป็นการแสดง วัน เวลา วันที่ จำนวนผู้ใช้ระบบ และชื่อบัญชี ซึ่งถ้ารันดูก็จะ
ได้ผลดังนี้

```
$ loginfo
```

```
The current date and time : Thu Nov 5 9:45:10 BKK 1987
```

```
The number of users: 4
```

```
My account is: dks tty02 Nov 5 9:30:08
```

```
$
```

ในตัวอย่างได้กำหนดตัวแปรขึ้นมา 3 ตัวคือ TIME, NOUSERS และ ME และเนื่องจากค่า
ของตัวแปรนี้มีช่องว่างอยู่ระหว่างคำ จึงต้องใส่เครื่องหมายคำพูดครอบไว้ และอย่าลืมว่าการที่ใส่
เครื่องหมายคำพูดครอบชื่อไว้ตอน echo นี้ ทำให้เชลล์แปลความหมายของ \$ ว่าหมายถึงค่าตัวแปร
ส่วนการใส่ "\c" ไว้ท้ายข้อความ คือ การบังคับ echo ว่าเมื่อแสดงข้อความแล้วไม่ต้องขึ้นบรรทัดใหม่
การจบเชลล์สคริปต์ด้วยคำสั่ง exit 0 เป็นการจบการทำงานอย่างสมบูรณ์ คือ ส่งค่าสถานะเป็น 0
กลับไปให้เชลล์ ในยูนิกซ์นี้ ถ้าการทำงานของโปรแกรมใด ๆ จบอย่างไม่สมบูรณ์จะให้ค่าสถานะที่
มากกว่า 0 ออกมา

ตัวอย่างที่ 2: การใช้ตัวแปรตามตำแหน่ง ลองดูตัวอย่างของไฟล์ who tty ดังต่อไปนี้

```
$ cat whotty
```

```
who | grep $1
```

เป็นการออกคำสั่ง เช็คดูว่าผู้ใช้ระบบตามชื่อที่ถามหรือเปล่า เช่น ถ้าออกคำสั่ง

```
$ whotty dks
```

```
dks tty02 Nov 5 9:30
```

```
$ whotty somsak
```

```
$
```

จะให้ผลลัพธ์ว่า ตอนนี้ผู้ที่สนใจกำลังใช้ระบบอยู่หรือไม่ และถ้าใช่ จะอยู่ที่เทอร์มินัลตัว
ไหน ในที่นี้นำคำสั่ง who มาไปห้ grep โดยให้ grep ค้นหาตามทีระบุใน argument ตัวแรก
ซึ่งในตัวอย่างที่ 1 \$1 จะมีค่าเป็น dks และในตัวอย่างที่ 2 \$1 จะมีค่าเป็น somsak ในการใช้ตัวแปร
ตามตำแหน่งที่นี้ สามารถส่งจำนวน argument เข้าไปมากกว่าตัวแปรตามตำแหน่งที่ที่เรียกใช้ก็ได้
ซึ่งในกรณีนี้เชลล์ก็เพียงแต่ไม่ต้องสนใจส่วนที่เกินเท่านั้น เช่น ถ้าออกคำสั่ง

```
$ whotty dks junk
```

ก็จะให้ผลเหมือนเดิม

ตัวอย่างที่ 3: การใช้ตัวแปรที่กำหนดไว้แล้ว ลองดูเชลล์สคริปต์ที่บรรจุอยู่ในไฟล์ showpre และผลการทำงานดังแสดงต่อไปนี้

```
$ cat showpre.
echo The number of arguments is $#.
cc long.c &
echo The process id of the cc command was $!.
echo The process id of the shell is $$$.
who | grep dks
echo The return code from grep was $?..
echo The current set of flags is $ -.
$ showpre a1 a2 a3
The number of arguments is 3
The process id of the cc command was 4186.
The process id of this shell is 4103.
Dks tty02 Nov 5 9:30
The return code from grep was 0
The current set of flags is.
$
```

ลองมาดูผลที่ได้กันก่อน ใน echo อันแรกให้แสดงค่าของ \$# ซึ่งคือจำนวนของ argument อันได้แก่ 3 เพราะออกคำสั่งโดยส่ง argument ไป 3 ตัว (a1, a2, a3) ในลำดับต่อไป ก็ให้แสดงค่าหมายเลขของโปรเซสที่อยู่เบื้องหลังอันได้แก่ โปรเซส cc (คอมไพเลอร์ภาษา c) และโปรเซสของเชลล์ ต่อมาก็ออกคำสั่ง who ไฟฟ้าไปยัง grep ซึ่งในที่นี้การทำงานเสร็จสมบูรณ์ จึงส่งค่าสถานะคือ \$? เป็น 0 ออกมาและในส่วนสุดท้ายแสดงค่า \$ - คือ ค่า flag หรือทางเลือกไว้เลยถ้าออกคำสั่ง

```
$ sh -ek showpre
```

ก็จะได้ผลคล้ายคลึงกัน ยกเว้นค่า \$# ซึ่งตอนนี้จะเป็น 0 เพราะไม่มี argument เลย และจะได้ค่า \$ - เท่ากับ ek

ตัวอย่างที่ 4: การใช้ตัวแปรคำสั่ง การกำหนดตัวแปรให้มีค่าเป็นผลลัพธ์ที่ได้จากคำสั่งนั้นจะใช้เครื่องหมาย " " หรือ accents grave ล้อมกรอบส่วนของคำสั่งเอาไว้ ตัวแปรชนิดนี้มีความแตกต่างจากตัวแปรที่ผู้ใช้งานกำหนดขึ้นคือ เวลาที่เรียกใช้ตัวแปร ชื่อของมันจะถูกแทนค่าด้วยผลที่เกิดจากการทำงานของคำสั่งนั้นลองดูตัวอย่างต่อไปนี้

```
$ now='date'
$ echo $now1
The Nov 5 12:08:10 BKK 1987
$ now2="date"
```

\$ echo \$now2

Date

เป็นการแสดงให้เห็นความแตกต่างระหว่างตัวแปร 2 ชนิดนี้ โดย now1 เป็นตัวแปรคำสั่ง ซึ่งจะแทนที่ค่าตัวแปรด้วยผลลัพธ์ของคำสั่งนั้น ในขณะที่ now2 คือ ตัวแปรที่ผู้ใช้กำหนดขึ้น ตัวอย่างที่ 5 ตัวแปรพารามิเตอร์ ค่าของตัวแปรพารามิเตอร์ขึ้นอยู่กับค่าของพารามิเตอร์ที่ส่งให้กับ เซลล์สคริปต์ ที่มีรูปแบบการเขียน ซึ่งให้ความหมายที่แตกต่างกัน 4 รูปแบบ

ตาราง 5 แสดงรูปแบบการเขียน และความหมายของ Shell script

รูปแบบการเขียน	ความหมาย
<code>\${ พารามิเตอร์ - คำ }</code>	ตัวแปรจะมีค่าเท่ากับค่าของพารามิเตอร์ ถ้าได้มีการเซตไว้ มิฉะนั้นตัวแปรจะมีค่าเท่ากับค่า
<code>\${ พารามิเตอร์ = คำ }</code>	ถ้าค่าพารามิเตอร์ยังไม่ได้เซตเอาไว้ก็就会被เซตให้มีค่าเท่ากับคำนี้
<code>\${ พารามิเตอร์ ? คำ }</code>	ถ้าพารามิเตอร์ได้เซตไว้ก็จะใช้แทนค่าตัวแปร มิฉะนั้นจะพิมพ์ค่า และออกจากเซลล์ แต่ถ้าไม่มีค่าในส่วนของตัวแปรก็จะพิมพ์ ข้อความมาตรฐานออกมา
<code>\${ พารามิเตอร์ + คำ }</code>	ถ้าได้เซตค่าพารามิเตอร์ไว้ ก็จะแทนค่าด้วยค่ามิฉะนั้นก็จะไม่แทน ค่าด้วยอะไรทั้งสิ้นตัวอย่างการทำงานของตัวแปร

```
$ cat parasub
```

```
DIR=$ { 1 - $HOME }
```

```
Echo DIR=$DIR
```

```
Echo 2=${ 2?"Positional parameter must be set " }
```

```
TMP=/tmp/$$
```

```
T=${ TMP+"a legal value" }
```

```
Echo FRED=$FRED
```

```
Exit 0
```

```
$
```

ในบรรทัดแรกเป็นการกำหนดให้ตัวแปร DIR มีค่าเท่ากับไดเรกทอรีบ้าน หรือค่าของ พารามิเตอร์ตัวแรกขึ้นอยู่กับว่า ในคำสั่งที่เรียกใช้ parasub ได้ใส่พารามิเตอร์ไว้หรือไม่ ส่วนใน บรรทัดที่ 3 ถ้าพารามิเตอร์ตัวที่ 2 ในตอนที่เรียกใช้ก็จะแสดงค่าของพารามิเตอร์ตัวนั้นออกมา มิฉะนั้นก็จะพิมพ์ข้อความหลัง " ? " ออกมา เช่น ถ้าออกคำสั่ง

```
$ parasub p1
```

```
DIR=p1
```

```
Parasub : 2 : Positional parameter must be set
```

ก็ได้ผลตามที่คิดไว้ เนื่องจากไม่มีพารามิเตอร์ตัวที่ 2 ดังนั้นจึงมีการพิมพ์ข้อความในส่วนหลัง "?" และจะเลิกทำงานทันที ดังนี้

```
$ parasub p1 p2
```

```
DIR=p1
```

```
2=p2
```

```
TMP=a legal value
```

```
FRED=
```

```
$
```

ตอนนี้มีพารามิเตอร์ตัวที่ 2 จึง echo 2=p2 ออกมา และในประโยคที่ 4 ได้กำหนดค่าตัวแปร TMP เอาไว้ จะได้ว่า \$\$ หมายถึงเบอร์ของโปรแกรมที่กำลังทำงานอยู่

เงื่อนไขอย่างง่าย ๆ ส่วนประกอบที่สำคัญของภาษาโปรแกรมทั่วๆ ไปคือ ความสามารถในการทดสอบเงื่อนไข และความสามารถที่จะระบุให้ทำงานส่วนไหน ถ้าผลจากเงื่อนไขเป็นจริง และจะทำงานอย่างไรถ้าการทดสอบเงื่อนไขให้ผลเท็จ เงื่อนไขที่เชลล์ทดสอบได้อันเป็นสิ่งที่จะพูดถึงเป็นอันดับแรกคือ การทดสอบสถานะจากการทำงาน ได้ทราบแล้วว่า ในการทำงานของคำสั่งใด ๆ ถ้าการทำงานเสร็จเรียบร้อยสมบูรณ์ ระบบจะกำหนดค่าของสถานะการทำงาน (Exit status) ให้เป็น 0 และในทางตรงกันข้าม ค่าของสถานะการทำงานจะไม่ใช่ 0 ถ้าคำสั่งนั้นทำงานไม่สำเร็จ ตัวอย่างเช่น การออกคำสั่ง

```
$ cd /usr/acct/sumalee
```

ถ้าสามารถเคลื่อนย้ายไปที่บ้านของ sumalee ได้สำเร็จก็จะได้ค่าสถานะของการทำงานของคำสั่งนี้เป็น 0 แต่ถ้าเกิดว่าไม่ได้รับอนุญาตให้เคลื่อนไปได้หรือไม่มีไคเรกเทอร์รี่ที่นั่น อยู่ในระบบ สถานะการทำงานของ cd ก็จะมีค่าเป็น 1 สำหรับผู้ที่รู้จักภาษา C คงจะทราบแล้วว่าค่า "ถูก" ในภาษา C คือค่าที่ไม่ใช่ 0 และค่าเท็จคือค่า 0 ดังนั้น ก็ขอให้ตั้งสติให้ดีสำหรับผู้ที่ยังคุ้นเคยกับ C เพราะเชลล์จะให้ค่าถูกและค่าผิดตรงกันข้ามกับ C ขอย้ำอีกทีว่าเป็นระบบยูนิกซ์ ค่าถูกคือ 0 และค่าผิดคือค่าที่ไม่ใช่ 0 เกี่ยวกับเรื่องของเงื่อนไขนี้ โดยถ้าเขียนเครื่องหมายตัวคือ " && " และ " || " ที่ใช้งานในเรื่องนี้ โดยถ้าเขียนเครื่องหมาย " && " คั่นกลางระหว่างคำสั่ง 2 คำสั่งจะยังผลให้คำสั่งที่ 2 ทำงานก็ต่อเมื่อผลจากการทำงานของคำสั่งแรกให้ค่าสถานะเป็น 0 ตัวอย่างเช่น

```
$ test -d /usr/acct && echo success!
```

จะพิมพ์ข้อความว่า "success!" ถ้า /usr/acct เป็นไคเรกเทอร์รี่ ในตัวอย่างนี้จะเห็นว่าใช้ test เป็นตัวทดสอบ โดยใส่ทางเลือก -d เพื่อทดสอบความเป็นไคเรกเทอร์รี่ คำสั่ง test นี้ให้ประโยชน์

ในการใช้งานมาก ในทางที่ตรงกันข้าม การใช้ "|" คั่นกลางระหว่าง 2 คำสั่ง จะยังผลให้คำสั่งที่ 2 ทำงานก็ต่อเมื่อคำสั่งแรกส่งผลให้สถานะที่ค่าที่ไม่ใช่ 0 เช่น

```
$ test -d /usr/acct || echo failure!
```

จะพิมพ์ข้อความ " failure! " ถ้าปรากฏว่า /usr/acct ไม่ใช่ไดเรกทอรีที่มีอยู่แล้วในระบบ
ประโยคเงื่อนไข – if

การใช้ "&&" และ "|" จะเหมาะกับเฉพาะเวลาที่สร้างเงื่อนไขการทำงานอย่างง่าย ๆ เท่านั้น ในเวอร์ชันแรก ๆ ของเชลล์ยังไม่มีประโยค if ใช้ แต่หลังจากที่ได้เริ่มมีการออกแบบพัฒนาประโยค if โดยเริ่มใส่ไว้ในเวอร์ชัน 6 เป็นต้นมา ก็ได้ผลมหาศาลในการเขียนโปรแกรมเชลล์ที่เดียวรูปแบบของประโยค if คือ

```
if เงื่อนไข
then กลุ่มคำสั่ง
else กลุ่มคำสั่ง
fi
```

ซึ่งจะทำการทดสอบคำสั่งตัวสุดท้ายในเงื่อนไข ถ้าให้ผลสถานะเป็น 0 ก็จะลงมาทำงานในส่วน of then มิฉะนั้นก็จะลงมาทำงานในส่วน of else ส่วน of else นี้ จะมีหรือไม่มีก็ได้ คำที่ขีดเส้นใต้เอาไว้บ่งบอกว่าเป็นคำเฉพาะ (reserved word) ที่ภาษาเชลล์บังคับไว้ แต่เวลาใช้งานจริง ๆ ก็ไม่ต้องขีดเส้นใต้ตัวเหล่านี้ การนำประโยค if มาใช้ร่วมกับคำสั่ง test จะเป็นเรื่องที่พบเห็นกันบ่อย ๆ เช่น

```
date > /dev/1p00
if test -r errfile
then
    cat errfile > /dev/1p00
    rm errfile
else
    echo " No errors this hour " >/dv/1p00
fi
```

เริ่มด้วยการส่งวันและวันที่ออกไปยังเครื่องพิมพ์ และเช็คว่ามีไฟล์ errfile อยู่หรือไม่ ถ้ามี ก็จะส่งข้อมูลในไฟล์นี้ไปออกที่เครื่องพิมพ์เช่นกัน มิฉะนั้นก็จะส่งข้อความว่าไม่มี error ไปออกที่เครื่องพิมพ์แทน อาจจะมีประโยค if ซ้อนกันอยู่หลาย ๆ ประโยคก็ได้ตามรูปแบบนี้

```
if
then.....
else if .....
fi
```

ซึ่งแทนที่จะเขียน else if อาจเขียนย่อๆ ได้ด้วย elif ซึ่งถ้าใช้อย่างนี้ก็ไม่ต้องมี fi ปิดท้าย
ดังนี้

```
if...
then...
elif...
.
fi
```

อีกตัวอย่าง

```
NAME1=$( 1- ROOT)
NAME2=$( 2- ROOT)
BB=/dev/null
If who | grep $NAME1 > $BB 2>$BB
then
    echo $NAME1 is currently using the system
elif who | grep $NAME2 > $BB 2>$BB
```

ในตัวอย่างนี้ได้นำเอาตัวแปรแบบพารามิเตอร์ มาใช้ "2>..." หมายถึงการเบี่ยงเบนทางแสดงข้อผิดพลาด

คำสั่ง test เราจะเห็นการใช้ test กันอยู่บ่อยๆ ในโปรแกรมเชลล์ เพราะมันสามารถใช้การทดสอบอะไรได้หลายอย่าง ขึ้นอยู่กับทางเลือกที่ใส่ไว้ในคำสั่ง โดยถ้าการทดสอบได้ผลเป็นจริง ก็จะส่งผลเป็นค่าสถานะเท่ากับ 0 ทางเลือกต่างๆที่ใช้ร่วมกับคำสั่ง test ที่ควรรู้จักมีดังนี้

