โครงงานวิศวกรรมคอมพิวเตอร์

เรื่อง

เครื่องมือตรวจวิเคราะห์ระบบดีเอ็นเอส
DNS Analyzer Tool

นายอภิวัฒน์ เปลี่ยนจิตรดี

เสนอ

ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรบัณฑิต
พ.ศ. 2544

ใบรับรองโครงงานวิศวกรรม
ภาควิชาวิศวกรรมคอมพิวเตอร์

เรื่อง

เครื่องมือตรวจวิเคราะห์ระบบดีเอ็นเอส

DNS Analyzer Tool

โดย

นายอภิวัฒน์ เปลี่ยนจิตรดี เลขประจำตัว 40056749

พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาโครงงาน ……..

(ผศ. สุรศักดิ์ สงวนพงษ์)

หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์ ……..

(อ. ประดนเดช นีละคุปต์)

วันที่……...เดือน……..….พ.ศ……

นาย อภิวัฒน์ เปลี่ยนจิตรดี 2544 : เครื่องมือตรวจวิเคราะห์ระบบดีเอ็นเอส
โครงงานวิศวกรรมคอมพิวเตอร์ ปริญญาวิศวกรรมศาสตร์บัณฑิต (วิศวกรรมคอมพิวเตอร์)
อาจารย์ที่ปรึกษา : ผศ. สุรศักดิ์ สงวนพงษ์

บทคัดย่อ

โครงงานนี้กล่าวถึงการพัฒนาเครื่องมือตรวจวิเคราะห์ความผิดพลาดในแฟ้มข้อมูลดีเอ็นเอสที่ผู้ดูแลระบบต้องจัดเตรียมไว้ โครงงานนี้จะนำข้อมูลมาจากการขอถ่ายโอนโซนจากโซนที่ต้องการวิเคราะห์ และจะวิเคราะห์โดยแบ่งความผิดพลาดในการวิเคราะห์ออกเป็นหัวข้อตามเรคอร์ดต่างๆ ดังนี้ A, CNAME, MX, NS, SOA, HINFO, PTR นอกจากนี้ยังเพิ่มการตรวจสอบข้อผิดพลาดเกี่ยวกับการมอบอำนาจ (delegation information) ระบบความปลอดภัย (security) และเรื่องเบ็ดเตล็ดอื่นๆ (Miscellaneous) หลังจากวิเคราะห์โซนที่ต้องการเสร็จสิ้นแล้วจะแสดงผลลัพธ์ออกทาง GUI (Graphic User Interface) เพื่อสะดวกต่อการใช้งาน การแสดงข้อผิดพลาดนั้นจะแบ่งเป็นหัวข้อ ซอฟแวร์ที่พัฒนาขึ้นในโครงงานนี้พัฒนาโดยใช้ภาษาจาวาและสามารถใช้ได้ในทุกระบบปฏิบัติการที่สนับสนุน JVM (JAVA Virtual Machine)

สารบัญ

สารบัญรูป

หน้า

สารบัญตาราง

หน้า

บทที่ 1
บทนำ

การใส่ข้อมูลของโฮสต์ในแฟ้มข้อมูลดีเอ็นเอสนั้น ในปัจจุบันต้องใช้โปรแกรมเมอร์ในการใส่ข้อมูล ดังนั้นจึงอาจเกิดความผิดพลาดในการใส่ข้อมูลได้ (ลักษณะของความผิดพลาดจะกล่าวในหัวข้อ 3.4 ลักษณะของความผิดพลาดและวิธีตรวจสอบ ซึ่งเป็นขอบเขตของโครงงานด้วย) ถ้าเราใช้โปรแกรมเมอร์มาหาความผิดพลาดเหล่านี้ด้วยตนเองนั้นจะทำได้ยาก เพราะแฟ้มข้อมูลดีเอ็นเอสของบางโซนอาจมีข้อมูลขนาดใหญ่มาก ดังนั้นโครงงานนี้จึงพัฒนาซอฟแวร์มาเพื่อหาข้อผิดพลาดต่างๆ เหล่านี้ โดยซอฟแวร์จะขอถ่ายโอนโซนจากโซนที่ต้องการวิเคราะห์มาวิเคราะห์ว่ามีข้อผิดพลาดตามที่ได้ระบุว่าเป็นข้อผิดพลาดหรือไม่ ถ้ามีก็จะแสดงผลให้แก่ผู้ใช้ซอฟแวร์ทราบ แต่ถ้าโซนที่เราต้องการวิเคราะห์ไม่อนุญาตให้เราขอถ่ายโอนโซนเราก็ไม่สามารถตรวจข้อผิดพลาดได้

1.1 ปัญหาในการติดตั้งระบบ DNS

ปัญหาเหล่านี้เป็นปัญหาทั่วๆ ไปที่เกิดขึ้นบ่อยครั้งในการติดตั้งระบบ DNS เราเรียกปัญหาเหล่านี้ว่า “Unlucky Thirteen” [2]

1.1.1 ลืมเพิ่มซีเรียลนัมเบอร์

หัวใจหลักของปัญหานี้ก็คือสเลฟเนมเซอร์ฟเวอร์จะไม่นำไฟล์ db ที่ผู้ใช้แก้ไขในไพรมารี่มาเก็บไว้แทนไฟล์ db เก่าของมันที่เป็นไฟล์ db ของไพรมารี่ก่อนที่จะแก้ไข เพราะสเลฟเนมเซอร์ฟเวอร์คิดว่าข้อมูลของโซนไม่ได้เปลี่ยนแปลงเนื่องมาจากซีเรียลนัมเบอร์ยังคงมีค่าเท่าเดิม

1.1.2 ลืมส่งสัญญาณให้ไพร์มารี่มาสเตอร์เซอร์ฟเวอร์

มีบางครั้งที่ผู้ใช้อาจลืมส่งสัญญาณไปบอกกับไพรมารี่มาสเตอร์เนมเซอรฟ์เวอร์ หลังจากที่ผู้ใช้ได้แก้ไขไฟล์ conf หรือไฟล์ db เนมเซอร์ฟเวอร์จะไม่รู้ว่ามีการแก้ไขไฟล์เหล่านี้ เพราะมันจะไม่ตรวจ timestamp ของไฟล์ด้วยตัวมันเองและมันจะไม่สังเกตว่าไฟล์เหล่านี้มีการเปลี่ยนแปลงหรือไม่ การเปลี่ยนแปลงที่ผู้ใช้ทำไปจะไม่มีผลกับข้อมูลในเนมเซอร์ฟเวอร์ คือ โซนใหม่ไม่ถูกโหลดขึ้นไปแทนข้อมูลเก่า

1.1.3 สเลฟเซอร์ฟเวอร์ไม่สามารถโหลดข้อมูลของโซนได้

ปัญหานี้มีอยู่ 3 สาเหตหลัก คือ 1. สูญเสียการติดต่อกับมาสเตอร์เนมเซอร์ฟเวอร์เนื่องจากเครือข่ายไม่ทำงาน (network failure) 2. ไอพีแอดเดรสของมาสเตอร์เซอร์ฟเวอร์ในไฟล์ conf ไม่ถูกต้อง 3. การเขียน syntax ผิดในไฟล์ข้อมูลของโซนบนมาสเตอร์เซอร์ฟเวอร์

1.1.4 เพิ่มชื่อในไฟล์ฐานข้อมูล แต่ลืมเพิ่ม เรคอร์ด PTR

เนื่องจากการเปลี่ยนจากชื่อโฮสต์ไปเป็นไอพีแอดเดรสนั้น มีความสัมพันธ์กับการเปลี่ยนจากไอพีแอดเดรสเป็นชื่อโฮสต์ ดังนั้นเมื่อผู้ใช้เพิ่มเรคอร์ด A เข้าไปในไฟล์ฐานข้อมูลโฮสต์แล้วผู้ใช้ก็จำเป็นต้องเพิ่มเรคอร์ด PTR สำหรับโฮสต์ที่เพิ่มเข้าไปใหม่ในไฟล์ฐานข้อมูลผกผันด้วย

1.1.5 เขียน Syntax ผิดในไฟล์ Conf หรือ ไฟล์ฐานข้อมูล

การเขียน syntax ผิดในไฟล์ conf และในไฟล์ฐานข้อมูลโซนนั้นเป็นธรรมดาที่เกิดขึ้นได้ แต่จะเกิดขึ้นมากหรือน้อยนั้นขึ้นอยู่กับประสบการณ์ของผู้เขียน ข้อผิดพลาดในไฟล์ conf นั้นจะเป็นเหตุให้เนมเซอร์ฟเวอร์ไม่สามารถโหลดโซนต่างๆ ได้

1.1.6 ไม่ได้เติมจุดที่อยู่ในตอนท้ายของชื่อโฮสต์ในไฟล์ฐานข้อมูล DNS

มันเกิดขึ้นได้ง่ายมากที่จะลืมเติมจุดในตอนท้ายของชื่อโฮสต์ในขณะที่เราแก้ไขไฟล์ db เพราะถ้าเราไม่เติมจุดในตอนท้ายของชื่อโฮสต์ named จะต่อท้ายชื่อโฮสต์ด้วยชื่อของโดเมน ทำให้ชื่อโฮสต์ที่ตั้งไว้ผิดพลาดไปและจะไม่สามารถอ้างถึงชื่อโฮสต์นั้นได้

1.1.7 ข้อมูลแคชหาย

ถ้าผู้ใช้ลืมติดตั้งไฟล์แคชบนโฮสต์ของผู้ใช้หรือถ้าผู้ใช้ไปลบมันโดยบังเอิญ เนมเซอร์ฟเวอร์ของผู้ใช้จะไม่สามารถหาชื่อที่อยู่นอกเหนือจากข้อมูลที่มันมีอำนาจหน้าที่ได้

1.1.8 สูญเสียการเชื่อมต่อกับระบบเครือข่าย

ไม่สามารถติดต่อกับเนมเซอร์ฟเวอร์ที่อยู่ในโดเมนอื่นได้

1.1.9 โดเมนย่อยหายไป

InterNIC สามารถทำกระบวนการต่างๆ ที่ผู้ใช้ร้องขอไปนั้นได้เร็วที่สุดนั้นก็ต้องใช้เวลา 1 หรือ 2 วัน เพื่อทำให้โดเมนที่ได้รับการมอบอำนาจของผู้ใช้ปรากฏในรูทเนมเซอร์ฟเวอร์ แต่ถ้า InterNIC ไม่จัดการให้กับโดเมนของผู้ใช้นั้น ผู้ใช้ก็ต้องรอต่อไป ถ้าข้อมูลการมอบอำนาจของผู้ใช้ยังไม่ปรากฏในเนมเซอร์ฟเวอร์ซึ่งเป็นโดเมนแม่ของผู้ใช้ ผู้ใช้ก็สามารถทำได้แต่เพียงมองดูข้อมูลต่างๆ ในอินเทอร์เน็ตเนมสเปซได้ แต่จะไม่มีใครบนอินเทอร์เน็ตซึ่งอยู่นอกโดเมนของผู้ใช้สามารถมาดูข้อมูลของผู้ใช้ได้

1.1.10 โดเมนย่อยไม่ถูกต้อง

โดเมนย่อยไม่ถูกต้องนั้นก็เป็นปัญหาที่พบได้บ่อยในอินเทอร์เน็ต เมื่อเราต้องการแก้ไขการมอบอำนาจต่างๆ นั้นเราจำเป็นต้องบอกโดยใช้คน โดยแจ้งให้ผู้ดูแลระบบที่เป็นโซนแม่ของผู้ใช้ทราบว่าเราได้แก้ไขอะไรบ้างในการติดตั้งเนมเซอร์ฟเวอร์ เช่น การเพิ่มเนมเซอร์ฟเวอร์ในโซน การเปลี่ยนไอพีแอดเดรส

