การผนวกเน็ตบีนส์ปลั๊กอินเพื่อศึกษาการใช้ดีเทรส

นายวรวุฒิ ไพเราะ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2554
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR)
are the thesis authors' files submitted through the Graduate School.

INTEGRATION OF NETBEANS PLUGIN FOR DTRACE USAGE STUDY

Mr. Woravut Pairoh

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2011

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การผนวกเน็ตบีนส์ปลั๊กอินเพื่อศึกษาการใช้ดีเทรส
โดย	นายวรวุฒิ ไพเราะ
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	อาจารย์ ดร.ยรรยง เต็งอำนวย

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

…….. คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

…… ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา)

…….. อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(อาจารย์ ดร. ยรรยง เต็งอำนวย)

…… กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.ชวลิต ศรีสถาพรพัฒน์)

วรวุฒิ ไพเราะ : การผนวกเน็ตบีนส์ปลั๊กอินเพื่อศึกษาการใช้ดีเทรส. (INTEGRATION OF NETBEANS PLUGIN FOR DTRACE USAGE STUDY) อ. ที่ปรึกษาวิทยานิพนธ์หลัก : อ.ดร.ยรรยง เต็งอำนวย, 46 หน้า.

สมรรถนะและประสิทธิภาพมีความสำคัญมากในการทำงานของระบบขนาดใหญ่ การวัดประสิทธิภาพและการตรวจหาปัญหาของระบบจึงเป็นสิ่งสำคัญที่ผู้ดูแลระบบต้องทราบ ดีเทรสเป็นเครื่องมือที่มีประสิทธิภาพเพื่อการนี้ แต่เนื่องด้วยการใช้งานที่ค่อนข้างยากต่อผู้เริ่มศึกษา ในงานวิจัยนี้จึงนำเสนอการผนวกชุดเครื่องมือดีเทรส เพื่อช่วยให้ผู้ดูแลระบบเข้าใจการใช้งาน อีกทั้งยังใช้ในการวิเคราะห์สมรรถนะของระบบที่ใช้งานจริงได้ด้วย

วิทยานิพนธ์นี้ได้ศึกษาการใช้งาน การทำงานของดีเทรสบนระบบปฏิบัติการโซลาริส และทำการพัฒนาปลั๊กอินเพื่อศึกษาการใช้ดีเทรสโดยนำส่วนประกอบหลายด้านที่นำมาผนวกเข้าเป็นชุดเครื่องมือดีเทรสโดยมี 3 องค์ประกอบคือ (1) ไอดีอี คือเน็ตบีนส์ สามารถติดตั้งได้หลายระบบปฏิบัติการอีกทั้งยังใช้ในการพัฒนาซอฟต์แวร์และรองรับการติดตั้งปลั๊กอินที่หลากหลาย (2) ซอฟต์แวร์ คือส่วนที่เป็นดีเทรส โดยทำการเลือกดีเทรสที่ได้ทำการพัฒนาเป็นจียูไอที่เป็นเน็ตบีนส์ปลั๊กอินเพราะมีชุดของดีสคริปต์และปลั๊กอินถูกพัฒนาให้ติดตั้งเฉพาะบนเน็ตบีนส์ (3) เอกเซอร์ไซเซอร์ คือโปรแกรมที่ทำหน้าที่รบกวนหรือทดสอบระบบ โดยจะให้ระบบทำงานหนักขึ้นเพื่อให้ผู้ใช้เห็นได้ชัดว่าส่วนใดกำลังมีปัญหาด้านสมรรถนะ ซึ่งทั้งหมดรวมกันพัฒนาเป็นโปรแกรมชื่อ ดีทีเอ็กซ์ (DTEx - DTrace Exerciser) ผลการทดลองใช้งานพบว่ามีการใช้งานง่าย และผู้ใช้สามารถศึกษาได้ด้วยตนเองเป็นสื่อการเรียนการสอนและนำไปใช้ตรวจติดตามสมรรถนะของระบบงานจริงได้

ภาควิชา .....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต .....

สาขาวิชา .....วิทยาศาสตร์คอมพิวเตอร์..... ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา .....2554.....

5271445921 : MAJOR COMPUTER SCIENCE

KEYWORDS: DTRACE / DTRACE GUI / DTRACE TOOLKIT/NETBEANS PLUGIN

WORAVUT PAIROH : INTEGRATION OF NETBEANS PLUGIN FOR DTRACE USAGE STUDY. ADVISOR : YUNYONG TENG-AMNUAY,Ph.D., 46 pp.

Performance and efficiency of large production system are very important. Performance measurement and tracing are important to administrator. DTrace is the powerful tool to measure performance and assist in debugging complex application but the usage is difficult for beginners. In this research, we present an integration of DTrace Toolkit suite to help train administrator and can also be used in tracking the performance of production system.

This work is based on Dtrace's solaris operating system. There are various components to be integrated into the dtrace toolkit which can be divided into three main categories. The first category is netbeans which can install multiple operating systems and various plug-ins. Secondly, is the Dtrace GUI that is developed on netbeans plug-in since it has a collection of dscript and plug-ins that have been developed to be installed only on the netbeans. The third is an interface program that will be used in system test. It will make the system work harder so that sectors that lack efficiency stands out. All of this will be developed under the program named DTEx (DTrace Exerciser). The trial result has shown ease to use, where users can learn on their own, and ultimately instruct and to monitor the performance of the real system.

Department: .....Computer Engineering..... Student's Signature.....

Field of Study: .....Computer Science..... Advisor's Signature.....

Academic Year: .....2011.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ กราบขอขอบพระคุณอาจารย์ ดร.ยรรยง เต็งอำนวย ที่ให้คำแนะนำ ข้อคิดเห็น แนวทางคำปรึกษาต่างๆ ตลอดจนความช่วยเหลือต่างๆ มาโดยตลอดระยะเวลาการศึกษาและการวิจัย ทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี ขอขอบพระคุณอาจารย์เป็นอย่างมากทั้งให้คำปรึกษาดีๆ ทั้งเรื่องเรียนและเรื่องทั่วไป ประสบการณ์ต่างๆ ที่นำไปใช้ในการดำเนินชีวิตได้

ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา และผู้ช่วยศาสตราจารย์ ดร.ชวลิต ศรีสถาพรพัฒน์ ที่ให้ข้อชี้แนะ คำแนะนำ ข้อคิดเห็นต่างๆ เพื่อนำไปใช้ในการปรับปรุงวิทยานิพนธ์ฉบับนี้

ขอขอบพระคุณคณาจารย์ทุกท่านที่ดำเนินการสอน ให้คำแนะนำ ให้ความรู้ต่างๆ ในแต่ละวิชา ความรู้ในแต่ละสาขาต่างๆ มากมาย

ขอบคุณเพื่อนๆ พี่ๆ น้องๆ ทั้งในชั้นเรียน ในรุ่นเดียวกันและในห้องปฏิบัติการวิศวกรรมระบบสารสนเทศ ที่คอยเอื้อเฟื้อในหลายๆ อย่าง ให้คำแนะนำ ให้กำลังใจกัน ช่วยเหลือกันในทุกๆ ครั้ง

ขอกราบขอบพระคุณคุณพ่อ คุณแม่ คุณปู่ คุณย่า และญาติทุกคน ที่ให้คำแนะนำ ส่งเสริม สนับสนุนให้ได้มีโอกาสได้ศึกษาเรียนรู้ และเป็นกำลังใจที่ดีเรื่อยเสมอมา

สารบัญ

หน้า

บทคัดย่อภาษาไทย…… ง
บทคัดย่อภาษาอังกฤษ…… จ
กิตติกรรมประกาศ…… ฉ
สารบัญ…… ช
สารบัญตาราง…… ฌ
สารบัญภาพ…… ญ

บทที่ 1 บทนำ…… 1
1.1 ความเป็นมาและความสำคัญของปัญหา…… 1
1.2 วัตถุประสงค์ของการวิจัย…… 2
1.3 ขอบเขตของการวิจัย……2
1.4 ประโยชน์ที่คาดว่าจะได้รับ……2
1.5 วิธีการดำเนินการวิจัย…… 2
1.6 ผลงานตีพิมพ์……3
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง……4
2.1 แนวคิดและทฤษฎี……4
2.1.1 ดีเทรส……5
2.1.2 โปรแกรมภาษา “ดี”……8
2.1.3 ชุดเครื่องมือดีเทรส……9
2.1.4 เน็ตบีนส์……10
2.1.5 ปลั๊กอินเน็ตบีนส์ดีเทรสจียูไอ……10
2.1.6 ไซม์……10
2.1.7 เอกเซอร์ไซเซอร์……10
2.2 งานวิจัยที่เกี่ยวข้อง……11
2.2.1 งานวิจัยที่เกี่ยวกับดีเทรส……11
2.2.2 งานวิจัยที่เกี่ยวกับเครื่องมือที่เกี่ยวข้อง……11
บทที่ 3 แนวทางการวิจัยการออกแบบ……12
3.1 แนวทางการวิจัย……12

หน้า

3.2 การออกแบบการผนวกชุดเครื่องมือดีเทรสตามแนวทางการวิจัยที่ได้วางไว้.....12
3.3 ชุดเครื่องมือดีทีเอ๊กซ์.....15
3.3.1 โมดูลเน็ตบีนส์จียูไอ.....16
3.3.2 โมดูลดีสคริปต์.....16
3.3.2.1 ดีเทรส.....16
3.3.2.2 ไซม์.....18
3.4 โมดูลปลั๊กอิน.....20
บทที่ 4 การทดสอบและผลการทดลอง.....22
4.1 เครื่องมือ โปรแกรมและการติดตั้งโปรแกรมในการทดลอง.....22
4.2 การพัฒนาชุดโปรแกรมดีทีเอ๊กซ์.....23
4.3 เปรียบเทียบการใช้งานด้วยมือกับชุดเครื่องมือ.....23
4.3.1 ใช้งานดีเทรสด้วยมือ.....23
4.3.2 ใช้งานดีเทรสด้วยชุดเครื่องมือ.....24
4.4 การทดลอง.....26
4.4.1 หน่วยประมวลผล.....26
4.4.2 หน่วยความจำหลัก.....31
4.4.3 พีเอชพี.....34
4.4.4 รูบี้.....36
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....41
5.1 สรุปผลการวิจัย.....41
5.2 ปัญหาและข้อจำกัด.....42
5.3 ข้อเสนอแนะ.....42
รายการอ้างอิง.....43
ประวัติผู้เขียนวิทยานิพนธ์.....46

สารบัญตาราง

หน้า

ตารางที่ 3.1 หมวดหมู่ของดีสคริปต์ในเครื่องมือดีเทรส.....17
ตารางที่ 3.2 หมวดหมู่ของดีสคริปต์ในใชม์.....19
ตารางที่ 4.1 เปรียบเทียบการใช้งานด้วยมือกับชุดเครื่องมือ.....26

สารบัญภาพ

หน้า

ภาพที่ 2.1 แสดงส่วนประกอบต่างๆ ของดีทีเอ็กซ์.....4
ภาพที่ 2.2 ตัวอย่างการใช้ดีเทรส.....5
ภาพที่ 2.3 สถาปัตยกรรมดีเทรส.....6
ภาพที่ 2.4 ตัวอย่างลำดับชั้นโครงสร้างของโปรแกรม.....7
ภาพที่ 2.5 การรวบรวม (Aggregate) ในลำดับชั้นของการเรียกโปรแกรมย่อย.....8
ภาพที่ 2.6 โครงสร้างของภาษา “ดี”.....9
ภาพที่ 2.7 ตัวอย่างสคริปต์เขียนโดยภาษา “ดี” 14
ภาพที่ 3.1 แสดงแผนผังการใช้งานโดยรวมของผู้ใช้งานชุดโปรแกรม.....14
ภาพที่ 3.2 องค์ประกอบของชุดโปรแกรมเอกเซอร์ไซเซอร์.....14
ภาพที่ 3.3 ขั้นตอนการใช้งานชุดเครื่องมือดีทีเอ็กซ์.....15
ภาพที่ 3.4 จียูไอของดีสคริปต์ปลั๊กอิน.....16
ภาพที่ 3.5 จียูไอของไซม์.....18
ภาพที่ 3.6 แสดงจียูไอของเอกเซอร์ไซเซอร์.....20
ภาพที่ 3.7 แสดงจียูไอของชุดเครื่องมือดีทีเอ็กซ์.....21
ภาพที่ 4.1 ดีสคริปต์ sysprobes.d 23
ภาพที่ 4.2 ผลลัพธ์ของการใช้คำสั่งดีสคริปต์.....24
ภาพที่ 4.3 จียูไอของไซม์.....24
ภาพที่ 4.4 แสดงผลลัพธ์เมื่อใช้ไซม์.....25
ภาพที่ 4.5 ผลการใช้ดีเทรสแสดงสมรรถนะของหน่วยประมวลผล.....26
ภาพที่ 4.6 โปรแกรมของเอกเซอร์ไซเซอร์ของหน่วยประมวลผล.....27
ภาพที่ 4.7 โปรแกรมที่เอกเซอร์ไซเซอร์ทำให้หน่วยประมวลผลทำการประมวลผล.....27
ภาพที่ 4.8 แสดงเอกเซอร์ไซเซอร์ขณะใช้งาน.....28
ภาพที่ 4.9 แสดงการเลือกดีสคริปต์และไซม์ของหน่วยประมวลผล.....29
ภาพที่ 4.10 ผลการใช้ดีเทรสแสดงสมรรถนะของหน่วยประมวลผล.....30
ภาพที่ 4.11 แสดงผลการใช้ไซม์.....30
ภาพที่ 4.12 ผลจากโปรแกรม System Monitor.....31
ภาพที่ 4.13 แสดงจียูไอของเอกเซอร์ไซเซอร์ของหน่วยความจำหลัก.....32

สารบัญภาพ

หน้า

ภาพที่ 4.14 แสดงโปรแกรมเอกเซอร์ไซเซอร์ของหน่วยความจำหลัก……32
ภาพที่ 4.15 แสดงโปรแกรมที่ทำการจองพื้นที่ในหน่วยความจำหลัก……..…….... 33
ภาพที่ 4.16 แสดงเอกเซอร์ไซเซอร์ขณะใช้งาน……..……... 33
ภาพที่ 4.17 การใช้ดีเทรสตรวจดูหน่วยความจำหลัก……..……..33
ภาพที่ 4.18 จียูไอของเอกเซอร์ไซเซอร์พีเอชพี……..……..34
ภาพที่ 4.19 แสดงการเลือกดีสคริปต์ของพีเอชพี……..…… 34
ภาพที่ 4.20 แสดงโปรแกรมของเอกเซอร์ไซเซอร์พีเอชพี……..…… 35
ภาพที่ 4.21 แสดงเอกเซอร์ไซเซอร์ขณะใช้งาน……..……36
ภาพที่ 4.22 ผลการใช้ดีเทรสในพีเอชพี……..……..…… 36
ภาพที่ 4.23 แสดงดีสคริปต์และไซม์ของรูบี้……..……37
ภาพที่ 4.24 จียูไอของเอกเซอร์ไซเซอร์รูบี้……..……38
ภาพที่ 4.25 แสดงโปรแกรมภาษารูบี้……..……..……38
ภาพที่ 4.26 แสดงเอกเซอร์ไซเซอร์ขณะใช้งาน……..……39
ภาพที่ 4.27 ผลของดีเทรสสำหรับรูบี้……..……..……39
ภาพที่ 4.28 ผลการใช้ไซม์ของรูบี้……..……40

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ทุกวันนี้ระบบเซิร์ฟเวอร์ที่ใช้งานจริง (Production system) และโปรแกรมระบบงานมีขนาดที่ใหญ่และซับซ้อนมาก และความต้องการระบบหรือโปรแกรมที่มีประสิทธิภาพยิ่งมีมากโดยเฉพาะ จะเห็นได้ว่าในปัจจุบันการให้ความสำคัญกับสมรรถนะและประสิทธิภาพในการทำงานมีมากกว่าการให้ความสำคัญกับการสร้างระบบหรือโปรแกรม [1] ในช่วงที่ทำการพัฒนาระบบนั้น การวัดสมรรถนะ ประสิทธิภาพของระบบหรือโปรแกรมจะทำได้แต่การจำลองสถานการณ์เท่านั้น แต่ในการใช้งานจริงจะพบกับปัญหาหลากหลายรูปแบบซึ่งยากแก่การจำลองหรือทำได้ ดังนั้นการที่จะทราบได้ว่าระบบหรือโปรแกรม เมื่อเกิดปัญหาขึ้น ยิ่งในระบบการผลิตที่นำไปใช้งานแล้วจะต้องแก้ในสภาพแวดล้อมของการใช้งานจริง และสิ่งที่จะมาช่วยในการแก้ไขปัญหาของระบบนั้นจะต้องไม่มีผลกระทบกับระบบ ปัจจุบันมีชุดเครื่องมือจำนวนมากที่ใช้วิเคราะห์และวัดประสิทธิภาพของระบบปฏิบัติการยูนิกซ์และลีนุกซ์ เช่น Dtrace [2] Cacti [3] Nagios [4] Gnome System Monitor [5] เป็นต้น แต่ดีเทรสมีความแตกต่างจากเครื่องมือทั่วไปคือ ดีเทรสใช้ในการวิเคราะห์สมรรถนะ ประสิทธิภาพ ตรวจจับพฤติกรรมการทำงานของระบบ และตรวจสอบหาปัญหาที่เป็นสาเหตุต่อระบบแบบทันกาล (Real time) ทั้งในระดับโปรแกรมประยุกต์ และเคอร์เนล (kernel) โดยไม่มีผลต่อสมรรถนะของระบบปฏิบัติการ [6]

ดีเทรส (DTrace) เป็นเครื่องมือที่ใช้วิเคราะห์และติดตามพฤติกรรมของระบบแบบทันกาล (Dynamic Tracing) รวมทั้งสมรรถนะในส่วนต่างๆ ของระบบปฏิบัติการ โดยดีเทรสถูกพัฒนาโดยบริษัท ซัน ไมโครซิสเต็ม (Sun Microsystems) เพื่อทำการช่วยแก้ปัญหา (Troubleshooting) และวิเคราะห์ปัญหา (Analysis) ในระดับผู้ใช้ (User land) และระดับเคอร์เนล (Kernel) วิเคราะห์ปัญหาเพื่อติดตาม สืบหาสาเหตุสำคัญของสิ่งที่ทำให้สมรรถนะของระบบที่ใช้งานจริงมีปัญหาในขณะนั้น [7] ในการติดตามระบบดีเทรสจะทำการติดตามโดยใช้ตัวตรวจวัด (Probe) ทำการติดตามระบบหรือซอฟต์แวร์ที่ต้องการติดตามที่กำลังทำงานในจุดที่กำลังมีปัญหา (Hot spot) เพื่อติดตามหาสาเหตุ ต้นตอของปัญหา ที่ทำให้ระบบหรือซอฟท์แวร์เกิดปัญหา เช่น ระบบทำงานไม่เต็มที่ ดีเทรสสามารถติดตามการทำงานในส่วนต่างๆของระบบปฏิบัติการทั้งในระดับล่าง เช่น หน่วยความจำ หน่วยประมวลผลกลาง ระบบแฟ้มข้อมูล และทรัพยากรเครือข่าย เป็นต้น และใน