ตาราง 6 แสดงการใช้งานคำสั่ง Test

คำสั่ง	ทางเลือก
test -f file	ทดสอบว่า เป็น file (ไม่ใช่ไดเรกทอรี) ที่มี อยู่แล้ว
test s	ทดสอบว่า s ไม่ใช่ข้อความว่าง
test -r file	ทดสอบว่าสามารถ "อ่าน" file ได้
test -w file	ทดสอบว่าสามารถ "เขียน" หรือทำการแก้ไข file ได้
test -d file	ทดสอบว่า file เป็นไดเรกทอรี
test -s file	ทดสอบว่าเป็น file ที่มีอยู่แล้ว และมีขนาดมากกว่า 0 ไบต์
test -c file	ทดสอบว่า file เป็นไฟล์ที่มีอยู่แล้ว และเป็น ไฟล์พิเศษแบบคาแรกเตอร์
test -b file	ทดสอบว่า file เป็นไฟล์ที่มีอยู่แล้ว และเป็นไฟล์พิเศษแบบบล็อก

ที่ผ่านมาเป็นการแสดงทางเลือกที่ใช้ในการทดสอบไฟล์ เอาไว้ นอกจากนี้ เรายังสามารถใช้คำสั่ง `test` ในการเปรียบเทียบ ค่าตัวเลขได้ โดยมีถึง 6 ทางเลือก

ตาราง 7 แสดงความหมายและทางเลือกของพารามิเตอร์

ทางเลือก	ความหมาย
<code>eq</code>	เท่ากับ
<code>ne</code>	ไม่เท่ากับ
<code>gt</code>	มากกว่า
<code>ge</code>	มากกว่าหรือเท่ากับ
<code>Lt</code>	น้อยกว่า
<code>Le</code>	น้อยกว่าหรือเท่ากับ

และในท้ายสุดเกี่ยวกับคำสั่ง `test` คือ ความสามารถในการเปรียบเทียบข้อความ โดยมีทางเลือกดังต่อไปนี้

ตาราง 8 แสดงความหมายและทางเลือกของพารามิเตอร์

ทางเลือก	ความหมาย
<code>=</code>	เท่ากับ
<code>!=</code>	ไม่เท่ากับ
<code>-z</code>	ข้อความ ข้อความที่มีความยาว 0
<code>-n</code>	ข้อความ ข้อความที่มีความยาวมากกว่า 0

และก็ต้องใส่ข้อความที่จะทดสอบไว้ในเครื่องหมายคำพูดเสมอ และการใช้คำสั่ง `test` นี้สามารถใช้ร่วมกับทางเลือก `-a`, `-o` และ `!` อันหมายถึง "และ" "หรือ" และ "ไม่" ได้อีกด้วย ดังตัวอย่างต่อไปนี้

```
if test $ # -ne 1 -o -z "$1" #ถ้าจำนวน argument ไม่เท่ากับ 1 หรือ argument ตัวแรกเป็นข้อความที่มีความยาวเป็น 0
```

```
then
```

```
    echo usage: $0 non-zero-string
```

```
RC=1
```



```

else
    echo "The name you have select is \c"
if test "$1"=bob -o "$1"!=Bob # ทดสอบว่า argument ตัวที่1 เป็นข้อความbob หรือ
Bob?
then
    echo "$1,"
elif test "$1"!=tom -a "$1"!=Tom # ทดสอบว่าไม่ใช่ tom และไม่ใช่ Tom?
then
    echo " Not tom or Tom,it's $1,"
else
    echo "$1,"
fi
RC=0
fi
exit $RC

```

ในภาษาเชลล์จะถือว่าข้อความที่อยู่หลังเครื่องหมาย "#" เป็นข้อความ comment หรือคำอธิบายว่ากำลังทำอะไรอยู่ แต่ ที่ยังไม่ได้กล่าวถึง คือการกำหนดค่าให้กับ RC จะเห็นว่าในกรณีที่เงื่อนไขแรกเป็นจริง (คือลักษณะการใช้คำสั่งที่ไม่มี argument ตัวที่ 1 หรือ argument ตัวที่ 1 มีข้อความยาว 0) ค่า RC จะถูกกำหนดให้เป็น 1 เพราะเปรียบเสมือนการส่งงานที่ไม่ถูกต้องสมบูรณ์ ในขณะที่ถ้าทำงานสมบูรณ์ดีแล้วโปรแกรมของจะกำหนดให้ RC มีค่าเป็น 0 และเมื่อจบการทำงานก็จะส่งค่าสถานะออกมาให้ตามความเหมาะสมด้วยประโยค exit \$RC

การวนลูปอย่างมีเงื่อนไข : ประโยค **while** และ **until**

การใช้ประโยค **while** และ **until** ทำให้สามารถทำงานกลุ่มคำสั่งได้หลายๆครั้งตรงเท่าที่ผลจากการทดสอบเงื่อนไขหลัง **while** หรือ **until** ให้สถานะที่เป็นจริง รูปแบบของประโยค **while** คือ

```

while เงื่อนไข
do กลุ่มคำสั่ง
done

```

สมมุติว่า ต้องการเขียนโปรแกรมที่จะหยุดการทำงานไว้จนกว่าบางไฟล์จะถูกลบทิ้ง ก็
สามารถนำประโยค **while** มาใช้ได้ดังนี้

```

while test -f lockfile
do sleep 5
done

```

ในที่นี้ถ้ายังมีไฟล์ lockfile อยู่ ก็จะรอไป 5 วินาที ก่อนที่จะทดสอบใหม่ เมื่อใดก็ตามที่ lockfile ถูกลบทิ้งไปแล้ว ก็จะกระโดดออกมานอก loop ประโยค until นั้นก็คล้ายคลึงกับประโยค while แต่ until จะทำงานในกลุ่มคำสั่งก่อนที่จะมีการทดสอบเงื่อนไข เช่น ถ้าต้องการรอจนกว่าจะมีไฟล์ที่ระบุไว้ถูกสร้างขึ้นมา (ด้วยคนอื่น) ก็สามารถเขียนโปรแกรมโดยใช้ประโยค while และ until ดังนี้

```
while test ! -f proceedfile; do sleep 1 ; done
```

หรือ

```
until test -f proceedfile; do sleep 1 ; done
```

ซึ่งทั้ง 2 โปรแกรมนี้จะทำงานเหมือนกันคือ ตราบใดที่ยังไม่มีไฟล์ proceedfile ก็รอไป 1 วินาที และกลับขึ้นมาทดสอบใหม่ ในตัวอย่างนี้ใช้เครื่องหมาย ' ; ' คั่นระหว่างประโยค เพราะสำหรับเชลล์แล้วไม่ว่าจะเขียนโดยการขึ้นบรรทัดใหม่ หรือใช้เครื่องหมาย " ; " ก็จะทำให้ผลเหมือนกัน ตัวอย่าง ที่จะแนะนำให้รู้จักการใช้โปรแกรม expr และคำสั่ง shift ด้วย

```
$ cat pgm
```

```
NUMBER=0
```

```
until test $#=0
```

```
do
```

```
    NUMBER= 'expr $NUMBER + 1 '
```

```
    echo Argument $NUMBER is $1
```

```
    shift
```

```
done
```

expr คือ โปรแกรมที่รับค่า argument และทำการคำนวณผลลัพธ์ตามที่ระบุใน argument และแสดงผลพร้อมออกทางจอภาพ ในตัวอย่างนี้ได้กำหนดให้ตัวแปรคำสั่ง NUMBER เก็บผลคำสั่ง expr ซึ่งจะทำการเพิ่มค่าของ NUMBER ทีละ 1 ส่วน shift เป็นคำสั่งที่ใช้ในการเลื่อนตำแหน่งที่ของพารามิเตอร์ คือ ถ้าออกคำสั่ง shift ที่หนึ่ง พารามิเตอร์ตัวที่ 2 หรือ \$2 ก็จะเป็น \$1 และ \$3 กลายเป็น \$2 เป็นต้น ดังนั้นผลจากการทำงานของโปรแกรมนี้คือ

```
$ pgm a1 a2 a3
```

```
Argument 1 is a1
```

```
Argument 2 is a2
```

```
Argument 3 is a3
```

```
$
```

หรือถ้าออกคำสั่งโดยไม่มีพารามิเตอร์เลยก็จะไม่พิมพ์อะไรออกมาเลย เพราะพอเริ่มต้น \$# ก็เท่ากับ 0 แล้วจึงไม่ได้ลงมาทำใน loop ดังนี้

```
$ pgm
```

การใช้ here document หน้าหนึ่งของ here document จะใช้ในการเขียนแบบสือข้อมูลมาตรฐานในโปรแกรมเชลล์ รูปแบบของการใช้ here document คือ

คำสั่ง << สัญลักษณ์

.

. ส่วนของ here document

.

สัญลักษณ์การระบุการใช้ here document ทำได้โดยอาศัยเครื่องหมาย "<<" ซึ่งมีความหมาย คือ คำสั่งจะทำการอ่านข้อมูลจากส่วนที่เป็น here document และอ่านไปเรื่อยๆ จนกว่าจะเจอสัญลักษณ์ที่ระบุไว้ ตัวอย่าง เช่น

```
mail root << !
```

```
* * * * PROBLEMS * * * *
```

```
I have accidentally removed a few files listing
```

```
as follows :
```

```
ass1.c
```

```
main1.c
```

```
Is it possible to recover these files from back-up?
```

```
If so, please kindly handle.
```

```
Thank you very much.
```

```
!
```

เป็นการส่งจดหมายไปหา root โดยส่งข้อความไปจนกว่าจะเจอสัญลักษณ์ " ! "

ประโยค for ทำให้สามารถกระทำกลุ่มคำสั่งเป็นจำนวนครั้งตามที่กำหนดรูปแบบของประโยคคือ

```
for name in word1 word2.....
```

```
do กลุ่มคำสั่ง
```

```
done
```

โดยเริ่มแรกด้วยการทำงานในกลุ่มคำสั่งโดยกำหนดให้ name มีค่าเป็น word1 เสร็จแล้วก็จะเปลี่ยนค่า name เป็น word2 และทำกลุ่มคำสั่งซ้ำอีกเรื่อยๆไปจนกระทั่ง name มีค่าครบตามที่ระบุไว้ เช่น

```
for i in 1 2 3 4 5 6 7 8 9 0
```

```
do
```

```
echo
```

```
done
```

จะได้ผลลัพธ์เมื่อรันโปรแกรมเป็นดังนี้

```
1
```

```
2
```

3
4
5
6
7
8
9
0

อาจใช้ประโยค for โดยไม่มีคำว่า " in " ก็ได้ ซึ่งจะมีรูปแบบ ดังนี้

```
for name
do กลุ่มคำสั่ง
done
```

ซึ่งจะมีผลให้ทำกลุ่มคำสั่งหลายๆ รอบตามจำนวนของพารามิเตอร์ที่ใส่ไว้ในเวลาที่ออกคำสั่ง เช่น ตัวอย่างต่อไปนี้จะใช้ประโยค for ทำการพิมพ์ค่าพารามิเตอร์โดยการสลับตำแหน่งจากท้ายมาออกหัว

```
list = ""
for arg
do
    list = "$arg $list"
done
echo $list
```

โดยเริ่มต้นด้วยให้ list เป็นข้อความว่าง และอ่านค่าพารามิเตอร์ทีละตัวไปปะเข้าไปทางหัวของ list เมื่อวนจนครบ ก็จะได้ว่าพารามิเตอร์ตัวสุดท้ายจะอยู่ที่หัวของ list ในลักษณะที่สลับกับของเดิมนั่นเอง

ประโยค Case ในกรณีที่มีทางเลือกหลายๆทาง ถ้าเรามัวมาใช้ประโยค if ซ้อนกันอยู่เยอะๆ ก็จะทำให้โปรแกรมดูรุงรัง และยากที่จะเข้าใจ ประโยค case จึงได้รับการพัฒนาขึ้นเพื่องานนี้โดยเฉพาะ โดยมีรูปแบบของ case คือ

```
case word in
รูปแบบ 1) กลุ่มคำสั่ง;;
รูปแบบ 2) กลุ่มคำสั่ง;;
.
.
esac
```

โดยจะเปรียบเทียบค่าของ word กับรูปแบบต่างๆ ว่า word มีค่าตรงกับรูปแบบใด รูปแบบแรกที่มีค่าตรงกับ word จะได้รับการทำงาน และเมื่อทำงานเสร็จแล้วก็จะกระโดดออกมา นอกประโยค case เราสามารถใช้ตัวย่อ " * " " ? " และ [. .] ประกอบขึ้นเป็นรูปแบบได้ และนอกจากนี้ยังสามารถนำรูปแบบย่อยๆ มาประกอบเป็นรูปแบบได้ โดยใช้เครื่องหมาย " | " คั่น เพื่อแสดงว่าเพียงแต่ค่าของ word ตรงกับรูปแบบใดรูปแบบหนึ่งก็ได้เช่น

```
$ cat mycal
case $# in
0) set 'date'; m = $2 ; y = $6 ; ; # noargs : use today
1) m = $1 ; set 'date' ; y = $6 ; ; # 1 arg : use this year
2) m = $1 ; y = $2 ; ; # 2 args : month and year
esac

case $m in
jan *| Jan*) m = 1; ;
feb *| Feb*) m = 2; ;
mar *| Mar*) m = 3; ;
apr *| Apr*) m = 4; ;
may *| May*) m = 5; ;
jun *| Jun*) m = 6; ;
jul *| Jul*) m = 7; ;
aug *| Aug*) m = 8; ;
sep *| Sep*) m = 9; ;
oct *| Oct*) m = 10; ;
nov *| Nov*) m = 11; ;
dec *| Dec*) m = 12; ;
[ 1-9 ] |10|11|12) ; ; #numeric month
* ) y = $m ; m = "" ; ; #plain year
esac

/usr/bin/cal $m $y # run the real one

$
```

ในระบบยูนิกซ์มีคำสั่งเชลล์อยู่คำสั่งหนึ่งชื่อว่า cal (ย่อมาจาก calendar) ซึ่งใช้ในการพิมพ์ปฏิทิน

การทำงานของ Cal

```
$ cal
```

```
usage : cal [month] year
```

```
$ cal october 1987
```

```
Bad argument
```

```
$ cal 10 1987
```

```
October 1987
```

```
S M Tu W Th F S
```

```
1 2 3
```

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

```
11 12 13 14 15 16 17
```

```
18 19 20 21 22 23 24
```

```
25 26 27 28 29 30 31
```

```
$
```

ก็คงจะเห็นแล้วว่าการใช้คำสั่ง `cal` นั้นต้องระบุเดือนด้วยตัวเลข และระบุปีไว้ด้วย ในโปรแกรมที่เขียนขึ้นมานี้จะทำให้การทำงานของ `cal` ดีขึ้น คือ ถ้าออกคำสั่ง `mycal` เหยงก็จะแสดงปฏิทินของเดือน และ ปีที่ออกมา และยังสามารถใช้ `mycal` ได้ทั้งกับชื่อเดือน หรือหมายเลขเดือนด้วย โดยถ้าไม่ได้กำหนดปีเอาไว้ `mycal` จะแสดงผลปฏิทินของปีนี้ การทำงานของ `mycal` จะกำหนดค่าที่เหมาะสมให้แก่ตัวแปร `m` และ `y` และทำการเรียกคำสั่ง `cal` โดยส่งค่า `m` เป็นเลขที่เดือน และ `y` เป็นเลขที่ปี โปรแกรมนี้ ส่วนแรกที่แปลกตาขึ้นมาก็คือคำสั่ง `set` ซึ่งเป็นคำสั่งที่ใช้กำหนดค่าให้กับตำแหน่งพารามิเตอร์ โดยใช้ตัวอย่างต่อไปนี้ประกอบ

```
$ date แสดงวันและวันที่
```

```
Tue Nov 5 17:30:48 BKK 1987
```

```
$ set 'date' หลังจากออกคำสั่ง set
```

```
$ echo $1 ตำแหน่งพารามิเตอร์ก็จะมีค่า Tue จากผลของคำสั่งที่ระบุใน set
```

```
$ echo $4
```

```
17:30:48
```

```
$
```

ดังนั้น หลังจากที้ออกคำสั่ง `set 'date'` และกำหนดค่า `m` และ `y` จะได้ว่า `m` คือ ชื่อย่อของเดือน และ `y` คือปีของวันนี้ ส่วนที่เหลือของโปรแกรม คือ การกำหนดค่าให้กับ `m` เป็นเลขที่ของเดือนตามชื่อย่อที่มีค่าอยู่ ขอให้สังเกตรูปแบบสุดท้ายของประโยค `case` อันได้แก่ `" * "` ซึ่งใส่ไว้เพื่อเป็นการป้องกันกรณีที่ `m` ไม่ตรงกับรูปแบบก่อนหน้านี้เลย การทำงานของ `mycal` จะเป็นดังนี้

```
$ date
```

```
The Nov 5 17:35:46 BKK 1987
```

```
$ mycal
```

```
November 1987
```

```

S M Tu W Th F S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
$ mycal dec
December 1987
S M Tu W Th F S
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

```

การใช้ **break** และ **continue** ใช้ในการบังคับทิศทางการทำงานของลูปต่างๆ โดยประโยค **break** จะทำให้การทำงานหลุดออกจากลูปที่ครอบประโยคเอาไว้ ในขณะที่ **continue** จะทำให้การทำงานกลับไปยังลูปเพื่อจะวนกลับมาทำงานต่อในรอบใหม่ ตัวอย่างต่อไปนี้ แสดงการทำงานที่ให้ผลเหมือนกับโปรแกรมที่เคยแสดงไปแล้ว คือจะทำการแสดงค่า **argument** โดยสลับลำดับที่จากตัวหลังมา ตัวหน้า โดยการใช้ **while**

```

Loop
count = $#
cmd = echo
while test "$count" -gt 0
do
cmd = "$cmd$${count}"
count = `expr $count - 1`
done
eval $cmd