ฯลฯ เพื่อให้ผู้ดูแลระบบของโซนแม่ของผู้ใช้แก้ไขข้อมูลที่อยู่ในการดูแลของเขาให้ตรงกับข้อมูลที่ผู้ใช้ได้แก้ไขไปแล้ว หากว่าผู้ใช้เปลี่ยนชื่อเนมเซอร์ฟเวอร์ของผู้ใช้โดยไม่บอกโดเมนแม่ของผู้ใช้ เวลามีคนต้องการดูข้อมูลในโซนของผู้ใช้ก็จะดูผ่านโดเมนแม่ของผู้ใช้ เขาก็จะเห็นชื่อเนมเซอร์ฟเวอร์เก่าที่ผู้ใช้ยังไม่ได้แก้ไข

1.1.11 เขียน Syntax ผิดใน resolv.conf

แม้ว่า syntax ของไฟล์ resolv.conf นั้นจะง่ายแต่ก็มีโอกาสที่จะเกิดความผิดพลาดได้เมื่อไปแก้ไขมัน และรีโซลเวอร์จะไม่สนใจบรรทัดที่มีข้อผิดพลาดของ syntax ใน resolv.conf เช่น ถ้าบรรทัดที่เป็นข้อมูลของ serch list ผิดพลาด รีโซลเวอร์ของผู้ใช้ก็ไม่สามารถจะร้องขอข้อมูลจากเนมเซอร์ฟเวอร์ที่ถูกต้องได้ หรือจะไม่สามารถร้องขอข้อมูลเนมเซอร์ฟเวอร์ได้เลยซักตัว

1.1.12 ไม่ได้กำหนดดีฟอล์โดเมน

ถ้าเราไม่ได้กำหนดดีฟอล์โดเมนแล้วก็จะไม่เป็นที่พอใจกับผู้ที่ชอบใช้ single-label names (การใส่เพียงชื่อโฮสต์ของเซอร์ฟเวอร์เท่านั้น)

1.1.13 มีการตอบกลับจากเครื่องที่ไม่ได้คาดหวัง

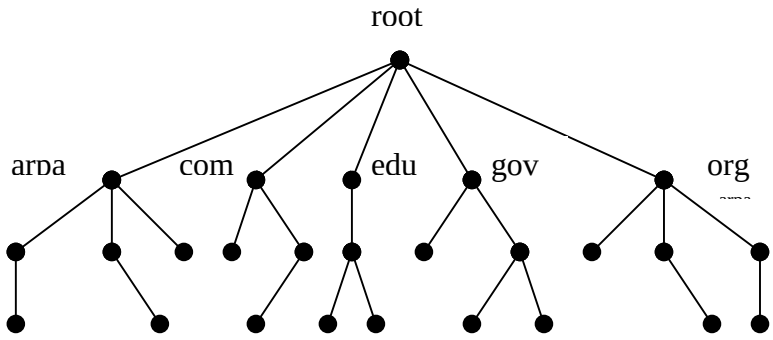
การตอบกลับนั้นไม่ได้มาจากไอพีแอดเดรสของเซอร์ฟเวอร์ที่ผู้ใช้ร้องขอข้อมูลไป เมื่อ BIND เซอร์ฟเวอร์ส่งการร้องขอข้อมูลไปยังรีโมทเซอร์ฟเวอร์ BIND จะต้องแน่ใจว่าคำตอบนั้นมาจากไอพีแอดเดรสของเซอร์ฟเวอร์นั้น ซึ่งการทำแบบนี้จะช่วยลดการรับคำตอบที่ถูกปลอมแปลงมา และ BIND เซอร์ฟเวอร์จะพยายามส่งคำตอบกลับไปยังอินเทอร์เฟสเน็ตเวิร์คเดียวกันกับที่มันได้รับการร้องขอเข้ามา

บทที่ 2
ทฤษฎีพื้นฐาน

2.1 โครงสร้างดีเอ็นเอส

ดีเอ็นเอสเป็นระบบชื่อที่มีฐานข้อมูลแบบกระจาย ซึ่งแต่ละเครือข่ายในอินเทอร์เน็ตจะมีดีเอ็นเอสเซอร์ฟเวอร์เก็บรักษาฐานข้อมูลและบริหารข้อมูลอย่างอิสระ เพื่อให้ไคลเอ็นต์ขอบริการสอบถามข้อมูลตามแบบโปรโตคอลที่กำหนด

อินเทอร์เน็ตดีเอ็นเอสมีโครงสร้างตามลำดับชั้นแบบโครงสร้างต้นไม้กลับหัวดังรูปที่ 1.1 ส่วนปลายสุดเป็นจุดที่ไม่สามารถแตกกิ่งออกไปได้อีกจะเป็น ชื่อโฮสต์ เช่น pirun.ku.ac.th หมายถึงโฮสต์ pirun ของ ku.ac.th โครงสร้างต้นไม้ทั้งโครงสร้างเรียกโดยทั่วไปว่า **โดเมนเนมสเปซ** หรือเรียกสั้นๆ ว่า **เนมสเปซ** ดังรูปที่ 2.1



รูปที่ 2.1 โดเมนเนมสเปซ

2.2 ชื่อโดเมน

คือชื่อที่กำกับประจำโหนดและเรียกชื่อโดยไล่ลำดับจากโหนดนั้นตามเส้นทางขึ้นไปยังราก การเขียนชื่อโดนเมนให้เขียนคั่นแต่ละโหนดด้วยจุด เช่น ku.ac.th

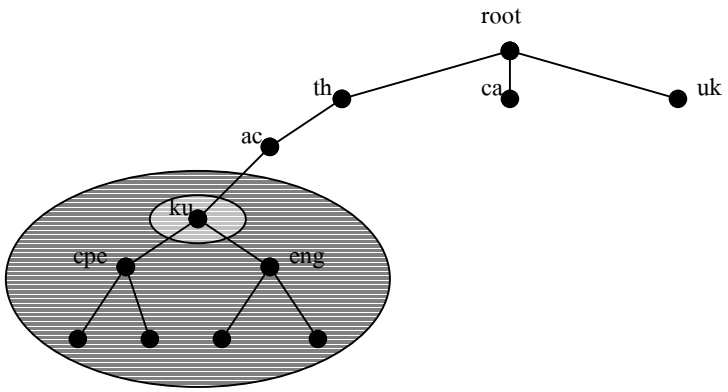
2.3 โดเมนและโซน

โซน หมายถึงต้นไม้ย่อยในดีเอ็นเอสที่มีการมอบอำนาจหน้าที่ให้มีผู้ดูแลเฉพาะ และภายในโซนอาจจะมีการแบ่งให้มีโซนย่อยออกไปอีกได้ตามคณะหรือหน่วยงาน แต่ละโซจะมีเนมเซอร์ฟเวอร์ทำหน้าที่เก็บรักษาข้อมูลประจำโซนไว้

เพื่อให้เห็นถึงความแตกต่างของโดเมนและโซนให้ชัดเจนยิ่งขึ้น ขอให้พิจารณาโดเมน ku.ac.th ในรูปที่ 2.2 ผู้ดูแลโดเมนได้รับมอบอำนาจมาเป็นลำดับเพื่อบริหารโดเมน

โดเมน ku ในรูปประกอบด้วยโดเมนระดับล่าง 2 โดเมนคือ cpe, eng สมมติให้ผู้ดูแลโดเมน ku บริหารเฉพาะโดเมน ku อย่างเดียว และมอบอำนาจให้โดเมน cpe, eng มีผู้ดูแลต่างหาก โดเมน ku จะถูกจัด

แบ่งออกเป็นโซน ku ซึ่งครอบคลุมเฉพาะโหนด ku เท่านั้น หรือกล่าวอีกนัยหนึ่งคือ โดเมน ku จะมีข้อมูล ku รวมทั้งข้อมูลใน cpe, eng แต่โซน ku จะมีเพียงข้อมูลของ ku เท่านั้น



รูปที่ 2.2 โดเมนและโซน ku

2.4 แฟ้มฐานข้อมูลดีเอ็นเอส

ซอฟแวร์ที่ให้บริการดีเอ็นเอสที่จะยกตัวอย่างต่อไปนี้ คือ ชุดซอฟแวร์ BIND (Berkeley Internet Name Domain) ซึ่งใช้บนระบบปฏิบัติการยูนิกซ์โดยมีเดมอน named เป็นโปรเซสเซอร์ฟเวอร์ให้บริการดีเอ็นเอส โดยผู้ดูแลระบบต้องจัดเตรียมแฟ้มข้อมูลต่อไปนี้คือ [1]

2.4.1 แฟ้มกำหนดงานเริ่มต้น (boot file)

เริ่มต้นเนมดีจะอ่านแฟ้มกำหนดงานจาก /etc/named.conf ภายในแฟ้มนี้จะบรรจุชื่อเซอร์ฟเวอร์และกำหนดที่เก็บแฟ้มข้อมูลอื่น ซึ่งได้แก่ ฐานข้อมูลบรรจุชื่อโฮสต์ ฐานข้อมูลบรรจุแอดเดรสผกผัน ฐานข้อมูลบรรจุแอดเดรสลูปแบ็ค

2.4.2 แฟ้มฐานข้อมูลโฮสต์ (host database file)

เป็นแฟ้มที่เก็บข้อมูลที่เนมเซอร์ฟเวอร์มีอำนาจดูแล ข้อมูลหลักในแฟ้มคือชื่อโฮสต์และไอพีแอดเดรสประจำโฮสต์รวมทั้งอาจมีข้อมูลอื่นเกี่ยวกับโฮสต์นั้นแทรกเพิ่มเติมอยู่ได้ รูปที่ 2.3 แสดงถึงตัวอย่างแฟ้มข้อมูลโฮสต์

```
; host database file for ku.ac.th
; name                  class type    server
ku.ac.th.                   IN      SOA   ns.ku.ac.th.        root.ns.ku.ac.th. (
                                                2000061300    ;serial
                                                10800         ;refresh
                                                3600          ;retry
                                                2592000 ;expire
                                                86400)    ;minimum TTL
;name servers
ku.ac.th.                   IN      NS      ns.ku.ac.th.
;mail  Hubs for the Domain
ku.ac.th.                   IN      NS      10      mail.ku.ac.th.
;host information
localhost.ku.ac.th.     IN  A       127.0.0.1
ns.ku.ac.th.                IN      A       158.108.2.67
www.ku.ac.th.               IN      A       158.108.2.69
nontri.ku.ac.th.  IN        A       158.108.2.71
;end of file
```

รูปที่ 2.3 ตัวอย่างแฟ้มฐานข้อมูลโฮสต์ในโดเมน ku.ac.th

ข้อมูลในรูปที่ 2.3 จะแบ่งออกเป็น เรคอร์ด หรือ รีซอร์สเรคอร์ด แต่ละเรคอร์ดมีคลาส IN ซึ่งหมายถึงคลาสเพื่อการใช้งานในอินเทอร์เน็ต ดีเอ็นเอสได้รับการออกแบบให้ทำงานโดยไม่จำกัดเฉพาะอินเทอร์เน็ต แต่ในที่นี้จะกล่าวเฉพาะคลาสอินเทอร์เน็ตเท่านั้น ชื่อเรคอร์ดสามารถใช้อักขระตัวเล็กและตัวใหญ่โดยไม่มีความแตกต่าง ยกเว้นชื่อแฟ้มที่ยูนิกซ์ถือว่ามีความแตกต่าง ทุกเรคอร์ดต้องเริ่มต้นที่คอลัมน์แรกของบรรทัดนั้นๆ เสมอ

2.4.3 แฟ้มฐานข้อมูลแอดเดรสผกผัน (reverse database file)

แฟ้มนี้ใช้เก็บข้อมูลการแปลงไอพีแอดเดรสไปเป็นชื่อโดเมน เรคอร์ดที่ใช้ในแฟ้มจะเป็นเรคอร์ด PTR ดังตัวอย่างในรูปที่ 2.4

```
;reverse address database for 108.158.in-addr.arpa
;name                          class   type    server
108.158.in-addr.arpa.          IN      SOA     ns.ku.ac.th.        root.ns.ku.ac.th. (
                                               2000061300          ;serial
                                               10800               ;refresh
                                               3600                ;retry
                                               2592000             ;exprie
                                               86400)              ;minimum TTL
;name servers
67.2.108.158.in-addr.arpa.     IN      NS      ns.ku.ac.th.
67.2.108.158.in-addr.arpa.     IN      PTR     ns.ku.ac.th.
69.2.108.158.in-addr.arpa.     IN      PTR     www.ku.ac.th.
71.2.108.158.in-addr.arpa.     IN      PTR     nontri.ku.ac.th.
;end of file
```

รูปที่ 2.4 แฟ้มแอดเดรสผกผัน 108.158.in-addr.arpa.