ระดับที่เป็นเครื่องเสมือน (Virtual Machine) เช่นจาวา (Java) ไพธอน (Python) เป็นต้น ที่ทำงานในแต่ละโพรเซส (Processes) เพื่อนำไปวิเคราะห์หาสาเหตุสำคัญของปัญหาได้

ปัญหา

ในปัจจุบันมีบทความและเว็บไซต์จำนวนมากที่สอนการใช้งานดีเทรส [8][9][10] ซึ่งในการสอนนั้นเป็นการสอนแบบให้ฝึกการใช้งานด้วยมือ (Manual) หรือแยกส่วนซอฟต์แวร์ เช่น การเขียนสคริปต์ซึ่งต้องใช้ความรู้เรื่องภาษา “ดี” (D language) รวมถึงการใช้คำสั่งต่างๆ บนระบบปฏิบัติการยูนิกซ์และลีนุกซ์ ซึ่งโดยภาพรวมเป็นการสอนและการใช้งานที่ค่อนข้างยากสำหรับผู้ที่ไม่มีพื้นฐานด้านยูนิกซ์และลีนุกซ์และความรู้ด้านการเขียนโปรแกรม

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 สร้างชุดโปรแกรมดีเทรส

1.2.2 ทดลองและทดสอบโปรแกรมกับระบบในรูปแบบต่างๆ

1.3 ขอบเขตของการวิจัย

1.3.1 สร้างจียูไอ Graphic User Interface (GUI) มาใช้ในการทำงานบนดีเทรสเพื่อให้ง่ายต่อการใช้งาน (Easy to use)

1.3.2 ใช้ทรัพยากรของระบบน้อย สามารถใช้งานบนโน้ตบุ๊กได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้ชุดโปรแกรมดีเทรสให้ผู้ศึกษาเข้าใจการใช้งานดีเทรสง่ายขึ้น

1.4.2 ใช้เป็นคอมพิวเตอร์ช่วยสอน (Computer Aid Instruction)

1.4.3 เพื่อให้ผู้ใช้งานมีความเข้าใจในการใช้ดีเทรสเพื่อทำการวัดประสิทธิภาพ รวมถึงการทดสอบหาปัญหาของโปรแกรมที่จะทำการตรวจวัดให้ใช้งานง่าย

1.4.4 เพื่อเป็นการสนับสนุนซอฟต์แวร์และดีเทรส

1.4.5 เพื่อนำไปประยุกต์ใช้ในระบบขององค์กร

1.5 วิธีการดำเนินการวิจัย

1.5.1 ศึกษาและทดลองการใช้งานดีเทรส

1.5.2 ออกแบบจียูไอส่วนต่างๆ ของเอกเซอร์ไซเซอร์ (Exerciser) เช่น ในส่วนของหน่วยประมวลผลกลาง หรือในส่วนของหน่วยความจำ เป็นต้น มีรายการในการเลือก

สคริปต์ในการจับค่าในรูปแบบต่างๆ มีปุ่ม (Button) ที่จะปรับแต่งค่า (Tune) พารามิเตอร์เพื่อทดลองวัดประสิทธิภาพได้ พัฒนาบนเน็ตบีนส์ (Netbeans)

1.5.3 ในส่วนของการแสดงผลลัพธ์ที่ได้ ทำการจัดรูปแบบที่เข้าใจง่าย

1.5.4 พัฒนาซอฟต์แวร์ในส่วนที่ติดต่อกับจียูไอและดีเทรส เพื่อทำการควบคุม เชื่อมต่อกับส่วนของโปรแกรมทดสอบส่วนต่างๆ ของระบบ

1.5.5 พัฒนาเอกเซอร์ไซเซอร์

1.5.6 รวบรวมเป็นชุดเครื่องมือดีทีเอ็กซ์ (DTEx - DTrace Exerciser) เป็นเน็ตบีนส์ปลั๊กอิน

1.5.7 ปรับปรุงแก้ไขงานวิจัย

1.5.8 สรุปผลงานวิจัย และจัดทำวิทยานิพนธ์

ผลงานตีพิมพ์

บทความชื่อ “Integration of DTrace Toolkit Suite for Self Leaning and Training of System Administrators” โดย วรวุฒิ ไพเราะ และ ยรรยง เต็งอำนวย ในงานประชุมวิชาการชื่อ “The 8th National Conference on Computing and Information Technology (NCCIT2012)” ซึ่งจัดขึ้นในวันที่ 9-10 พฤษภาคม 2555 ณ เมืองพัทยา จ.ชลบุรี ประเทศไทย

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดและทฤษฎี

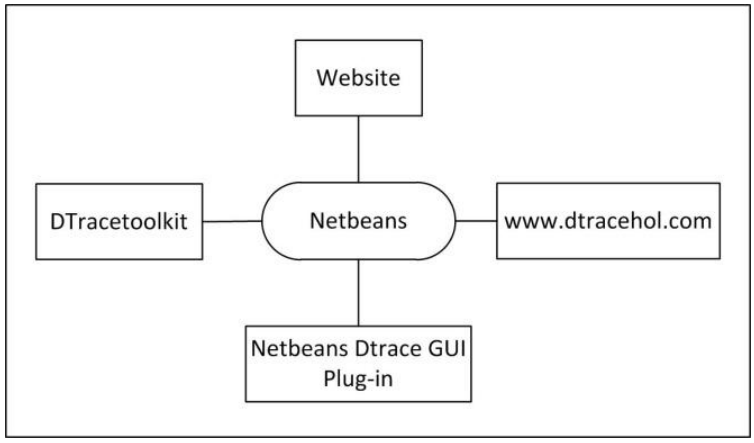
งานวิจัยชิ้นนี้มีองค์ประกอบหลายสามด้านที่นำมาผนวกเข้าเป็นชุดเครื่องมือดีเทรส ได้แก่

1) ไอดีอี (IDE-Integrated Development Environment) ในปัจจุบันไอดีอีมีอยู่มาก แต่เลือกใช้เน็ตบีนส์ เพราะสามารถติดตั้งได้หลายระบบปฏิบัติการอีกทั้งยังใช้ในการพัฒนาซอฟต์แวร์และรองรับการติดตั้ง สร้างปลั๊กอินได้

2) ซอฟต์แวร์ คือส่วนที่เป็นดีเทรส โดยทำการเลือกดีเทรสที่ได้ทำการพัฒนาเป็นจียูไอที่เป็นเน็ตบีนส์ปลั๊กอินเพราะมีชุดของดีสคริปต์และปลั๊กอินถูกพัฒนาให้ติดตั้งเฉพาะบนเน็ตบีนส์ [11]

3) เอกเซอร์ไซเซอร์ คือโปรแกรมที่ทำหน้าที่รบกวนหรือทดสอบระบบ โดยจะให้ระบบทำงานหนักขึ้นเพื่อให้ผู้ใช้เห็นได้ชัดว่าส่วนใดกำลังมีปัญหาด้านสมรรถนะ ทำการสร้างด้วยจาวาเน็ตบีนส์จียูไอ และจัดทำเป็นเน็ตบีนส์ปลั๊กอิน

ทั้งสามองค์ประกอบรวมกันพัฒนาเป็นโปรแกรมชื่อ ดีทีเอกซ์ (DTEx-DTrace Exerciser) ภาพที่ 2.1 แสดงส่วนประกอบของชุดเครื่องมือ โดยมีเน็ตบีนส์เชื่อมต่อกับชุดเครื่องมือดีเทรสเพื่อทำการพัฒนาให้เป็นปลั๊กอิน ปลั๊กอินเน็ตบีนส์ดีเทรสจียูไอ และเว็บไซต์ซึ่งเป็นแหล่งข้อมูลเพื่อรวบรวมทำเป็นเอกเซอร์ไซเซอร์

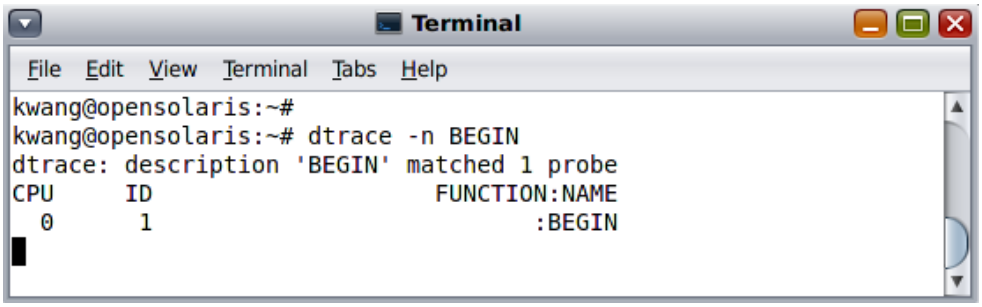


ภาพที่ 2.1 แสดงส่วนประกอบต่างๆ ของดีทีเอ็กซ์

- เน็ตบีนส์ เป็นเครื่องมือกลางที่ใช้ติดตั้งเน็ตบีนส์ปลั๊กอินจียูไอ ใช้ในการสร้างจียูไอของเอกเซอร์ไซเซอร์
- เว็บไซต์ คือ แหล่งข้อมูลซึ่งนำแบบฝึกหัดของเว็บไซต์ต่างๆ มาสร้างในส่วนประกอบของเอกเซอร์ไซเซอร์ เช่น
- www.dtracehol.com คือ เว็บไซต์หลักที่นำแบบฝึกหัดมาสร้างเป็นเอกเซอร์ไซเซอร์ในส่วนของ
- ชุดเครื่องมือดีเทรส (DTrace Toolkit) คือชุดเครื่องมือที่ผู้สร้างดีเทรส [13] รวบรวมดีสคริปต์ซึ่งนำไปใช้งานทั่วไป นำมารวบรวมให้เป็นหมวดหมู่
- เน็ตบีนส์ดีเทรสจียูไอปลั๊กอิน (Netbeans GUI Plug-in) โดยเป็นปลั๊กอินที่เผยแผ่บนเว็บไซต์ของเน็ตบีนส์ [11] พัฒนาโดย N. Nouri

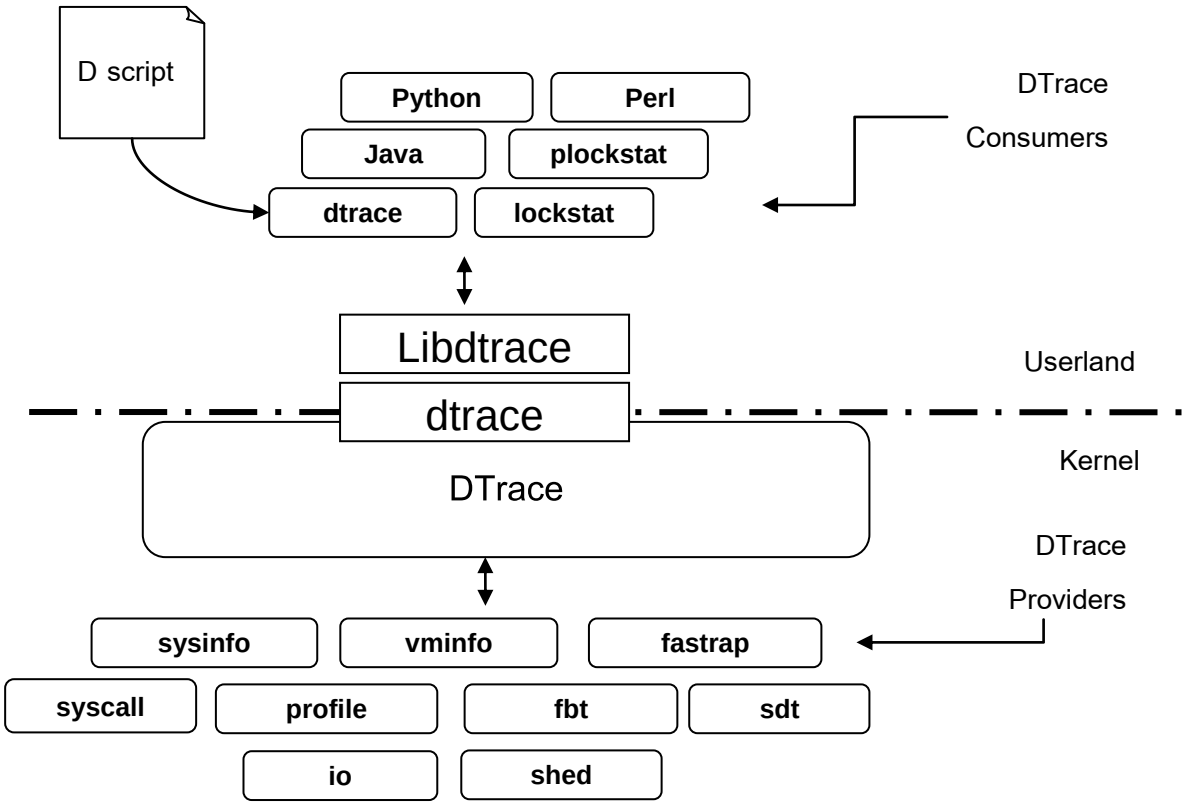
2.1.1 ดีเทรส

ดีเทรส หรือ Dynamic Tracing เป็นซอฟต์แวร์ที่ผู้ดูแลระบบใช้ในการติดตามดูพฤติกรรมของระบบ ทั้งโปรแกรมของผู้ใช้และระบบปฏิบัติการ ซึ่งจะได้ผลลัพธ์เพื่อใช้ในการระบุปัญหาข้อบกพร่องที่แท้จริงของระบบ โดยสามารถวิเคราะห์สมรรถนะในระดับล่าง (Low level) ได้ เช่น หน่วยประมวลผลกลาง, หน่วยความจำ, ทรัพยากรเครือข่าย เป็นต้น และในระดับบน (High level) เช่น จาวา, ไพธอน โดยดีเทรสถูกออกแบบและพัฒนาโดยบริษัท ซัน ไมโครซิสเต็ม (Sun System) ซึ่งออกแบบมาให้ใช้งานกับระบบปฏิบัติการโซลาริส (Solaris) และในภายหลังได้พัฒนาให้ใช้กับระบบปฏิบัติการอื่น เช่น ฟรีบีเอสดี (FreeBSD), แมคโอเอสเอ็กซ์ (Mac OS X) เป็นต้น ตัวอย่างแสดงการใช้คำสั่งในดีเทรสดังภาพที่ 2.2 เป็นคำสั่งเรียกใช้ตัวติดตาม (Probe) เริ่มต้น โดยจะเห็นได้ว่าผู้ใช้ต้องมีความรู้ ความเข้าใจในการใช้คำสั่งของดีเทรส



ภาพที่ 2.2 ตัวอย่างการใช้ดีเทรส

ในส่วนของสถาปัตยกรรมของดีเทรสดังภาพที่ 2.3 [6] จะประกอบด้วยส่วนที่ติดต่อกับผู้ใช้ (User land) และส่วนของเคอร์เนล (Kernel) โดยดีเทรสจะใช้ชุดคำสั่งจากไลบรารี่ชื่อ ลิบดีเทรส (libdtrace) ที่ชี้ไปยังดีเทรสโพรไวเดอร์ (DTrace Providers) ที่ทำงานในส่วนของระดับผู้ใช้และในส่วนของเคอร์เนลเพื่อทำการเรียกตัวติดตาม และใช้ร่วมกับดีสคริปต์ (D-Script) ในการสร้างตัวติดตาม มาสั่งให้ดีเทรสทำการเลือกโพรไวเดอร์ที่ต้องการติดตามในส่วนต่างๆของระบบ โดยดีสคริปต์พัฒนาด้วยภาษาดี (D-Language) ในการโดยดีเทรสจะต้องไม่ไปรบกวนระบบ ใช้โปรเซสในการทำงานน้อยเพื่อให้ได้ผลการติดตามของระบบดีที่สุด



ภาพที่ 2.3 สถาปัตยกรรมดีเทรส

ตัวตรวจวัด (Probe) ซึ่งอยู่ในแต่ละโพรไวเดอร์ จะกำหนดให้ชี้ไปยังเคอร์เนล (Kernel) เมื่อโปรแกรมที่จะทำการติดตามผ่านตัวติดตาม ตัวติดตามทำการวัดค่า (Fired) เพื่อรับค่าที่ได้และทำ

การเก็บค่าตรงโปรเซสนั้นมาแสดงผล เช่นในไลบรารี่ซี มีฟังก์ชันที่ใช้จองหน่วยความจำ (malloc) โดยตัวติดตามโปรเซสเมื่อมีการเรียก malloc หรือมีการเรียกเพื่อส่งค่าคืน ซึ่งจะเขียนดีสคริปต์เพื่อกำหนดตัวติดตามได้ว่า libc:malloc:return สำหรับในแต่ละตัวติดตามนั้นประกอบด้วย 4 ส่วน โดยมีชื่อที่เกี่ยวข้องกับปัญหาการล่มของเคอร์เนล (Clashes in the kernel) แต่ละส่วนใช้ชื่อเรียกแตกต่างกัน ได้แก่ โพรไวเดอร์ (Provider) โมดูล (Module) ฟังก์ชัน (Function) และชื่อ (Name) โดยทำการระบุตัวติดตามได้ดังนี้ provider:module:function:name

โพรไวเดอร์ (Provider) คือชื่อของกลุ่มของตัวติดตามหลายตัวรวมกัน โดยโพรไวเดอร์จะถูกกำหนดโดยผู้พัฒนาส่วนนั้นของระบบ ด้วยการวางตัวตรวจวัดไว้ในจุดต่างๆของโปรแกรมที่ได้พัฒนาขึ้น โพรไวเดอร์เป็นผู้กำหนดกรอบการทำงาน และควบคุมการทำงานของตัวติดตาม ตัวอย่างโพรไวเดอร์ เช่น ซีสคอลโพรไวเดอร์ (syscall) เอฟบีทีโพรไวเดอร์ (fbt) จาวา (Java) พีเอชพี (Php) เป็นต้น