```

คงจะต้องอธิบายการทำงานของตัวอย่างนี้ ก่อนที่จะแสดงตัวอย่างที่ใช้ **break** และ **continue** ในโปรแกรมนี้มีเพียงสิ่งเดียวเท่านั้นที่ยังไม่รู้จักกัน ได้แก่ ประโยค **eval** ซึ่งมีความหมายว่า **evaluate** นั่นก็คือ **eval** จะทำหน้าที่ตามคำสั่งที่ระบุไว้ สำหรับผู้ที่ยังไม่ค่อยมั่นใจว่าจะเข้าใจการทำงานอย่างถ่องแท้ คงจะเป็นเพราะยังไม่มีคามแม่นยำกับคุณลักษณะของเชลล์ที่นำมาใช้เริ่มด้วยการกำหนดค่าของตัวแปร **count** ให้เท่ากับจำนวนของ **argument** นั่นก็คือจะทำการรวนลูป และลด

ค่า count ที่ละ 1 ตราบจน count มีค่าเป็น 0 ได้กำหนดค่าเริ่มต้นของ cmd เป็น echo และถ้าสมมุติให้จำนวน argument คือค่าแรกเริ่มของ count เท่ากับ 4 การวนลูปในครั้งแรกจะกำหนดให้ cmd มีค่าเท่ากับ " echo \$4 " ความหมายของ \\$\$count คือ \$4 เพราะ \\$ จะคงค่า \$ เอาไว้ และ \$count มีค่าเป็น 4 ดังนั้นเมื่อเขียนรวมกัน \\$\$count จึงมีค่าเป็น \$4 จากนั้นก็ลดค่า count ลง 1 และวนลูปเป็นครั้งที่ 2 คราวนี้เนื่องจากค่าของ count คือ 3 ดังนั้น cmd จึงมีค่ากลายเป็น " echo \$4 \$3 " นั่นเอง จะวนลูปจนกระทั่งค่าของ count กลายเป็น 0 นั่นก็หมายความว่าค่าสุดท้ายของ cmd นี้ก็จะแสดง argument ที่ละตัว โดยแสดงตัวสุดท้ายก่อนนั่นเอง

การทำงานของโปรแกรมโดยใช้ประโยค break

```
count = $#
cmd = echo
while true
do
    cmd = " cmd \$$count "
    count = ' expr $count -1 '
    if test $count -eq 0
    then break
    fi
done
eval $cmd
```

ได้เปลี่ยนแปลงเงื่อนไขในส่วน of while โดยใส่ true ไว้ซึ่งเท่ากับให้วนลูปไปเรื่อยๆ อย่างไม่สิ้นสุด แต่ในส่วน of ประโยค if ในลูป ได้ทดสอบดูว่าค่าของ count ลดลงจนเป็น 0 หรือยัง ซึ่งถ้าใช้ก็จะ break คือบังคับให้กระโดดออกนอกลูปของ while นั่นเอง การทำงานเช่นนี้อาจจะใช้ continue แทนก็ได้ ดังนี้

```
count = $#
cmd = echo
while true
do
    cmd = " $cmd \$$count "
    count = ' expr $count -1 '
    if test $count -gt 0
    then continue
    fi
eval $cmd
```


exit

done

ในทางที่ตรงกันข้ามกับ break ประโยค continue จะบังคับให้วนลูปในรอบใหม่ตรงเท่าที่ค่าของ count ยังมากกว่า 0 อยู่ ดังนั้นเมื่อ count ลดลงจนเป็น 0 ก็จะกระโดดมาประโยคถัดจาก if และทำการแสดงผลลัพธ์แล้วจึงจบการทำงานด้วยประโยค exit

การดีบั๊กโปรแกรมเชลล์ ในกรณีที่โปรแกรมเชลล์ที่เราเขียนขึ้นมา เกิดทำงานได้ไม่ตรงกับความต้องการ การค้นหาข้อผิดพลาดมีเทคนิคอยู่ 2 วิธี คือ วิธีแรก โดยการสอดแทรกประโยค echo ไปตามจุดต่างๆ เพื่อเป็นการทดสอบแต่ละระยะว่าทำงานถึงจุดไหนและค่าตัวแปรเป็นเท่าไร ซึ่งแม้จะเป็นวิธีการที่ใช้ได้ผล แต่ก็สร้างความรำคาญให้แก่ผู้เขียนโปรแกรมเป็นอย่างยิ่ง เพราะมันไม่รวดเร็วทันใจ และยังทำให้โปรแกรมดูรุงรัง

ดังนั้นจึงใช้เทคนิคแบบที่ 2 คือ การใส่ทางเลือก -x ในขณะที่รันโปรแกรมด้วยคำสั่ง sh ซึ่งจะมีผลให้ค้นพบจุดผิดพลาดได้ง่ายขึ้น รูปแบบของการใช้งานคือการออกคำสั่ง

sh -x ชื่อโปรแกรมเชลล์

คำสั่ง read คำสั่งที่มีประโยชน์มากในการเขียนโปรแกรมเชลล์อีกคำสั่งหนึ่ง ได้แก่ คำสั่ง read ซึ่งจะทำการอ่านค่าข้อมูลให้แก่ตัวแปร โดยไม่มีการบังคับว่าข้อมูลที่อ่านเข้ามานั้นจะเป็นข้อมูลชนิดใด ตัวอย่าง เช่น

```
$ read greeting
```

```
hello, world
```

```
$ echo $greeting
```

```
hello, world
```

```
$
```

คำสั่ง read จะอ่านข้อมูลจาก เทอร์มินัล ดังนั้น เมื่อพิมพ์ข้อความ hello world ตัวแปร greeting ก็จะมีค่าตามข้อความนั้น คำสั่งนี้มีคุณสมบัติพิเศษ คือ ไม่สามารถทำการเบี่ยงเบนข้อมูลกับคำสั่ง read ได้ read จะอ่านข้อมูลจากเทอร์มินัลเท่านั้นดังตัวอย่างต่อไปนี้

```
$ read greeting < /etc/passwd พยายามเบี่ยงเบนให้อ่าน goodbye ข้อมูลจากไฟล์/
etc/passwd
```

```
$ echo $greeting แต่ปรากฏว่าก็ต้องพิมพ์ข้อมูลgoodbye ให้อยู่ดี
```

```
$
```

งานวิจัยที่เกี่ยวข้อง

ซิลเวียร์; ฟิลลิปป์;และ เอ็ดมุนโด (Silvia; Philippe;& Edmundo. 2005) หลายครั้งที่ การศึกษาการ monitor traffic มีการเปลี่ยนแปลงสูง (บางครั้งไม่คงที่) และหลายครั้งที่เกิดการ ผิดพลาดของข้อมูล นั้นเป็นเหตุของความเสียหายสำหรับการควบคุมคุณภาพของการบริการ (QoS) ถ้าเราสามารถทำความเข้าใจความผิดพลาดนั้น เกิดเป็นการเปลี่ยนแปลงที่ถูกต้อง เพราะผู้ใช้งานเกิดการ ใช้ งานแบบทันที หรือเพราะผู้ใช้งานมีจำนวนมาก เหตุผลอย่างอื่น ๆ อาจเกิดจาก DoS (Denide of Service) การโจมตีที่ทำให้ไม่สามารถใช้งานได้ บทความนี้แสดงถึงการใช้เทคนิคการวัดและ ตรวจสอบสำหรับปรับปรุงเครือข่าย ตัวอย่างวิธีการควบคุม TCP/IP ในระดับของ Traffic engineering mechanisms และวิธีใช้การตรวจสอบวิธีการรับมือจากการโจมตี (DoS Attacks) การ กำหนดการช่วยเหลือทั้งหมดและการบำรุงรักษาจากต้นทางถึงปลายทางของการควบคุมคุณภาพ (QoS) สำหรับความผิดพลาดของ Traffic จนถึงความต้องการของผู้ใช้งานหรือลักษณะที่ยุ่งร้าย

เกาวกัง;และ คนอื่นๆ. (Gaogang;& et al. 2003) ความถูกต้องและการวัดที่ real time ของ link traffic เป็นพื้นฐานของ Traffic Engineering การตรวจสอบลักษณะของเครือข่ายและ วางแผนเกี่ยวกับปริมาณการใช้งาน วิธีการตรวจสอบ link traffic ด้วยวิธีการให้คะแนนทุกอุปกรณ์ สำหรับเครือข่ายขนาดใหญ่จะมีผลกระทบต่อประสิทธิภาพของ routers และ ทำให้ก่อเกิดการใช้งาน Traffic ปริมาณมากในระบบเครือข่าย นอกจากนี้วิธีการนี้ไม่สามารถพบความต้องการของสเกล เวล ตั้งแต่อุปกรณ์ที่สนับสนุน SNMP จำนวนมากในระบบเครือข่าย ที่อยู่ภายใต้การจัดการจะ ตัดสินใจอย่างไรที่จะตัดสินใจอย่างไรที่จะตั้งค่าอุปกรณ์ให้น้อยที่สุดในการสอบถามด้วยหลักฐานของ ทุกๆ link traffic ที่จะยืนยันวิธีที่จะแก้ไขปัญหา ในบทความนี้วิธีการตรวจสอบวัด traffic นี้เป็นการ แนะนำและเป็นเงื่อนไขที่เพียงพอ สามารถรวมถึงเงื่อนไขจุดอ่อนที่สุด ครอบคลุมการเสนอของ Yuri Breitbart นโยบายของวิธีการตรวจวัด traffic เป็นการออกแบบบนพื้นฐานของเงื่อนไขที่เพียงพอ ตัวอย่างนี้ใช้วิธีการอธิบายกระบวนการของนโยบาย วิธีการวัดสามารถลดจำนวน nodes ในการ ตรวจสอบและลดปริมาณ traffic ของการตรวจสอบ นโยบายนี้มีการใช้จริงกับระบบการจัดการ เครือข่ายสำหรับโครงการการศึกษาทางไกลสนับสนุนโดย NEPC

บราวน์ลี (N. Brownlee. 2001) วิธีการตรวจวัดการใช้งานของระบบเครือข่ายสามารถให้ ประโยชน์สำหรับการลดปริมาณข้อมูลเกือบจะเป็นปัจจุบัน RTFM เป็นมาตรฐานของ IETF ได้ ออกแบบโครงสร้างสำหรับการวัดปริมาณข้อมูลที่ไหลอยู่บนเครือข่าย และ NeTraMet เป็น แหล่งข้อมูลเปิด(Open-source) เป็นเครื่องมือของการทำงานเบื้องต้น ผลสำรวจจากการใช้งาน NeTraMet โดยส่วนประกอบสามารถใช้ในการสร้างระบบการจัดการการทำงานของข้อมูลบน เครือข่ายสำหรับหลายๆ อุปกรณ์ที่มาต่อร่วมกันบนเครือข่าย Nifty เป็น X Windows ที่วิเคราะห์และ อธิบายการใช้งานเครือข่ายที่ใกล้เคียงเวลาปัจจุบัน โดยประสบการณ์การใช้ NeTraMet โดยหัวข้อ สำคัญที่เป็นประโยชน์ในการปรับกับระบบการจัดการข้อมูลเครือข่ายในอาคาร

งานวิจัยที่เกี่ยวข้องที่ได้ศึกษามานั้น จะเป็นการตรวจสอบโดยการทำงานของระบบ เพื่อที่จะปรับปรุงช่วยเหลือและควบคุมคุณภาพ (QoS) และออกแบบวิธีการตรวจวัดการใช้งาน เครือข่ายให้มีประสิทธิภาพยิ่งขึ้นด้วยการใช้หลักการ Traffic Engineering โดยการวิจัยของผู้จัดทำนี้ จะเป็นการพัฒนาและเพิ่มประสิทธิภาพของระบบการจัดการประสิทธิภาพส่วนของสถานีฐาน โดยที่ ข้อมูลระบบการจัดการประสิทธิภาพส่วนของสถานีฐานนั้นนำมาใช้ประเมินการใช้งานและแนวโน้ม การใช้งาน ของผู้ใช้งานแต่ละพื้นที่ รวมถึงสามารถทำให้ทราบถึงประสิทธิภาพหรือความผิดปกติ ของสถานีฐานแต่ละสถานี เพื่อที่จะนำค่าที่ได้มาปรับปรุงหรือปรับเปลี่ยนและประเมินการขยาย อุปกรณ์ของสถานีฐานนั้นๆ ด้วยการพัฒนาจะเป็นการใช้โปรแกรม Shell script และ SQL ในการ ตรวจสอบและแก้ไขปัญหา เพื่อช่วยเพิ่มประสิทธิภาพของระบบการจัดการประสิทธิภาพส่วนของ สถานีฐานให้ข้อมูลมีความถูกต้อง ครบถ้วน และสามารถทราบปัญหา อีกทั้งเป็นเครื่องมือในการ แก้ไขปัญหาที่เกิดขึ้นได้อย่างรวดเร็ว

บทที่ 3

วิธีการดำเนินงานวิจัย

จากเอกสารและงานวิจัยที่เกี่ยวข้อง ในบทนี้จะกล่าวถึงรายละเอียดในการดำเนินการวิจัย ตั้งแต่การศึกษาขั้นตอนการทำงาน การวางแผน การพัฒนาโปรแกรมพัฒนาประสิทธิภาพของระบบ การจัดการสมรรถนะส่วนสถานี่ฐาน ตลอดจนการทดสอบและประเมินผลในงานวิจัยนี้สามารถแบ่ง ขั้นตอนการดำเนินการวิจัยออกเป็น 6 ขั้นตอนหลัก ดังต่อไปนี้

1. ประชากรและกลุ่มตัวอย่าง
2. เครื่องมือที่ใช้ในงานวิจัย
3. การวิเคราะห์และออกแบบ
4. การทดสอบเครื่องมือในการวิจัย
5. ประเมินผลเครื่องมือในการวิจัย
6. การวิเคราะห์ค่าทางสถิติ

ประชากรและกลุ่มตัวอย่าง

ประชากรและกลุ่มตัวอย่างที่ใช้ในงานวิจัยนี้ จะแบ่งเป็น 2 กลุ่มคือ

1. ประชากรและกลุ่มตัวอย่างที่ใช้ในการตรวจสอบซึ่งก็คือ BSC ทั่วประเทศจำนวน 383

BSC

2. ประชากรและกลุ่มตัวอย่างที่ใช้ในการแก้ไขปัญหาเบื้องต้น ซึ่งจะเป็น BSC ยี่ห้อ Huawei รุ่น B32 จำนวน 31 BSC

เนื่องมาจากการสำรวจเดือน มกราคม ถึง เดือนกันยายน พ.ศ. 2552 พบปัญหาเกิดขึ้น จำนวน 592 ครั้ง และเป็นปัญหาที่เกิดจากระบบการจัดการสมรรถนะส่วนสถานี่ฐาน จำนวน 204 ครั้ง เฉลี่ยในแต่ละเดือนจะเกิดขึ้นจำนวน 66 ครั้ง และพบปัญหาดังกล่าวทุกๆเดือน โดยรูปแบบของการพบปัญหาไม่ได้เป็นรูปแบบที่แน่นอนชัดเจน

ดังนั้นกลุ่มระยะเวลาในการทดสอบงานวิจัยนี้จะใช้การตรวจสอบเครื่องมือวิจัยจะใช้ ช่วงเวลาตั้งแต่เดือนกุมภาพันธ์ – มีนาคม พ.ศ. 2553

เครื่องมือที่ใช้ในการวิจัย

เครื่องมือที่ใช้ในการวิจัยครั้งนี้ แบ่งออกเป็น 2 ส่วน คือ

1. โปรแกรม Shell Script ในการสร้างโปรแกรมหลักที่ใช้ในการตรวจสอบและแก้ไขปัญหาเบื้องต้น และ ภาษา SQL ในการค้นหาข้อมูล(Database) ร่วมกันประมวลผลเพื่อให้ได้ผลลัพธ์ตรงตามวัตถุประสงค์ของการวิจัย

2. แบบสอบถามเกี่ยวกับการประเมินผลความพึงพอใจ และสมรรถนะของระบบที่สร้างขึ้น ซึ่งประเมินโดยผลวิศวกรผู้เชี่ยวชาญที่มีตำแหน่งตั้งแต่วิศวกรอาวุโส (Senior Engineer) ขึ้นไป โดยลักษณะของแบบสอบถามเป็นแบบมาตราส่วนการประมาณค่า (Rating Scale), การเลือกตอบ 1 คำตอบและคำถามปลายเปิดแบบเขียนให้ตอบ แบ่งเป็น 3 ตอนคือ

ตอนที่ 1 เป็นแบบสอบถามเกี่ยวกับสถานภาพทั่วไปของผู้ตอบแบบสอบถาม

ตอนที่ 2 เป็นแบบสอบถามเกี่ยวกับการประเมินความพึงพอใจในการใช้งานระบบ ลักษณะของแบบสอบถามเป็นมาตราส่วนการประมาณค่า 5 ระดับ ดังนี้