2.4.4 แฟ้มลูปแบ็ค (loopback file)

แฟ้มนี้ใช้กำหนดแอดเดรสลูปแบ็ค ค่าโดยปกติที่ใช้คือ 127.0.0.1 แฟ้มนี้มีข้อมูลดังตัวอย่างในรูปที่ 2.5

```
;loopback file 127.0.0.1
;name                    class   type    server
0.0.127.in-addr.arpa.    IN      SOA     ns.ku.ac.th.        root.ns.ku.ac.th. (
                                         2000061300          ;serial
                                         10800               ;refresh
                                         3600                ;retry
                                         259200              ;exprie
                                         86400)              ;minimum TTL
;name servers
0.0.127.in-addr.arpa.    IN      NS      ns.ku.ac.th
;host information
1. 0.0.127.in-addr.arpa. IN      PTR     localhost.
;end of file
```

รูปที่ 2.5 แฟ้มลูปแบ็ค

2.4.5 แฟ้มแคช (cache file)

แฟ้มแคชใช้เป็นแคชเก็บโดเมนหรือโฮสต์ที่มักใช้ประจำ เพื่อให้เนมดีนำไปใช้โดยไม่ต้องผ่านกระบวนการเรโซลูชัน ข้อมูลในแคชจึงมักเป็นข้อมูลซึ่งไม่เปลี่ยนแปลงบ่อยนัก แฟ้มแคชมักใช้บรรจุแอดเดรสของรูทเนมเซอร์ฟเวอร์

2.5 เมล์รีเลย์

วิธีรีเลย์เริ่มจากเอ็มทีเอต้นทางไปยังเอ็มทีเอระหว่างทางซึ่งจะเก็บเมลไว้ และนำส่งต่อตามจังหวะเวลาที่เหมาะสมจนกระทั่งเมลไปถึงปลายทาง ระบบเมล์ที่ใช้วิธีส่งต่อเป็นทอดๆ นี้เรียกว่า ระบบเก็บและส่งต่อ เมล์รีเลย์ประจำโดเมนหนึ่งๆ เรียกว่า ตัวแลกเปลี่ยนเมล์ ซึ่งกำหนดในดีเอ็นเอสด้วยเรคอร์ด MX

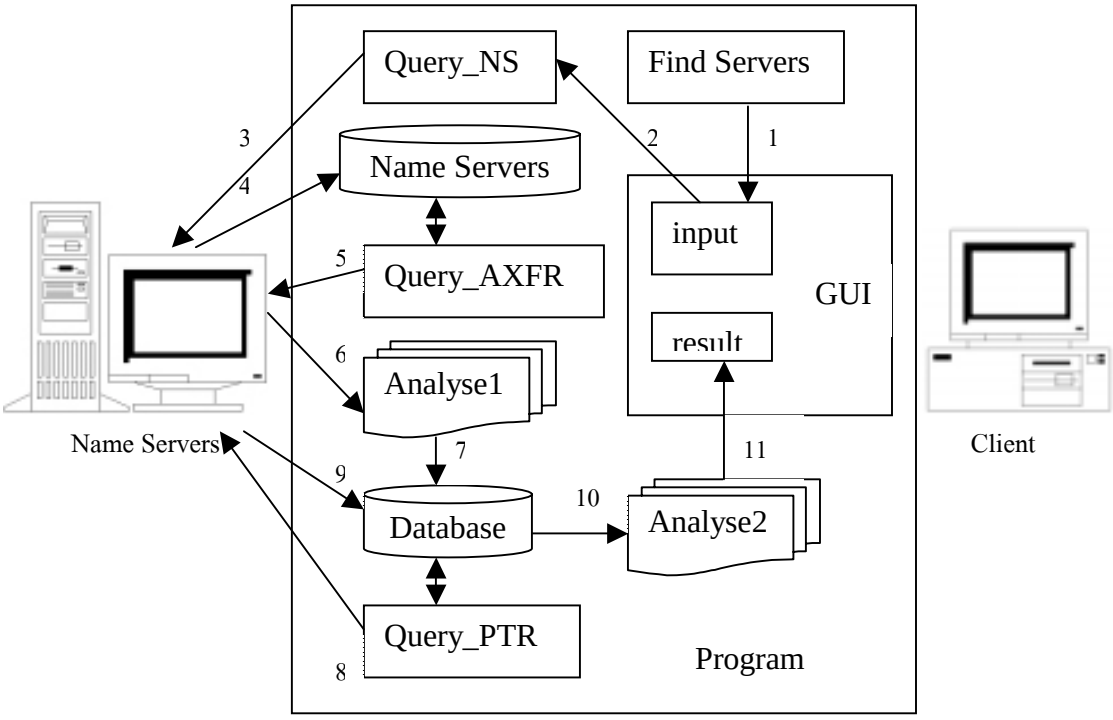
บทที่ 3
แนวคิดและการออกแบบ

3.1 เครื่องมือที่ใช้ในการพัฒนาซอฟแวร์

เครื่องคอมพิวเตอร์ 1 เครื่อง, โปรแกรม JBILDER 4 (ใช้สำหรับพัฒนาโปรแกรมหลัก), โปรแกรม Freeman Installer v3 32bit (ใช้เพื่อทำให้โปรแกรมสามารถติดตั้งแบบอัตโนมัติได้)

3.2 โครงสร้างของระบบ

โปรแกรมที่พัฒนานี้ให้ชื่อว่า "DNS Analyzer" เมื่อเริ่มการทำงานโปรแกรมจะหาเนมเซอร์ฟเวอร์ของเครื่องโดยหาว่าเครื่องที่รันโปรแกรมได้ตั้งเนมเซอร์ฟเวอร์ของเครื่องไว้ที่ไหน เมื่อรู้แล้วก็จะส่งคำร้องขอไปยังโซนที่ต้องการเพื่อขอว่าเนมเซอร์ฟเวอร์ของโซนมีเครื่องใดบ้าง จากนั้นจะส่งคำร้องขอโซนทรานสเฟอร์ไปยังเนมเซอร์ฟเวอร์ที่ได้รับมา (โปรแกรมจะวนส่งคำร้องขอโซนทรานสเฟอร์ไปยังเนมเซอร์ฟเวอร์ของโซนที่ต้องการทุกเนมเซอร์ฟเวอร์ เพื่อหาว่ามีเนมเซอร์ฟเวอร์ตัวไหนบ้างที่ให้ขอโซนทรานสเฟอร์ เมื่อพบก็จะรับข้อมูลมาและจะไม่หาเนมเซอร์ฟเวอร์ต่อไปอีก ถ้าไม่พบเลยจะแสดงข้อความว่าไม่มีเนมเซอร์ฟเวอร์ที่ให้ขอโซนทรานเฟอร์) เมื่อได้รับข้อมูลมาโปรแกรมก็จะเก็บข้อมูลไว้ พอรับหมดก็จะขอข้อมูลจากโดเมนผกผันเพื่อรับ PTR record ด้วย ดังรูปที่ 3.1 จากนั้นจะวิเคราะห์ข้อมูลที่ได้มาว่ามีข้อผิดพลาดใดบ้างดังที่ได้กล่าวมาแล้วในลักษณะของความผิดพลาดและวิธีตรวจสอบ



รูปที่ 3.1 โครงสร้างของระบบ DNS Analyzer

3.2.1 โมดูล Find Server

เมื่อรันโปรแกรม DNS Analyzer แล้ว โปรแกรมจะหาชื่อของเซอร์ฟเวอร์ที่ระบุไว้ในเครื่องที่รันโปรแกรมทันที (โปรแกรมต้องการเซอร์ฟเวอร์เพื่อจะใช้มันในการติดต่อขอเนมเซอร์ฟเวอร์ของโซนที่ต้องการวิเคราะห์) เนื่องจากแต่ละระบบปฏิบัติการจะเก็บชื่อของเซอร์ฟเวอร์ไว้ต่างกัน ดังนั้นโปรแกรมต้องหาว่าเป็นระบบปฏิบัติการใดก่อน จากนั้นก็จะหาไฟล์ที่เก็บข้อมูล โดยแต่ละระบบปฏิบัติการจะมีวิธีการหาชื่อของเซอร์ฟเวอร์ดังนี้

ระบบปฏิบัติการวินโดว์ 95 จะหาโดยใส่คำสั่ง "winipcfg /all /batch " + ชื่อไฟล์เอาท์พุต เนื่องจากวิธีการนี้จะนำเอาท์พุตไปใส่ไว้ในไฟล์ที่เรากำหนด ซึ่งช่วยให้การอ่านข้อมูลทำได้ง่ายขึ้น โดยอ่านไฟล์เอาท์พุตทีละบรรทัดเพื่อหาบรรทัดที่มีคำว่า "DNS Servers" ซึ่งหลังคำนี้ก็คือชื่อของเซอร์ฟเวอร์นั่นเอง เมื่อได้ข้อมูลแล้วก็ลบไฟล์ที่เราสร้างขึ้นมาทิ้งไป

ระบบปฏิบัติการวินโดว์ NT และ 2000 จะหาโดยใส่คำสั่ง "ipconfig /all" วินโดว์ NT และวินโดว์ 2000 ไม่สามารถนำข้อมูลใส่ลงไฟล์ได้จึงต้อง ใช้คำสั่งนี้แล้วอ่านข้อมูลทีละบรรทัดเพื่อหาบรรทัดที่มีคำว่า "DNS Servers" หลังจากคำนี้ก็คือชื่อของเซอร์ฟเวอร์ซึ่งก็คล้ายกับการอ่านข้อมูลในระบบปฏิบัติการวินโดว์ 95 แต่วิธีการนี้จะทำให้เกิดหน้าต่างดอสขึ้นมาที่หน้าจอเครื่องคอมพิวเตอร์ชั่วขณะหนึ่งซึ่งเป็นที่รำคาญแก่ผู้ใช้ แต่ตอนนี้ก็ยังหาวิธีการหาชื่อของเซอร์ฟเวอร์ที่ดีกว่านี้ไม่ได้

ระบบปฏิบัติการยูนิกซ์ จะหาจาก "/etc/resolv.conf" การหาชื่อของเซอร์ฟเวอร์ในระบบปฏิบัติการยูนิกซ์นั้นก็หาคล้ายกับระบบปฏิบัติการวินโดว์ แต่หาบรรทัดที่มีคำว่า “nameserver”

3.2.2 โมดูล input ในส่วนของ GUI

ทำหน้าที่รับโซนที่ผู้ใช้ต้องการวิเคราะห์

3.2.3 โมดูล Query_NS

ขั้นตอนถัดมาหลังจากรู้โซนที่ต้องการวิเคราะห์และเซอร์ฟเวอร์ที่ใช้เพื่อติดต่อได้แล้ว ก็จะติดต่อผ่านเซอร์ฟเวอร์ที่หามาได้จากโมดูล Find Server ให้มันหาชื่อเนมเซอร์ฟเวอร์ของโซนที่ต้องการวิเคราะห์ โดยส่งคำร้องขอชนิด NS ไปยังเซอร์ฟเวอร์ที่หามา