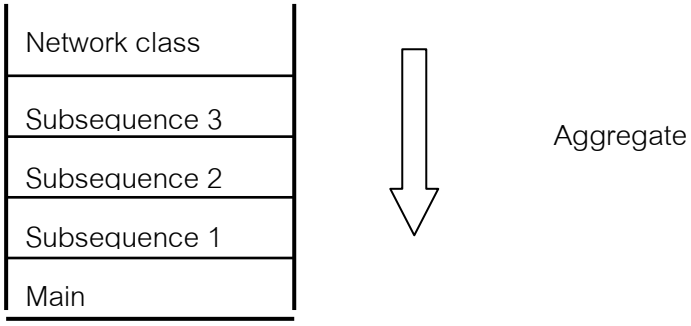
การรวบรวม (Aggregation) คือทำการวิเคราะห์ซอฟต์แวร์เมื่อทำการใช้ดีเทรสแล้ว จะทำการแบ่งชั้นเพื่อนำไปวิเคราะห์โครงสร้างลำดับชั้น (Stack) โดยจุดประสงค์เพื่อทำให้สามารถวิเคราะห์หาสาเหตุที่แท้จริงของปัญหาของระบบเพื่อแก้ไขได้ตรงจุด [12] โดยจะใช้ในการวิเคราะห์ในระดับต่างๆจากชั้นบนสุดลงไปยังชั้นล่างสุด จากนั้นจะได้รวบรวม ผลของการติดตามในทุกระดับชั้น ผลลัพธ์ที่ได้คือจุดที่เป็นปัญหาของระบบ (Hot spot) ตัวอย่างเช่น มีแอพพลิเคชันซึ่งมีปัญหาในการส่งกลุ่มข้อมูล (Packet) ของเครือข่ายช้า มีโครงสร้างดังภาพที่ 2.4

```
Application
        main() {
                Subsequence 1 ();
        }
        Subsequence 1 () {
                Subsequence 2 ();
        }
        Subsequence 2 () {
                Subsequence 3 ();
        }
```

```
Subsequence 3 () {
        Network class ();
}
Network class () {
}
```

ภาพที่ 2.4 ตัวอย่างลำดับชั้นโครงสร้างของโปรแกรม

เมื่อทำการรันแอปพลิเคชัน และใช้ดีเทรสในการติดตาม จากนั้นทำการนำส่วนต่างๆของแอปพลิเคชันมาเรียงลำดับชั้นในการเรียก (Call Stack) ดังภาพที่ 2.5 เพื่อสืบย้อนกลับไปยังลำดับชั้นที่เรียกมา (Call) เพื่อหาต้นเหตุของปัญหาด้านสมรรถนะนั้น



ภาพที่ 2.5 การรวบรวมในลำดับชั้นของการเรียกโปรแกรมย่อย

ตัวอย่างเช่น เกิดปัญหากับการส่งแพ็กเก็ตของเครือข่ายช้า เมื่อทำการรวบรวมลงไปยังชั้นที่ต่ำกว่าเพื่อดูอาการของปัญหา แล้วย้อนขึ้นไปจนถึงชั้นที่เป็นสาเหตุของปัญหานั้น เช่นฟังก์ชันหลักของแอปพลิเคชัน (Main function) ที่ทำการสร้างกลุ่มข้อมูล (Generate packet) เป็นต้น

2.1.2 โปรแกรมภาษา “ดี”

โปรแกรมภาษา “ดี” (D Programming Language) มีลักษณะคล้ายภาษา “ซี” และ “ซีพลัสพลัส” (C/C++) ไว้ใช้ในการเขียนสคริปต์สร้างตัวติดตามและกำหนดการรับผลให้ดีเทรส

โดยจะเขียนให้มีพารามิเตอร์ กำหนดสิ่งที่ต้องตรวจวัด ในการเก็บค่าต่างๆในส่วนของระบบที่ต้องการเฝ้าดู ข้อดีคือไม่มีการวนซ้ำ (Loop) โดยมีโครงสร้างของภาษาดี ดังภาพที่ 2.6

```
Probe-description
/predicate/
{
        action 1; action 2; … ; action n;

}
```

ภาพที่ 2.6 โครงสร้างของภาษา “ดี”

รายละเอียดตัวตรวจวัด (Probe-description) เป็นการกำหนดตัวติดตามที่สนใจจะเก็บข้อมูล เพรดิเคต (Predicate) คือส่วนที่ใช้เปรียบเทียบว่าตรงตามเงื่อนไขหรือไม่ ถ้าตรงให้เข้าไปทำงานในส่วนที่อยู่หลังเพรดิเคต ถ้าไม่ใช้ให้ไปเปรียบเทียบกับเพรดิเคตถัดไป การปฏิบัติ (Action) คือการกำหนดว่าจะให้ตัวติดตามนี้ดำเนินการอะไร เช่น เก็บข้อมูลแบบใด เป็นต้น ภาพที่ 2.7 แสดงตัวอย่างดีสคริปต์ที่เขียนโดยภาษา “ดี”

```
syscall : : : entry
/pid == $target/
    {
        @calls[pid, “syscall”, probefunc] = count();
    }
```

ภาพที่ 2.7 ตัวอย่างสคริปต์เขียนโดยภาษา “ดี”

2.1.3 ชุดเครื่องมือดีเทรส

ชุดเครื่องมือดีเทรส [13] เป็นชุดเครื่องมือที่รวบรวมดีสคริปต์ที่ใช้มากและเป็นประโยชน์ โดยได้รวบรวมไว้เป็นหมวดหมู่ต่างๆ เช่น ซีพียู หน่วยความจำหลัก จาวา และพีเอชพี เป็นต้น โดยการแบ่งหมวดหมู่นี้เป็นในแต่ละส่วนของระดับซอฟต์แวร์ เพื่อใช้ในการวิเคราะห์และรวบรวมในแต่ละระดับชั้นเพื่อหาต้นตอของปัญหาที่เกิดต่อระบบ [12] อย่างไรก็ตามการใช้ดีเทรสไม่ใช่ชุดโปรแกรมที่ระบุปัญหาได้เอง แต่เป็นตัวช่วยในการวิเคราะห์และสรุปรวบยอดข้อมูลให้เท่านั้น [14]

ดังนั้นผู้ใช้งานต้องมีความรู้โครงสร้างการทำงานของดีเทรสและระบบปฏิบัติการยูนิกซ์และลีนุกซ์ได้ในระดับหนึ่ง การใช้งานชุดเครื่องมือดีเทรสนั้นต้องทราบถึงคำสั่งต่างๆของดีเทรสเพราะชุดเครื่องมือไม่มีจียูไอ โดยชุดเครื่องมือดีเทรสนั้นเป็นเหมือนตัวช่วยในการเลือกใช้ดีเทรสให้ตรงกับความต้องการเท่านั้น

2.1.4 **เน็ตบีนส์**

เน็ตบีนส์ [15] เป็นกรอบการทำงานของโปรแกรมประยุกต์จาวา และซอฟต์แวร์ที่ใช้ในการพัฒนา โดยใช้ในการติดตั้งปลั๊กอินเน็ตบีนส์ดีเทรสจียูไอ (NetBeans DTrace GUI Plug-in) [11] เพื่อเป็นกรอบการทำงานให้กับชุดเครื่องมือดีเทรส และใช้ในการพัฒนาในส่วนของเอกเซอร์ไซเซอร์ที่เป็นจียูไอ โดยได้ทำการพัฒนาให้เป็นปลั๊กอิน เพื่อให้สามารถนำไปใช้งานได้ง่าย

2.1.5 **ปลั๊กอินเน็ตบีนส์ดีเทรสจียูไอ**

ปลั๊กอินเน็ตบีนส์ดีเทรสจียูไอ คือโปรแกรมปลั๊กอินที่ได้พัฒนาและเผยแพร่บนเว็บไซต์ของเน็ตบีนส์เพื่อให้ผู้สนใจได้นำไปติดตั้งบนเน็ตบีนส์ไอดีอีและใช้งาน โดยชุดเครื่องมือนี้ประกอบด้วย 2 ส่วนคือ (1) ส่วนของดีสคริปต์ซึ่งเป็นการนำดีสคริปต์ ต่างๆของชุดเครื่องมือดีเทรสมาใช้งานและจัดหมวดหมู่ให้ใช้งานง่าย (2) ไชม์ (Chime) คือชุดเครื่องมือที่แสดงผลเป็นกราฟิก โดยชุดปลั๊กอินนี้สามารถเพิ่มเติมและสร้างดีสคริปต์ได้ด้วย

2.1.6 **ไชม์**

ไชม์ (Chime) คือชุดเครื่องมือสำหรับการรวบรวม (Aggregation) ด้วยดีเทรส โดยแสดงผลลัพธ์ทางกราฟิก ไชม์เป็นส่วนหนึ่งของปลั๊กอินเน็ตบีนส์จียูไอ ประโยชน์คือมีการแสดงผลที่ง่าย ผู้ใช้สามารถเรียงลำดับในคอลัมน์ (Column) ได้ เหมาะกับการใช้ดูพฤติกรรมของระบบและแอพพลิเคชั่นได้ ผู้ใช้สามารถสร้างดีสคริปต์สำหรับรวบรวมได้ และสะดวกใช้งานบนไชม์ได้

2.1.7 **เอกเซอร์ไซเซอร์**

เอกเซอร์ไซเซอร์ (Exerciser) คือชุดโปรแกรมที่ทำการพัฒนาให้เป็นปลั๊กอินบนเน็ตบีนส์ สร้างด้วยภาษาจาวาบนเน็ตบีนส์ จุดประสงค์ของเอกเซอร์ไซเซอร์คือ ทำการทดสอบ จำลองเหตุการณ์ต่างๆ ต่อระบบ เช่น รบกวน บริหารระบบ กระตุ้นระบบ ทำให้ระบบทำงานหนัก เป็นต้น ประกอบด้วยโปรแกรมภาษาต่างๆ ที่มีในกลุ่มของดีสคริปต์ และมีรูปแบบการทำงานให้มีหลายรูปแบบ โดยได้ทำการรวบรวมจากเว็บไซต์ [9]

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัยที่เกี่ยวกับดีเทรส

งานวิจัย [1] ของ Bryan M. และคณะ นำเสนอดีเทรสเป็นครั้งแรก เป็น Dynamic instrumentation ที่ทำงานทั้งในระดับผู้ใช้ (User-level) และระดับเคอร์เนล (Kernel-level) เสนอว่าดีเทรสจะไม่มีผลกระทบต่อระบบ ไม่ใช้โปรเซสมาก (Zero probe effect) คือเป็นซอฟต์แวร์ที่ทำงานโดยไม่มีผลกระทบต่อระบบ ผลกระทบเป็นศูนย์ เป็นซอฟต์แวร์ที่ติดตามส่วนต่างๆได้มากกว่าหมื่นจุด ใช้ประเมินประสิทธิภาพที่จะเกิดขึ้นต่อไปของระบบ ระบุปัญหาของประสิทธิภาพของระบบ

งานวิจัย [12] ของ B. Cantril นำเสนอที่มา ส่วนประกอบต่างๆ ของดีเทรส เสนอแนวทางการวิเคราะห์ปัญหาของซอฟต์แวร์ที่ใช้ดีเทรสทดสอบเพื่อให้ได้ปัญหาที่เป็นปัญหาที่แท้จริง โดยทำการแบ่งซอฟต์แวร์ออกเป็นระดับชั้น และทำการวิเคราะห์ไปจนถึงชั้นที่มีปัญหา และให้ทำการรวบรวม (Aggregation) ข้อมูลลงไปยังชั้นสูงกว่า จนพบสาเหตุที่ทำให้ระบบมีปัญหา อีกทั้งงานวิจัยนี้ได้ยกตัวอย่างการใช้งานดีเทรส ทดลองและแสดงผล ระบุปัญหาว่าเกิดจากอะไร และทำการเพิ่มโปรแกรมที่ใช้เพื่อแก้ปัญหาที่พบ

งานวิจัย [16] ของ Q. Ni และคณะ นำเสนอการพัฒนาดีเทรสจียูไอ ให้ดีเทรสมีกราฟิกติดต่อกับผู้ใช้ โดยเสนอการใช้งานที่ง่าย (Easy to use) ใช้ต้นทุนต่ำในการพัฒนาและให้ประสิทธิภาพสูงในการใช้งาน โดยได้เสนอการใช้จีทีเคพลัส (GTK+) ในการพัฒนาจียูไอขึ้น ใช้เกลด (Glade) ในการเชื่อมต่อในส่วนของผู้ใช้ผ่านทางจียูไอ เพื่อเรียกใช้งานสคริปต์ต่างๆด้วยเอ็กซ์เอ็มแอล (XML) และใช้ลิปเกลด (Libglade) เป็นไลบรารี่เพื่อใช้สร้างอินเตอร์เฟสกับไฟล์เอ็กเอมแอล

งานวิจัย [6] ของ R. Cecil นำเสนอนิยาม การทำงาน การใช้งานของดีเทรสบนระบบปฏิบัติการโซลาริส รวมถึงสคริปต์ ภาษา “ดี” รูปแบบการสร้างสคริปต์ และตัวอย่างสคริปต์ส่วนประกอบและตัวแปรต่างๆ ที่ทำงานกับดีเทรส

2.2.2 งานวิจัยที่เกี่ยวกับเครื่องมือที่เกี่ยวข้อง

งานวิจัย [7] ของ R. McDougall และคณะ นำเสนอชุดเครื่องมือดีเทรส (DTracetoolkit) คือชุดเครื่องมือที่รวบรวมดีสคริปต์ที่ใช้งานบ่อยและเป็นประโยชน์ รวบรวมและจัดหมวดหมู่ โดยชุดเครื่องมือนี้ต้องติดตั้งบนระบบปฏิบัติการโซลาริส สำหรับการใช้งานผู้ใช้ต้องใช้งานผ่าน

เทอร์มินอลด้วยการใช้คำสั่ง (Command line) ซึ่งผู้ใช้งานต้องมีความรู้เกี่ยวกับคำสั่งของยูนิกซ์และดีเทรสอีกด้วย

งานวิจัย [11] ของ N. Nouri นำเสนอปลั๊กอินเน็ตบีนส์ดีเทรสจียูไอ (Netbeans DTrace GUI Plug-in) คือเครื่องมือจียูไอสำหรับดีเทรส โดยสามารถติดตั้งได้บน ซัน สตูดิโอ ไอดีอี (Sun Studio IDE) เน็ตบีนส์ ไอดีอี เวอร์ชัน 6.0 ขึ้นไป โดยปลั๊กอินนี้แบ่งเป็นส่วนของ ดีเทรส และไซม์ ส่วนของดีเทรสคือส่วนที่รวบรวมและจัดหมวดหมู่ของดีสคริปต์ให้ง่ายต่อการใช้งาน ส่วนของไซม์คือส่วนที่ใช้แสดงผลกราฟิกของการใช้ดีเทรสในการรวบรวม โดยสามารถแสดงผลได้แบบทันกาล (Real time) ผู้ใช้สามารถทำการสร้างดีสคริปต์เพื่อใช้งานดีเทรส และไซม์ได้ด้วย

งานวิจัย [17] ของ B. Gregg นำเสนอชุดเครื่องมือดีเทรสแทสทูล (DTraceTazTool) คือเครื่องมือจียูไอที่แสดงการทำงานของดิสก์ (Disk Activity) โดยเครื่องมือนี้ใช้การทำงานของดีเทรสในด้านเกี่ยวกับการตรวจวัดข้อมูลของดิสก์ โดยทำการแสดงผลลัพธ์ออกเป็นรูปแบบของกราฟ การใช้งานเครื่องมือดีเทรสแทสทูลนั้นใช้งานได้เฉพาะเกี่ยวกับดิสก์เท่านั้น และการใช้งานผู้ใช้ต้องมีความรู้เกี่ยวกับระบบของดิสก์บนยูนิกซ์ และต้องศึกษาการใช้งานเครื่องมือด้วย

งานวิจัย [18] ของ A. Rice นำเสนอชุดเครื่องมือดีไลท์ (DLight) คือเครื่องมือที่ใช้ดีเทรสในการหาปัญหาและวิเคราะห์ประสิทธิภาพของฟังก์ชันการทำงานของระบบปฏิบัติการโซลาริส โดยดีไลท์ทำการติดตั้งบนโซลาริส สตูดิโอ ไอดีอี (Solaris Studio IDE) ซึ่งเครื่องมือนี้ใช้ทดสอบกับโปรแกรมที่สร้างด้วยภาษา “ซี” หรือ “ซีพลัสพลัส” (C/C++) ทำการแสดงผลลัพธ์เป็นกราฟิกแสดงประสิทธิภาพ และการใช้ทรัพยากรของโปรแกรม

บทที่ 3

แนวทางการวิจัยและการออกแบบ

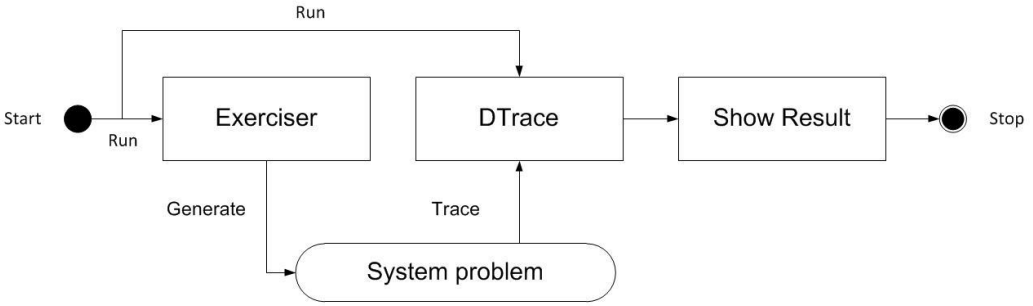
การทดสอบวัดประสิทธิภาพของระบบเซิร์ฟเวอร์ที่ใช้งานจริงมีเครื่องมือจำนวนมากที่สามารถใช้งานได้ ดีเทรสเป็นเครื่องมือที่เหมาะสมในการใช้งาน แต่เนื่องด้วยการใช้งานดีเทรสด้วยมือ (Manual) ที่ผู้ใช้ต้องมีความรู้และศึกษาการใช้งานหลายด้าน เช่น ความรู้เกี่ยวกับระบบปฏิบัติการยูนิกซ์ ความรู้ด้านของดีเทรส เป็นต้น ด้วยการใช้งานที่ยากนี้งานวิจัยนี้จึงได้นำเสนอแนวทางในการพัฒนาให้ดีเทรสมีการใช้งานง่ายขึ้น เพื่อให้ผู้ใช้งาน ผู้สนใจ เข้าใจและทำการศึกษาด้วยตนเองได้