คะแนน 5 หมายถึง ความพึงพอใจในระดับมากที่สุด

คะแนน 4 หมายถึง ความพึงพอใจในระดับมาก

คะแนน 3 หมายถึง ความพึงพอใจในระดับปานกลาง

คะแนน 2 หมายถึง ความพึงพอใจในระดับน้อย

คะแนน 1 หมายถึง ความพึงพอใจในระดับน้อยที่สุด

ตอนที่ 3 เป็นแบบสอบถามเกี่ยวกับการประเมินสมรรถนะของระบบที่สร้างขึ้น ลักษณะของแบบสอบถามเป็นการให้เลือกเพียงคำตอบเดียวคือ เห็นด้วย หรือไม่เห็นด้วย

วิเคราะห์และออกแบบ

การวิเคราะห์และออกแบบระบบพัฒนาประสิทธิภาพระบบการจัดการประสิทธิภาพส่วนของสถานีฐานสามารถแบ่งได้เป็นหัวข้อหลักๆ ได้ 2 หัวข้อ ดังนี้

1. เป็นการนำข้อมูลที่เก็บรวบรวมได้ มาดำเนินการออกแบบพัฒนาระบบช่วยตรวจสอบและแก้ไขปัญหา โดยใช้เครื่องมือคือ

1.1 ส่วนโปรแกรม จะใช้ภาษา Shell Script, Standard relational database Query Language (SQL)

1.2 ส่วนฐานข้อมูล (Database) จะใช้ Oracle Database

2 หลักการทำงานของโปรแกรมพัฒนาประสิทธิภาพระบบการจัดการประสิทธิภาพส่วนของสถานีฐาน จะมีขั้นตอนดังต่อไปนี้

2.1 จะมีการตั้งเวลาในการเรียกโปรแกรมพัฒนาประสิทธิภาพระบบการจัดการประสิทธิภาพส่วนของสถานีฐาน เพื่อเข้าทำการตรวจสอบ ข้อมูลในระบบส่วน Pre-calculate (Pcal) Database

2.2 เมื่อทำการตรวจสอบแล้วจะแยกเป็น 2 กรณี คือ

กรณีที่1 ข้อมูลในระบบส่วน Pre-calculate (Pcal) Database ของแต่ละภาคครบถ้วน สมบูรณ์มากกว่า 95 % โปรแกรมจะทำการบันทึกเก็บเป็นไฟล์ (.log) ไว้ และทำการจบการทำงาน ของโปรแกรม

กรณีที่2 ข้อมูลในระบบส่วน Pre-calculate (Pcal) Database ของแต่ละภาคครบถ้วน สมบูรณ์น้อยกว่า 95 % โปรแกรมจะทำการตรวจสอบว่าเป็นของภาคอะไร และเกี่ยวข้องกับ Raw Database ตารางไหนบ้าง จากนั้นทำการเรียกโปรแกรมที่ใช้ในการตรวจสอบ Raw Database ต่อไป

2.3 ทำการตรวจสอบ Raw Database ตารางที่เกี่ยวข้อง เพื่อทำการตรวจสอบว่า ข้อมูลของภาคที่มีข้อมูลใน Pre-calculate (Pcal) Database ครบถ้วนสมบูรณ์น้อยกว่า 95 % นั้น ข้อมูลส่วนของ Raw Database นั้นพบข้อมูลหรือไม่ โดยจะแยกเป็น 2 กรณี

กรณีที่1 พบข้อมูลดังกล่าว ดังนั้นแสดงว่ากระบวนการคำนวณข้อมูลจาก Raw Database ไปยัง Pre-calculate (Pcal) Database นั้นมีความผิดปกติ โปรแกรมจะทำการบันทึก ข้อมูลและส่งแจ้งเป็น SMS แจ้งไปยังวิศวกรที่ดูแลระบบเพื่อเข้าไปทำการตรวจสอบต่อไป

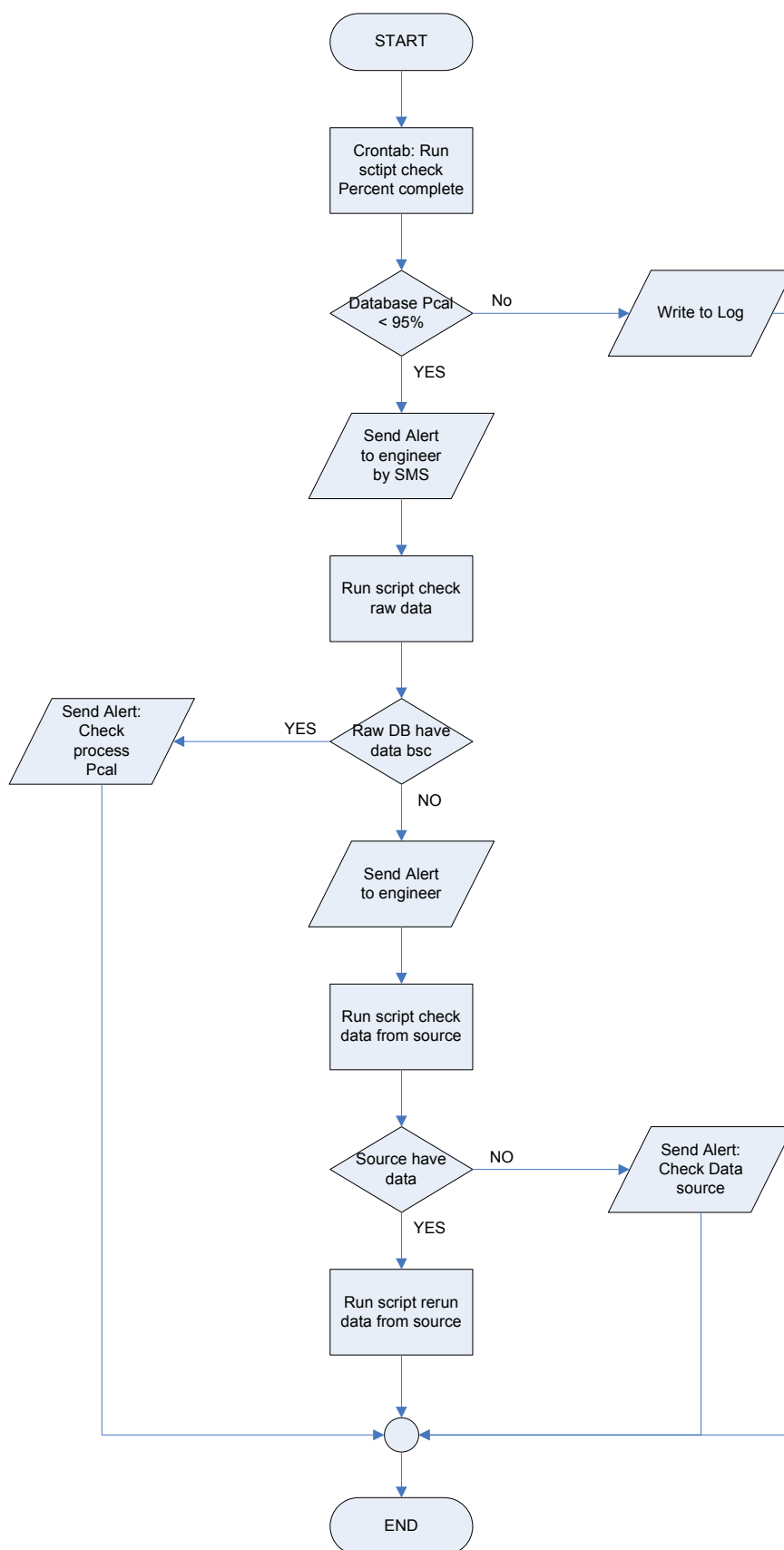
กรณีที่2 ไม่พบข้อมูลในตารางของ Raw Database โปรแกรมจะทำการเรียกโปรแกรม ที่เข้าไปตรวจสอบข้อมูลที่ชุมสายต่อไป

2.4 โปรแกรมที่เข้าไปตรวจสอบข้อมูลที่ชุมสาย จะทำการตรวจสอบข้อมูลจาก Server ที่ชุมสายว่ามีข้อมูลที่ผิดปกติหรือไม่ โดยจะแยกเป็น 2 กรณี

กรณีที่1 พบข้อมูลดังกล่าว ดังนั้นแสดงว่ากระบวนการรับ - ส่งข้อมูลจากชุมสาย ไปยังระบบคำนวณลง Raw Database นั้นมีความผิดปกติ โปรแกรมจะทำการบันทึกข้อมูลและส่งแจ้ง เป็น SMS แจ้งไปยังวิศวกรที่ดูแลระบบและทำการเรียกโปรแกรมในการเข้าไปดึงข้อมูลมาให้อีกครั้ง

กรณีที่2 ไม่พบข้อมูลหรือไม่สามารถติดต่อชุมสายได้ โปรแกรมจะทำการบันทึกข้อมูล และส่งแจ้งเป็น SMS แจ้งไปยังวิศวกรที่ดูแลระบบเพื่อเข้าไปทำการตรวจสอบต่อไป

โดยสามารถเขียนเป็นแผนผังการทำงานของระบบดังนี้



ภาพประกอบ 5 แผนผังการทำงานของโปรแกรม

การทดสอบระบบช่วยตรวจสอบและแก้ไขปัญหา

โดยระบบที่ได้รับการออกแบบและพัฒนาจะต้องเป็นไปตามวัตถุประสงค์ คือ การพัฒนาประสิทธิภาพระบบการจัดการประสิทธิภาพส่วนของสถานีฐานที่พัฒนาขึ้น ตรวจสอบและแก้ไขปัญหาค่าตัววัด (Measurement) เบื้องต้นของเครือข่ายโทรศัพท์เคลื่อนที่ได้อย่างถูกต้องและสามารถแก้ไขปัญหาเบื้องต้นได้ ดังนี้

1. ระบบที่ออกแบบและพัฒนาสามารถคำนวณ เปอร์เซนต์ความครบถ้วนของข้อมูลในส่วนของ Per-calculate ได้ตามที่กำหนดไว้
2. ระบบที่ออกแบบและพัฒนา สามารถระบุถึงปัญหาจากความผิดปกติ และสามารถส่งแจ้งเตือนไปยังผู้ดูแลระบบได้
3. ระบบที่ออกแบบและพัฒนา สามารถทำการแก้ไขปัญหาเบื้องต้นได้ คือสามารถทำการเรียกโปรแกรมให้ดำเนินการดึงข้อมูลที่ขาดหายไปในระบบการจัดการประสิทธิภาพส่วนของสถานีฐานได้

การประเมินผลเครื่องมือในการวิจัย

การประเมินผลเครื่องมือในการวิจัยนี้แบ่งออกเป็น 2 ส่วนคือ

1. การประเมินผลเพื่อทดสอบความเชื่อมั่นของเครื่องมือในการวิจัย
2. การประเมินผลความสามารถของระบบและทดสอบสมมติฐานการวิจัย

ซึ่งการประเมินผลเครื่องมือในแต่ละส่วนมีรายละเอียด ดังนี้

1. การประเมินผลเพื่อทดสอบความเชื่อมั่นของเครื่องมือในการวิจัย

การประเมินผลเพื่อทดสอบความเชื่อมั่นของเครื่องมือในการวิจัย แบ่งออกเป็น 2 ส่วนคือ

- การทดสอบความเชื่อมั่นของโปรแกรมที่สร้างขึ้น ใช้ผลสรุปจากการประเมินผลของผู้เชี่ยวชาญโดยใช้แบบสอบถาม โดยหาค่าเฉลี่ย (Mean) และหาค่าความเบี่ยงเบนมาตรฐาน (Standard Deviation) ซึ่งการแปลความหมายค่าเฉลี่ยน้ำหนักแบ่งเป็น 5 ระดับ ดังนี้

- | | |
|-----------------------------|---|
| ค่าเฉลี่ย 4.51-5.00 หมายถึง | ผู้เชี่ยวชาญมีความพึงพอใจในการใช้งานระบบสนับสนุนการตัดสินใจระดับมากที่สุด |
| ค่าเฉลี่ย 3.51-4.50 หมายถึง | ผู้เชี่ยวชาญมีความพึงพอใจในการใช้งานระบบสนับสนุนการตัดสินใจระดับมาก |
| ค่าเฉลี่ย 2.51-3.50 หมายถึง | ผู้เชี่ยวชาญมีความพึงพอใจในการใช้งานระบบสนับสนุนการตัดสินใจระดับปานกลาง |
| ค่าเฉลี่ย 1.51-2.50 หมายถึง | ผู้เชี่ยวชาญมีความพึงพอใจในการใช้งานระบบสนับสนุนการตัดสินใจระดับน้อย |
| ค่าเฉลี่ย 1.00-1.50 หมายถึง | ผู้เชี่ยวชาญมีความพึงพอใจในการใช้งานระบบสนับสนุนการตัดสินใจระดับน้อย |

การตัดสินใจระดับน้อยที่สุด

- การทดสอบความเชื่อมั่นของแบบสอบถามที่ผู้วิจัยสร้างขึ้น ใช้วิธีสัมประสิทธิ์แอลฟาของครอนบาค ซึ่งแบบสอบถามจะมีความเชื่อมั่นได้ เมื่อค่าสัมประสิทธิ์แอลฟา (α -coefficient) ที่คำนวณได้มีค่ามากกว่าหรือเท่ากับ 0.80 (ยุทธ ไกยวรรณ. 2545: 174)

2. การประเมินผลเกี่ยวกับความสามารถของระบบและทดสอบสมมติฐานการวิจัย

การประเมินผลความสามารถของระบบที่สร้างขึ้นนั้นมี 2 ตัวชี้วัด คือ

- ความถูกต้องจากการประมวลผลของโปรแกรมที่ได้เมื่อเปรียบเทียบกับผลการพิจารณาด้วยวิธีปกติจากค่าทางสถิติที่เกี่ยวข้องของผู้ออกแบบระบบเองภายใต้เงื่อนไขเดียวกัน ซึ่งความแตกต่างของทั้งสองวิธีนี้จะต้องไม่เกินค่ามาตรฐานที่ผู้ออกแบบระบบใช้ในการออกแบบนั่นคือ ใช้การหาค่าเฉลี่ย (Mean) ของความแตกต่างของเปอร์เซ็นต์ส่วนแบ่งระหว่าง 2 วิธีซึ่งต้องมีค่าแตกต่างกันไม่เกิน 2 และหาค่าความเบี่ยงเบนมาตรฐาน (Standard Deviation)

- ระยะเวลาจากการประมวลผลของโปรแกรมที่ได้เมื่อเปรียบเทียบกับผลการพิจารณาด้วยวิธีปกติจากค่าทางสถิติที่เกี่ยวข้องของผู้ดูแลระบบเองภายใต้เงื่อนไขเดียวกัน โดยการใช้การหาค่าเฉลี่ย (Mean) ของระยะเวลาในการประมวลผลของแต่ละวิธีแล้วนำมาเปรียบเทียบกับ และหาค่าความเบี่ยงเบนมาตรฐาน (Standard Deviation)

การทดสอบสมมติฐานการวิจัย จากสมมติฐานการวิจัยที่ตั้งขึ้น คือ

1. ผลลัพธ์จากการประมวลผลของระบบที่สร้างขึ้น มีความถูกต้องสอดคล้องกับการพิจารณาด้วยวิธีปกติของผู้ดูแลระบบเอง นั่นคือค่าเฉลี่ยรวมของค่าความแตกต่างของเปอร์เซ็นต์ส่วนแบ่งระหว่างการประมวลผลของโปรแกรม กับการพิจารณาด้วยวิธีปกติของผู้ดูแลระบบ

2. ระบบที่สร้างขึ้นใช้ระยะเวลาในการประมวลผล น้อยกว่าระยะเวลาในการพิจารณาด้วยวิธีปกติของผู้ดูแลระบบเอง

การวิเคราะห์ค่าทางสถิติ

1. สถิติพื้นฐาน

สถิติพื้นฐานที่ใช้ในการวิจัยนี้ประกอบด้วย ค่าเฉลี่ยเลขคณิต และค่าส่วนเบี่ยงเบนมาตรฐาน

1.1 ค่าเฉลี่ยเลขคณิต (ชูศรี วงศ์รัตนะ. 2546: 35)

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$$

โดย

$\bar{X}$ = ค่าเฉลี่ยคะแนนของหัวข้อที่ประเมิน

$\sum_{i=1}^n Xi$ = ผลรวมค่าคะแนนของหัวข้อที่ประเมินได้จากกลุ่มตัวอย่างในแต่ละกลุ่ม

n = จำนวนทั้งหมดของกลุ่มตัวอย่าง

1.2 ค่าส่วนเบี่ยงเบนมาตรฐาน (ชูศรี วงศ์รัตนะ. 2546: 64)

$$SD = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}}$$

โดย

SD = ค่าเฉลี่ยคะแนนของหัวข้อที่ประเมิน

X_i = ผลรวมค่าคะแนนของหัวข้อที่ประเมินได้จากกลุ่มตัวอย่างในแต่ละกลุ่ม

$\bar{X}$ = จำนวนค่าเฉลี่ยทั้งหมดของกลุ่มตัวอย่าง

n = จำนวนทั้งหมดของกลุ่มตัวอย่าง

2. สถิติสำหรับการวิเคราะห์คุณภาพ

สถิติสำหรับการวิเคราะห์คุณภาพเพื่อหาค่าความเชื่อมั่นของแบบสอบถามซึ่งเป็นแบบมาตราส่วนการประมาณค่า (Rating Scale) ใช้วิธีการหาค่าสัมประสิทธิ์แอลฟา (α -coefficient) ของครอนบาค (เกียรตินาค สุทธิสุข. 2549: 160)

$$\alpha = \left[\frac{n}{n-1} \right] \left[1 - \frac{\sum S_i^2}{S_t^2} \right]$$