3.2.4 โมดูล Name Server

จะเก็บข้อมูลของเนมเซอร์เวอร์ที่ได้มาจากโมดูล Query_NS เพื่อให้โมดูล Query_AXFR ใช้ขอถ่ายโอนโซน

3.2.5 โมดูล Query_AXFR

เนื่องจากในบางโซนอาจมีเนมเซอร์ฟเวอร์บางตัวที่ไม่อนุญาตให้ถ่ายโอนโซน หรือไม่มีเนมเซอร์ฟเวอร์ตัวใดอนุญาตให้ถ่ายโอนโซนเลย ดังนั้นเมื่อได้ชื่อของเนมเซอร์ฟเวอร์ที่มีอำนาจหน้าที่ในโซนที่

ต้องการวิเคราะห์ก็ต้องขอถ่ายโอนโซน(ร้องขอข้อมูลชนิด AXFR)ไปทีละตัวจนกว่าจะมีเนมเซอร์ฟเวอร์ที่อนุญาตให้ขอถ่ายโอนโซนได้ หรือจนกว่าชื่อของเนมเซอร์ฟเวอร์ที่มีอำนาจหน้าที่ที่ได้รับมาหมด (ในกรณีนี้คือไม่มีเนมเซอร์ฟเวอร์ตัวใดในโซนที่ต้องการวิเคราะห์อนุญาตให้ขอถ่ายโอนโซน โปรแกรมจะแสดงข้อความว่าไม่มีเนมเซอร์ฟเวอร์ตัวใดตอบกลับมา)

3.2.6 โมดูล Query_PTR

หลังจากเก็บข้อมูลที่ได้รับมาทั้งหมดแล้ว ก็จะดูข้อมูลของ A record ใน Struct ทุกตัวที่เก็บอยู่ใน hash เพื่อนำแอดเดรสของ A record นั้นมาทำเป็นแอดเดรสผกผันก่อนแล้วจึงส่งคำร้องขอชนิด PTR ไปยังมาสเตอร์เนมเซอร์ฟเวอร์ของโซนนั้น โดยการจะดูว่าเนมเซอร์ฟเวอร์ไหนเป็นมาสเตอร์เนมเซอร์ฟเวอร์นั้นจะดูในฟิลด์เซอร์ฟเวอร์ที่อยู่ใน SOA record

3.2.7 โมดูล Analyse 1, Analyse 2

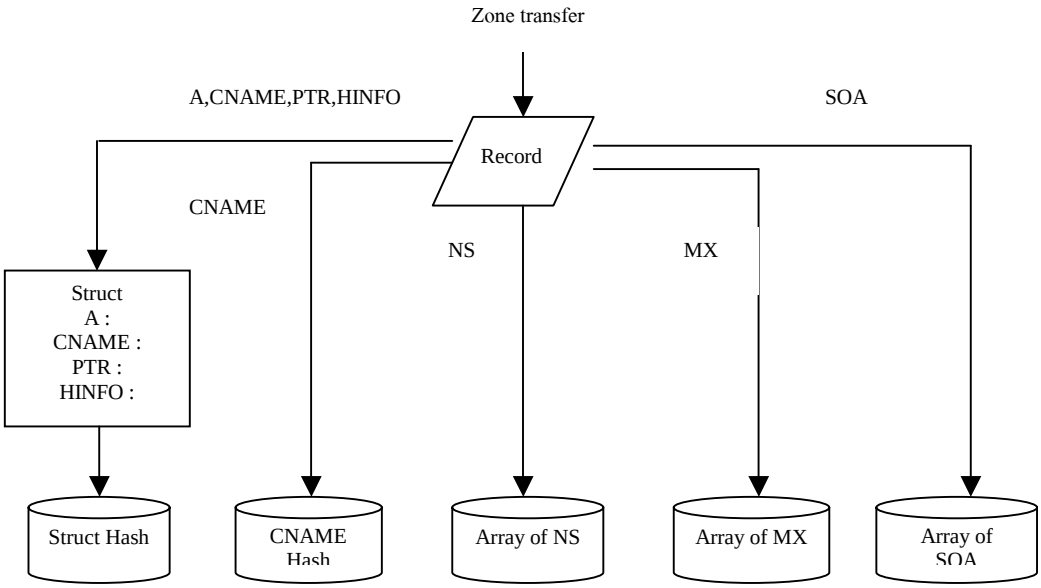
เมื่อมีเนมเซอร์ฟเวอร์อนุญาตให้โปรแกรมขอถ่ายโอนโซน เนมเซอร์ฟเวอร์ตัวนั้นจะส่งข้อมูลในโซนที่โปรแกรมขอถ่ายโอนโซนมาให้ ซึ่งโปรแกรมจะรับข้อมูลและเก็บข้อมูลทั้งหมดไว้ในบัฟเฟอร์ก่อน จากนั้นจะอ่านข้อมูลมาทีละเรคอร์ด จากนั้นจะดูว่าเป็นเรคอร์ดชนิดใด และจะแยกวิเคราะห์เรคอร์ดแต่ละชนิด (รายละเอียดทั้งหมดสามารถอ่านได้ในหัวข้อ 3.4 ลักษณะของความผิดพลาดและวิธีตรวจสอบ)

3.2.8 โมดูล Result ในส่วนของ GUI

ทำหน้าที่รับข้อมูลและแสดงผลการวิเคราะห์ (รายละเอียดของการใช้งานและการแสดงผลสามารถดูได้ใน ภาคผนวก ข. คู่มือการใช้งาน ในส่วนของ ตัวอย่างการใช้งาน)

3.3 การจัดเก็บข้อมูล

ส่วนนี้เป็นส่วนที่อธิบายการเก็บข้อมูลในโมดูล Analyse1 ซึ่งเป็นการรับข้อมูลจากการขอถ่ายโอนโซนแล้วแยกเก็บข้อมูลตามเรคอร์ดต่างๆ และมี Struct ซึ่งทำหน้าที่เก็บเรคอร์ด A,CNAME,PTR,HINFO ไว้ภายในตัวมัน ดังรูปที่ 3.2



รูปที่ 3.2 การจัดเก็บข้อมูลของ DNS Analyzer

เมื่อโปรแกรมรับข้อมูลของโซนที่ได้ส่งคำร้องขอไป โปรแกรมก็จะแยกข้อมูลที่ได้รับมาเป็นเรคอร์ดต่างดังรูป คือถ้าเป็น A,PTR,HINFO ก็จะเก็บข้อมูลลง Struct จากนั้นจึงนำไปเก็บไว้ใน Hash table โดยใช้ชื่อโฮสต์ของ A record เป็นคีย์ ส่วนถ้าเป็น CNAME ก็จะเก็บลงใน Struct และเก็บแยกต่างหากใน Hash table ของมันเองโดยใน Hash table นี้จะเก็บโดยใช้ชื่อของโฮสต์ที่มันอ้างถึงเป็นคีย์ สำหรับ NS, MX, SOA จะเก็บในอาร์เรย์ เพราะว่าเรคอร์ดเหล่านี้มีข้อมูลน้อยไม่เหมาะสมที่จะเก็บใน Hash table เพราะเวลาค้นหาจะใช้เวลานานกว่าในอาร์เรย์ และที่ใช้ Struct ในการเก็บ A,CNAME,PTR,HINFO record ก็เพื่อให้ง่ายต่อการตรวจสอบความผิดพลาด เช่นถ้ามีข้อมูลซ้ำเข้ามาก็สามารถรู้ได้ว่าข้อมูลผิดพลาด แล้วโปรแกรมก็จะเก็บข้อมูลที่ผิดพลาดนี้ไว้ในอาร์เรย์ของความผิดพลาดแบบนั้น เนื่องจากข้อมูลบางชนิดมีขนาดใหญ่จึงเลือกใช้ Hash table ในการเก็บข้อมูลก็เพราะง่ายต่อการค้นหาข้อมูลทั้งในการจัดเก็บและหาความผิดพลาด

3.4 ลักษณะของความผิดพลาดและวิธีตรวจสอบ

ก่อนที่จะอธิบายถึงลักษณะของความผิดพลาดขอให้เข้าใจว่าโปรแกรมนี้เก็บข้อมูลใส่ลงใน Hash table โดยจะมีการเก็บที่เป็นลักษณะพิเศษคือ จะเก็บ A record, CNAME record, PTR record และ HINFO record ไว้ใน Struct ก่อนจากนั้นจึงเก็บลงใน Hash table โดยใช้ชื่อโฮสต์ของ A record เป็นคีย์ในการเก็บและสามารถเรียกใช้ได้โดยใช้คีย์นี้ ลักษณะของความผิดพลาดและวิธีตรวจสอบที่จะนำมาใช้วิเคราะห์ในโครงงานนี้ จะมีลักษณะดังต่อไปนี้ [3]

3.4.1 A records

- **มี A record มากกว่า 1 เรคอร์ดที่ชื่อโฮสต์เหมือนกัน**

การตรวจนี้จะดูว่าในโซนนั้นมี 2 A record ที่มีชื่อโฮสต์เหมือนกันหรือไม่

<u>วิธีตรวจ</u> ข้อมูลที่รับมาจะเก็บใน Hash โดยใช้ชื่อโฮสต์เป็นคีย์ ดังนั้นเมื่อรับโฮสต์ที่ชื่อโฮสต์ซ้ำเข้ามาก็จะเก็บเป็นค่าความผิดพลาดเอาไว้

- **มี A record มากกว่า 1 เรคอร์ดที่แอดเดรสเหมือนกัน**

  การตรวจนี้จะดูว่าในโซนนั้นมี 2 A record ที่มีไอพีแอดเดรสเหมือนกันหรือไม่

  <u>วิธีตรวจ</u> ข้อมูลที่รับมาจะเก็บใน Hash โดยใช้แอดเดรสเป็นคีย์ ดังนั้นเมื่อรับโฮสต์ที่แอดเดรสซ้ำเข้ามาก็จะเก็บเป็นค่าความผิดพลาดเอาไว้

- **A record ซ้ำกัน**

  การตรวจนี้จะดูว่าในโซนมี A record ที่ซ้ำกันหรือไม่

  <u>วิธีตรวจ</u> เหมือนกับการตรวจ A record แบบแรก แต่จะตรวจต่อไปว่า เมื่อชื่อโฮสต์ซ้ำแล้วแอดเดรสซ้ำหรือไม่ ถ้าซ้ำก็เก็บความผิดพลาดว่าเป็น A record ซ้ำ

- **A record 1 เรคอร์ด มี PTR record มากกว่า 1 เรคอร์ด**

  การตรวจนี้จะดูในโดเมนผกผันว่ามี PTR record มากกว่า 1 เรคอร์ด ที่อ้างถึง A record 1 เรคอร์ดในโซนหรือไม่

  <u>วิธีตรวจ</u> เมื่อรับ PTR record เข้ามาก็จะใช้ชื่อโฮสต์หาใน Hash และตรวจดูว่าใน Struct มี PTR record อยู่หรือไม่ ถ้ามี PTR record อยู่แล้ว ก็จะเก็บค่าความผิดพลาดไว้

- **A record ไม่มี PTR record**

  การตรวจนี้จะดูว่าทุก A record มี PTR record ที่ตรงกันหรือไม่

  <u>วิธีตรวจ</u> การตรวจนี้จะทำเมื่อรับข้อมูลมาครบทั้งหมดและตรวจ Struct ทุกตัวที่อยู่ใน Hash ว่ามี PTR record อยู่หรือไม่ ถ้า Struct ใดไม่มีก็จะเก็บความผิดพลาดไว้

- **ชื่อโฮสต์ของ A record ไม่ตรงกับ PTR record**

  การตรวจนี้จะดูว่าชื่อโฮสต์ใน PTR record ตรงกับชื่อโฮสต์ของ A record ในโซนหรือไม่

  <u>วิธีตรวจ</u> จะนำชื่อของ PTR record ไปตรวจกับ Hash ของ A record ที่เรียงด้วยแอดเดรส ว่าชื่อตรงกันหรือไม่