3.1 แนวทางการวิจัย

จากการศึกษาถึงทฤษฎีและงานวิจัยที่เกี่ยวข้อง รวมไปถึงการใช้งานดีเทรสพบว่า การทำความเข้าใจกับการใช้งานนั้นค่อนข้างยาก ต้องทำการศึกษาความรู้ในหลายด้านประกอบกันจึงจะเข้าใจถึงกระบวนการการทำงานของดีเทรสและการใช้งาน รวมถึงผลลัพธ์ที่ได้จากการทดสอบระบบนั้นยังอยู่ในรูปแบบที่เข้าใจยาก ซึ่งผู้ใช้ต้องนำข้อมูลที่ได้นั้นไปทำการจัดรูปแบบเอง เช่น นำไปเขียนกราฟเอง เป็นต้น งานวิจัยนี้ได้นำแนวทางของจียูไอที่ทำให้ผู้ใช้งานดีเทรสใช้งานง่ายขึ้น แต่เป็นการใช้งานตามสคริปต์ที่ผู้พัฒนาวางไว้ และไม่สามารถปรับแต่ง (Tune) พารามิเตอร์ต่างๆ ได้ อีกทั้งการแสดงผลลัพธ์ทีได้จะแสดงผลลัพธ์ที่ได้โดยตรงซึ่งไม่ได้แสดงออกเป็นกราฟหรือบอกถึงจุดที่เป็นปัญหา

3.2 การออกแบบการผนวกชุดเครื่องมือดีเทรสตามแนวทางการวิจัยที่ได้วางไว้

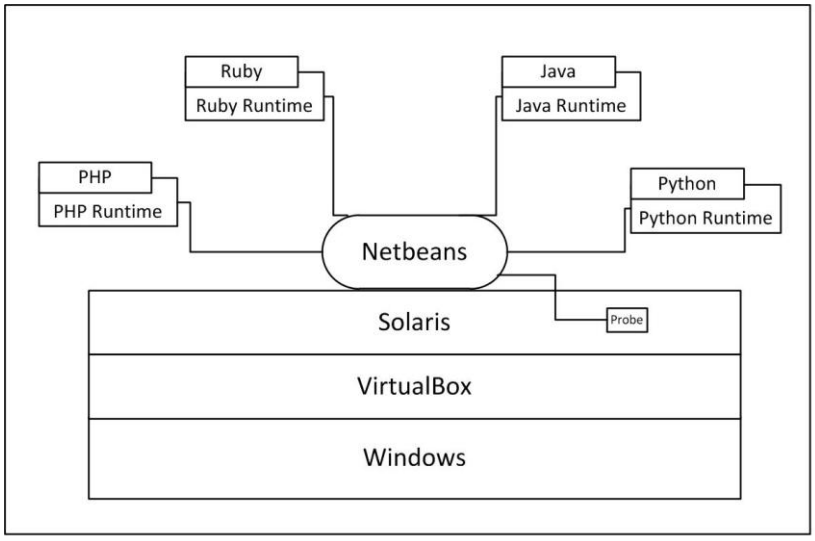
การออกแบบได้ทำการรวบรวมส่วนประกอบต่างๆ โดยให้ชื่อชุดโปรแกรมว่า “ดีทีเอ็กซ์” (DTEx – DTrace Exerciser) คือชุดโปรแกรมที่รวบรวมปลั๊กอินเน็ตบีนส์ดีเทรสจียูไอและทำการพัฒนาในส่วนของเอกเซอร์ไซเซอร์ โดยจุดประสงค์ของเอกเซอร์ไซเซอร์ คือ เป็นโปรแกรมสร้างเหตุการณ์จำลองต่อระบบ เช่น รบกวน บริหาร กระตุ้นระบบ ทำให้ระบบทำงานหนัก เพื่อเป็นแบบฝึกหัดจำลองให้มีปัญหารบกวนระบบเกิดขึ้นในระบบ และให้ผู้ใช้ทดลองใช้งานดีเทรสตรวจวัดหาสาเหตุที่ทำให้ระบบมีปัญหารบกวนที่เอกเซอร์ไซเซอร์สร้างขึ้นและทำการแสดงผลลัพธ์ ภาพที่ 3.1 แสดงแผนผังการใช้งานโดยรวมของผู้ใช้งานชุดโปรแกรม



ภาพที่ 3.1 แสดงแผนผังการใช้งานโดยรวมของผู้ใช้งานชุดโปรแกรม

ผู้ใช้ทำการเลือกใช้งานและเปิดดีเทรสหรือเอกเซอร์ไซเซอร์ จากนั้นเอกเซอร์ไซเซอร์ทำการรบกวนระบบ ดีเทรสทำการตรวจวัดข้อมูลที่ได้จากระบบ เมื่อเอกเซอร์ไซเซอร์ทำงานเสร็จ ดีเทรสทำการแสดงผลลัพธ์

ในเอกเซอร์ไซเซอร์ประกอบด้วยโปรแกรมแบ่งเป็นกลุ่มสำหรับหลายระดับชั้นทั้งในด้านฮาร์ดแวร์และซอฟต์แวร์ คือ หน่วยประมวลผล หน่วยความจำหลัก จาวา พีเอชพี รูบี้และไพธอน ภาพที่ 3.2 แสดงองค์ประกอบของชุดเครื่องมือ โดยเน็ตบีนส์ทำการเรียกตัวตรวจวัด (Probe) ซึ่งฝังตัวอยู่ในระบบปฏิบัติการโซลาริส รวมถึงรันไทม์ของภาษาต่างๆ ผ่านทางเน็ตบีนส์เพื่อทำการเก็บข้อมูลที่ได้นำไปแสดงผลบนเน็ตบีนส์

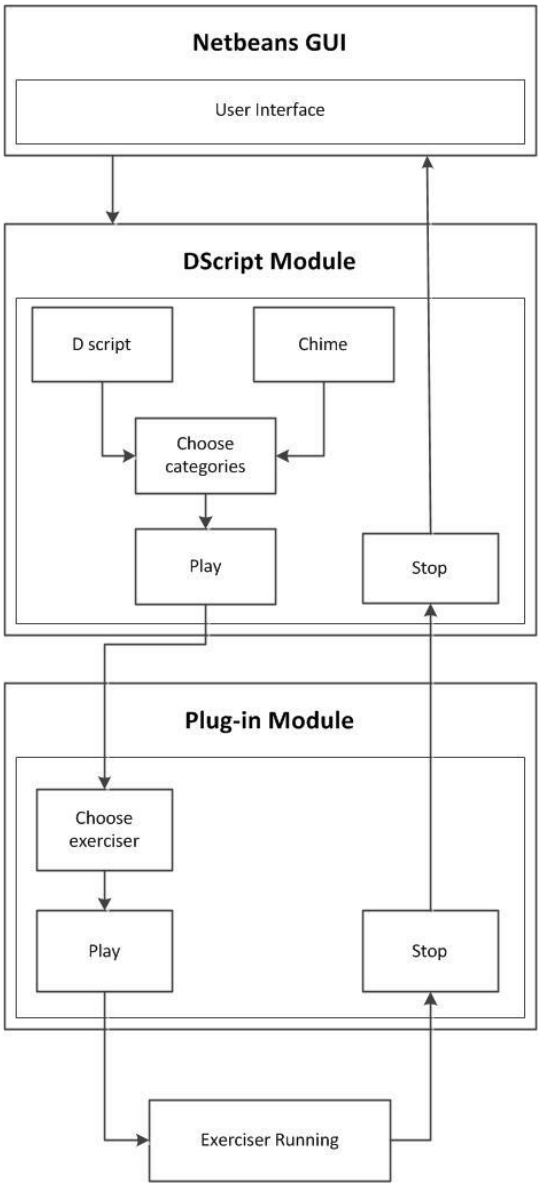


ภาพที่ 3.2 องค์ประกอบของชุดโปรแกรมเอกเซอร์ไซเซอร์

สำหรับสิ่งที่ผู้ใช้ได้รับจากการใช้งานชุดเครื่องมือดีทีเอ็กซ์นั้น ผู้ใช้ได้เข้าใจการใช้งานดีเทรส โดยทำการทดสอบระบบด้วยเอกเซอร์ไซเซอร์ในหมวดหมู่ต่างๆ ซึ่งผู้ใช้สามารถเลือกเอกเซอร์ไซเซอร์ที่ทดสอบระบบ และทดลองการใช้งานดีเทรสและดูผลลัพธ์ที่ได้จากการใช้งานดีเทรส โดยเป็นการศึกษาการใช้งานดีเทรสด้วยวิธีที่ง่ายขึ้น และใช้เอกเซอร์ไซเซอร์ทดสอบเป็นแบบฝึกหัดที่แสดงให้เห็นในด้านต่างๆ ที่หลากหลาย

3.3 ชุดเครื่องมือดีทีเอ็กซ์

งานวิจัยนี้มีองค์ประกอบที่กล่าวไว้ข้างต้น โดยการใช้งานชุดเครื่องมือดีทีเอกซ์ได้ทำการออกแบบไว้ดังภาพที่ 3.3 โดยแบ่งเป็น 3 ส่วน ดังนี้



ภาพที่ 3.3 ขั้นตอนการใช้งานชุดเครื่องมือดีทีเอ็กซ์

3.3.1 โมดูลเน็ตบีนส์จียูไอ

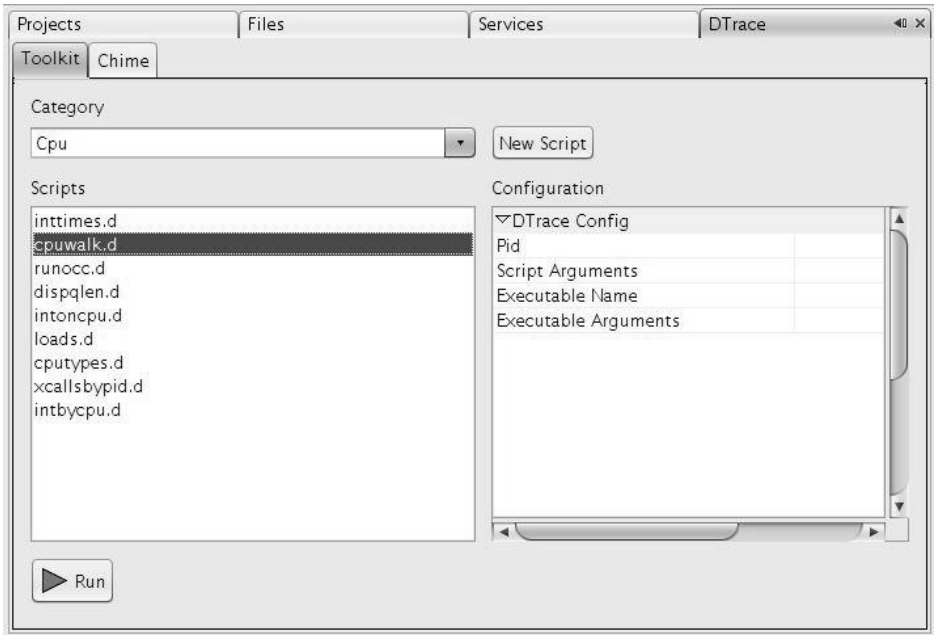
ในส่วนแรกทำการออกแบบ และสร้างจียูไอเพื่อติดต่อกับผู้ใช้ โดยทำการพัฒนาภาษาจาวาบนเน็ตบีนส์ โดยกำหนดส่วนประกอบคือ ส่วนของดีสคริปต์และส่วนของเอกเซอร์ไซเซอร์ โดยส่วนของดีสคริปต์มีการเลือกและใช้งานดีสคริปต์ และอีกส่วนคือส่วนของเอกเซอร์ไซเซอร์

3.3.2 โมดูลดีสคริปต์

โมดูลดีสคริปต์คือโมดูลที่ทำการติดตั้งปลั๊กอินเน็ตบีนส์ดีเทรสจียูไอ [11] โดยเมื่อทำการติดตั้งเสร็จจะปรากฏในส่วนของดีเทรสบนเน็ตบีนส์ดังภาพที่ 3.4 ประกอบด้วยแท็บเครื่องมือดีเทรส (Toolkit) คือส่วนที่รวบรวมและใช้งานในส่วนของดีสคริปต์ และแท็บของไชม์ (Chime) ของในส่วนนี้ให้ผู้ใช้ทำการเลือกใช้ดีสคริปต์หรือใช้สคริปต์ของไชม์ ดังนี้

3.3.2.1 ดีเทรส

ผู้ใช้เลือกใช้งานส่วนของเครื่องมือดีเทรส โดยหมวดหมู่ (Category) คือส่วนที่ผู้สร้าง [11] แบ่งดีสคริปต์เป็นประเภทของแต่ละส่วนในระบบดังตารางที่ 3.2 ช่องสคริปต์ (Scripts) แสดงดีสคริปต์ต่างๆในแต่ละหมวดหมู่ ช่องตั้งค่า (Configuration) ใช้ในการกำหนดตัวแปรในแต่ละดีสคริปต์ที่เลือก เพื่อเปิดการใช้งานดีเทรสโดยกดปุ่มให้ทำงาน (Run) ดีสคริปต์จะเริ่มทำงานจนกว่าโมดูลปลั๊กอินทำงานเสร็จ



ภาพที่ 3.4 จียูไอของดีสคริปต์ปลั๊กอิน

ในปลั๊กอินดีสคริปต์จียูไอดังภาพที่ 3.4 ทางผู้สร้าง [11] ได้ทำการแบ่งหมวดหมู่การใช้งานดีเทรสเพื่อใช้งานในการเลือกกับเอกเซอร์ไซเซอร์ โดยผู้ใช้เลือกเอกเซอร์ไซเซอร์แล้วจึงเลือกหนวดหมู่ที่เกี่ยวข้องกับเอกเซอร์ไซเซอร์ เช่น

- หน่วยประมวลผล เลือกดีสคริปต์ในหมวดของ Cpu
- หน่วยความจำหลัก เลือกดีสคริปต์ในหมวดของ Mem
- ระบบไฟล์ เลือกดีสคริปต์ในหมวดของ FS
- จาวา เลือกดีสคริปต์ในหมวดของ Java
- พีเอชพี เลือกดีสคริปต์ในหมวดของ Php
- ไพธอน เลือกดีสคริปต์ในหมวดของ Python
- รูบี้ เลือกดีสคริปต์ในหมวดของ Ruby

นอกจากนี้ผู้ใช้สามารถใช้ดีสคริปต์ตรวจวัดระบบทั่วไปที่นอกเหนือจากในเอกเซอร์ไซเซอร์ได้ ซึ่งแนะนำให้เลือกในหมวดของ TopMostUseful Proc System เป็นต้น

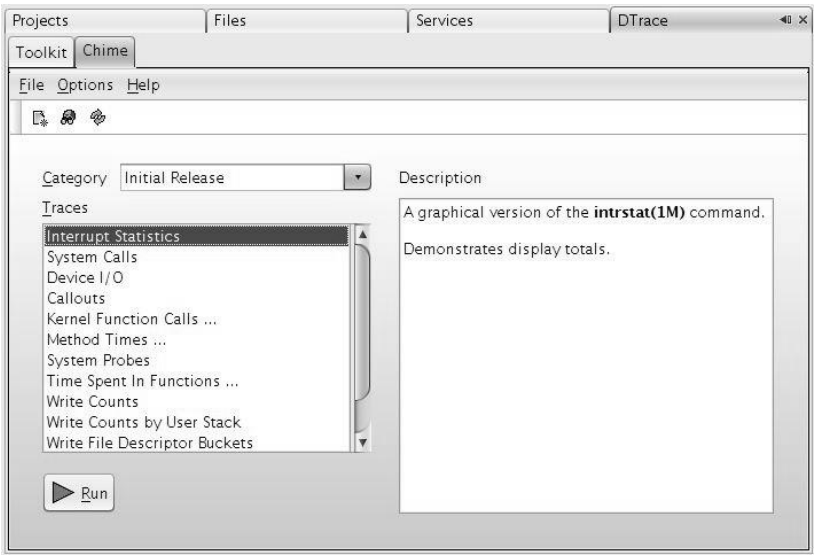
ตารางที่ 3.1 หมวดหมู่ของดีสคริปต์ในเครื่องมือดีเทรส

Categories	Description
Apps	ตรวจสอบในระดับแอปลิเคชัน ดูพฤติกรรมของเซอร์วิซ
TopMostUseful	รวมดีสคริปต์ที่นิยมใช้กับระบบปฏิบัติการ
Cpu	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของหน่วยประมวลผล
Disk	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของฮาร์ดดิสก์
FS	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของระบบจัดเก็บไฟล์
Java	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของโปรแกรมภาษาจาวา
JavaScript	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของจาวาสคริปต์
Kernel	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ที่มีการเรียกผ่านเคอร์เนล
Locks	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ Locks
Mem	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของหน่วยความจำ
Net	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของระบบเครือข่าย

Perl	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของโปรแกรมภาษาเพิล
Php	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของโปรแกรมภาษาพีเอชพี
Proc	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของโปรเซสบนระบบปฏิบัติการ
Python	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของโปรแกรมภาษาไพธอน
Ruby	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของโปรแกรมภาษารูบี้
Shell	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของ Bourne shell
System	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของระบบ
Tcl	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของ TCL
User	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของผู้ใช้
Zones	ใช้ตรวจสอบและเก็บค่าพฤติกรรมต่างๆ ของโซนในระบบปฏิบัติการ

3.3.2.2 **ไชม์**

ไชม์ (Chime) คือ เครื่องมือแสดงผลทางกราฟิก ที่แสดงในส่วนของการรวบรวมของดีเทรส (DTrace Aggregations) โดยทำงานเหมือนเครื่องมือที่เรียกคำสั่ง แสดงผลข้อมูลที่ได้ต่อหน่วยเวลา โดยแสดงให้เห็นถึงพฤติกรรมของระบบและแอพพลิเคชั่นได้ ภาพที่ 3.5 แสดงจียูไอของไชม์



ภาพที่ 3.5 จียูไอของ ไชม์

ผู้ใช้ทำการเลือกหมวดหมู่ของดีสคริปต์ (Category) จากนั้นเลือกดีสคริปต์ในช่องเทรส (Trace) โดยสามารถดูคำอธิบายในแต่ละดีสคริปต์ได้ที่ช่องอธิบาย (Description) ด้านขวา ชุดการทำงานของไซม์ แบ่งออกเป็นหมวดหมู่โดยผู้ใช้ทำการเลือกใช้ไซม์คือ

- หน่วยประมวลผล เลือกดีสคริปต์ในหมวดของ DTraceToolkit
- หน่วยความจำหลัก เลือกดีสคริปต์ในหมวดของ DTraceToolkit
- ระบบไฟล์ เลือกดีสคริปต์ในหมวดของ DTraceToolkit
- จาวา เลือกดีสคริปต์ในหมวดของ DTraceToolkit
- พีเอชพี เลือกดีสคริปต์ในหมวดของ Php
- ไพธอน เลือกดีสคริปต์ในหมวดของ Python
- รูบี้ เลือกดีสคริปต์ในหมวดของ Ruby