โดย

α = ค่าสัมประสิทธิ์ความเชื่อมั่น

n = จำนวนข้อของแบบทดสอบ

$\sum S_i^2$ = ผลรวมของความแปรปรวนแต่ละข้อ

S_t^2 = ความแปรปรวนของคะแนนรวม

3. สถิติที่ใช้ทดสอบสมมติฐาน

สถิติที่ใช้ในการทดสอบทดสอบสมมติฐานของการวิจัยนี้ใช้สถิติค่า Z-test (ชูศรี วงศ์รัตนะ. 2546: 145) ซึ่งเป็นการทดสอบในกรณีที่มีกลุ่มตัวอย่างเดียว (One Sample test) เป็นการทดสอบความแตกต่างระหว่างค่าเฉลี่ยของกลุ่มตัวอย่างกับค่าคงที่ค่าใดค่าหนึ่ง และจำนวนกลุ่มตัวอย่าง > 30 ซึ่งขั้นตอนในการทดสอบสมมติฐาน มีดังนี้

3.1 ตั้งสมมติฐาน H_0 , H_1

- สมมติฐานด้านความถูกต้อง

$$H_0 : \mu < 2$$

$$H_1 : \mu > 2$$

โดย μ คือค่าเฉลี่ย (Mean) ของความแตกต่างของเปอร์เซ็นต์ส่วนแบ่งระหว่าง 2

วิธี

- สมมติฐานด้านระยะเวลา

$$H_0 : \mu < \text{เวลาที่ใช้ในการประมวลผลด้วยวิธีพิจารณาปกติของผู้ดูแลระบบ}$$

$$H_1 : \mu > \text{เวลาที่ใช้ในการประมวลผลด้วยวิธีพิจารณาปกติของผู้ดูแลระบบ}$$

โดย μ คือค่าเฉลี่ย (Mean) ของระยะเวลาที่ใช้ในการประมวลผลจากโปรแกรม

3.2 คำนวณค่า Z จากสูตร

$$Z = \frac{\bar{X} - \mu_0}{\sigma / \sqrt{n}}$$

โดย

$\bar{X}$ = ค่าเฉลี่ยของกลุ่มตัวอย่าง

μ_0 = ค่าคงที่

σ = ส่วนเบี่ยงเบนมาตรฐานของประชากร (กรณีไม่ทราบค่า σ ให้ใช้ค่าส่วนเบี่ยงเบนมาตรฐานของกลุ่มประชากรแทน)

n = จำนวนกลุ่มตัวอย่าง

3.3 กำหนดระดับนัยสำคัญทางสถิติ (α)

การวิจัยนี้กำหนดระดับนัยสำคัญ $\alpha = 0.05$

3.3 สร้างเขตปฏิเสธ

จากค่านัยสำคัญที่กำหนด และการทดสอบสมมติฐานนี้เป็นการทดสอบทางเดียวทางซ้าย จึงสามารถสร้างเขตปฏิเสธ ได้ดังนี้

เขตปฏิเสธ : จะปฏิเสธสมมติฐาน H_0 ถ้า $Z < -Z_{\alpha=0.05}$ หรือ $Z < -1.645$

3.4 สรุปผลการทดสอบ

ในการทดสอบสมมติฐาน สามารถเกิดความคลาดเคลื่อนในการตัดสินใจได้ 2 แบบ คือ

- ความคลาดเคลื่อนแบบที่ 1 (Type I Error): เป็นการปฏิเสธสมมติฐานศูนย์ที่ถูกความน่าจะเป็นในการปฏิเสธสมมติฐานศูนย์ทั้ง ๆ ที่สมมติฐานศูนย์นั้นเป็นจริง เขียนแทนด้วยสัญลักษณ์ α หรือ P หรือเรียกว่าระดับนัยสำคัญ (Level of significant) ดังนั้นความน่าจะเป็นในการยอมรับสมมติฐานศูนย์ที่ถูกต้องจึงมีค่าเท่ากับ $1 - \alpha$ หรือเรียกว่าระดับความเชื่อมั่น (Level of confidence)

- ความคลาดเคลื่อนแบบที่ 2 (Type II Error): เป็นการยอมรับสมมติฐานศูนย์ที่ผิดความน่าจะเป็นในการยอมรับสมมติฐานศูนย์ทั้ง ๆ ที่สมมติฐานศูนย์นั้นเป็นผิดหรือไม่เป็นจริง เขียนแทนด้วยสัญลักษณ์ β ดังนั้นความน่าจะเป็นในการปฏิเสธสมมติฐานศูนย์ที่ผิดต้องจึงมีค่าเท่ากับ $1 - \beta$ หรือเรียกว่าอำนาจของการทดสอบ (Power of testing)

ความคลาดเคลื่อนทั้ง 2 แบบ แสดงดังตาราง 9

ตาราง 9 ความคลาดเคลื่อนในการตัดสินใจ

การตัดสินใจ	สภาพความเป็นจริงของ H_0	
	H_0 เป็นจริง	H_0 ไม่เป็นจริง
ยอมรับ H_0	การตัดสินใจถูกต้อง ($1 - \alpha$) (ระดับความเชื่อมั่น)	ความคลาดเคลื่อนแบบที่ 2 (Type II Error : β)
ปฏิเสธ H_0	ความคลาดเคลื่อนแบบที่ 1 (Type I Error : α) (ระดับนัยสำคัญ)	การตัดสินใจถูกต้อง ($1 - \beta$) (อำนาจของการทดสอบ)

บทที่ 4

ผลการดำเนินงานวิจัย

ผลการวิจัยในวิทยานิพนธ์ฉบับนี้มีวัตถุประสงค์เพื่อ แสดงผลลัพธ์ของการพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานี่ฐาน ซึ่งจากการดำเนินการวิจัยสามารถแสดงผลได้ 3 ส่วนดังนี้

1. ลักษณะการทำงานและผลลัพธ์จากการพัฒนาประสิทธิภาพของของระบบการจัดการสมรรถนะส่วนสถานี่ฐาน
2. ผลของการประเมินผลเพื่อทดสอบความเชื่อมั่นของเครื่องมือในการวิจัย
3. ผลของการประเมินผลเกี่ยวกับความสามารถของระบบและทดสอบสมมติฐานการวิจัย

ลักษณะการทำงานและผลลัพธ์จากโปรแกรมพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานี่ฐาน

จากการการพัฒนาระบบโดยใช้โปรแกรม Shall script และ SQL สามารถแสดงผลทั้งในส่วนติดต่อผู้ใช้และรูปแบบการแสดงผลของโปรแกรมได้ดังนี้

1. ลักษณะการทำงานของโปรแกรมโดยเริ่มจากการตั้งเวลา เพื่อให้โปรแกรมทำงานตามลำดับ (Crontab) แสดงดังภาพประกอบที่ 6

```
xtserver:/home/narangss/Monitor/per-complete/script$ crontab -l
#### Script Check and rerun data less than 95 % ####
40 8 * * * /home/narangss/Monitor/per-complete/script/1_comp_conclude_less95.cron > /tmp/1_comp_conclude_less95.log 2>&1
43 8 * * * /home/narangss/Monitor/per-complete/script/2_BSC_less95.cron > /tmp/2_BSC_less95.log 2>&1
45 8 * * * /home/narangss/Monitor/per-complete/script/3_Match_find_BSC.sh > /tmp/3_Match_find_BSC.log 2>&1
48 8 * * * /home/narangss/Monitor/per-complete/script/4_Query_Raw_incom.cron > /tmp/4_Query_Raw_incom.log 2>&1
50 8 * * * /home/narangss/Monitor/per-complete/script/5_Decition_SendAlarm.sh > /tmp/5_Decition_SendAlarm.log 2>&1
55 8 * * * /home/narangss/Monitor/per-complete/script/6_Rerun_data.sh > /tmp/6_Rerun_data.log 2>&1
xtserver:/home/narangss/Monitor/per-complete/script$
```

ภาพประกอบ 6 แสดงการตั้งเวลาทำงานของโปรแกรม

2. ผลจากการทำงานของโปรแกรมที่ช่วยหาค่าข้อมูลที่มีค่าน้อยกว่า 95 % โดยจะเห็นได้ว่า หากความครบถ้วนของข้อมูลน้อยกว่า 95% ระบบจะทำการแสดงผลเก็บเป็น Log file โดยจะระบุยี่ห้อของซุ่มสาย และ ภาค (Vendor and Region)

```
xtserver:/home/narangss/Monitor/per-complete/output/1_comp_concludeless95$ more 20100316-20100316_08:40.txt
```

TABLE_NAME	DATETIME	SUPPLIERCODE	REGIONCODE	PERC_COMPLETE	VENDORNA	REGIONN
D_OVERBLOCK_MNC	2010-03-16 00:00:00	2	2	92.85	Nokia	Central

TABLE_NAME	DATETIME	SUPPLIERCODE	REGIONCODE	PERC_COMPLETE	VENDORNA	REGIONN
D_DROPS_MNC	2010-03-16 00:00:00	2	2	92.85	Nokia	Central

no rows selected

no rows selected

TABLE_NAME	DATETIME	SUPPLIERCODE	REGIONCODE	PERC_COMPLETE	VENDORNA	REGIONN
D_CALLLENGTH_MNC	2010-03-16 00:00:00	2	2	92.85	Nokia	Central

TABLE_NAME	DATETIME	SUPPLIERCODE	REGIONCODE	PERC_COMPLETE	VENDORNA	REGIONN
D_OHANDOVERS_MNC	2010-03-16 00:00:00	2	2	92.85	Nokia	Central

TABLE_NAME	DATETIME	SUPPLIERCODE	REGIONCODE	PERC_COMPLETE	VENDORNA	REGIONN
D_SDTTRAFFIC_MNC	2010-03-16 00:00:00	4	1	61.2	Huawei	North
D_SDTTRAFFIC_MNC	2010-03-16 00:00:00	4	3	30.92	Huawei	South

ภาพประกอบ 7 แสดงผลจากการทำงานของโปรแกรมที่ช่วยหาค่าข้อมูลที่มีค่าน้อยกว่า 95 %

3. ผลจากการทำงานหาค่า BSC ที่มีค่าน้อยกว่า 95 % โดยจะเห็นได้ว่า หาความครบถ้วนของข้อมูลแต่ละ BSC มีข้อมูลน้อยกว่า 95% ระบบจะทำการแสดงผลเก็บเป็น Log file โดยจะระบุยี่ห้อของชุมสาย ภาค และชื่อ BSC (Vendor, Region and BSC name)

```
xtserver:/home/narangss/Monitor/per-complete/output/2_BSC_less95$ more 20100317-20100317_08:43.txt
```

TABLE_NAME	DATETIME	SUPPLIERCODE	REGIONCODE	BSCNAME	PERC_COMPLETE	VENDORNA	REGIONN
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	1	BBOHN	93.4	Nokia	North
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	1	BCMCP6	94.82	Nokia	North
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	1	BCMUG1	87.38	Nokia	North
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	1	BDRMU1	91.71	Nokia	North
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	1	BMCLP5	94.43	Nokia	North
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	1	BMERM	94.93	Nokia	North
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	1	BNNMRT	90.54	Nokia	North
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	1	BNNAP1	94.1	Nokia	North
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	1	BPHE3	94.6	Nokia	North
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	1	BTAP4	94.23	Nokia	North
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	1	BTAP5	94.93	Nokia	North

TABLE_NAME	DATETIME	SUPPLIERCODE	REGIONCODE	BSCNAME	PERC_COMPLETE	VENDORNA	REGIONN
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	1	BYSMG	90.64	Nokia	North
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	2	BSC1KRI	0	Nokia	Central
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	2	BSC1NPT	0	Nokia	Central
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	2	BSC4PKN	41.38	Nokia	Central
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	2	BSC4RBR	53.9	Nokia	Central
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	2	BSC5SKN	87.11	Nokia	Central
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	2	BSC6NPT	88.91	Nokia	Central
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	2	BSC6RBR	87.33	Nokia	Central
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	2	BSC7NPT	94.63	Nokia	Central
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	2	BSC8KRI	94.69	Nokia	Central
D_OVERBLOCK_MNC	2010-03-17 00:00:00	2	2	BSC8NPT	39.09	Nokia	Central

ภาพประกอบ 8 แสดงผลจากการทำงานหาค่า BSC ที่มีค่าน้อยกว่า 95 %

4. โปรแกรมจะทำการตรวจสอบข้อมูล BSC ที่ต้นทาง ว่าข้อมูลดังกล่าวมีหรือไม่ โดยจะเข้าไประบุตรวจสอบจากฐานข้อมูล (Database) ที่เป็นส่วนของการประมวลผลเก็บ ยี่ห้อ ภาค ชุมสาย ค่าตัววัด (Measurement) รวมถึงช่วงเวลา ในส่วนข้อมูลที่หายไป โดยแสดงดังภาพ

TABLE_NAME	BSCNAME	REGION	DATE_TIME
KPI	UTTH	1	17-MAR-10
KPI	UTTH	1	17-MAR-10
KPI	UTTH	1	17-MAR-10
KPI	UTTH	1	17-MAR-10
KPI	UTTH	1	17-MAR-10
KPI	UTTH	1	17-MAR-10
KPI	UTTH	1	17-MAR-10
KPI	UTTH	1	17-MAR-10
KPI	UTTH	1	17-MAR-10
SDCCH	UTTH	1	17-MAR-10
SDCCH	UTTH	1	17-MAR-10
SDCCH	UTTH	1	17-MAR-10

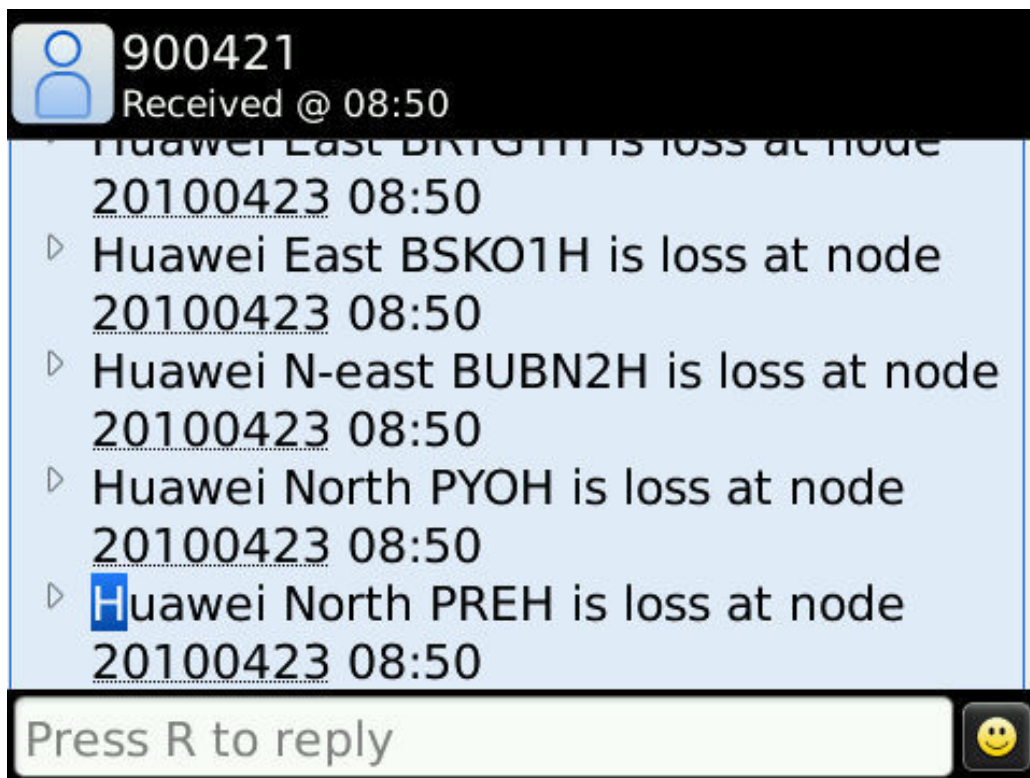
ภาพประกอบ 9 แสดงผลการตรวจสอบข้อมูล BSC ที่ต้นทาง

โดยจากการใช้โปรแกรม Query ข้อมูลในฐานข้อมูลดังกล่าว จะแสดงยี่ห้อ ภาคว ชุมสาย รวมถึงช่วงเวลา ในส่วนข้อมูลที่หายไป โดยจะสามารถแสดงบน Website ได้ดังนี้

NO	DATE_TIME	TABLE_NAME	VENDOR_NAME	REGION_NAME	BSCNAME	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	2010-03-28	KPI	HUAWEI	North	BTPHN1H																								
2	2010-03-28	KPI	HUAWEI	North	NANH																								
3	2010-03-28	KPI	HUAWEI	Central	SAT																								