- **ไม่มี A record ที่ชื่อเหมือนกับโซน**

  การตรวจนี้จะดูว่าในโซนนั้นมี A record ที่มีชื่อเหมือนกับโซนหรือไม่

  <u>วิธีตรวจ</u> เมื่อรับข้อมูลทั้งหมดมาแล้ว จะใช้ชื่อโซนหาว่ามี A record หรือไม่

3.4.2 CNAME records

- **CNAME chains**

  การตรวจนี้จะดูว่า CNAME record ในโซนชี้ไปยัง CNAME reord อื่นหรือไม่

  <u>วิธีตรวจ</u> ในขณะที่เก็บก็จะตรวจทุก CNAME record ที่มีขณะนั้น โดยดูว่าชื่อเล่นของเรคอร์ดนั้นตรงกับชื่อโฮสต์ของเรคอร์ดที่มีอยู่หรือไม่

- **มี CNAME record ที่ชื่อเหมือนกับชื่อโซน**
  การตรวจนี้จะดูว่าในโซนมี CNAME record ที่มีชื่อเหมือนกับโซนหรือไม่
  วิธีตรวจ จะหาใน Hash ของ CNAME ว่ามีชื่อเหมือนกับโซนหรือไม่
- **มี CNAME record ซ้ำกัน**
  การตรวจนี้จะดูว่าในโซนมี CNAME record ซ้ำกันหรือไม่
  วิธีตรวจ จะหาใน Hash ของ CNAME ว่าซ้ำกันหรือไม่
- **มี CNAME record มากกว่า 1 เรคอร์ดที่อ้างถึงโฮสต์เดียวกัน**
  การตรวจนี้จะดูว่าในโซนมี CNAME record มากกว่า 1 เรคอร์ด ที่อ้างถึงโฮสต์เดียวกันหรือไม่
  วิธีตรวจ จะรู้ได้ขณะที่รับข้อมูลเข้ามาเมื่อตรวจดู Struct ของ Hash ว่ามี CNAME record อยู่หรือไม่

3.4.3 MX records

- **ไม่มี MX records**
  การตรวจนี้จะดูว่าในโซนมี MX records หรือไม่
  วิธีตรวจ เมื่อรับข้อมูลหมดแล้ว ก็จะไปดูในอาร์เรย์ของ MX ว่ามี MX record หรือไม่
- **มี MX record 1 เรคอร์ด**
  การตรวจนี้จะดูว่าในโซนมี MX record เพียงเรคอร์ดเดียวหรือไม่
  วิธีตรวจ ต่อจากข้อข้างต้น ดูว่ามี MX record เพียงเรคอร์ดเดียวหรือไม่
- **มี MX record มากกว่า 1 เรคอร์ด ที่ค่าความลำดับความสำคัญเท่ากัน**
  การตรวจนี้จะดูว่าในโซนมี MX record มากกว่า 1 เรคอร์ด ที่มีค่าลำดับความสำคัญเท่ากันหรือไม่
  วิธีตรวจ จะดูในอาร์เรย์ของ MX record ที่เป็นชื่อของโซนว่ามีค่าลำดับความสำคัญเท่ากันหรือไม่
- **มี MX record ซ้ำกัน**
  การตรวจนี้จะดูว่าในโซนมี MX record ซ้ำกันหรือไม่
  วิธีตรวจ ตรวจในอาร์เรย์ของ MX record ว่ามีเรคอร์ดซ้ำกันหรือไม่
- **เมลล์เซอร์ฟเวอร์อ้างถึง A record มากกว่า 1 เรคอร์ด**
  การตรวจนี้จะดูว่าเมลล์เซอร์ฟเวอร์อ้างโฮสต์ที่มี A record มากกว่า 1 เรคอร์ด ซึ่งไอพีแอดเดรสต่างกัน
  วิธีตรวจ โปรแกรมจะเก็บ A record ที่มีชื่อโฮสต์ซ้ำเอาไว้ และการตรวจนี้จะไปหาในอาร์เรย์ตัวนี้
- **เมลล์เซอร์ฟเวอร์เดียวกันแต่มีค่าลำดับความสำคัญต่างกัน**

การตรวจนี้จะดูว่าในโซนได้ใส่ MX record มากกว่า 1 เรคอร์ด โดยมีค่าลำดับความสำคัญต่างกัน

วิธีตรวจ จะดูในอาร์เรย์ของ MX record ว่ามีเรคอร์ดไหนที่เป็นเมลล์เซอร์ฟเวอร์เดียวกันแต่มีค่าลำดับความสำคัญแตกต่างกันบ้าง

- **MX record ไม่มี A record**

  การตรวจนี้จะดูว่าในโซนมี MX record ที่ไม่มี A record รองรับหรือไม่

  วิธีตรวจ จะหาว่าโฮสต์ที่ MX record อ้างถึงว่ามี A record หรือไม่ โดยใช้ชื่อโฮสต์หาใน Hash ของ Struct

3.4.4 NS record

- **ไม่มี NS record**

  การตรวจนี้จะดูว่าในโซนมี NS record หรือไม่

  วิธีตรวจ เมื่อรับข้อมูลหมดแล้ว ก็จะไปดูในอาร์เรย์ของ NS ว่ามี NS record หรือไม่

- **มี NS record 1 เรคอร์ด**

  การตรวจนี้จะดูว่าโซนมี NS record เพียง 1 เรคอร์ดหรือไม่

  วิธีตรวจ ต่อจากข้อข้างต้น ดูว่ามี NS record เพียงเรคอร์ดเดียวหรือไม่

- **มี NS record ซ้ำกัน**

  การตรวจนี้จะดูว่ามี NS record ซ้ำกันหรือไม่

  วิธีตรวจ ตรวจในอาร์เรย์ของ NS record ว่ามีเรคอร์ดซ้ำกันหรือไม่

- **เนมเซอร์ฟเวอร์อ้างถึง A record มากกว่า 1 เรคอร์ด**

  การตรวจนี้จะดูว่าเนมเซอร์ฟเวอร์อ้างโฮสต์ที่มี A record มากกว่า 1 เรคอร์ด ซึ่งไอพีแอดเดรสต่างกัน

  วิธีตรวจ โปรแกรมจะเก็บ A record ที่มีชื่อโฮสต์ซ้ำเอาไว้ และการตรวจนี้จะไปหาในอาร์เรย์ตัวนี้

3.4.5 SOA record

- **ค่า Refresh ไม่เหมาะสม**

  การตรวจนี้จะดูว่าฟิลด์ Refresh ใน SOA record มีค่าเหมาะสมหรือไม่

  วิธีตรวจ ดูว่ามีค่าอยู่ระหว่าง 20 นาที – 12 ชั่วโมงหรือไม่

- **ค่า Retry ไม่เหมาะสม**

  การตรวจนี้จะดูว่าฟิลด์ Retry ใน SOA record มีค่าเหมาะสมหรือไม่

  วิธีตรวจ ดูว่ามีค่าอยู่ระหว่าง 20 นาที – 12 ชั่วโมงหรือไม่

- **ค่า Expire ไม่เหมาะสม**

การตรวจนี้จะดูว่าฟิลด์ Expire ใน SOA record มีค่าเหมาะสมหรือไม่

<u>วิธีตรวจ</u> ดูว่ามีค่าอยู่ระหว่าง 2 – 4 สัปดาห์หรือไม่

- **ค่า Minimum TTL ไม่เหมาะสม**

  การตรวจนี้จะดูว่าฟิลด์ Minimum TTL ใน SOA record มีค่าเหมาะสมหรือไม่

  <u>วิธีตรวจ</u> ดูว่ามีค่าอยู่ระหว่าง 1 – 2 วันหรือไม่

3.4.6 HINFO records

- **มี HINFO record มากกว่า 1 เรคอร์ดที่ชื่อเหมือนกัน**

  การตรวจนี้จะดูว่าในโซนนั้นมี HINFO record มากกว่า 1 เรคอร์ด ที่มีชื่อเหมือนกันหรือไม่

  <u>วิธีตรวจ</u> ขณะที่รับเข้ามาก็จะดูว่าใน Struct มี HINFO record แล้วหรือยัง และค่าข้อมูลต้องไม่ซ้ำกัน

- **มี HINFO records ซ้ำกัน**

  การตรวจนี้จะดูว่ามี HINFO records ซ้ำกันหรือไม่

  <u>วิธีตรวจ</u> จากการตรวจ HINFO records แรก ถ้าข้อมูลซ้ำกันด้วย

- **มี HINFO records เพียง 1 เรคอร์ด หรือมากกว่านั้น**

  การตรวจนี้จะดูว่าโซนมี HINFO record หรือไม่

  <u>วิธีตรวจ</u> จะดูในอาร์เรย์ของ HINFO record ว่ามีขนาดมากกว่า 1 หรือไม่

3.4.7 PTR records (โดเมนผกผัน)

- **มี PTR record ซ้ำกัน**

  การตรวจนี้จะดูว่าในโดเมนมี PTR record ที่ซ้ำกันหรือไม่

  <u>วิธีตรวจ</u> ขณะที่เก็บ PTR record เข้าไปใน Struct ก็จะดูว่า PTR record ซ้ำกันหรือไม่

- **มี PTR record มากกว่า 1 เรคอร์ดที่อ้างถึงโฮสต์เดียวกัน**

  การตรวจนี้จะดูในโดเมนผกผันว่ามี PTR record มากกว่า 1 เรคอร์ด ที่อ้างถึงโฮสต์เดียวกันหรือไม่

  <u>วิธีตรวจ</u> ขณะที่เก็บ PTR record เข้าไปใน Struct ก็จะดูว่ามีเรคอร์ดที่ชื่อโฮสต์ซ้ำหรือไม่

3.4.8 ข้อมูลการ Delegation

- **เวอร์ชันของโซนเหมือนกัน แต่ซีเรียลนัมเบอร์ต่างกัน**

  การตรวจนี้จะดูว่าเซอร์ฟเวอร์ที่มีอำนาจหน้าที่ของโซนมีข้อมูลเหมือนกันหรือไม่

  <u>วิธีตรวจ</u> ดูว่าเซอร์ฟเวอร์เครื่องที่มีซีเรียลนัมเบอร์ต่างกัน มีข้อมูลเหมือนกันหรือไม่

- **มี เซอร์ฟเวอร์ที่บรรจุข้อมูลเก่าของโซน**

การตรวจนี้จะดูว่าเซอร์ฟเวอร์ที่มีอำนาจหน้าที่บรรจุข้อมูลที่เป็นเวอร์ชันเก่าของโซนหรือไม่

วิธีตรวจ ตรวจดูซีเรียลนัมเบอร์ของเซอร์ฟเวอร์ว่ามีเซอร์ฟเวอร์ที่มีค่าน้อยกว่าเซอร์ฟเวอร์ตัวอื่นหรือไม่

- **ไพรมารี่เซอร์ฟเวอร์บรรจุข้อมูลเก่าของโซน**

  การตรวจนี้จะดูว่าเซอร์ฟเวอร์ที่เป็นไพรมารี่เซอร์ฟเวอร์บรรจุข้อมูลเก่าของโซนหรือไม่

  วิธีตรวจ สมมติให้ไพรมารี่เซอร์ฟเวอร์เป็นเซอร์ฟเวอร์ที่ได้มากับ SOA record และดูว่ามีซีเรียลนัมเบอร์น้อยกว่าเซอร์ฟเวอร์ตัวอื่นหรือไม่

- **มี เซอร์ฟเวอร์ที่มีอำนาจหน้าที่ มากกว่า 1 เซอร์ฟเวอร์ที่มีไอพีแอดเดรสเหมือนกัน**

  การตรวจนี้จะดูว่ามีเซอร์ฟเวอร์ที่มีอำนาจหน้าที่ของโซน มากกว่า 1 เซอร์ฟเวอร์ มีไอพีแอดเดรสเหมือนกันหรือไม่ ซึ่งแสดงว่าเป็นเครื่องเดียวกัน