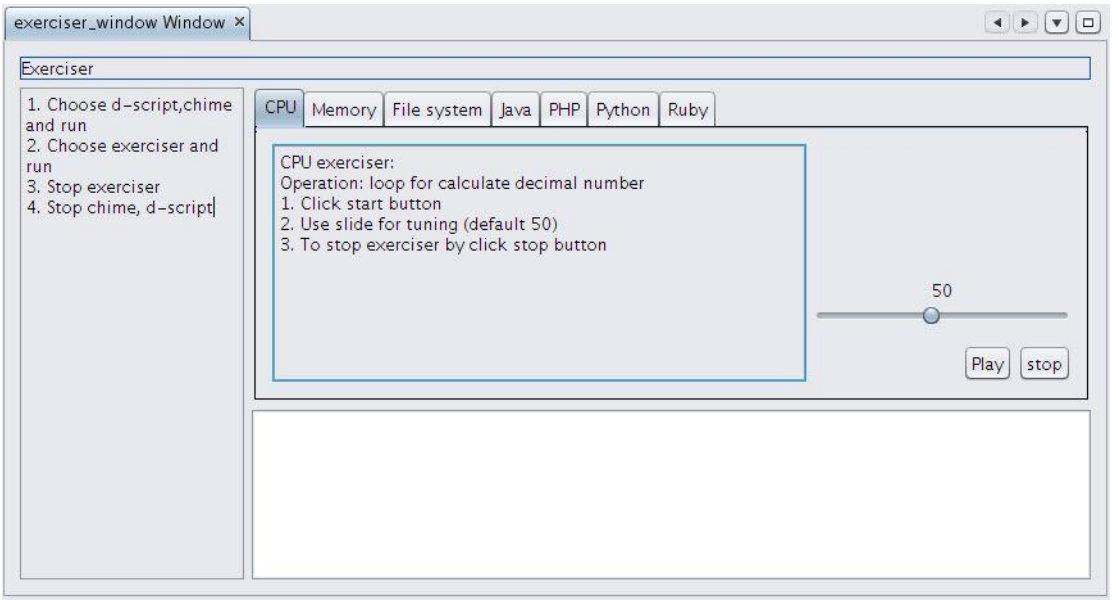
ตารางที่ 3.2 หมวดหมู่ของดีสคริปต์ใน ไซม์

Category	Description
Initial Release	สคริปต์ที่นิยมใช้ในการทดสอบระบบ
My Traces	คือสคริปต์ที่ผู้ใช้สร้างเองได้
DTraceToolkit	สคริปต์ที่ปรับปรุงมาจากชุดของดีสคริปต์
Python	คือชุดของสคริปต์ที่ใช้กับภาษาไพธอน
PHP	คือชุดของสคริปต์ที่ใช้กับภาษาพีเอชพี
Ruby	คือชุดของสคริปต์ที่ใช้กับภาษารูบี้
Zones	คือชุดของสคริปต์ที่ใช้กับโซน

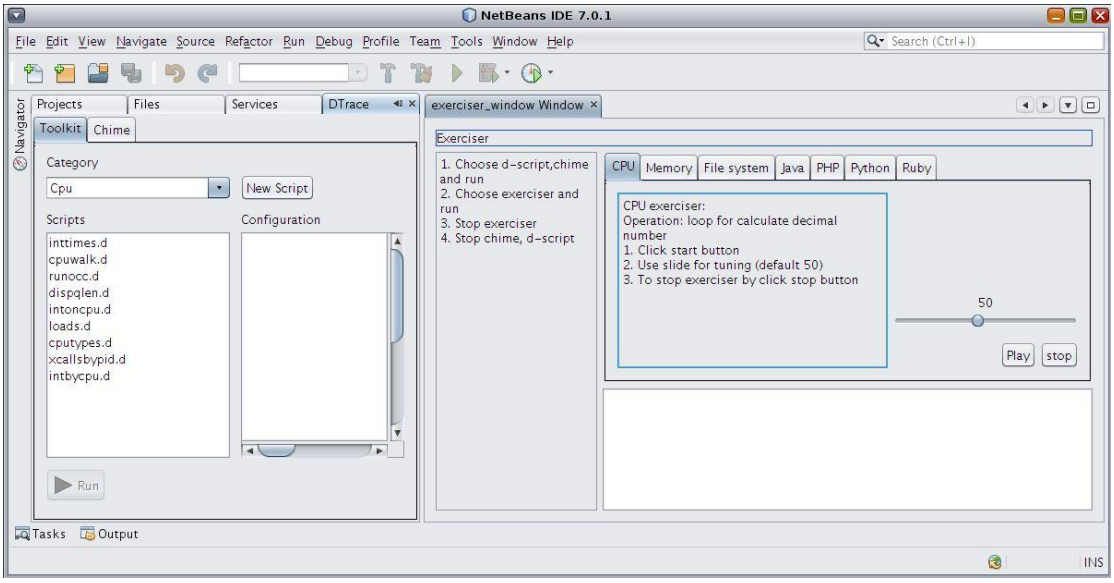
สำหรับการทำงานของโมดูลดีสคริปต์ ทำการเลือกต่อจากการเปิดใช้งานเอกเซอร์ไซเซอร์ ดีสคริปต์ทำการเก็บข้อมูล บางตัวสามารถดูผลลัพธ์ได้ทันที แต่บางตัวต้องปิดการใช้งานดีสคริปต์ก่อนถึงจะแสดงผลลัพธ์ เมื่อทำการใช้งาน และลองปรับเอกเซอร์ไซเซอร์เสร็จ ให้หยุดการทำงานและดูผลลัพธ์ที่ได้ของดีเทรส

3.4 โมดูลปลั๊กอิน

ในส่วนนี้ดำเนินการพัฒนาชุดโปรแกรมเอกเซอร์ไซเซอร์ให้เป็นจียูไอให้เป็นหมวดหมู่ที่เป็นรูปแบบเดียวกับปลั๊กอินเน็ตบีนส์ดีเทรสจียูไอเพื่อให้มีเอกเซอร์ไซเซอร์ที่ตรงกับดีสคริปต์ที่ใช้งาน ดังภาพที่ 3.6 พัฒนาให้ชุดเครื่องมือเป็นปลั๊กอิน [19] โดยทำการพัฒนาด้วยภาษาจาวาให้เป็นโปรแกรม dtex.nbm โดยในส่วนนี้ทำการออกแบบให้มีการเลือกเอกเซอร์ไซเซอร์ในหนวดหมู่ต่างๆ ที่หลากหลายบนแท็บ (Tap) ได้แก่ หน่วยประมวลผล (CPU) หน่วยความจำหลัก (Memory) ระบบไฟล์ (File system) จาวา (Java) พีเอชพี (PHP) ไพธอน (Python) และรูบี้ (Ruby) ซึ่งในแต่ละแท็บมีแถบเลื่อน (Slider bar) เพื่อทำการปรับ (Tune) เอกเซอร์ไซเซอร์ มีปุ่มเล่น (Play) เพื่อให้เอกเซอร์ไซเซอร์ทำงาน ปุ่มหยุด (Stop) เพื่อให้เอกเซอร์ไซเซอร์หยุดทำงาน ในส่วนด้านล่างคือส่วนที่ใช้แสดงผลขณะที่เอกเซอร์ไซเซอร์ทำงาน บอกสถานะขณะทำงาน ภาพที่ 3.7 คือจียูไอของชุดเครื่องมือดีทีเอ็กซ์ ซึ่งรวมส่วนประกอบของทุกโมดูลไว้บนเน็ตบีนส์



ภาพที่ 3.6 แสดงจียูไอของเอกเซอร์ไซเซอร์



ภาพที่ 3.7 แสดงจียูไอของชุดเครื่องมือดีทีเอ็กซ์

บทที่ 4

การทดสอบและผลการทดลอง

การทดสอบทำโดยนำชุดเครื่องมือดีทีเอ็กซ์และโปรแกรมต่างๆ ที่แสดงไว้ในบทที่ 3 มาทำการติดตั้งในแต่ละส่วนและพัฒนาชุดเครื่องมือและโปรแกรมในส่วนต่างๆ ตามที่ได้ออกแบบไว้ ทำการพัฒนาในส่วนของจียูไอบนเน็ตบีนส์ ติดตั้งโมดูลดีสคริปต์ตามปลั๊กอินเน็ตบีนส์ดีเทรสจียูไอ พัฒนาในส่วนของเอกเซอร์ไซเซอร์ เมื่อติดตั้งเสร็จทำการทดสอบโดยทำการทดลองใช้งานแบบมือ (Manual) ด้วยวิธีการใช้งานดีเทรสแบบปกติด้วยคำสั่ง (Command line) เทียบกับการใช้งานดีเทรสด้วยชุดเครื่องมือดีทีเอ็กซ์ เพื่อให้ได้ทราบถึงความแตกต่างในการใช้งานว่าแบบใดใช้งานง่ายกว่า จากนั้นทำการทดสอบโดยใช้งานชุดเครื่องมือในแต่ละแท็บของเอกเซอร์ไซเซอร์ ซึ่งการทดลองได้เลือกทดลองกับระบบย่อย 4 ระบบ ได้แก่ หน่วยประมวลผล หน่วยความจำหลัก พีเอชพี และไพธอน ด้วยการให้เอกเซอร์ไซเซอร์จำลองเหตุการณ์และใช้ดีสคริปต์ตรวจวัดในรูปแบบที่ต่างกัน

4.1 เครื่องมือ โปรแกรมและการติดตั้งโปรแกรมในการทดลอง

ในการทดลอง ใช้ซอฟต์แวร์และชุดเครื่องมือต่างๆ ดังนี้

1) VirtualBox [20] คือ ชุดโปรแกรมจำลองสภาพแวดล้อมเสมือนคอมพิวเตอร์ X86 ซึ่งปัจจุบันพัฒนาโดย Oracle Corporation โดยชุดโปรแกรมรองรับระบบปฏิบัติต่างๆ มากมาย เช่น Windows, Linux, Mac OS X, Solaris เป็นต้น โดยเมื่อติดตั้งชุดโปรแกรมเสร็จ ใช้โปรแกรมทำการจำลองสภาพแวดล้อมเสมือนคอมพิวเตอร์เพื่อลงระบบปฏิบัติการอื่น และในการทดลองได้ลงระบบปฏิบัติการ Solaris 11 ลงบน VirtualBox ซึ่งได้ติดตั้งบนระบบปฏิบัติการ Windows โดย VirtualBox ใช้ version 4.1.2

2) Oracle Solaris OS [21] คือ ระบบปฏิบัติการยูนิกซ์ ซึ่งสร้างโดย Sun Microsystems ปัจจุบันเป็นของ Oracle Corporation โดยมีสถาปัตยกรรมคือ SPARC และมีนวัตกรรมเฉพาะอยู่ในระบบปฏิบัติการมาด้วยคือ ดีเทรส ZFS และ Time Slider ซึ่งทาง Oracle ทำการพัฒนาดีเทรสใน Solaris มาตลอดและใช้งานได้ดีที่สุดในระบบปฏิบัติการโซลาริส จึงเป็นเหตุผลในการเลือกทดลองกับระบบปฏิบัติการนี้ โดยการทดลองใช้ Solaris version 11

3) Netbeans + JDK [15] คือ ชุดโปรแกรมไอ ดี อี (IDE-Integrated Development Environment) ที่ใช้ทำการพัฒนาทั้งโปรแกรมภาษาจาวา จาวาสคริปต์ พีเอชพี ไพธอน และอื่นๆ JDK (Java Development Kit) คือ ชุดคำสั่งในการพัฒนาโปรแกรมภาษาจาวา โดยการทดลองนี้ใช้ version 7.0.1

4) Netbeans Dtrace GUI Plug-in [11] คือ โปรแกรมปลั๊กอิน ซึ่งนำมาติดตั้งบนเน็ตบีนส์ โดยใช้ version 1.2

ทำการติดตั้งโปรแกรมดังนี้

1) ติดตั้งโปรแกรม VirtualBox บนระบบปฏิบัติการ Windows 7
2) ติดตั้งระบบปฏิบัติการ Solaris บน VirtualBox
3) เปิด Solaris ทำการติดตั้งโปรแกรม Netbeans + JDK
4) ติดตั้งปลั๊กอิน Netbeans DTrace GUI บน Netbeans

4.2 การพัฒนาชุดโปรแกรมดีทีเอ็กซ์

การพัฒนาชุดโปรแกรมดีทีเอ็กซ์ (DTEx - DTrace Exerciser) โดยภาพรวมได้องค์ประกอบ 3 ส่วนมารวมกัน คือ ดีเทรสจียูไอปลั๊กอิน เน็ตบีนส์ และเอกเซอร์ไซเซอร์ ซึ่ง 2 ส่วนแรกได้รวบรวมจากเว็บไซต์ทางด้านนี้ [9] ทำการพัฒนาในส่วนเอกเซอร์ไซเซอร์ โดยใช้เครื่องมือสร้างจียูไอด้วยภาษาจาวา

4.3 เปรียบเทียบการใช้งานด้วยมือกับชุดเครื่องมือ

เมื่อทำการติดตั้งชุดเครื่องมือดีทีเอ็กซ์แล้ว ได้ทำการทดลองเปรียบเทียบการใช้งานดีเทรสด้วยมือ (Manual) กับการใช้งานชุดเครื่องมือ ดังนี้

4.3.1 ใช้งานดีเทรสด้วยมือ

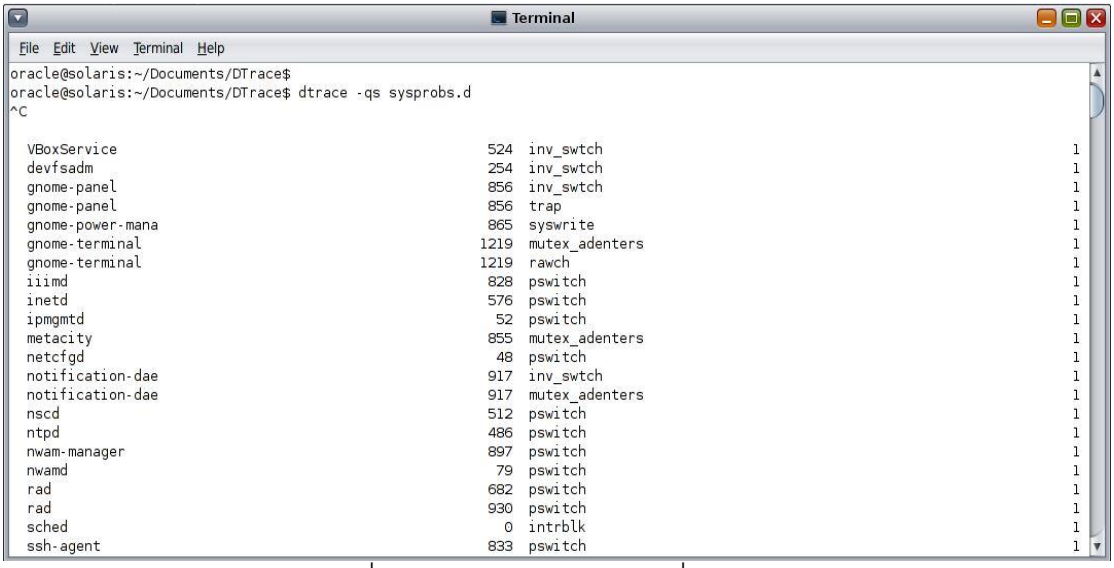
1) สร้างดีสคริปต์ชื่อ sysprobes.d [11] ด้วยการเขียนโปรแกรมดังภาพที่ 4.1

```
sysinfo:::
/(pid != $pid) && (probename != “readch”) && (probename != “writch”)/
{
        @a[execname, pid, probename] = sum(arg0);
        @b[execname, pid, probename] = sum(arg0);
}
```

รูปที่ 4.1 ดีสคริปต์ sysprobes.d

ดีสคริปต์นี้นำมาจากดีสคริปต์ของเน็ตบีนส์ปลั๊กอินจียูไอเพื่อให้การทดลองด้วยมือกับใช้เครื่องมือใช้ดีสคริปต์ที่เหมือนกัน นำมาสร้างภายนอกและเรียกใช้ผ่านเทอร์มินอล เพื่อให้ได้ผลลัพธ์ที่ใกล้เคียงกัน

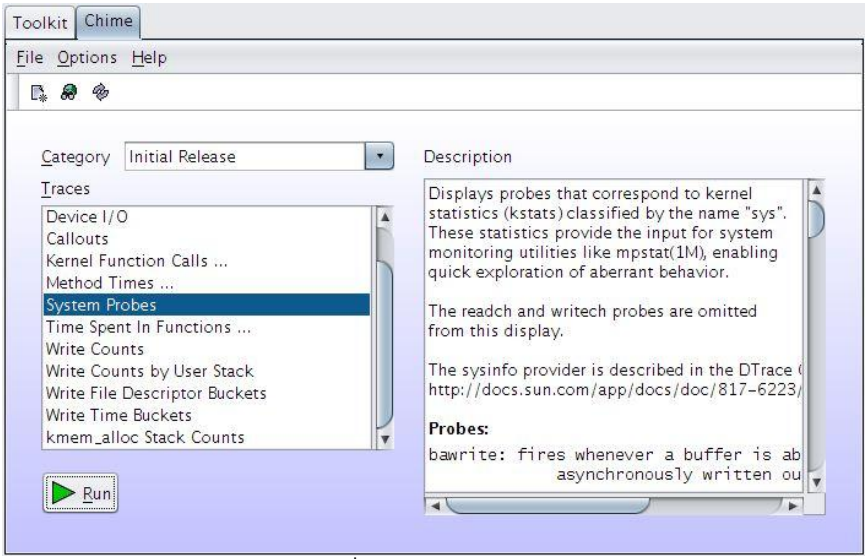
2) เรียกเทอร์มินอลและไปยังไดเร็กทอรี่ที่ดีสคริปต์อยู่ ใช้คำสั่งดังภาพที่ 4.2