ภาพประกอบ 10 แสดง Website การแสดงผล Query ข้อมูลในฐานข้อมูล

5. โปรแกรมจะทำการส่งค่าความผิดปกติโดย ระบุสาเหตุให้กับวิศวกร ผู้รับผิดชอบ ปัจจุบันผู้พัฒนาได้ทำการส่งผลลัพธ์ในรูปแบบข้อความสั้น (SMS) โดยจะระบุยี่ห้อ ภาคว ชุมสาย (BSC) รวมทั้งสาเหตุความไม่ครบถ้วนของข้อมูลดังกล่าว



ภาพประกอบ 11 แสดงผลการส่งผลลัพธ์ในรูปแบบข้อความสั้น (SMS)

6. โปรแกรมจะทำการประมวลผลดึงข้อมูล BSC ที่ขาดหายไป เนื่องจากสาเหตุข้อมูลที่ต้นทางมีปัญหาใหม่ สามารถดูผลใน Log ไฟล์ โดยจะระบุถึง ชุมสาย Measurement ที่เข้าไปทำการดึงข้อมูล รวมถึงช่วงเวลาและระยะเวลาที่เข้าไปทำการดึงข้อมูล

```

Connected to 10.215.121.101.
220 ais-ygpa10lu8by Microsoft FTP Service (Version 5.0).
331 Password required for administrator.
230 User administrator logged in.
Interactive mode off.
200 Type set to I.
Local directory now /home/narangss/Monitor/per-complete/script/rerun/data/SDC
200 PORT command successful.
150 Opening BINARY mode data connection for /west-AYA1H/SDC/AYA1H03192010000000.SDC(58619 bytes).
226 Transfer complete.
58619 bytes received in 0.07 seconds (804.90 Kbytes/s)
221
Connected to 10.215.121.101.
220 ais-ygpa10lu8by Microsoft FTP Service (Version 5.0).
331 Password required for administrator.
230 User administrator logged in.
Interactive mode off.
200 Type set to I.
Local directory now /home/narangss/Monitor/per-complete/script/rerun/data/TCH
200 PORT command successful.
150 Opening BINARY mode data connection for /west-AYA1H/TCH/AYA1H03192010000000.TCH(71400 bytes).
226 Transfer complete.
71400 bytes received in 0.09 seconds (818.08 Kbytes/s)
221

```

ภาพประกอบ 12 แสดงผลการประมวลผลดึงข้อมูล BSC ที่ขาดหายไป

ผลของการประเมินผลเพื่อทดสอบความเชื่อมั่นของเครื่องมือในการวิจัย

1. การทดสอบความเชื่อมั่นของเครื่องมือในการวิจัย

จากผลการทดลองใช้และการผลประเมินด้วยแบบสอบถามจากผู้เชี่ยวชาญทั้ง 3 ท่าน สามารถสรุปผลเกี่ยวกับการประเมินความพึงพอใจในการใช้งานระบบซึ่งลักษณะของแบบสอบถามเป็นมาตราส่วนการประมาณ ดังตาราง 10

ตาราง 10 ผลการประเมินความพึงพอใจในการใช้งานระบบจากผู้เชี่ยวชาญ

รายการประเมิน	$\bar{X}$	SD	ระดับความพึงพอใจ
1. ความสะดวกรวดเร็วในการใช้งาน	4.67	0.58	มากที่สุด
2. สามารถเรียนรู้การใช้งานได้ง่าย	4.67	0.58	มากที่สุด
3. การจัดหน้าจอการแสดงผลของโปรแกรมมีความเหมาะสม	4.67	0.58	มากที่สุด
4. การแสดงผลหน้าจอของโปรแกรมตัวหนังสือสามารถอ่านได้ง่าย ชัดเจน	4.00	0.00	มาก
5. ระบบมีการแบ่งขั้นตอนการประมวลผลที่สามารถเข้าใจได้ง่ายและชัดเจน	4.67	0.57	มากที่สุด
6. ผลลัพธ์ที่ได้จากโปรแกรมสอดคล้องกับหลักการในการประเมิน	5.00	0.00	มากที่สุด
7. ข้อมูลที่ได้จากการประมวลผลของโปรแกรมสามารถนำไปใช้ประโยชน์ต่อไปได้	5.00	0.00	มากที่สุด
8. ความเหมาะสมในการทำงานของโปรแกรมโดยรวม	5.00	0.00	มากที่สุด
ค่าเฉลี่ยรวมจากการประเมินทุกหัวข้อ	4.71	0.33	มากที่สุด

เมื่อนำคะแนนเฉลี่ยของแต่ละหัวข้อมาหาค่าทางสถิติพบว่า ค่าเฉลี่ยทั้งหมดอยู่ที่ 4.71 ซึ่งอยู่ในระดับความพึงพอใจมากที่สุด ค่าเบี่ยงเบนมาตรฐานทั้งหมดอยู่ที่ 0.33 ส่วนการประเมินพึงพอใจในด้านสะดวกรวดเร็วในการใช้งาน, ความพึงพอใจในการจัดหน้าจอการแสดงผลของโปรแกรมมีความเหมาะสม, ผลลัพธ์ที่ได้จากโปรแกรมสอดคล้องกับหลักการในการประเมิน, ความเหมาะสมในการทำงานของโปรแกรมโดยรวม, ความพึงพอใจในด้านสามารถเรียนรู้การใช้งานได้ง่ายและระบบมีการแบ่งขั้นตอนการประมวลผลที่สามารถเข้าใจได้ง่ายชัดเจน และข้อมูลที่ได้จากการ

ประมวลผลของโปรแกรมสามารถนำไปใช้ประโยชน์ต่อไปได้อยู่ในระดับมากที่สุด และการแสดงผลหน้าจอของโปรแกรมตัวหนังสือสามารถอ่านได้ง่าย ชัดเจนอยู่ในระดับมาก

การประเมินผลด้วยแบบสอบถามจากผู้เชี่ยวชาญ 3 ท่านในส่วนของการประเมินสมรรถนะ (Performance) ของระบบสามารถสรุปผลได้ดังตาราง 11

ตาราง 11 ผลการประเมินสมรรถนะ (Performance) ของระบบจากผู้เชี่ยวชาญ

รายการประเมิน	จำนวนผู้เชี่ยวชาญ ที่เห็นด้วย	จำนวนผู้เชี่ยวชาญ ที่ไม่เห็นด้วย
1. โปรแกรมสามารถช่วยลดระยะเวลาในการประเมินได้	3	0
2. ระบบสามารถประมวลได้อย่างรวดเร็ว	3	0
3. ผลลัพธ์ที่ได้จากโปรแกรมมีความถูกต้อง เชื่อถือได้	3	0
4. ผู้ประเมินคิดว่าสามารถใช้โปรแกรมนี้แทนการประเมินด้วยวิธีปกติ (Manual) ได้	3	0
ค่าเฉลี่ยรวมจากการประเมินทุกหัวข้อ	3	0

จากผลการประเมินผู้เชี่ยวชาญมีความคิดเห็นว่ารบบที่สร้างขึ้นสามารถประมวลผลได้อย่างรวดเร็ว ซึ่งส่งผลให้ลดระยะเวลาในการตรวจสอบและแก้ไขเบื้องต้นได้ โดยผลลัพธ์ที่ได้จากโปรแกรมมีความถูกต้องเชื่อถือได้ และสามารถนำไปปรับใช้เป็นเครื่องมือในการตรวจสอบและแก้ไขเบื้องต้นในระบบจริงแทนการตรวจสอบและแก้ไขด้วยวิธีปกติ (Manual) ได้ นอกจากนี้ผู้เชี่ยวชาญได้ให้ข้อเสนอแนะสำหรับการพัฒนาระบบให้มีประสิทธิภาพมากยิ่งขึ้นในอนาคต คือ การใช้ให้เหมาะสมยิ่งขึ้น (Optimize) อื่นๆ เช่น การจัดการตัวแปร (Parameter) ของการใช้งานแบบโปรแกรมเดี่ยว (Single Script) รวมกับการจัดคู่ผลลัพธ์ที่ได้จากโปรแกรม (Map exit code) ในการแยกแยะเงื่อนไขต่างๆ

2. การทดสอบความเชื่อมั่นของแบบสอบถาม

การทดสอบความเชื่อมั่นของแบบสอบถามที่ผู้วิจัยสร้างขึ้น ใช้วิธีสัมประสิทธิ์แอลฟา (α - coefficient) ของครอนบาค โดยการหาค่าความแปรปรวนในแต่ละข้อ และความแปรปรวนของคะแนนรวม ดังตาราง 3 และนำไปคำนวณหาค่าสัมประสิทธิ์แอลฟา ซึ่งจากการคำนวณพบว่าแบบสอบถามในการวิจัยนี้มีค่าสัมประสิทธิ์แอลฟาเท่ากับ 1.14 ซึ่งมากกว่าค่ามาตรฐาน 0.80 ดังนั้นแบบสอบถามในการวิจัยนี้มีความเที่ยงโดยมีความเชื่อมั่นเท่ากับ 1.14

ตาราง 12 การหาค่าความแปรปรวนของแบบสอบถามเพื่อใช้คำนวณสัมประสิทธิ์แอลฟา

ผู้เชี่ยวชาญ คนที่	รายการประเมินข้อที่								คะแนน รวม
	1	2	3	4	5	6	7	8	
1	4	5	4	4	5	5	5	5	37
2	5	4	5	4	5	5	5	5	38
3	5	5	5	4	4	5	5	5	38
ความแปรปรวน (S^2)	0.33	0.33	0.33	0.00	0.33	0.00	0.00	0.00	0.33

ผลของการประเมินผลเกี่ยวกับความสามารถของระบบและทดสอบสมมติฐานการวิจัย

1. ผลของการประเมินผลเกี่ยวกับความสามารถของระบบ

จากการพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานี่ฐานขึ้น และทดสอบการทำงานสามารถตอบสนองตามวัตถุประสงค์คือ

1.1 พัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานี่ฐาน เพื่อช่วยตรวจสอบและแก้ไขปัญหาค่าตัววัด (Measurement) เบื้องต้นของเครือข่ายโทรศัพท์เคลื่อนที่ โดยได้ทำการพัฒนาและได้ทำการทดสอบระบบ ด้วยวิธีการตั้งเวลาให้ระบบที่ทำงานดำเนินการตรวจสอบและแก้ไขปัญหาเบื้องต้นในส่วนของการดึงข้อมูล ตั้งแต่เดือนกุมภาพันธ์ ถึงเดือนมีนาคม พ.ศ. 2553 จำนวน 59 วัน สามารถตรวจสอบระบบการจัดการประสิทธิภาพส่วนของสถานี่ฐาน พบความผิดปกติของข้อมูลดังตาราง 13

ตาราง 13 จำนวนและค่าเฉลี่ยของ BSC ที่ผิดปกติและต้องทำการแก้ไข

Data	All BSC		Only Huawei B32 (rerun)			
	Complete < 95%	ที่ต้องแก้ไข	Process by Engineer	Process by Program	% Complete	% Difference
20100201	22	3	3	3	100.00	0.00
20100202	21	2	1	1	100.00	0.00
20100203	20	3	3	3	100.00	0.00
20100204	19	2	2	2	100.00	0.00
20100205	21	3	3	3	100.00	0.00
20100206	22	3	3	3	100.00	0.00
20100207	21	3	3	3	100.00	0.00
20100208	18	3	2	2	100.00	0.00
20100209	21	4	3	3	100.00	0.00
20100210	16	1	1	1	100.00	0.00
20100211	22	1	1	1	100.00	0.00
20100212	14	21	17	17	100.00	0.00
20100213	42	10	2	2	100.00	0.00
20100214	21	0	3	3	100.00	0.00
20100215	20	2	4	4	100.00	0.00
20100216	20	0	0	0	100.00	0.00
20100217	39	0	0	0	100.00	0.00
20100218	38	3	3	3	100.00	0.00
20100219	42	0	0	0	100.00	0.00
20100220	23	1	1	1	100.00	0.00
20100221	20	1	1	1	100.00	0.00
20100222	22	1	1	1	100.00	0.00
20100223	19	0	0	0	100.00	0.00
20100224	19	0	0	0	100.00	0.00
20100225	19	0	0	0	100.00	0.00
20100226	52	50	31	30	96.77	3.23
20100227	20	4	4	4	100.00	0.00
20100228	27	4	4	4	100.00	0.00
20100301	27	1	1	1	100.00	0.00
20100302	21	18	18	17	94.44	5.56
20100303	21	15	15	15	100.00	0.00
20100304	20	4	4	4	100.00	0.00
20100305	19	4	4	4	100.00	0.00
20100306	29	5	3	3	100.00	0.00
20100307	21	6	5	5	100.00	0.00
20100308	36	18	15	15	100.00	0.00
20100309	20	5	5	5	100.00	0.00
20100310	23	8	5	5	100.00	0.00
20100311	19	1	1	1	100.00	0.00
20100312	20	3	3	3	100.00	0.00
20100313	69	51	3	3	100.00	0.00
20100314	20	4	3	3	100.00	0.00
20100315	73	59	8	8	100.00	0.00
20100316	22	3	3	3	100.00	0.00
20100317	21	2	2	2	100.00	0.00
20100318	20	3	3	3	100.00	0.00

ตาราง 13 (ต่อ)

Data	All BSC		Only Huawei B32 (rerun)			
	Complete < 95%	ที่ต้องแก้ไข	Process by Engineer	Process by Program	% Complete	% Difference
20100319	19	2	2	2	100.00	0.00
20100320	21	3	3	3	100.00	0.00
20100321	22	3	1	1	100.00	0.00
20100322	21	3	2	2	100.00	0.00
20100323	18	3	3	3	100.00	0.00
20100324	21	4	4	4	100.00	0.00
20100325	16	1	1	1	100.00	0.00
20100326	54	31	30	29	96.67	3.33
20100327	24	6	3	3	100.00	0.00
20100328	54	7	6	6	100.00	0.00
20100329	54	31	25	25	100.00	0.00
20100330	39	12	12	12	100.00	0.00
20100331	24	13	13	13	100.00	0.00
Summary	1568	454	302	299	5887.89	12.11
Average	26.58	7.69	5.12	5.07	99.79	0.21
SD	12.98	12.62	6.92	6.76	0.93	0.93

ตาราง 14 สรุปจำนวนและค่าเฉลี่ยของ BSC ที่ผิดปกติและต้องทำการแก้ไข

	จำนวน BSC ทั้งหมด 383 BSC		จำนวน BSC (Huawei B32 ทั้งหมด 31 BSC)
	BSC ที่มีข้อมูลมีปัญหา (% Complete <95%)	BSC ที่ต้องมีการแก้ไข (ทุกยี่ห้อ)	BSC ที่ต้องมีการแก้ไข (Huawei B32)
จำนวน BSC (ก.พ. – มี.ค. 2553)	1568	454	302
ค่าเฉลี่ยต่อหนึ่งวัน	26.58	7.7	5.12
% จาก BSC ทั้งหมด	6.94 %	2.01 %	1.34 %
% จาก BSC ที่มีปัญหา	100 %	28.95 %	19.26 %
SD	12.98	12.62	6.92

โดยระบบสามารถทำการตรวจสอบ BSC ข้อมูลที่มีความครบถ้วนของข้อมูลน้อยกว่า 95 เปอร์เซนต์ จำนวน 1568 BSC และทำการตรวจสอบว่าข้อมูลดังกล่าวเกิดจากความผิดปกติจากไหนจุดต้นทางต้องทำการแก้ไขเบื้องต้นจำนวน 454 BSC ซึ่งเป็น BSC ยี่ห้อ Huawei รุ่น B32 จำนวน 302 BSC ซึ่งคิดเป็น 19.26% ของ BSC ทั้งหมดที่พบปัญหาด้านความครบถ้วน และ คิดเป็น 16.52% ของ BSC ยี่ห้อ Huawei รุ่น B32 ที่พบปัญหาด้านความครบถ้วน โดยมีค่าเบี่ยงเบน (SD) เท่ากับ 6.92