  วิธีตรวจ ตรวจในอาร์เรย์ของ NS record หาเซอร์ฟเวอร์ของโซน จากนั้นหาว่ามี A record ซ้ำกันหรือไม่

- **ไม่มี glue data**

  การตรวจนี้จะดูในโซนมี glue data หรือไม่

  วิธีตรวจ ดูว่าเซอร์ฟเวอร์ที่อยู่โซนนั้นและเซอร์ฟเวอร์ที่ได้รับมอบอำนาจหน้าที่นั้นมี A record หรือไม่

3.4.9 ระบบความปลอดภัย

- **เมลล์เซอร์ฟเวอร์เป็นเมลล์รีเลย์**

  การตรวจนี้จะดูว่าเมลล์เซอร์ฟเวอร์ของโซนสามารถใช้เป็นเมลล์รีเลย์ได้หรือไม่

  วิธีตรวจ เมื่อจะตรวจว่ามันเป็นไปได้หรือเปล่าที่จะใช้เมลเซอร์ฟเวอร์เป็น third-party mail relay เราจะติดต่อไปยังพอร์ต SMTP ของเมลเซอร์ฟเวอร์และส่งคำสั่งต่อไปนี้

  HELO dnsanalyzer.com

  MAIL FROM: <xteex@hotmail.com>

  RCPT TO: <b40avpd@anreg.cpe.ku.ac.th>

  RSET

  QUIT

  ถ้าเมลเซอร์ฟเวอร์ไม่สร้างข้อความผิดพลาดขณะทำคำสั่งนี้ แสดงว่าเมลเซอร์ฟเวอร์สามารถใช้เป็นเมลรีเลย์ได้

3.4.10 อื่นๆ

- **มีเรคอร์ดชนิดที่เป็น Experimental**

  การตรวจนี้จะดูว่าในโซนมีเรคอร์ดชนิดที่เป็น Experimental หรือไม่

<u>วิธีตรวจ</u> ขณะที่รับมา ถ้าเป็นเรคอร์ดชนิดที่เป็น Experimental คือ MB, MG, MINFO, MR, NULL ก็จะเก็บค่าเอาไว้

- **มีเรคอร์ดชนิดที่เป็น Obsolete**
  การตรวจนี้จะดูว่าในโซนมีเรคอร์ดชนิดที่เป็น Obsolete หรือไม่
  <u>วิธีตรวจ</u> ขณะที่รับมา ถ้าเป็นเรคอร์ดชนิดที่เป็น Obsolete คือ MD, MF ก็จะเก็บเอาไว้
- **มีเรคอร์ดชนิดที่เป็น Unknown**
  การตรวจนี้จะดูว่าในโซนมีเรคอร์ดชนิดที่เป็น Unknown หรือไม่
  <u>วิธีตรวจ</u> ขณะที่รับมา ถ้าเป็นเรคอร์ดชนิดที่เป็น Unknown คือไม่รู้จักก็จะเก็บเอาไว้

3.5 ปัญหาและการแก้ไข

เนื่องจากในการหาเซอร์ฟเวอร์ในเครื่องที่ใช้ระบบปฏิบัติการ WINDOWS 2000 โปรแกรมนี้จำเป็นต้องใช้คำสั่งเป็น command line เพื่อหาเซอร์ฟเวอร์ ในขณะที่เปิดโปรแกรมก็จะเห็นหน้าต่างที่เป็น DOS-PROMPT ขึ้นมาชั่วขณะหนึ่ง ส่วนการแก้ไขนั้นขณะนี้กำลังหาวิธีการอยู่

บทที่ 4
บทสรุปและแนวทางในการพัฒนา

4.1 การประยุกต์ใช้โปรแกรม

โปรแกรมนี้สามารถนำไปประยุกต์ใช้ในการตรวจสอบความผิดพลาดในการใส่ข้อมูลในแฟ้มข้อมูลดีเอ็นเอส และแก้ไขข้อมูลที่ผิดพลาดต่อไป

4.2 แนวทางในการพัฒนา

ในขณะนี้โปรแกรมสามารถตรวจได้เฉพาะในโซนที่ต้องการจะตรวจสอบเท่านั้น แต่โปรแกรมนี้ยังสามารถพัฒนาให้ตรวจสอบข้อมูลได้ทุกโซนภายใต้โซนที่ต้องการตรวจสอบได้ ซึ่งจะต้องพัฒนาโปรแกรมต่อไป นอกจากนี้ยังสามารถเพิ่มความสามารถของโปรแกรมส่วนอื่นได้อีก เช่น ทำให้โปรแกรมสามารถเปลี่ยนจากภาษาไทย-อังกฤษ หรือ อังกฤษ-ไทย ได้, หรือทำให้โปรแกรมสามารถพิมพ์ผลลัพธ์ที่ได้จากการวิเคราะห์โซนออกทางเครื่องพิมพ์ได้

เอกสารอ้างอิง

[1] สุรศักดิ์ สงวนพงษ์ , *สถาปัตยกรรมและโปรโตคอลทีซีพี/ไอพี* , ซีเอ็ดยูเคชั่น , 2543.

[2] Albitz, P. , and Cricket, L., *DNS and BIND* , O'Reilly & Associates, Inc,1998.

[3] http://www.menandmice.com.

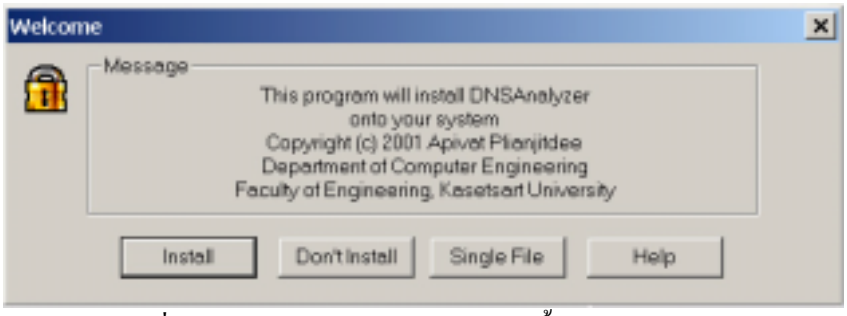
ภาคผนวก ก. การติดตั้งโปรแกรม

การติดตั้งบนระบบปฏิบัติการ Windows 9x/2000/NT

เนื่องจากโปรแกรมนี้สามารถทำงานได้ทั้งในหลายๆ ระบบ วิธีการติดตั้งจะแตกต่างกันไปตามชนิดของระบบปฏิบัติการ ในระบบปฏิบัติการ Windows 9x/2000/NT จะติดตั้งได้ด้วยโปรแกรม setup.exe ดังในรูปที่ ซึ่งมี Java Runtime Environment (JRE) ของ Microsoft อยู่ด้วย JRE ที่มากับโปรแกรมนี้เป็นของ Sun Microsystems, Inc. รุ่น 1.3

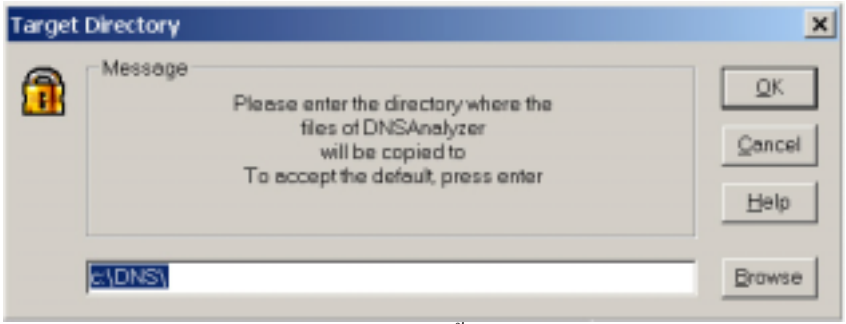
รูปที่ ก.1 ไอคอนของโปรแกรมติดตั้ง DNS Analyzer

หลังจากโปรแกรมติดตั้งเริ่มทำงาน จะมีหน้าต่างดังรูปที่ ถ้าต้องการติดตั้งต่อไปให้เลือก Install แต่ถ้าต้องการยกเลิกการติดตั้งสามารถเลือก Don't Install เพื่อออกจากโปรแกรมติดตั้งได้



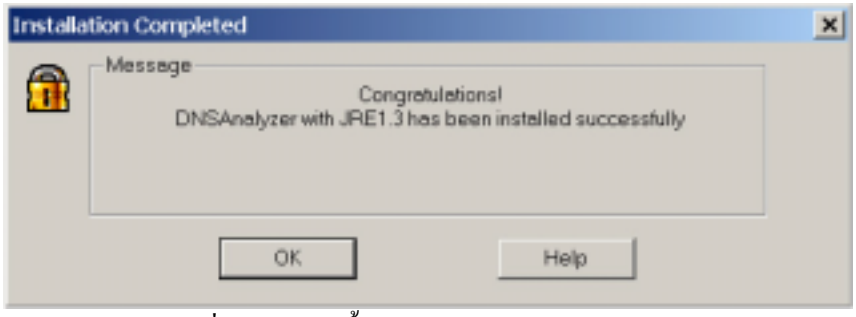
รูปที่ ก.2 หน้าต่างแรกของโปรแกรมติดตั้ง DNS Analyzer

จากนั้นจะมีหน้าต่างให้ผู้ใช้เลือกไดเรคทอรี่ที่จะติดตั้งโปรแกรมนี้ลงไป ดังในรูปที่ โปรแกรมและข้อมูลทั้งหมดของโปรแกรมจะถูกติดตั้งลงในไดเรคทอรี่ที่ระบุนี้ เลือก OK เพื่อดำเนินการติดตั้งต่อ จากนั้นโปรแกรมติดตั้งจะทำสำเนาข้อมูลเพื่อติดตั้งลงไปภายในที่ๆ ระบุไว้

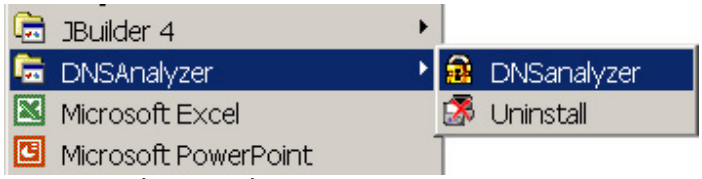


รูปที่ ก.3 เลือกไดเรคทอรี่ที่จะติดตั้งโปรแกรม DNS Analyzer

เมื่อการติดตั้งเสร็จสิ้นลง จะมีหน้าต่างดังรูปที่ เลือก OK เพื่อออกจากโปรแกรมติดตั้ง หลังจากนั้น ผู้ใช้สามารถเรียกโปรแกรมได้จากเมนู dnsanalyzer ดังรูปที่ (นอกจากจะเลือกไว้เป็นอย่างอื่น)



รูปที่ ก.4 การติดตั้ง DNS Analyzer เสร็จสมบูรณ์



รูปที่ ก.5 เมนูที่ใช้เรียกโปรแกรม DNS Analyzer

การติดตั้งบนระบบปฏิบัติการ Linux (glibc 2.1)

การติดตั้งบนระบบปฏิบัติการ Linux สามารถติดตั้งได้สองรูปแบบคือ ติดตั้งเป็นโปรแกรมของระบบที่ทุกคนในระบบสามารถเรียกใช้ได้ กับติดตั้งเฉพาะสำหรับผู้ใช้แต่ละคนไป การติดตั้งในแบบแรกนั้น ผู้ติดตั้งจำเป็นต้องมีสิทธิในการเขียนไฟล์ในไดเรคทอรี่ /usr/local (ในกรณีทั่วไปหมายความว่าผู้ติดตั้งต้องเป็น root ของเครื่องนั้นๆ) ส่วนในแบบหลัง ผู้ติดตั้งจะต้องมีโฮมไดเรคทอรี่ที่สามารถอ่านเขียนได้ตามปกติ