ภาพที่ 4.2 ผลลัพธ์ของการใช้คำสั่งดีสคริปต์

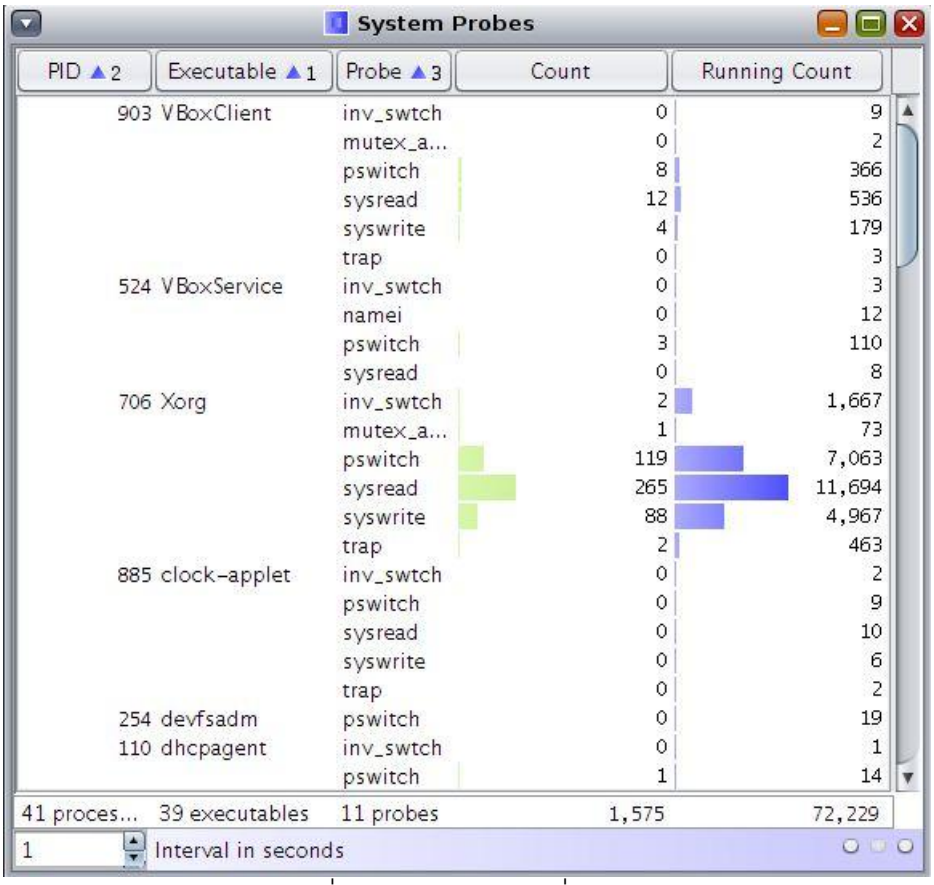
4.3.2 ใช้งานดีเทรสด้วยชุดเครื่องมือดีทีเอ็กซ์

1) เปิดโปรแกรมเน็ตบีนส์ เลือกโมดูลดีเทรส เลือกใช้ไชม์ และเลือกดีสคริปต์ System Probes ซึ่งดีสคริปต์นี้เขียนเหมือนกันกับ sysprobes.d จากนั้นกดปุ่มเล่น (Run) เพื่อใช้งานดีสคริปต์ดังภาพที่ 4.3



ภาพที่ 4.3 จียูไอของ ไชม์

2) ผลลัพธ์ที่ได้จากการใช้ไซม์ ดังภาพที่ 4.4



ภาพที่ 4.4 แสดงผลลัพธ์เมื่อใช้ ไซม์

โดยไซม์แสดงผลลัพธ์แบบทันกาล (Real time) และแบ่งตัวแปรเป็นคอลัมน์ ซึ่งมีการแสดงผลเป็นกราฟได้ตลอดตามเวลา แต่ในช่วงที่ใช้ด้วยมือ ดีเทรสยังไม่แสดงผลลัพธ์ใดๆ จนกว่าผู้ใช้หยุดดีเทรสด้วยการกด Ctrl + c แล้วดีเทรสจึงแสดงผลลัพธ์ออกมา ซึ่งผลลัพธ์ที่ได้ ไม่ได้แสดงผลทางกราฟิกและเรียงผลลัพธ์เหมือนในไซม์

จากการทดลองเปรียบเทียบการใช้งานดีเทรสด้วยมือกับการใช้เครื่องมือ ผู้ใช้ที่ใช้งานแบบด้วยมือต้องมีความรู้หลายด้าน เช่น คำสั่งยูนิกซ์ คำสั่งดีเทรส เข้าใจโครงสร้างของระบบปฏิบัติการยูนิกซ์ รวมทั้งโซลาริส สำหรับการใช้งานผู้ใช้ต้องทราบว่ามีดีสคริปต์อะไร อยู่ที่ไดเร็คทอรี่ไหน เรียกใช้งานอย่างไร เป็นต้น ในการใช้งานผ่านชุดเครื่องมือดีทีเอ็กซ์นั้น ผู้ใช้ไม่จำเป็นต้องทราบถึงคำสั่งของยูนิกซ์และคำสั่งของดีเทรส ซึ่งทำให้ผู้ใช้ที่ยังไม่มีความรู้ด้านนี้สามารถใช้งานดีเทรสได้ อีกทั้งความสะดวกในการใช้งาน สามารถเลือกใช้งานดีสคริปต์พร้อมคำอธิบายในการใช้งาน งานวิจัยนี้ทำการเปรียบเทียบการใช้งานด้วยมือกับชุดเครื่องมือดีทีเอ็กซ์ดังตารางที่ 4.1

ตารางที่ 4.1 เปรียบเทียบการใช้งานด้วยมือกับชุดเครื่องมือ

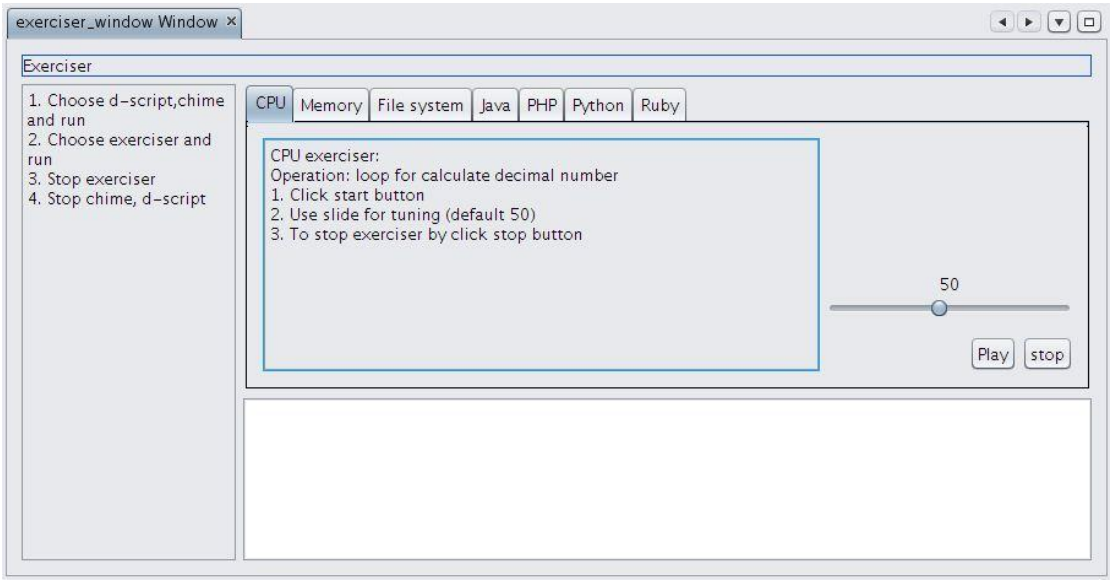
คุณลักษณะ	ใช้งานด้วยมือ	ใช้ชุดเครื่องมือดีทีเอ็กซ์
ทราบคำสั่งดีเทรส	ใช่	ไม่
ทราบคำสั่งยูนิกส์	ใช่	ไม่
มีดีสคริปต์พร้อมใช้	ไม่	มี
มีจียูไอที่ใช้ง่าย	ไม่	มี
แสดงผลกราฟิก	ไม่	ได้

4.4 การทดลอง

การทดลองทำกับระบบย่อย 4 ระบบ ได้แก่ หน่วยประมวลผล หน่วยความจำหลัก พีเอชพี และไพธอน

4.4.1 หน่วยประมวลผล

ในการทดลองทำการทดสอบต่อระบบหน่วยประมวลผลนั้น จุดประสงค์คือให้เอกเซอร์ไซเซอร์ทำให้หน่วยประมวลผลทำงานหนัก และให้ดีเทรสเก็บข้อมูลเพื่อมาวิเคราะห์หาปัญหา



ภาพที่ 4.5 แสดงจียูไอของเอกเซอร์ไซเซอร์ในส่วนของหน่วยประมวลผล

ในการทดลอง เลือกแทบ CPU ดังภาพที่ 4.5 เพื่อทำการรบกวนระบบ โดยผู้ใช้สามารถเลือกปุ่มเลื่อนเพื่อเพิ่มหรือลดความเร็วในการทำงานของเอกเซอร์ไซเซอร์ สำหรับการทำงานของ

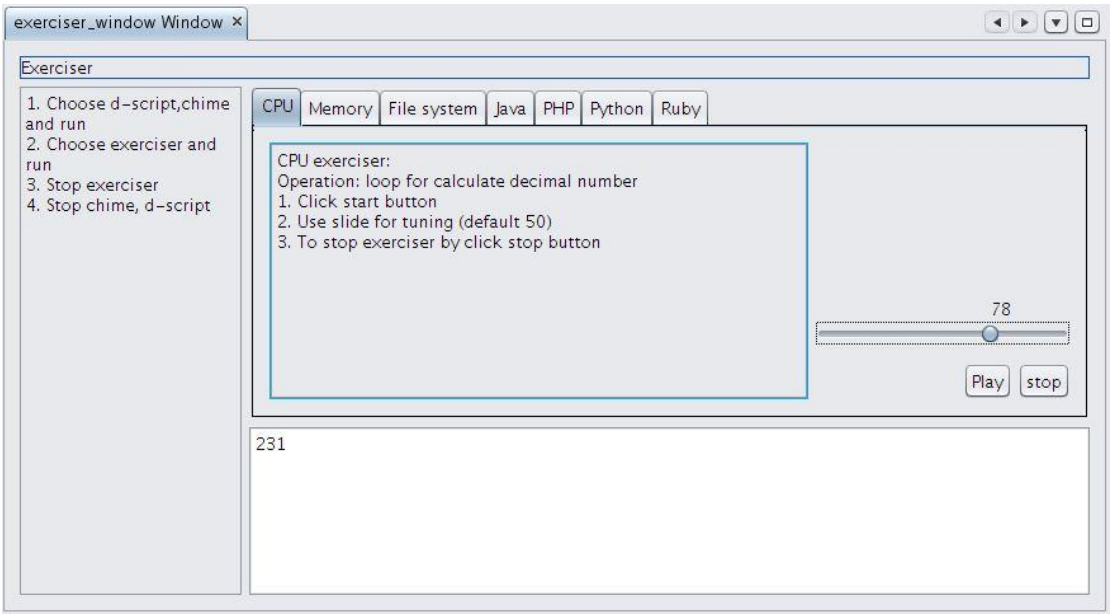
เอกเซอร์ไซเซอร์ ดังภาพที่ 4.6 คือ โปรแกรม [22] เมื่อทำการกดปุ่มเล่น โปรแกรมสั่งให้ timer [23] ทำงานไปเรียกโปรแกรมในภาพที่ 4.7 ทำการคำนวณเรียกคลาส BigDecimal เพื่อคำนวณเลขยกกำลังสิบ (10^x) ทุกครั้งที่มี timer นับเวลาและหน่วยประมวลผลคำนวณทำการประมวลในทุกครั้ง ซึ่งตัวแปรที่ทำให้ timer นับช้าหรือเร็วนั้นมาจากตัวแปร delay ในบรรทัดที่ 472 ซึ่งได้จากการที่กำหนดค่า maxDelay โดยให้เท่ากับ 10000 และ timer มีหน่วยเป็น millisecond [24] ดังนั้น ค่าของ delay เมื่อนำค่าของ slider คือ 1-100 นำมาหาร maxDelay ได้ 10000-100 คือ 10 วินาที ถึง 0.01 วินาที ซึ่งยิ่ง delay มากทำให้ timer ช้า เมื่อทำการกดปุ่มเล่น เอกเซอร์ไซเซอร์ทำงานโดยแสดงผลลัพธ์ที่ด้านล่างดังภาพ 4.8 โดยแสดงการนับเลข โดยความเร็วในการนับเลขขึ้นอยู่กับการเลือกแถบเลื่อนโดยการนับหนึ่งครั้งคือการทำงานของ timer ในแต่ละครั้ง

```
468  private void jButton1_CPU_playActionPerformed(java.awt.event.ActionEvent evt) {
469  // TODO add your handling code here:
470      menu = "CPU";
471      //Calculate initial delay
472      delay = maxDelay / jSlider1_CPU.getValue();
473      //initial value
474      jTextPane2.setText("0");
475      amount = new BigDecimal(0);
476      //initial timer
477      if (timer == null) {
478          timer = new Timer(delay, timerListener);
479      }
480      //Set delay to timer
481      timer.setInitialDelay(delay * wait);
482      timer.start();
483  }
484
```

ภาพที่ 4.6 โปรแกรมของเอกเซอร์ไซเซอร์ของหน่วยประมวลผล

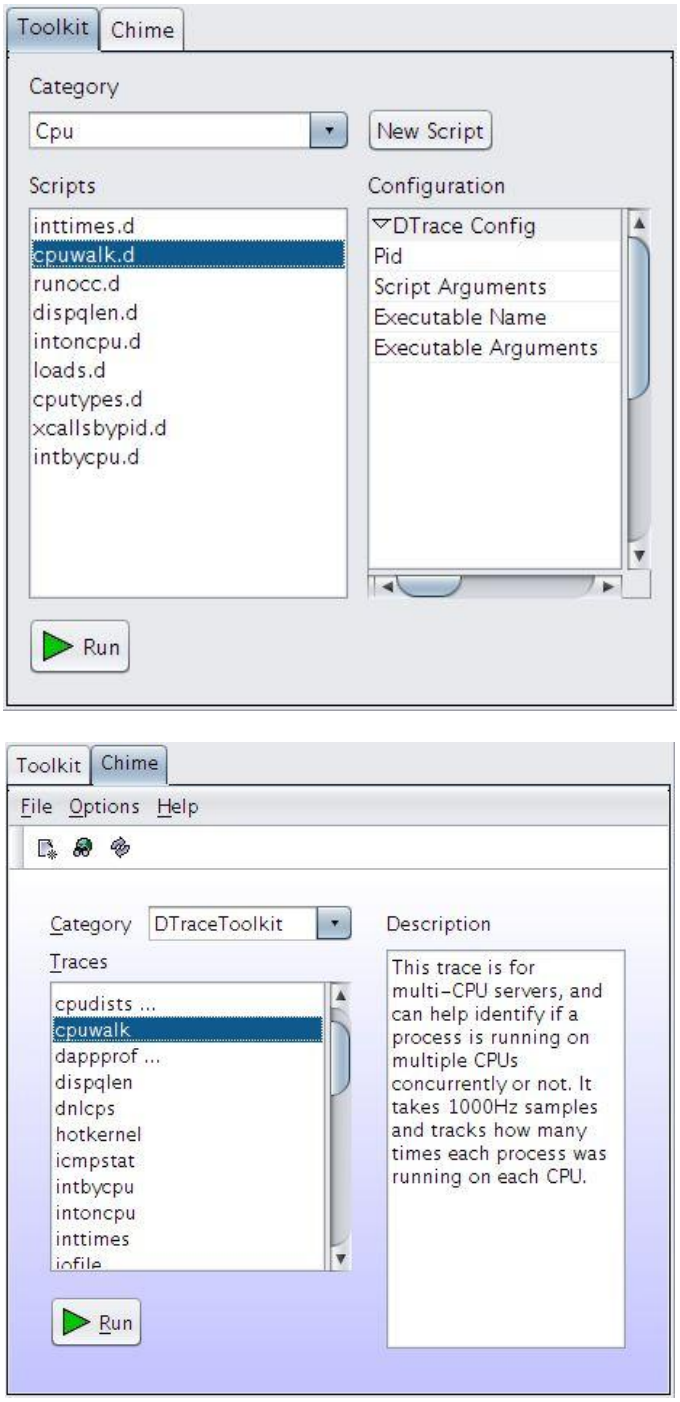
```
if ("CPU".equals(menu)) {
    int i = Integer.parseInt(jTextPane2.getText());
    i++;
    jTextPane2.setText(String.valueOf(i));
    amount.add(new BigDecimal(1000));
}
```

ภาพที่ 4.7 โปรแกรมที่เอกเซอร์ไซเซอร์ทำให้หน่วยประมวลผลทำการประมวลผล



ภาพที่ 4.8 แสดงเอกเซอร์ไซเซอร์ขณะใช้งาน

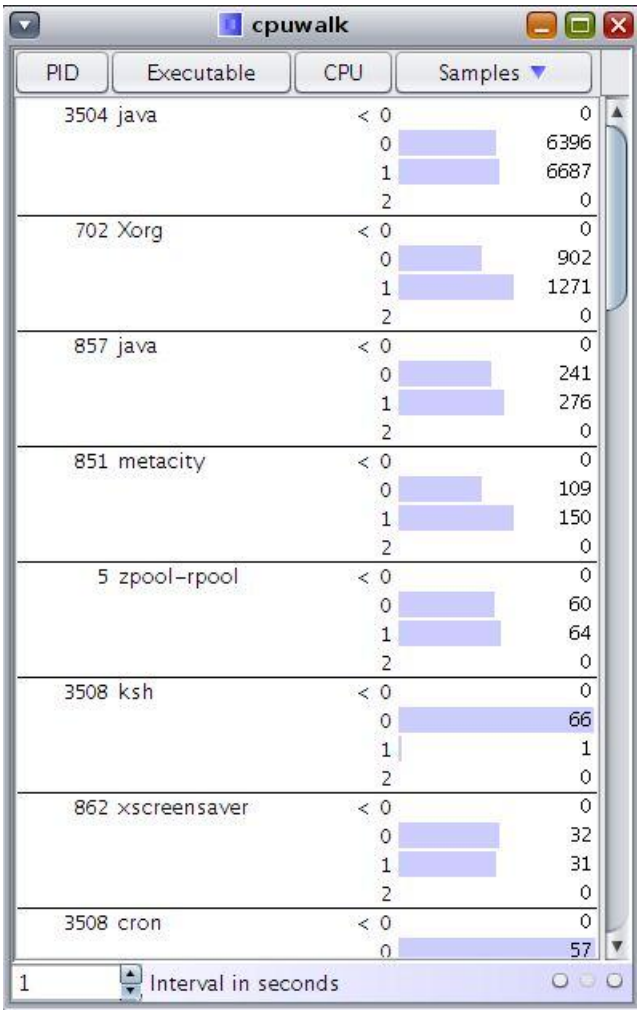
ขั้นต่อมา ทำการเลือกแทบดีเทรส ดังภาพที่ 4.9 และทำการเลือกดีสคริปต์ cpuwalk.d [11] และไซม์ cpuwalk [11] เมื่อทำการเปิดใช้ดีเทรสและแสดงผลของโปรเซสไอดี (PID) ดังภาพที่ 4.10 ซึ่งการทดลองนี้ใช้หน่วยประมวลผลแบบแกนคู่ (Dual Core) โดยผลการทดลองพบว่าจาวา PID ที่ 3504 เป็น process ที่ใช้ทรัพยากรของหน่วยประมวลผลมากที่สุด โดยดูจากภาพรวมของทั้งหน่วยประมวลผล 0 และ 1 โดยดีเทรสเก็บข้อมูลได้รวมกัน 787 + 943 = 1730 ซึ่งมากที่สุดเมื่อเทียบกับโปรเซสอื่นเพราะเอกเซอร์ไซเซอร์ถูกเขียนด้วยภาษาจาวา ภาพที่ 4.11 คือผลของการใช้ไซม์ โดยแสดงผลกราฟิก จาวามีการใช้ทรัพยากรมากที่สุด