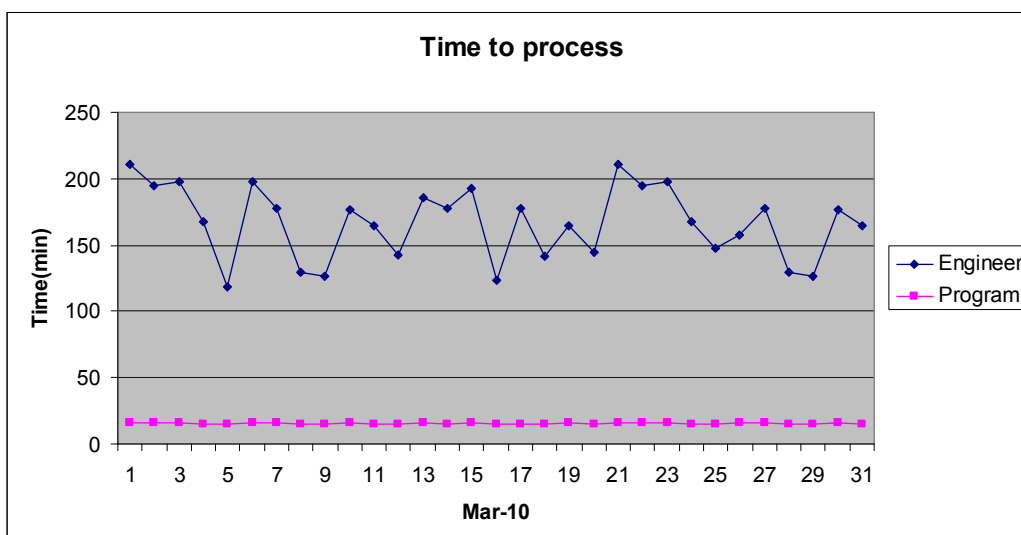
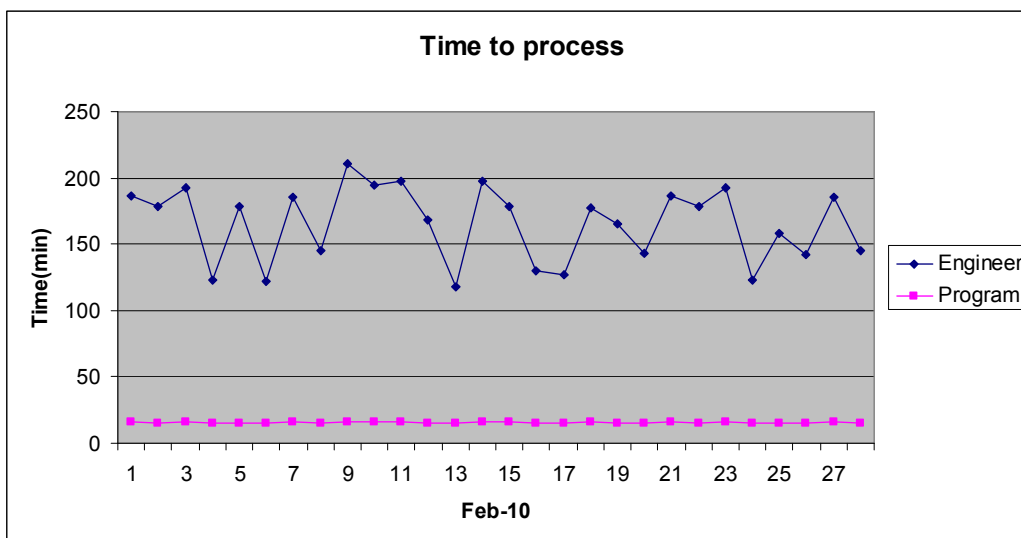
1.2 ได้ระบบพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนฐานที่ฐาน ที่ช่วยเพิ่มประสิทธิภาพในการตรวจสอบความถูกต้องและสามารถแก้ไขปัญหาเบื้องต้นได้รวดเร็ว โดยจากการทดสอบระบบเปรียบเทียบกับผลการพิจารณาด้วยวิธีปกติจากค่าทางสถิติการตรวจสอบข้อมูลโดยใช้วิศวกรเองภายใต้เงื่อนไขเดียวกัน พบว่าระบบที่พัฒนาขึ้นสามารถทำการตรวจสอบข้อมูลได้ถูกต้องเทียบเท่า อีกทั้งระยะเวลาจากการประมวลผลของโปรแกรมที่ได้เมื่อเปรียบเทียบกับผลการพิจารณาด้วยวิธีปกติจากค่าทางสถิติที่เกี่ยวข้องของผู้ดูแลระบบเองภายใต้เงื่อนไขเดียวกัน โดยใช้การหาค่าเฉลี่ย (Mean) ของระยะเวลาในการประมวลผลของแต่ละวิธีแล้วนำมาเปรียบเทียบกัน และหาค่าความเบี่ยงเบนมาตรฐาน (Standard Deviation) ซึ่งจากการทดสอบการทำงาน ก.พ. – มี.ค. พ.ศ. 2553 พบว่าค่าเฉลี่ยของการประมวลผลด้วยโปรแกรมมีค่าเท่ากับ 15.46 นาที ค่าส่วนเบี่ยงเบนมาตรฐานมีค่าเท่ากับ 0.50 ส่วนการประมวลผลด้วยวิธีปกติของผู้ดูแลระบบมีค่าเฉลี่ยเท่ากับ 166.12 นาที และค่าส่วนเบี่ยงเบนมาตรฐานมีค่าเท่ากับ 27.51 แสดงดังภาพประกอบ 13 ซึ่งค่าสถิติที่ได้นี้จะนำไปใช้ทดสอบสมมติฐานการวิจัยต่อไป

ตาราง 15 แสดงเวลาการประมวลผลในการตรวจสอบและแก้ไขปัญหา

Date	Time (Engineer)	Time (Program)	%Time Diff
20100201	186	16	8.60
20100202	178	15	8.43
20100203	193	16	8.29
20100204	123	15	12.20
20100205	178	15	8.43
20100206	122	15	12.30
20100207	185	16	8.65
20100208	145	15	10.34
20100209	211	16	7.58
20100210	195	16	8.21
20100211	198	16	8.08
20100212	168	15	8.93
20100213	118	15	12.71
20100214	198	16	8.08
20100215	178	16	8.99
20100216	130	15	11.54
20100217	127	15	11.81
20100218	177	16	9.04
20100219	165	15	9.09
20100220	143	15	10.49
20100221	186	16	8.60
20100222	178	15	8.43
20100223	193	16	8.29
20100224	123	15	12.20
20100225	158	15	9.49
20100226	142	15	10.56
20100227	185	16	8.65
20100228	145	15	10.34

ตาราง 15 (ต่อ)

Date	Time (Engineer)	Time (Program)	%Time Diff
20100301	211	16	7.58
20100302	195	16	8.21
20100303	198	16	8.08
20100304	168	15	8.93
20100305	118	15	12.71
20100306	198	16	8.08
20100307	178	16	8.99
20100308	130	15	11.54
20100309	127	15	11.81
20100310	177	16	9.04
20100311	165	15	9.09
20100312	143	15	10.49
20100313	186	16	8.60
20100314	178	15	8.43
20100315	193	16	8.29
20100316	123	15	12.20
20100317	178	15	8.43
20100318	142	15	10.56
20100319	165	16	9.70
20100320	145	15	10.34
20100321	211	16	7.58
20100322	195	16	8.21
20100323	198	16	8.08
20100324	168	15	8.93
20100325	148	15	10.14
20100326	158	16	10.13
20100327	178	16	8.99
20100328	130	15	11.54
20100329	127	15	11.81
20100330	177	16	9.04
20100331	165	15	9.09
Average	166.12	15.46	9.54
SD	27.51	0.50	1.49



ภาพประกอบ 13 ระยะเวลาเฉลี่ยในการประมวลผลด้วยโปรแกรมและด้วยผู้ดูแลระบบ

ตาราง 16 แสดงสถิติเปรียบเทียบระยะเวลาการทำงานเฉลี่ย

Time Process by	Min	% time to process	SD
Engineer	166.12	100 %	27.51
Program	15.46	9.54 %	0.50

จากตาราง 16 พบว่า ระบบที่พัฒนาขึ้นสามารถทำการประมวลผลด้วยระยะเวลาที่น้อยลงจากเดิมเฉลี่ย 90.46% เมื่อเทียบกับการทำงานด้วยวิธีเดิมคือการตรวจสอบโดยวิศวกรที่ใช้ระยะเวลาเฉลี่ย 166.12 นาที

2. ผลของการทดสอบสมมติฐานการวิจัย

ในการวิจัยนี้มีการตั้งสมมติฐาน 2 ข้อซึ่งเป็นการทดสอบทั้งในด้านของความถูกต้อง และด้านระยะเวลาจากการประมวลผลด้วยโปรแกรมเปรียบเทียบกับวิธีการแก้ไขปัญหของผู้ดูแลระบบดังนี้

2.1 ผลลัพธ์จากการประมวลผลของระบบที่สร้างขึ้น มีความถูกต้องสอดคล้องกับการพิจารณาด้วยวิธีการแก้ไขปัญหของผู้ดูแลระบบ นั่นคือค่าเฉลี่ยรวมของค่าความแตกต่างของเปอร์เซ็นต์ส่วนแบ่งระหว่างผลการประมวลผลของโปรแกรม กับการแก้ไขปัญหของผู้ดูแลระบบ

2.2 ระบบที่สร้างขึ้นใช้ระยะเวลาในการประมวลผล น้อยกว่าระยะเวลาในการแก้ไขปัญห ด้วยวิธีปกติของผู้ดูแลระบบเอง

ซึ่งกระบวนการทดสอบและผลของการทดสอบสมมติฐานทั้ง 2 ข้อ มีดังนี้

1. การทดสอบสมมติฐานด้านของความถูกต้อง

- จากสมมติฐานในข้อ 1. สามารถแปลงให้อยู่รูปของสมมติฐาน H_0 , H_1 ได้ดังนี้

$$H_0 : \mu < 2$$

$$H_1 : \mu > 2$$

- คำนวณค่าสถิติทดสอบจากสูตรการคำนวณค่า Z จากพารามิเตอร์

$\bar{X} = 0.21$, $SD = 0.93$, $\mu_0 = 2$ และ $n = 31$ ซึ่งผลการคำนวณได้ค่า $Z = -11.19$

- กำหนดระดับนัยสำคัญ $\alpha = 0.05$
- สร้างเขตปฏิเสธ เนื่องจากการทดสอบสมมติฐานนี้เป็นการทดสอบทางเดียว

ทางซ้าย จึงสามารถสร้างเขตปฏิเสธได้ดังนี้

เขตปฏิเสธ : จะปฏิเสธสมมติฐาน H_0 ถ้า $Z < -Z_{\alpha=0.05}$ หรือ $Z < -1.645$

- สรุปผลการทดสอบ

เนื่องจากค่าสถิติทดสอบ Z ที่คำนวณได้คือ -11.19 ซึ่งน้อยกว่า -1.646 จึงอยู่ในเขตปฏิเสธ H_0 แต่เนื่องจากสภาพความเป็นจริงของ H_0 นั้นมีค่าเป็นจริงเนื่องจาก ค่าเฉลี่ยรวมของค่าความแตกต่างของเปอร์เซ็นต์ส่วนแบ่งระหว่างผลการประมวลผลของโปรแกรม กับการพิจารณาด้วยวิธีปกติของผู้ดูแลระบบ เฉลี่ยไม่เกินค่ามาตรฐานที่ผู้ออกแบบระบบใช้ ($0.21 < 2$) จึงถือว่าเกิดความคลาดเคลื่อนในแบบที่ 1 (Type I Error) ที่ระดับนัยสำคัญ (α) เท่ากับ 0.05

ดังนั้นสามารถสรุปผลการทดสอบสมมติฐานการวิจัยด้านความถูกต้องได้ ดังนี้ ค่าเฉลี่ยรวมของค่าความแตกต่างของเปอร์เซ็นต์ส่วนแบ่งระหว่างผลการประมวลผลของโปรแกรมกับการพิจารณาด้วยวิธีปกติของผู้ดูแลระบบ มีค่าน้อยกว่าค่ามาตรฐาน อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 โดยมีค่าเฉลี่ยเท่ากับ 0.21, ค่าส่วนเบี่ยงเบนมาตรฐานเท่ากับ 0.93 หมายความว่า การ

ประมวลผลด้วยโปรแกรมกับการประมวลผลด้วยวิธีปกติของผู้ดูแลระบบมีความสอดคล้องใกล้เคียงกัน

2. การทดสอบสมมติฐานด้านของระยะเวลา

- จากสมมติฐานในข้อ 2. สามารถแปลงให้อยู่รูปของสมมติฐาน H_0 , H_1 ได้ดังนี้

$$H_0 : \mu < 30$$

$$H_1 : \mu > 30$$

- คำนวณค่าสถิติทดสอบจากสูตรการคำนวณหาค่า Z จากพารามิเตอร์

$\bar{X} = 15.46$, $SD = 0.50$, $\mu_0 = 30$ และ $n = 59$ ซึ่งผลการคำนวณได้ค่า $Z = -242.33$

- กำหนดระดับนัยสำคัญ $\alpha = 0.05$

- สร้างเขตปฏิเสธ เนื่องจากการทดสอบสมมติฐานนี้เป็นการทดสอบทางเดียวทางซ้าย จึงสามารถสร้างเขตปฏิเสธได้ดังนี้

เขตปฏิเสธ : จะปฏิเสธสมมติฐาน H_0 ถ้า $Z < -Z_{\alpha=0.05}$ หรือ $Z < -1.645$

- สรุปผลการทดสอบ

เนื่องจากค่าสถิติทดสอบ Z ที่คำนวณได้คือ -242.33 ซึ่งน้อยกว่า -1.646 จึงอยู่ในเขตปฏิเสธ H_0 แต่เนื่องจากสภาพความเป็นจริงของ H_0 นั้นมีค่าเป็นจริงเนื่องจาก ค่าเฉลี่ยของระยะเวลาในการประมวลผลด้วยโปรแกรม น้อยกว่าค่าเฉลี่ยของระยะเวลาในการพิจารณาด้วยวิธีปกติของผู้ออกแบบระบบเอง ($15.46 < 30$) จึงถือว่าเกิดความคลาดเคลื่อนในแบบที่ 1 (Type I Error) ที่ระดับนัยสำคัญเท่ากับ 0.05

ดังนั้นสามารถสรุปผลการทดสอบสมมติฐานการวิจัยด้านระยะเวลาได้ ดังนี้ ระบบที่สร้างขึ้นใช้ระยะเวลาในการประมวลผล น้อยกว่าระยะเวลาในการพิจารณาด้วยวิธีปกติของผู้ออกแบบระบบเองอย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 โดยมีค่าเฉลี่ยเท่ากับ 15.46 นาที, ค่าส่วนเบี่ยงเบนมาตรฐานเท่ากับ 0.50

บทที่ 5

สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

การวิจัยเรื่องระบบพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานี่ฐานสามารถสรุปผลโดยมีรายละเอียดตามหัวข้อต่าง ๆ ดังนี้

1. สรุปผลการวิจัย
2. อภิปรายผลการวิจัย
3. ข้อเสนอแนะ

สรุปผลการวิจัย

ผลการดำเนินงานวิจัยนี้แบ่งออกเป็น 3 ส่วน คือ

- การพัฒนาระบบพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานี่ฐาน
- การประเมินผลเพื่อทดสอบความเชื่อมั่นของเครื่องมือในการวิจัย
- การประเมินผลเกี่ยวกับความสามารถของระบบและทดสอบสมมติฐานการวิจัย

ซึ่งแต่ละส่วนสามารถสรุปผลได้ดังนี้

1. การพัฒนาระบบพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานี่ฐาน

การพัฒนาระบบใช้การออกแบบ จะใช้โปรแกรม Shell Script ในการตรวจสอบและแก้ไข โดยใช้วิธีการตั้งเวลาในการตรวจสอบแล้วทำการหาความครบถ้วนของข้อมูลน้อยกว่า 95% และผลลัพธ์ 2 ส่วน คือ

ส่วนที่ 1 การแสดงผลหลังจากการตรวจสอบผ่านทาง SMS และ มีการเขียน Log ไฟล์ โดยจะระบุถึง ยี่ห้อขอมุขมสาย, ภาค และ Measurement

ส่วนที่ 2 การทำงานแก้ไขปัญหาเบื้องต้นของระบบ ที่ข้อมูลขาดความครบถ้วน อันเนื่องมาจากสาเหตุของ Node ต้นทางเกิดปัญหา สามารถดูผลใน Log ไฟล์

2. การประเมินผลเพื่อทดสอบความเชื่อมั่นของเครื่องมือในการวิจัย

การประเมินผลเพื่อทดสอบความเชื่อมั่นของเครื่องมือที่ใช้ในการวิจัยแบ่งออกเป็น 2 ส่วน ตามรายการเครื่องมือที่ใช้ในการวิจัยซึ่งสามารถสรุปผล ได้ดังนี้

- การทดสอบความเชื่อมั่นของโปรแกรมพัฒนาประสิทธิภาพส่วนของสถานี่ฐาน ใช้ผลการประเมินความพึงพอใจในการใช้งานระบบและผลการประเมินสมรรถนะ (Performance) ของระบบ ด้วยแบบสอบถามจากผู้เชี่ยวชาญ เมื่อนำคะแนนเฉลี่ยของแต่ละหัวข้อมาผ่านระเบียบวิธีการหาค่าทางสถิติเพื่อหาค่าเฉลี่ย พบว่า ค่าเฉลี่ยทั้งหมดอยู่ที่ 4.71 ค่าเบี่ยงเบนมาตรฐานทั้งหมดอยู่ที่ 0.33 ในส่วนการประเมินพึงพอใจในด้านสะดวกรวดเร็วในการใช้งาน, ความพึงพอใจในด้านการจัดหน้าจอ

การแสดงผลของโปรแกรมมีความเหมาะสม, ผลลัพธ์ที่ได้จากโปรแกรมสอดคล้องกับหลักการในการประเมิน, ความเหมาะสมในการทำงานของโปรแกรมโดยรวม, ความพึงพอใจในด้านสามารถเรียนรู้การใช้งานได้ง่ายและระบบมีการแบ่งขั้นตอนการประมวลผลที่สามารถเข้าใจได้ง่ายชัดเจน และข้อมูลที่ได้จากการประมวลผลของโปรแกรมสามารถนำไปใช้ประโยชน์ต่อไปได้อยู่ในระดับมากที่สุด และการแสดงผลหน้าจอของโปรแกรมตัวหนังสือสามารถอ่านได้ง่าย ชัดเจนอยู่ในระดับมาก ในส่วนการประเมินสมรรถนะ (Performance) ของระบบ ผู้เชี่ยวชาญมีความคิดเห็นว่ารระบบที่สร้างขึ้นสามารถประมวลผลได้อย่างรวดเร็ว ซึ่งส่งผลให้ลดระยะเวลาในการประเมินได้ โดยผลลัพธ์ที่ได้จากโปรแกรมมีความถูกต้องเชื่อถือได้ และสามารถนำไปปรับใช้เป็นเครื่องมือในการประเมินการจัดเส้นทางเดินของปริมาณการใช้ในระบบจริงแทนการประเมินด้วยวิธีปกติ (Manual) ได้ นอกจากนี้ผู้เชี่ยวชาญได้ให้ข้อเสนอแนะสำหรับการพัฒนาระบบให้มีประสิทธิภาพมากยิ่งขึ้นในอนาคต คือ การใช้ให้เหมาะสมยิ่งขึ้น (Optimize) อื่นๆ เช่น การจัดการตัวแปร (Parameter) ของการใช้งานแบบโปรแกรมเดี่ยว (Single Script) รวมกับการจัดกลุ่มผลลัพธ์ที่ได้จากโปรแกรม (Map exit code) ในการแยกแยะเงื่อนไขต่างๆ