การติดตั้งทั้งสองแบบเริ่มด้วยการนำ CD-ROM ใส่ในเครื่องอ่าน จากนั้นเมาต์แผ่น CD-ROM เข้าเป็นส่วนหนึ่งในระบบ สมมุติว่าเครื่องอ่าน CD-ROM เป็นแบบ IDE ต่อเป็น Secondary Master อยู่ในระบบ (/dev/hdc) เราสามารถเมาต์แผ่น CD-ROM เข้าไปในไดเรคทอรี่ /mnt ได้ด้วยคำสั่ง (ในระบบทั่วไปผู้ใช้ไม่สามารถเมาต์แผ่น CD-ROM ด้วยตัวเองได้ ต้องให้ผู้ดูแลระบบเมาต์ให้)

```
# mount /dev/hdc /mnt –t iso9660
```

การติดตั้งเป็นโปรแกรมของระบบ

หลังจากเมาต์แผ่น CD-ROM เข้ามาในระบบเรียบร้อยแล้ว ให้เรียกโปรแกรมติดตั้งชื่อ install-system จากแผ่น CD-ROM ที่อยู่ในไดเรคทอรี่ Linux โดยเรียก

```
# /mnt/Linux/install-system
```

โปรแกรมติดตั้งจะสร้างไดเรคทอรี่ /usr/local/DNSAnalyzer และติดตั้งโปรแกรมลงในไดเรคทอรี่นั้น พร้อมกับสร้างลิงค์ไปหาโปรแกรมในไดเรคทอรี่ /usr/local/bin ในชื่อ DNSAnalyzer หลังจากการติดตั้งผู้ใช้สามารถเรียกโปรแกรมได้ในชื่อ DNSAnalyzer (ผู้ติดตั้งจะต้องแน่ใจว่าผู้ใช้มีไดเรคทอรี่ /usr/local/bin อยู่ภายใน PATH ด้วย)

การติดตั้งเฉพาะสำหรับผู้ใช้

การติดตั้งในแบบที่สองนี้ ให้ผู้ติดตั้งเรียกโปรแกรม install-user จากแผ่น CD-ROM ที่เมาต์อยู่จากในไดเรคทอรี่ Linux โดยเรียก

```
$ /mnt/Linux/install-user
```

โปรแกรมติดตั้งจะสร้างไดเรคทอรี่ DNSAnalyzer ผู้ใช้สามารถเรียกโปรแกรมได้โดยการเปลี่ยนไดเรคทอรี่ปัจจุบันไปเป็นไดเรคทอรี่ DNSAnalyzer ในโฮมของผู้ใช้เอง แล้วเรียก ./DNSAnalyzer

เปลี่ยนไดเรคทอรี่

```
$ cd ~/DNSAnalyzer
```

เรียกโปรแกรม

```
$ ./DNSAnalyzer
```

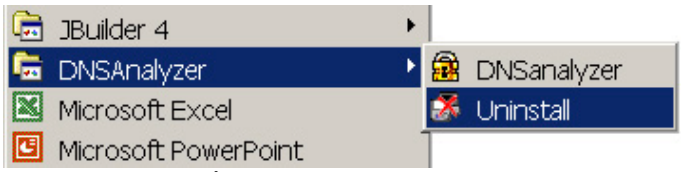
การติดตั้งในระบบปฏิบัติการอื่นๆ

ขั้นแรกจะต้องติดตั้ง JRE ลงบนระบบนั้นๆ ก่อน จากนั้นนำไฟล์ dnsanalyzer.jar ไปเก็บไว้ในไดเรคทอรี่ที่ต้องการ ตั้งให้ dnsanalyzer.jar อยู่ใน CLASSPATH แล้วเรียกโปรแกรมขึ้นมาด้วยคำสั่ง java dnsanalyzer.gui.Application1 ในบางระบบปฏิบัติการ เราสามารถเรียกโดยไม่ตั้ง CLASSPATH และไม่ต้องระบุชื่อคลาสเริ่มต้นได้ (เพราะสามารถหาได้เองในไฟล์ jar ได้) เช่น เราสามารถเรียก java –jar dnsanalyzer.jar ได้โดยตรง

การนำโปรแกรมออก

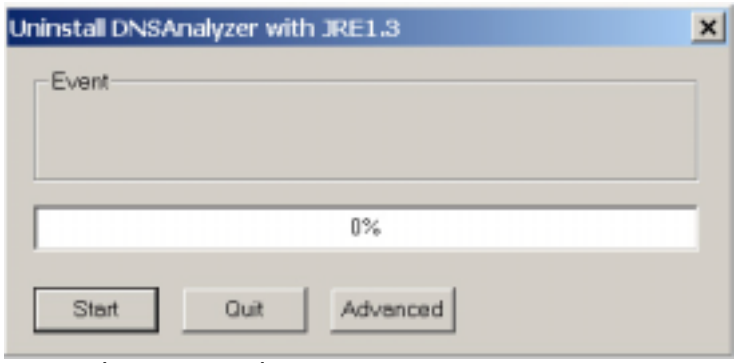
การนำโปรแกรมออกสำหรับระบบปฏิบัติการ Windows 9x/NT/2000

ถ้าโปรแกรม dnsanalyzer ยังทำงานอยู่ ให้ออกจากโปรแกรมก่อน จากนั้นเรียกโปรแกรม Uninstall จากเมนู Start/Programs/dnsanalyzer/Uninstall (หรือที่อื่นๆ ตามที่ได้เลือกไว้ในขั้นตอนการติดตั้ง) ดังในรูปที่

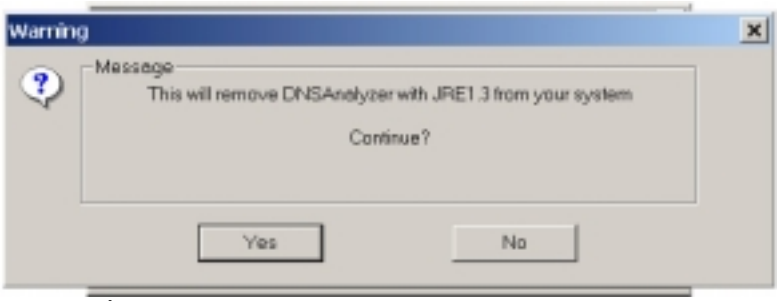


รูปที่ ก.6 เรียก Uninstall จากเมนู

จากนั้นจะมีหน้าต่างดังรูปที่ เลือก Start เพื่อเริ่มต้นการนำโปรแกรมออก หรือกด Quit เพื่อยกเลิกการนำโปรแกรมออก เมื่อกด Start แล้วจะมีหน้าต่างถามเพื่อยืนยันว่าจะนำโปรแกรมออกจริงหรือไม่ดังในรูปที่ กดเลือก Yes เพื่อดำเนินการต่อไป

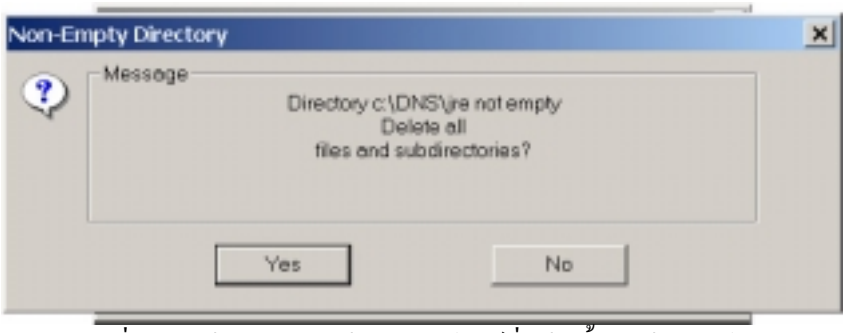


รูปที่ ก.7 หน้าต่างเริ่มการนำโปรแกรม DNS Analyzer ออก

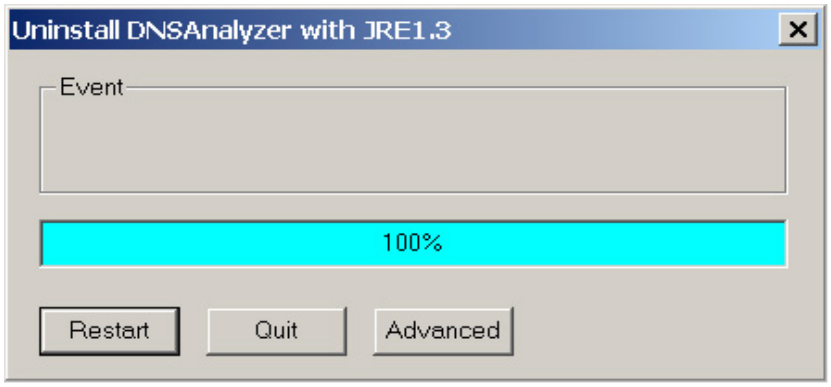


รูปที่ ก.8 หน้าต่างยืนยันการนำโปรแกรม DNS Analyzer ออก

จากนั้นโปรแกรมจะเริ่มกระบวนการนำโปรแกรมออกจากระบบ ในกรณีที่ผู้ใช้สร้างไฟล์ขึ้นมาเก็บไว้ จะมีหน้าต่างถามเพื่อยืนยันว่าจะลบไฟล์เหล่านั้นด้วยหรือไม่ ดังในรูปที่ เมื่อการนำโปรแกรมออกจากระบบเสร็จเรียบร้อยแล้วจะมีหน้าต่างดังรูปที่ กด Quit เพื่อสิ้นสุดการนำโปรแกรมออก



รูปที่ ก.9 หน้าต่างถามว่าต้องการลบไฟล์ที่สร้างขึ้นมาด้วยหรือไม่



รูปที่ ก.10 หน้าต่างแสดงว่าโปรแกรม DNS Analyzer ได้ถูกนำออกเสร็จสมบูรณ์

การนำโปรแกรมออกสำหรับระบบปฏิบัติการ Linux

ในกรณีติดตั้งเป็นโปรแกรมของระบบ

เรียกโปรแกรม uninstall จากในไดเรคทอรี่ /usr/local/DNSAnalyzer โปรแกรมจะลบตัวเองออกจากระบบโดยอัตโนมัติ (แต่ข้อมูลที่ผู้ใช้แต่ละคนสร้างเก็บไว้จะยังคงอยู่ ถ้าต้องการลบออกจากระบบด้วย ผู้ใช้ต้องลบไดเรคทอรี่ DNSAnalyzer ออกจากโฮมไดเรคทอรี่เอง)

```
# /usr/local/DNSAnalyzer/uninstall
```

ในกรณีติดตั้งเฉพาะสำหรับผู้ใช้

เรียกโปรแกรม uninstall จากไดเรคทอรี่ DNSAnalyzer ในโฮมไดเรคทอรี่ของผู้ใช้คนนั้นๆ โปรแกรมจะลบตัวเองออกจากโฮมไดเรคทอรี่โดยอัตโนมัติ รวมถึงข้อมูลที่ผู้ใช้คนนั้นๆ สร้างเก็บไว้ในไดเรคทอรี่นั้นด้วย