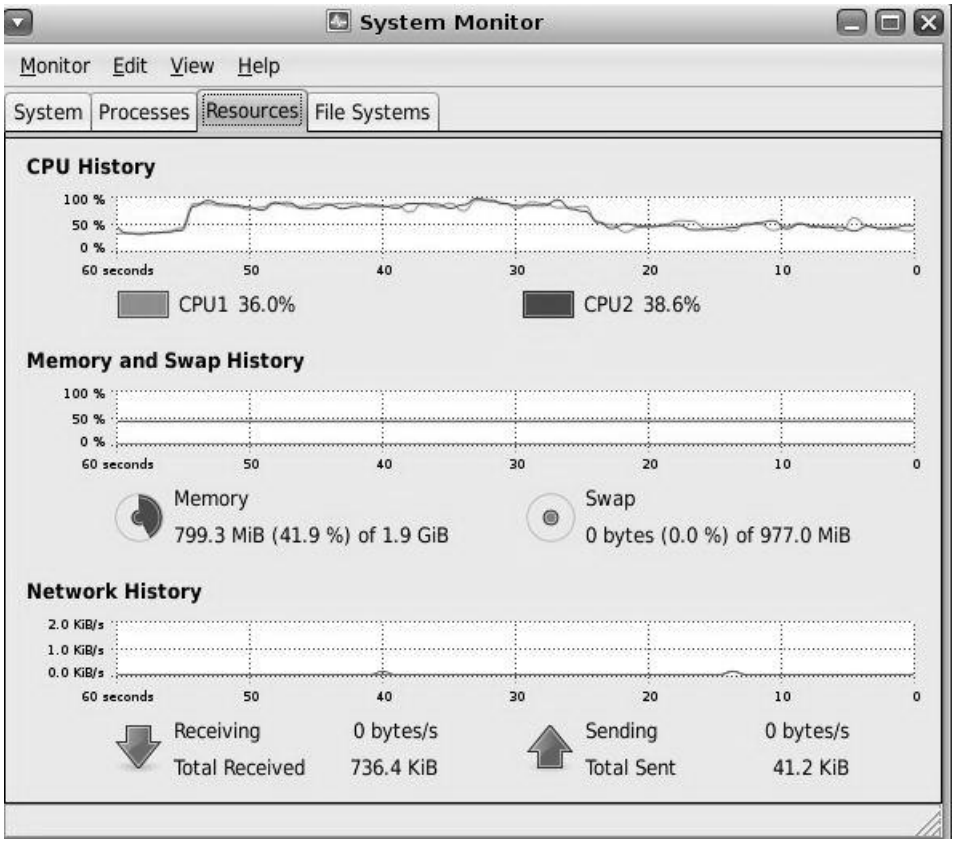
ภาพที่ 4.9 แสดงการเลือกดีสคริปต์และไซม์ของหน่วยประมวลผล

```
PID: 702      CMD: Xorg

      value  ------------- Distribution ------------- count
        < 0 |                                       0
          0 |@@@@@@@@@@@@@@@@@@@@                   243
          1 |@@@@@@@@@@@@@@@@@@@@@@                 256
          2 |                                       0

PID: 3504     CMD: java

      value  ------------- Distribution ------------- count
        < 0 |                                       0
          0 |@@@@@@@@@@@@@@@@@@@                    787
          1 |@@@@@@@@@@@@@@@@@@@@@@@                943
          2 |                                       0
```

ภาพที่ 4.10 ผลการใช้ดีเทรสแสดงสมรรถนะของหน่วยประมวลผล



ภาพที่ 4.11 แสดงผลการใช้ไซม์

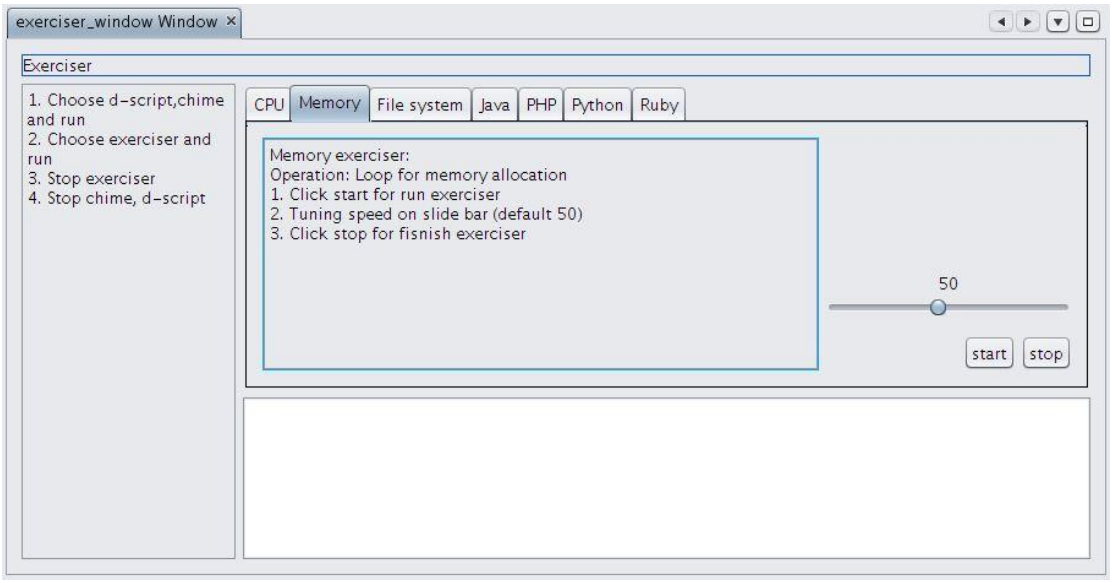


ภาพที่ 4.12 ผลจากโปรแกรม System Monitor

ทดลองใช้งาน System Monitor ซึ่งเป็นโปรแกรมมาตรฐานของระบบปฏิบัติการโซลาริส ดังภาพที่ 4.12 พบว่าโปรแกรมแสดงผลของการใช้หน่วยประมวลผลโดยรวม แสดงผลได้ไม่ละเอียดเมื่อเปรียบเทียบกับไซม์ โดยแสดงผลให้เห็นว่าโดยรวมมีการใช้งานหน่วยความจำในระดับไหนโดยแสดงเป็นกราฟ โดยไม่ได้บอกว่าโปรเซสใดที่ใช้ทรัพยากรของหน่วยประมวลผลมากจำนวนเท่าใด ซึ่งดีเทรสบอกได้ว่าที่โปรเซสใดมีการใช้ทรัพยากรของหน่วยประมวลผลเท่าใด โดยในการทดลองนี้ได้เปรียบเทียบและแสดงให้เห็นถึงผลลัพธ์เมื่อใช้ดีเทรสว่าแสดงผลลัพธ์ได้ลึกและละเอียดกว่า

4.4.2 หน่วยความจำหลัก

ในการทดลองทำการทดสอบต่อหน่วยความจำหลักนั้น จุดประสงค์คือให้หน่วยความจำหลักมีการจองหน่วยความจำเพิ่มขึ้นมาก โดยการจำลองให้เอกเซอร์ไซเซอร์ทำการจองหน่วยความจำเพิ่มขึ้นเรื่อยๆ และทำการให้ดีเทรสเก็บข้อมูลเพื่อมาวิเคราะห์หาปัญหาของหน่วยความจำหลัก



ภาพที่ 4.13 แสดงจียูไอของเอกเซอร์ไซเซอร์ของหน่วยความจำหลัก

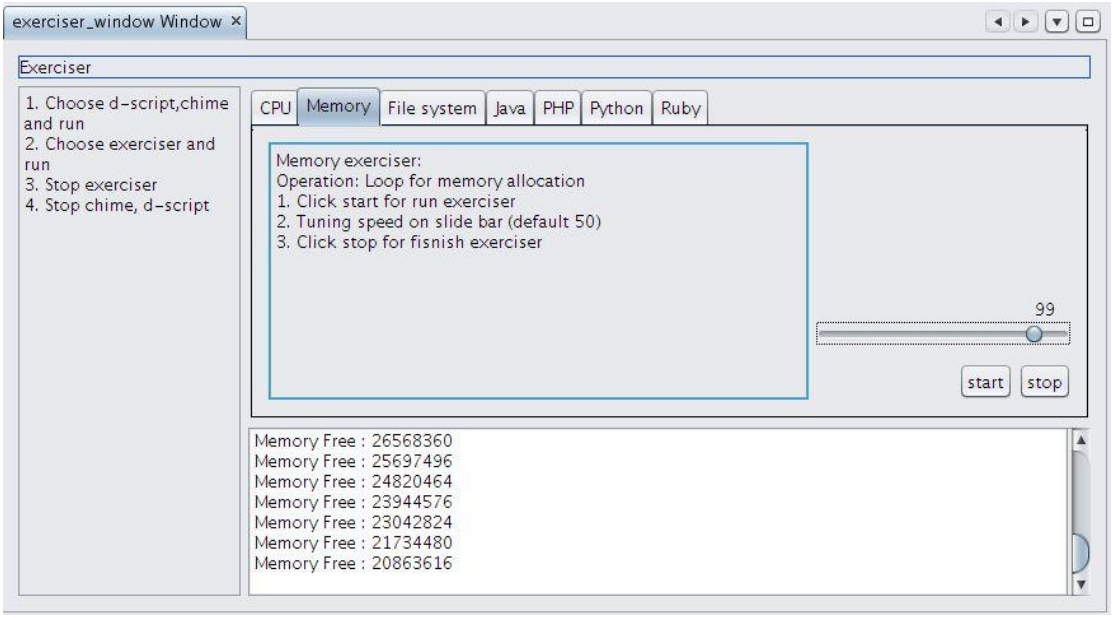
ในการทดลอง เลือกเอกเซอร์ไซเซอร์หน่วยความจำที่แทบ Memory ดังภาพที่ 4.13 โดยเลือก slider เพื่อปรับความเร็วในการทำงานของเอกเซอร์ไซเซอร์ โดยเมื่อทำการกดปุ่มเล่นโปรแกรมทำการเรียก timer และมีรูปแบบโปรแกรมและตัวแปรเหมือนของหน่วยประมวลผล เพราะใช้ timer เดียวกันทั้งโปรแกรม ดังภาพที่ 4.14 แสดงโปรแกรมเมื่อมีการกดปุ่มเล่น แต่โปรแกรมเรียก ActionListener [25] ดังภาพที่ 4.15 ให้โปรแกรมมีตัวแปร lst ซึ่งคือ LinkedList ในการเพิ่ม Object() จองพื้นที่ในหน่วยความจำทุกครั้งที่มีการนับ timer โดยแสดงผลลัพธ์บอกถึงว่าหน่วยความจำหลักมีพื้นที่ว่างเท่าใด โดยพื้นที่ว่างลดลงตามจำนวนของ timer เมื่อใช้งานเอกเซอร์ไซเซอร์ ผลขณะทำงานที่ได้ดังภาพที่ 4.16 โดยส่วนแสดงผลแสดงตัวเลขที่ว่างของหน่วยความจำหลัก ซึ่งมีขนาดลดลงตามรอบของ timer และเลือกใช้ความเร็วได้ตามแถบเลื่อนด้านบน

```
485  private void jButton1_memory_startActionPerformed(java.awt.event.ActionEvent evt) {
486  // TODO add your handling code here:
487      menu = "Memory";
488      //Calculate initial delay
489      delay = maxDelay / jSlider1_Memory.getValue();
490      //initial value
491      jTextPane2.setText("");
492      lst = new LinkedList();
493      //initial timer
494      if (timer == null) {
495          timer = new Timer(delay, timerListener);
496      }
497      //Set delay to timer
498      timer.setInitialDelay(delay * wait);
499      timer.start();
500  }
```

ภาพที่ 4.14 แสดงโปรแกรมเอกเซอร์ไซเซอร์ของหน่วยความจำหลัก

```
else if ("Memory".equals(menu)) {
    jTextPane2.setText(jTextPane2.getText() +
      "Memory Free : " + Runtime.getRuntime().freeMemory() + "\n");
    lst.add(new Object());
}
```

ภาพที่ 4.15 แสดงโปรแกรมที่ทำการจองพื้นที่ในหน่วยความจำหลัก



ภาพที่ 4.16 แสดงเอกเซอร์ไซเซอร์ขณะใช้งาน

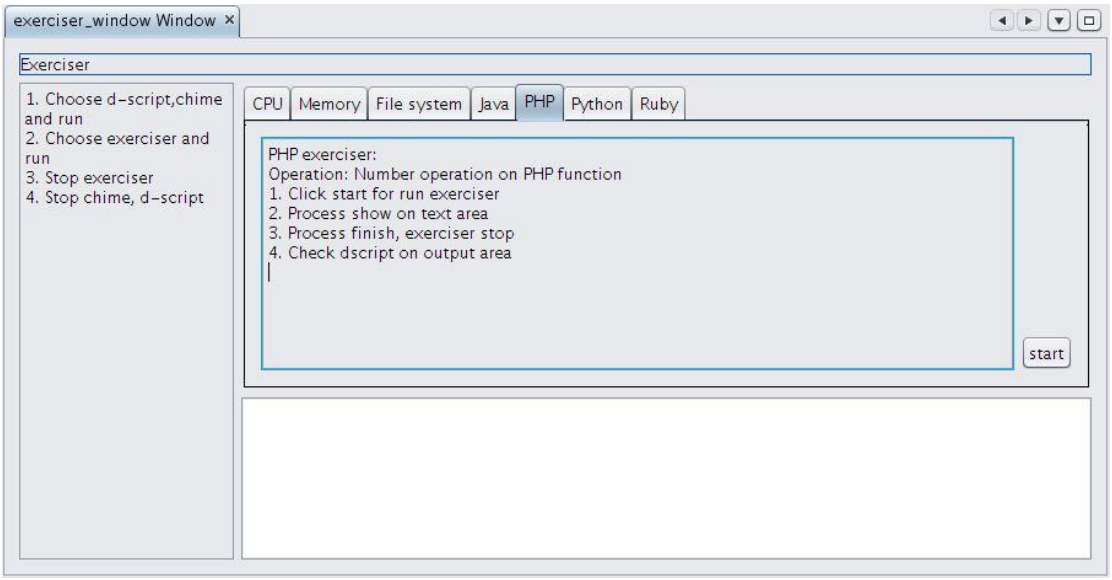
โดยเลือกใช้ดีสคริปต์ในกลุ่มของหน่วยความจำหลัก vmbypid.d [11] ที่แสดงการใช้หน่วยความจำหลัก ภาพที่ 4.17 มีโปรแกรมจาวาซึ่งเป็นเอกเซอร์ไซเซอร์ PID 1137 ใช้หน่วยความจำมากที่สุด

```
java     2475              as_fault      217
java     1137             cow_fault      296
Xorg      707              as_fault      314
java     1137                  zfod     1174
java     1137              as_fault     1257
java     2472                  zfod     5032
java     2472              as_fault     7463
java     1137            prot_fault    32045
```

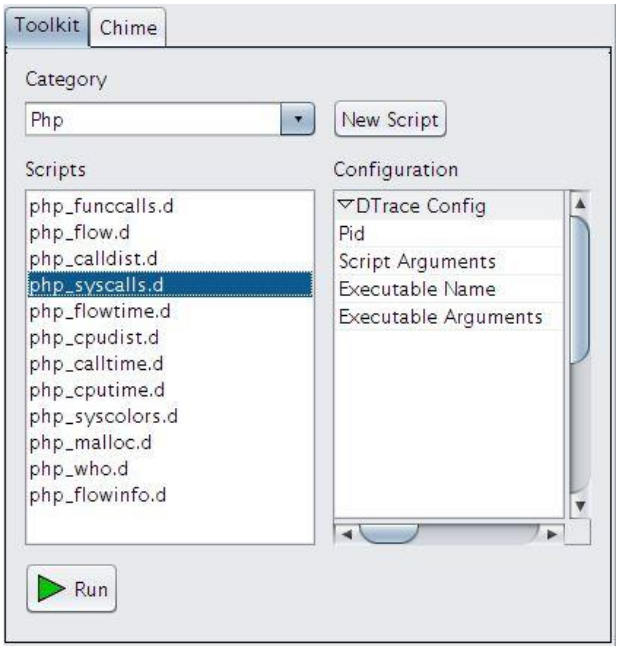
ภาพที่ 4.17 การใช้ดีเทรสตรวจดูหน่วยความจำหลัก

4.4.3 พีเอชพี

ในการทดลองทำการทดสอบต่อโปรแกรมที่พัฒนาด้วยภาษาพีเอชพีนั้น จุดประสงค์คือให้โปรแกรมภาษาพีเอชพีมีการเรียก syscall มาก โดยให้เอกเซอร์ไซเซอร์จำลองโปรแกรมภาษาพีเอชพี



ภาพที่ 4.18 จียูไอของเอกเซอร์ไซเซอร์พีเอชพี



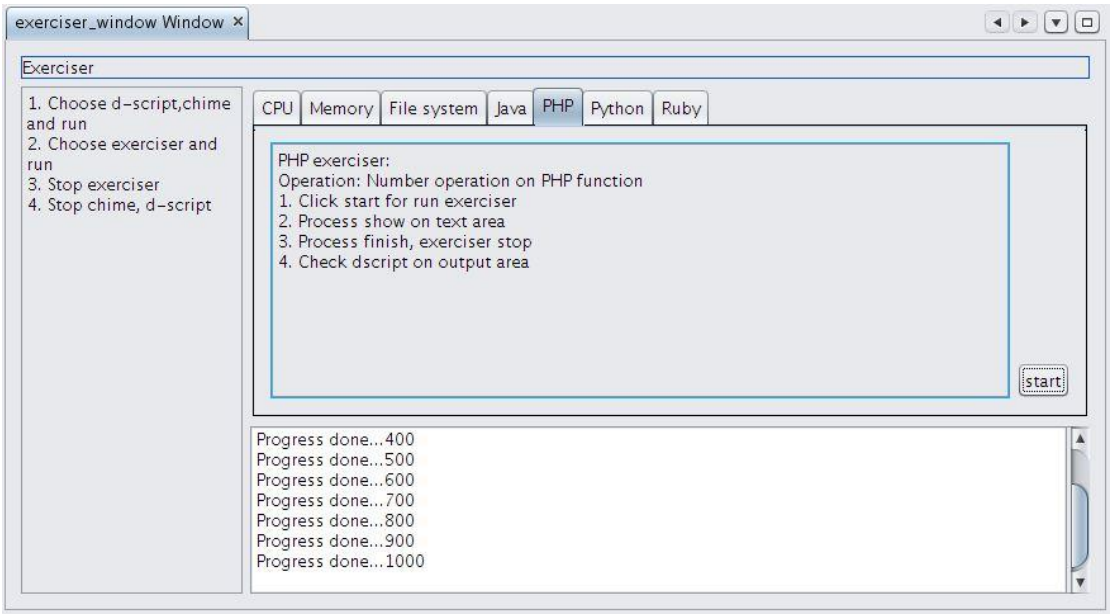
ภาพที่ 4.19 แสดงการเลือกดีสคริปต์ของพีเอชพี