- การทดสอบความเชื่อมั่นของแบบสอบถาม ใช้วิธีสัมประสิทธิ์แอลฟา (α -coefficient) ของครอนบาค พบว่าแบบสอบถามในการวิจัยนี้มีค่าสัมประสิทธิ์แอลฟาเท่ากับ 1.14 ซึ่งมากกว่าค่ามาตรฐาน 0.80 ดังนั้น แบบสอบถามในการวิจัยนี้มีความเที่ยงโดยมีความเชื่อมั่นเท่ากับ 1.14

3. การประเมินผลเกี่ยวกับความสามารถของระบบและทดสอบสมมติฐานการวิจัย

จากการทดสอบกลุ่มตัวอย่างและคำนวณค่าทางสถิติที่เกี่ยวข้องกับการทดสอบสมมติฐานพบว่า

- ค่าเฉลี่ยรวมของค่าความแตกต่างของเปอร์เซ็นต์ส่วนแบ่งระหว่างการประมวลผลของโปรแกรมกับการพิจารณาด้วยวิธีปกติของผู้ดูแลระบบ มีค่าน้อยกว่าค่ามาตรฐาน อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 โดยมีค่าเฉลี่ยเท่ากับ 1.10, และค่าส่วนเบี่ยงเบนมาตรฐานเท่ากับ 0.76 ซึ่งลักษณะของการวิจัยนี้ ค่าเฉลี่ยรวมของค่าความแตกต่างของเปอร์เซ็นต์ส่วนแบ่งระหว่างการประมวลผลของโปรแกรมกับการตรวจสอบและแก้ไขเบื้องต้นด้วยวิธีปกติของผู้ดูแลระบบยังมีค่าเข้าใกล้ศูนย์จะอยู่ในระดับดีเนื่องจากหมายความว่าผลการประมวลผลด้วยโปรแกรมกับการประมวลผลด้วยวิธีปกติของผู้ดูแลระบบมีความสอดคล้องใกล้เคียงกัน

- ระบบที่สร้างขึ้นใช้ระยะเวลาในการประมวลผล น้อยกว่าระยะเวลาในการพิจารณาด้วยวิธีปกติของผู้ดูแลระบบเองอย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 โดยมีค่าเฉลี่ยเท่ากับ 15.45 นาที, ค่าส่วนเบี่ยงเบนมาตรฐานเท่ากับ 0.51

อภิปรายผลการวิจัย

การอภิปรายผลของการสร้างระบบการพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะ ส่วนฐานนี้ฐาน โดยใช้โปรแกรม Shell Script และ SQL มีรายละเอียดดังนี้

1. จากผลการประเมินเกี่ยวกับความพึงพอใจของการใช้งานระบบและการประเมินสมรรถนะของระบบจากผู้เชี่ยวชาญ และการใช้งานระบบของผู้วิจัยเองพบว่า ระดับความพึงพอใจโดยรวมในด้านต่าง ๆ ของระบบอยู่ในระดับความพึงพอใจมาก และระบบที่สร้างขึ้นสามารถช่วยลดระยะเวลาในการตรวจสอบและแก้ไขปัญหาเบื้องต้นได้ โดยผลลัพธ์ที่ได้จากโปรแกรมมีความถูกต้องเชื่อถือได้ สามารถนำไปปรับใช้เป็นเครื่องมือในการใช้ในระบบจริงได้ แต่ตัวระบบยังมีข้อจำกัดความหลากหลายของยี่ห้อของ Node อยู่ เนื่องจากการตรวจสอบและการแก้ไขปัญหาไม่มีรูปแบบที่เหมือนกัน

2. ผลของการวิจัยนี้สอดคล้องกับสมมติฐานการวิจัยและวัตถุประสงค์ที่ตั้งไว้ โดยมีนัยสำคัญทางสถิติที่ระดับ 0.05 โดย โปรแกรมที่สร้างขึ้นสามารถสามารถช่วยลดระยะเวลาในการตรวจสอบและแก้ไขปัญหาเบื้องต้นได้ มีความถูกต้องเชื่อถือได้ และสอดคล้องกับวิธีการตรวจสอบวิธีปกติของผู้ดูแลระบบ

ข้อเสนอแนะ

1. ข้อจำกัดของการวิจัย

ระบบที่พัฒนาขึ้นปัจจุบันมีการพัฒนาขึ้นเฉพาะการตรวจสอบและแก้ไขยี่ห้อ Huawei รุ่น B32 ดังนั้นเพื่อให้สามารถใช้งานกับระบบได้อย่างสมบูรณ์ ต้องทำการพัฒนาในรุ่นอื่น เช่น Ericsson Nokia Siemens เป็นต้น

2. ข้อเสนอแนะสำหรับการวิจัยต่อไป

การวิจัยนี้มีข้อเสนอแนะอันจะเป็นประโยชน์ต่อผู้วิจัยที่ต้องการนำผลของการวิจัยไปพัฒนาหรือนำไปใช้ต่อ ดังนี้

- การทำงานปัจจุบัน ยังไม่สามารถระบุปัญหาเป็นรายชั่วโมงได้ ทำให้การแก้ไขปัญหาต้องทำเป็นรายวัน จึงทำให้ประสิทธิภาพของเครื่อง Server มากกว่าการทำการแก้ไขปัญหารายชั่วโมง ดังนั้นการพัฒนาระบบต่อไป

- การทำงานในปัจจุบันนั้น จะใช้การตั้งเวลาแต่ละขั้นตอนในการทำงาน ดังนั้นถ้าสามารถผูกขั้นตอนแต่ละขั้นตอนเป็น 1 ขั้นตอน จะทำให้ระยะเวลาของการทำงานของโปรแกรมจะเร็วขึ้น

บรรณานุกรม

บรรณานุกรม

- กรมที่ดิน. (2008). *Oracle Database Administration*. สืบค้นเมื่อ 2 สิงหาคม 2552, จาก http://www.dol.go.th/dol/images/medias/dol/example/news_dol/october-2008/mt0519-2108/1-dataBase-server/6.Oracle%20Database%2010g%20Administration.pdf.
- เกียรติสุดา ศรีสุข. (2549). *ระเบียบวิธีวิจัย*. เชียงใหม่: ภาควิชาประเมินผลและวิจัยการศึกษา มหาวิทยาลัยเชียงใหม่.
- จักรี จันทสร. (ม.ป.ป.). *สถานีฐาน Base Station*. เชียงราย: วิทยาลัยเทคนิคกาญจนาภิเษก เชียงราย. สืบค้นเมื่อ 2 สิงหาคม 2552, จาก http://www.geocities.com/chakri_cri/base_11.htm.
- ชนวัฒน์ ศรีสอาน. (2542). *การออกแบบและพัฒนาฐานข้อมูล*. นครราชสีมา: มหาวิทยาลัยเทคโนโลยีสุรนารี.
- ชูศรี วงศ์รัตน์. (2546). *เทคนิคการใช้สถิติเพื่อการวิจัย*. พิมพ์ครั้งที่ 9 กรุงเทพฯ: มหาวิทยาลัยศรีนครินทรวิโรฒ.
- ยุทธ ไกยวรรณ. (2545). *พื้นฐานการวิจัย*. พิมพ์ครั้งที่ 4 กรุงเทพฯ: สุวีริยาสาส์น.
- วิทยาลัยชุมชนภูเก็ต. (ม.ป.ป.). *Shell Script*. มหาวิทยาลัยสงขลานครินทร์ สืบค้นเมื่อ 2 สิงหาคม 2552, จาก <http://www.pcc.psu.ac.th/~rattana/344201/unix/main7.htm>.
- ศิริลักษณ์ ไรจนิกจ๋านวย. (2542). *ระบบฐานข้อมูล*. พิมพ์ครั้งที่ 3. กรุงเทพฯ: ดวงกลมสมัย.
- สถิตย์โชค โพธิ์สอาด. (2552). *การออกแบบและพัฒนาฐานข้อมูล*. สืบค้นเมื่อ 2 สิงหาคม 2552, จาก http://www.sut.ac.th/ist/coursesonline/204204/Lecture/204204_47_09.pdf.
- Brownlee N. (2001). *Using NeTraMet for Production Traffic Measurement*. New Zealand: In IEEE Xplore. Retrieved April 26, 2009, from [http://ieeexplore.ieee.org/search/freesrchabstract.jsp?arnumber=918033&isnumber=19847&punumber=7332&k2dockey=918033@ieeecnfs&query=\(using+netramet+for+production+traffic+measurement\)metadata&pos=0](http://ieeexplore.ieee.org/search/freesrchabstract.jsp?arnumber=918033&isnumber=19847&punumber=7332&k2dockey=918033@ieeecnfs&query=(using+netramet+for+production+traffic+measurement)metadata&pos=0).
- Cisco Corporation. (n.d.). *Cisco Mobile Exchange (CMX) Solution Guide*. สืบค้นเมื่อ 2 สิงหาคม 2552 จาก http://www.cisco.com/en/US/netsol/ns341/ns396/ns177/ns278/networking_solutions_design_guide_chapter09186a00801219ac.html.
- Farraposo, Slvia; Owezarski, Philippe; & Monteiro, Edmundo. (2005). *On the Use of Traffic Monitoring and Measurements for Improving Networking*. In IEEE Xplore. Retrieved April 26, 2009, from <http://ieeexplore.ieee.org/search/freesrchabstract.jsp?arnumber=1517664&isnumber=32498&punumber=10175&k2dockey=1517664>

@ieeecnfs&query=(on+the+use+of+traffic+monitoring+and+measurements+for+improving+networking)metadata&pos=0.

International Telecommunication Union. (2009). *The World in 2009: ICT FACTS AND FIGURES*. p.1 ITU Telecom World2009 Geneva.

Wikipedia. (2007). *Traffic measurement (Telecommunications)*. Retrieved July 12, 2009, from [http://en.wikipedia.org/wiki/Traffic_measurement_\(telecommunications\)](http://en.wikipedia.org/wiki/Traffic_measurement_(telecommunications))

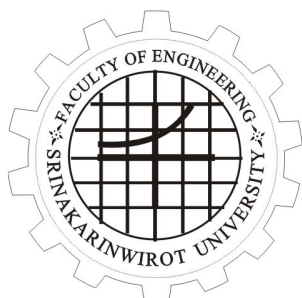
Xie, Gaogang; et al. (2003). *A Methodology of Effective Measurement for Link Traffic*. In IEEE Xplore. Retrieved April 26, 2009, from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=1209060&isnumber=27215

ภาคผนวก

แบบสอบถามและผลการประเมินความพึงพอใจของผู้ใช้ระบบ

1.แบบฟอร์มของแบบสอบถาม

แบบสอบถามสำหรับประเมินความพึงพอใจของผู้ใช้
ระบบพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานี่ฐาน



สาขาการจัดการวิศวกรรม มหาวิทยาลัยศรีนครินทรวิโรฒ ประสานมิตร
เลขที่ 114 สุขุมวิท 23 คลองเตยเหนือ เขตวัฒนา กรุงเทพมหานคร
10110 โทรศัพท์ : 02-649-5000

แบบประเมินความพึงพอใจของผู้ใช้ระบบ
พัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถานี่ฐาน

ส่วนที่ 1 ข้อมูลส่วนบุคคล

☐ นาย ☐ นาง ☐ นางสาว ชื่อ.....นามสกุล.....

สถานที่ทำงาน.....

เลขที่ หมู่ที่ ซอย ถนน.....

ตำบล/แขวงอำเภอ/เขต

จังหวัด รหัสไปรษณีย์.....

โทรศัพท์.....

ตำแหน่งงาน.....

ส่วนงาน.....

ส่วนที่ 2 ประเมินความพึงพอใจการใช้งานระบบ					
คำชี้แจง กรุณาใส่เครื่องหมาย / ลงในช่องตรงกับหมายเลข 5 , 4 , 3 , 2 หรือ 1 ที่ตรงกับความคิดเห็นของท่านมากที่สุดตามของระดับความพึงพอใจในแต่ละประเด็น					
ประเด็นในการประเมินความพึงพอใจ ในการใช้งานระบบ	ระดับความพึงพอใจ				
	มากที่สุด (5)	มาก (4)	ปานกลาง (3)	น้อย (2)	น้อยที่สุด (1)
1. ความสะดวกรวดเร็วในการใช้งาน					
2. สามารถเรียนรู้การใช้งานได้ง่าย					
3. การแสดงผลของโปรแกรมมีความเหมาะสม					
4. การแสดงผลหน้าจอของโปรแกรมตัวหนังสือสามารถอ่านได้ง่าย ชัดเจน					
5. ระบบมีการแบ่งขั้นตอนการประมวลผลที่สามารถเข้าใจได้ง่ายและชัดเจน					
6. ผลลัพธ์ที่ได้จากโปรแกรมสอดคล้องกับหลักการในการประเมิน					
7. ข้อมูลที่ได้จากการประมวลผลของโปรแกรมสามารถนำไปใช้ประโยชน์ต่อไปได้					
8. ความเหมาะสมในการทำงานของโปรแกรมโดยรวม					
ส่วนสำหรับผู้วิจัย					
คะแนนรวมแต่ละระดับความพึงพอใจ					
คะแนนรวมทั้งหมด (คะแนนเต็ม = 40)					
คะแนนเฉลี่ย (คะแนนเฉลี่ยเต็ม = 5)					

ส่วนที่ 3 ประเมิน Performance ของระบบ

คำชี้แจง กรุณาใส่เครื่องหมาย / ลงในช่อง ☐ ที่ตรงกับความคิดเห็นของท่านในแต่ละประเด็น

1. โปรแกรมสามารถช่วยลดระยะเวลาในการประเมินได้ ☐ เห็นด้วย ☐ ไม่เห็นด้วย
2. ระบบสามารถประมวลได้อย่างรวดเร็ว ☐ เห็นด้วย ☐ ไม่เห็นด้วย
3. ผลลัพธ์ที่ได้จากโปรแกรมมีความถูกต้องเชื่อถือได้ ☐ เห็นด้วย ☐ ไม่เห็นด้วย
4. ผู้ประเมินคิดว่าสามารถใช้โปรแกรมนี้แทนการประเมิน
ด้วยวิธีปกติ (Manual) ได้ ☐ เห็นด้วย ☐ ไม่เห็นด้วย

ส่วนที่ 4 ข้อเสนอแนะเพิ่มเติม

1. เกี่ยวกับด้านคุณลักษณะของการพัฒนาประสิทธิภาพของระบบการจัดการสมรรถนะส่วนสถา
ฐาน

.....

.....

.....

.....

.....

.....

2. เกี่ยวกับด้านผลลัพธ์ที่ได้จากการใช้โปรแกรมพัฒนาประสิทธิภาพของระบบการจัดการ
สมรรถนะส่วนสถาฐาน

.....

.....

.....

.....

.....

.....

3. ด้านอื่น ๆ

.....

.....

.....

.....

.....

.....

(ลายมือชื่อ)ผู้ประเมิน

(.....)

ประวัติย่อผู้วิจัย

ประวัติย่อผู้วิจัย

ชื่อ ชื่อสกุล	ณรังสรรค์ สุรเจริญชัยกุล
วันเดือนปีเกิด	30 กันยายน 2528
สถานที่เกิด	กรุงเทพมหานคร
สถานที่อยู่ปัจจุบัน	774 จรัญสนิทวงศ์ วัดท่าพระ บางกอกใหญ่ กรุงเทพฯ 10600
ตำแหน่งหน้าที่การงานปัจจุบัน	วิศวกร
สถานที่ทำงานปัจจุบัน	บริษัท แอดวานซ์ อินโฟร์ เซอร์วิส จำกัด (มหาชน)

ประวัติการศึกษา

พ.ศ. 2539	ชั้นประถมศึกษาปีที่ 6 จากโรงเรียนเสสสะเวชวิทยา กรุงเทพฯ
พ.ศ. 2542	ชั้นมัธยมศึกษาปีที่ 3 จากโรงเรียนทวีธาภิเศก กรุงเทพฯ
พ.ศ. 2545	ชั้นมัธยมศึกษาปีที่ 6 จากโรงเรียนทวีธาภิเศก กรุงเทพฯ
พ.ศ. 2549	วิศวกรรมศาสตรบัณฑิต สาขาไฟฟ้าสื่อสาร (วศ.บ) จากมหาวิทยาลัยศรีนครินทรวิโรฒ องค์กรักษ์ นครนายก