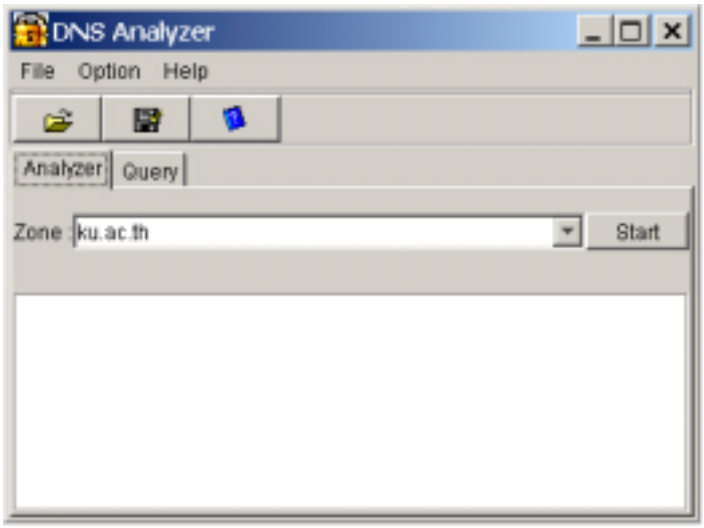
```
$ ~/DNSAnalyzer/uninstall
```

การนำโปรแกรมออกสำหรับระบบปฏิบัติการอื่นๆ

ในระบบปฏิบัติการอื่นๆ ผู้ใช้ต้องลบไฟล์ต่างๆ ที่สร้างขึ้น ไฟล์ที่ทำสำเนาไว้ และ ไดเรคทอรี่ที่สร้างขึ้นออกด้วยตัวเอง

ภาคผนวก ข. คู่มือการใช้งาน

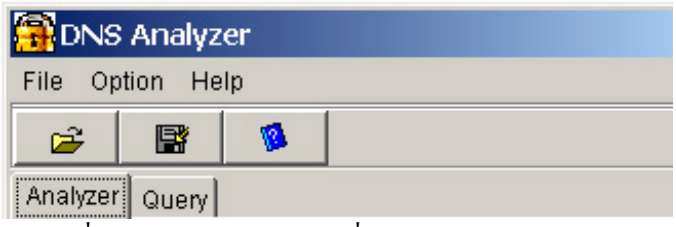
เมื่อโปรแกรมเริ่มทำงาน จะมีลักษณะของส่วนติดต่อกับผู้ใช้ดังรูปที่ ส่วนบนสุดของหน้าต่างจะเป็น Title Bar ซึ่งลักษณะและหน้าที่ของปุ่มและเมนูต่างๆ จะต่างกันไปตามระบบปฏิบัติการหรือตัวจัดการวินโดว์ที่ใช้ ซึ่งจะศึกษาได้จากคู่มือของระบบนั้นๆ (ในรูปเป็นการใช้งานบน Window 2000) ถัดลงมาจะเป็นแถบเมนูและแถบเครื่องมือตามลำดับ คำสั่งต่างๆ ในแถบเมนูและแถบเครื่องมือจะคล้ายคลึงกันซึ่งผู้ใช้สามารถเลือกใช้ได้ตามความสะดวก



รูปที่ ข.1 ส่วนติดต่อกับผู้ใช้เมื่อโปรแกรม DNS Analyzer เริ่มทำงาน

แถบเมนูและแถบเครื่องมือ

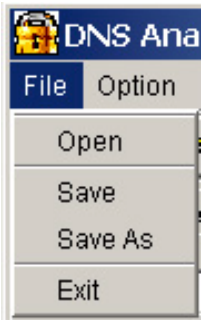
แถบเมนูและแถบเครื่องมือ (รูปที่) เป็นคำสั่งที่เกี่ยวกับไฟล์และการจำลองการทำงาน โดยคำสั่งแบ่งออกเป็นชุดๆ ตามแถบเมนู ดังนี้



รูปที่ ข.2 แถบเมนูและแถบเครื่องมือของ DNS Analyzer

แถบเมนูไฟล์ (File)

แถบเมนูไฟล์ (รูปที่) ประกอบไปด้วยคำสั่งต่างๆ ดังนี้



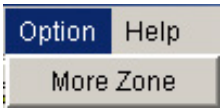
รูปที่ ข.3 แถบเมนูไฟล์ของ DNS Analyzer

ตารางที่ ข.1 แถบเมนูไฟล์

สัญลักษณ์	ชื่อ	คำอธิบาย
	Open	เปิดไฟล์เก่าขึ้นมาวิเคราะห์
	Save	บันทึกงานที่ทำลงไฟล์
	Save As	บันทึกงานที่ทำลงไฟล์ในชื่อใหม่
	Exit	ออกโปรแกรม

แถบเมนูออฟชัน (obtion)

แถบเมนูการจำลอง (รูปที่) ประกอบไปด้วยคำสั่งต่างๆ ดังนี้



รูปที่ ข.4 แถบเมนูออฟชันของ DNS Analyzer

ตารางที่ ข.2 แถบเมนูการจำลอง

สัญลักษณ์	ชื่อ	คำอธิบาย
	More Zone	เมื่อกดปุ่มจะมีชื่อของมหาวิทยาลัยต่างๆ ในรูปของไดอะล็อก ขึ้นมา เมื่อคลิก 2 ครั้งบนชื่อ ชื่อของโซนนั้นจะไปปรากฏอยู่ในอินพุทบล็อก

ไดอะล็อกเมื่อกดปุ่ม More Zone

รูป ข.5 ไดอะล็อก More Zone

แถบเมนูช่วยเหลือ (Help)

แถบเมนูช่วยเหลือ (รูปที่) ประกอบไปด้วยคำสั่งต่างๆ ดังนี้

รูปที่ ข.6 แถบเมนูช่วยเหลือของ DNS Analyzer

ตารางที่ ข.3 แถบเมนูช่วยเหลือ

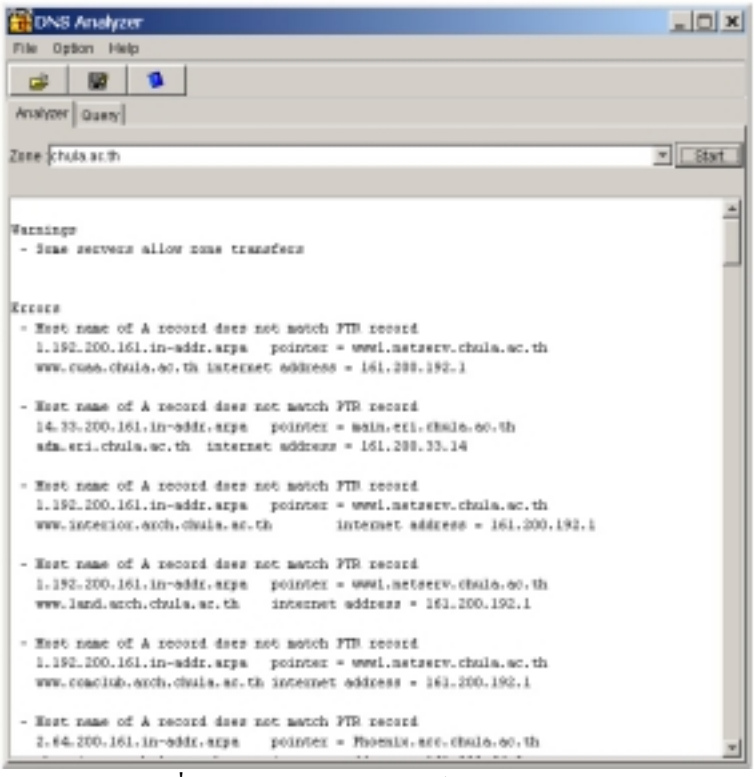
สัญลักษณ์	ชื่อ	คำอธิบาย
	About	แสดงข้อมูลเกี่ยวกับโปรแกรม

ตัวอย่างการใช้งาน

วิเคราะห์โซน chula.ac.th ของจุฬาลงกรณ์มหาวิทยาลัย

1. ดูว่าเป็นแท็บ Analyzer หรือไม่ ถ้าไม่ให้กดที่แท็บของ Analyzer ก่อน
2. กดบนปุ่ม 

3. รอสักครู่จนมีข้อมูลปรากฏขึ้น แสดงว่าโปรแกรมวิเคราะห์เสร็จแล้ว



รูปที่ ข.7 ผลของการวิเคราะห์โซน chula.ac.th

ผลการวิเคราะห์

หลังจากโปรแกรมวิเคราะห์โซนแล้ว โปรแกรมก็จะแสดงผลการวิเคราะห์ และเพื่อให้ง่ายต่อการใช้งานของผู้ใช้ก็จะให้โปรแกรมแสดงผลผ่านทาง GUI (Graphic User Interface) โดยผลการวิเคราะห์จะแบ่งเป็น

- **คำเตือน (warnings)** การผิดพลาดแบบนี้ไม่สำคัญเท่าใดนัก จะแก้ไขหรือไม่ก็ได้ เช่น Some servers allow zone transfers การผิดพลาดแบบนี้เป็นการเตือนว่ามีเซอร์ฟเวอร์บางตัวที่มีอำนาจหน้าที่ในโซนอนุญาตให้เซอร์ฟเวอร์ของโซนอื่นขอถ่ายโอนโซนได้ เพราะถ้าโซนนั้นต้องการรักษาความปลอดภัยของระบบก็จะต้องไม่ให้เซอร์ฟเวอร์ที่อยู่ในโดเมนอื่นขอถ่ายโอนโซนได้
- **ความผิดพลาด (errors)** ความผิดพลาดเหล่านี้จำเป็นต้องแก้ไขเพราะเป็นการผิดพลาดแบบรุนแรง เช่น Host name of A record does not match PTR record การผิดพลาดแบบนี้บอกว่าชื่อโฮสต์ของ A record ไม่ตรงกับชื่อโฮสต์ของ PTR เรคอร์ด จำเป็นต้องแก้ไขให้ชื่อโฮสต์ของ A record ให้ตรงกับชื่อโฮสต์ของ PTR record

โดยข้อความที่อยู่ใต้ข้อความคำเตือนหรือความผิดพลาดนั้นคือการผิดพลาดต่าง ๆ ที่เป็นประเภทของการผิดพลาดที่มันอยู่ใต้ข้อความนั้น บรรทัดที่มีเครื่องหมาย (-) เป็นการบอกว่าเป็นความผิดพลาดแบบใด บรรทัดถัดไปจะแสดงเรคอร์ดที่ผิดพลาดของชนิดความผิดพลาดในบรรทัดที่มีเครื่องหมาย (-) เช่น

- Host name of A record does not match PTR record

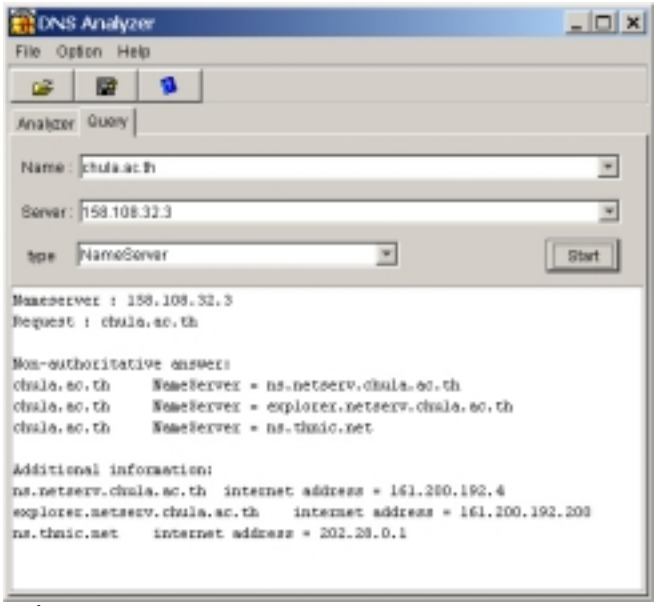
  1.192.200.161.in-addr.arpa pointer = www1.netserv.chula.ac.th

  www.cuaa.chula.ac.th internet address = 161.200.192.1

รูปที่ ข.8 ตัวอย่างการวิเคราะห์ข้อมูลที่ผิดพลาดของโซน

Query โซน chula.ac.th

1. ดูว่าเป็นแท็บ Query หรือไม่ ถ้าไม่ให้กดที่แท็บของ Query ก่อน
2. กดบนปุ่ม Start
3. รอสักครู่จนมีข้อมูลปรากฏขึ้น แสดงว่าโปรแกรม query เสร็จแล้ว



รูปที่ ข.9 ผลของการ query เนมเซอร์ฟเวอร์ของโซน chula.ac.th