ทำการทดลองโดย เลือกเอกเซอร์ไซเซอร์พีเอชพีดังภาพ 4.18 โดยต้องเปิดใช้งานดีเทรสก่อน เพื่อให้ดีเทรสรอตรวจวัดข้อมูล ดังภาพที่ 4.19 ใช้ดีสคริปต์ php_syscall.d [11] โดยดี

สคริปต์นี้ใช้ในการที่ดีเทรสทำการเก็บข้อมูลโปรเซสที่มีการเรียก syscall เปิดใช้งานโดยกดปุ่มเล่น จากนั้นจียูไอทำการเรียกโปรแกรมภาษาพีเอชพี โดยโปรแกรมดังภาพที่ 4.20 แบ่งเป็น 2 ฟังก์ชัน คือ เรียกฟังก์ชัน isprime เพื่อทำการตรวจเลขทีละครั้งตั้งแต่ 0 – 1000 โดยการตรวจเลข 1 ครั้ง ต้องเรียกฟังก์ชัน doesDivide เพื่อทำการ mod จำนวน 0 - $num ครั้งและส่งค่าที่ได้ว่าอยู่ในเลขจำนวนเฉพาะหรือไม่ เมื่อทำการตรวจเสร็จจึงตรวจเลขตัวต่อไปด้วยฟังก์ชัน isprime ทำให้ฟังก์ชัน doesDivide มีการใช้เรียกมาก และเอกเซอร์ไซเซอร์ขณะใช้งานแสดงผลดังภาพที่ 4.21

```php
<?php

$numprimes=1000;
$pArray[0]=2;
$test=3;
$num=0;

function doesDivide($x,$y)
{
        return($x % $y);
}


 function isprime($x)
 {
         global $pArray;
         global $num;
         $index=0;
         $check=1;

         while($check==1 && $index <= $num)
         {
                 if(doesDivide($x, $pArray[$index])==0){
                         $check=0;
                 }
                 else
                         $index++;
         }

         return($check);
 }

while($num<$numprimes)
{
        if(isPrime($test)==1){
                $num++;
                $pArray[$num]=$test;
                if($num%100==0){
                        printf("Progress done...%d\n",$num);
                }
        }
        $test++;
}


?>
```

ภาพที่ 4.20 แสดงโปรแกรมของเอกเซอร์ไซเซอร์พีเอชพี



ภาพที่ 4.21 แสดงเอกเซอร์ไซเซอร์ขณะใช้งาน

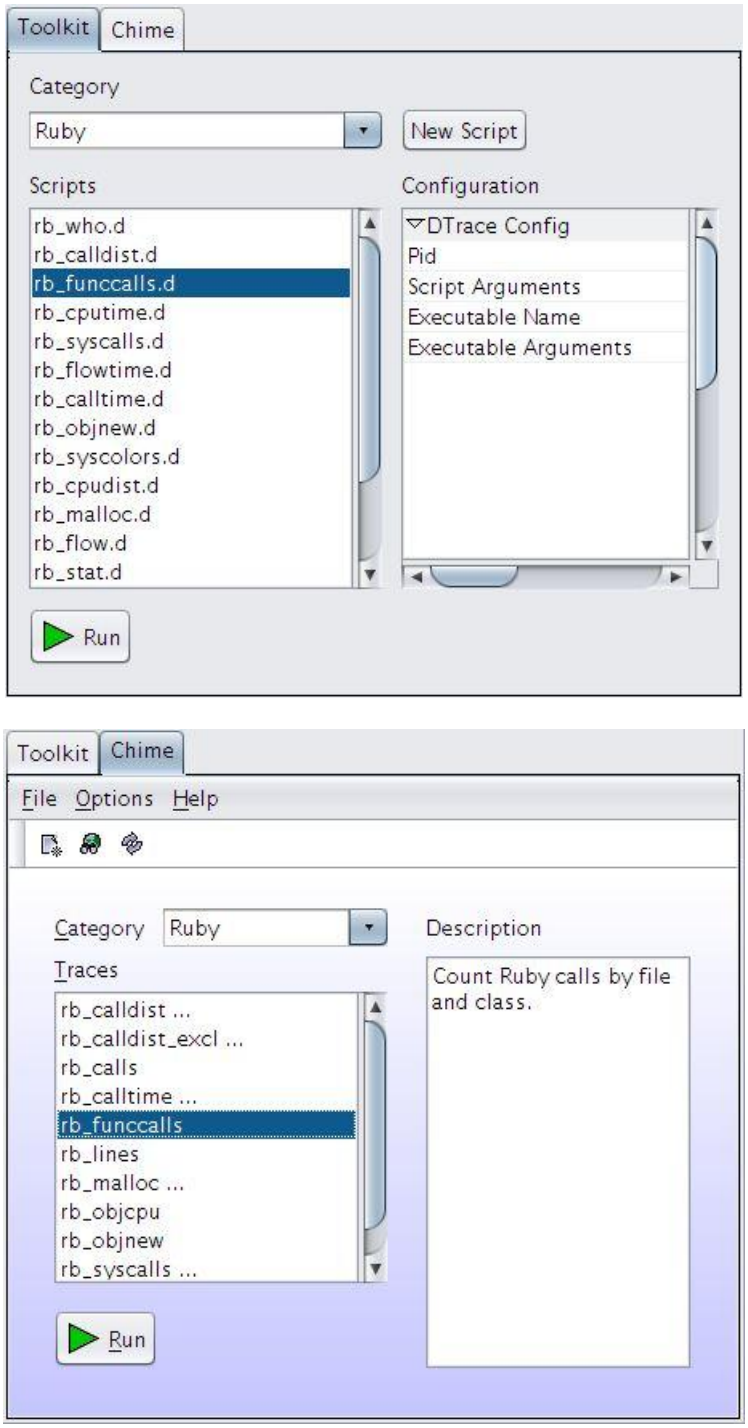
เมื่อดูผลลัพธ์ของดีเทรสดังภาพที่ 4.22 พบว่าเอกเซอร์ไซเซอร์มี PID หมายเลข 1522 มีคลาส (Class) เรียกฟังก์ชันคือ isprime จำนวน 7925 และ doesDivide จำนวน 517416 ซึ่งแสดงให้เห็นว่า doesDivide ถูกเรียกมากที่สุด ซึ่งสามารถย้อนกลับไปทำการแก้ไขโปรแกรม ให้มีประสิทธิภาพมากขึ้นได้

PID	FILE	TYPE	NAME	COUNT
1522	php_exerciser.php	func	isprime	7925
1522	php_exerciser.php	func	doesDivide	517416

ภาพที่ 4.22 ผลการใช้ดีเทรสในพีเอชพี

4.4.4 รูบี้

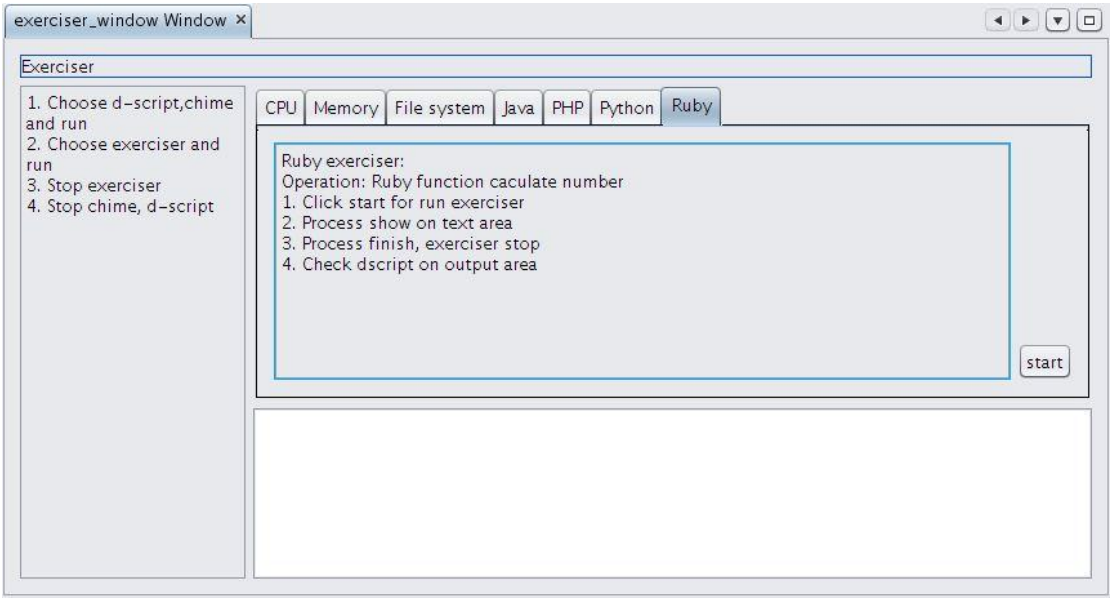
ในการทดลองทำการทดสอบต่อโปรแกรมที่พัฒนาด้วยภาษารูบี้ จุดประสงค์เพื่อให้เอกเซอร์ไซเซอร์จำลองโปรแกรมภาษารู้บี้ให้มีการเรียกเมธอดฟังก์ชัน (Function method call)



ภาพที่ 4.23 แสดงดีสคริปต์และไซม์ของรูบี้

ทำการทดลองโดยทำการเลือกดีสคริปต์ rb_funccalls.d [11] และไซม์ rb_funccalls [11] ดังภาพที่ 4.23 แล้วเปิดใช้ดีเทรสด้วยการกดปุ่มเล่น จากนั้นเลือกใช้เอกเซอร์ไซเซอร์รูบี้ ดังภาพที่ 4.24 และกดปุ่มเล่น โดยเอกเซอร์ไซเซอร์ทำงานโดยเอกเซอร์ไซเซอร์ทำงานโดยให้โปรแกรม [9] ทำการคำนวณหาจำนวนเฉพาะ 1000 ตัวแรกตั้งแต่ 1 ขึ้นไป โปรแกรมดังภาพที่ 4.25 โดยการทำงานของโปรแกรมคล้ายของพีเอชพี โดยคลาส isPrime ทำการเรียกตัวเลขแล้วเรียก

doesDivide โดยในแต่ละครั้งที่เรียก isPrime หนึ่งครั้งต้องเรียก doesDivide หลายครั้ง ซึ่งทำให้คลาสนี้มีการใช้ทรัพยากรมากกว่า



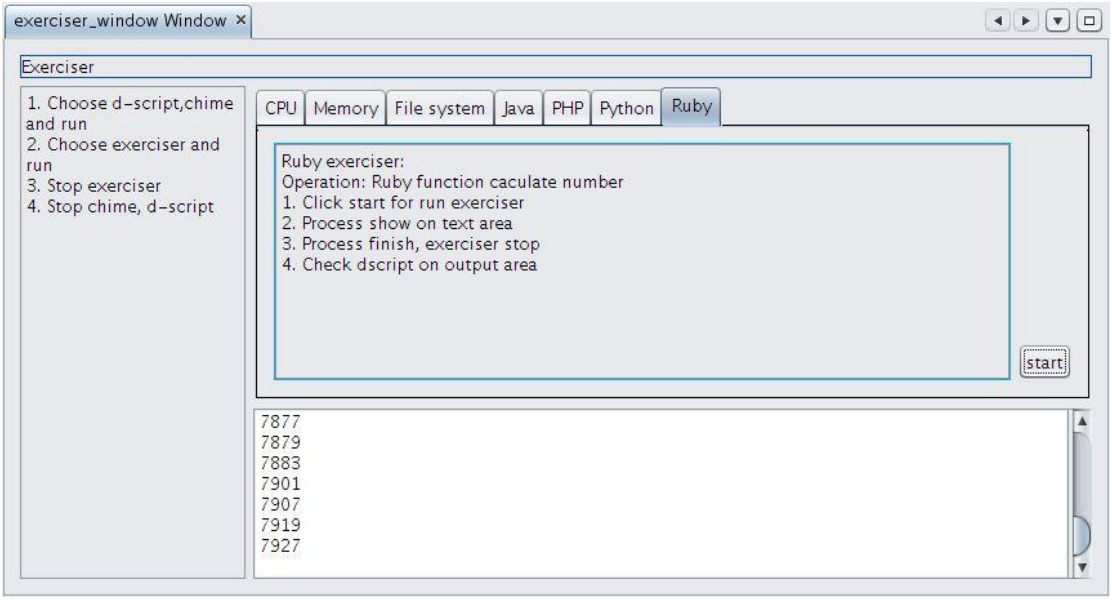
ภาพที่ 4.24 จียูไอของเอกเซอร์ไซเซอร์รูบี้

```
def doesDivide(num1,num2)
return true if num1%num2==0
return false
end

def isPrime(num)
for i in 2..num-1
        return false if doesDivide(num,i)
end
return true
end

totPrimes=0
num=3

while(totPrimes<1000)
        if isPrime(num)
                print num,"\n"
                totPrimes+=1
        end
        num+=1
end
```

ภาพที่ 4.25 แสดงโปรแกรมภาษารูบี้

เมื่อทำการใช้งานเอกเซอร์ไซเซอร์ เมื่อทำงานเสร็จได้ดังภาพที่ 4.26 แสดงจำนวนเฉพาะ ส่วนผลลัพธ์ที่ได้ของดีเทรสดังในภาพที่ 4.27 แสดงให้เห็นถึงฟังก์ชันของโปรแกรมเอกเซอร์ไซเซอร์รูบี้ทำงานอยู่ในระบบปฏิบัติการโซลาริส โดยผลลัพธ์ที่ได้คือ พบฟังก์ชันของรูบี้ doesDivide ทำงานมากกว่า isPrime และตัวแปรต่างๆ มีการเรียกเท่าใด นอกจากนี้ไซม์แสดงผลเป็นกราฟิก

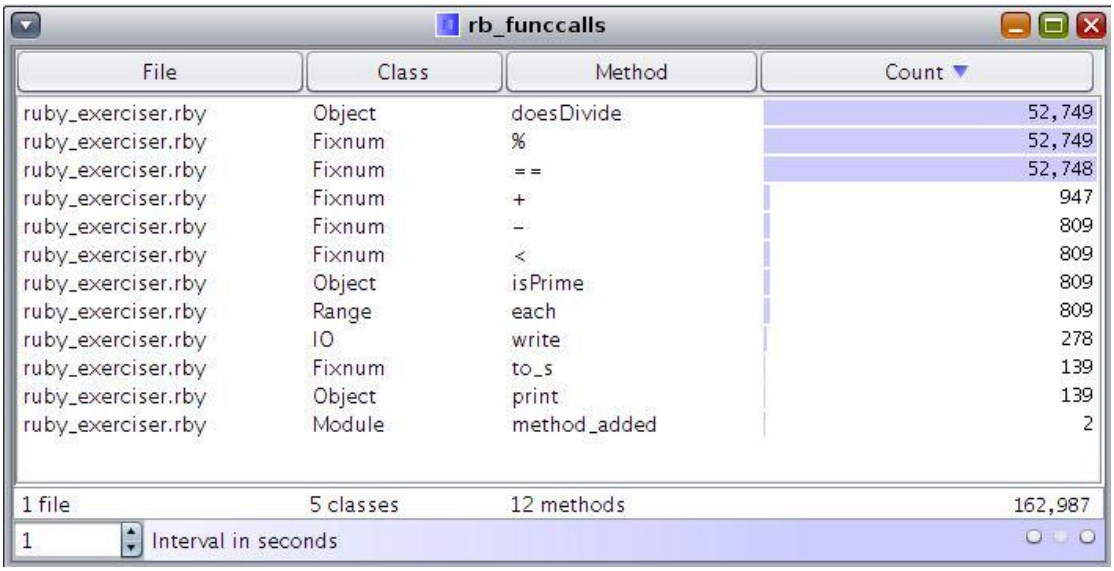
ดังภาพที่ 4.28 โดยผลลัพธ์ที่ได้คือพบโปรแกรมเอกเซอร์ไซเซอร์รูบี้แสดงว่าคลาสใดตัวแปรใดที่มีการเรียกฟังก์ชันเท่าใด ซึ่งเห็นได้ว่า doesDivide มีการเรียกใช้งานมากกว่า isPrime และโดยไซม์แสดงให้เห็นโดยละเอียดถึงในโปรแกรม ส่วนการใช้งานไซม์คือ เมื่อใช้ดีสคริปต์จะต้องรอให้เอกเซอร์ไซเซอร์ทำงานเสร็จและหยุดโปรแกรมก่อนจึงแสดงผลได้ ในการทดลองที่ดีเทรสมีจำนวนผลลัพธ์ไม่เท่ากับไซม์เนื่องจากต้องกด Ctrl + c เพื่อให้ดีเทรสหยุดการตรวจวัดก่อนถึงแสดงผลได้ แต่เมื่อใช้ไซม์สามารถแสดงผลได้แบบทันกาลในขณะที่โปรแกรมยังทำงานอยู่



ภาพที่ 4.26 แสดงเอกเซอร์ไซเซอร์ขณะใช้งาน

FILE	CLASS	METHOD	CALLS
ruby_exerciser.rby	Fixnum	to_s	13
ruby_exerciser.rby	Object	print	13
ruby_exerciser.rby	IO	write	26
ruby_exerciser.rby	Fixnum	-	92
ruby_exerciser.rby	Fixnum	<	92
ruby_exerciser.rby	Object	isPrime	92
ruby_exerciser.rby	Range	each	92
ruby_exerciser.rby	Fixnum	+	105
ruby_exerciser.rby	Fixnum	==	10466
ruby_exerciser.rby	Object	doesDivide	10466
ruby_exerciser.rby	Fixnum	%	10467

ภาพที่ 4.27 ผลของดีเทรสสำหรับรูบี้



ภาพที่ 4.28 ผลการใช้ไซม์ของรูบี้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยชิ้นนี้ได้ทำการศึกษาถึงการใช้งาน ทฤษฎี รวมทั้งส่วนประกอบต่างๆ ของดีเทรส เพื่อนำไปใช้งานพบว่า การใช้งานและทำความเข้าใจดีเทรสนั้นค่อนข้างยาก รวมถึงผลลัพธ์ที่ดีเทรสแสดงออกมายังอยู่ในรูปแบบที่เข้าใจยาก โดยในการใช้งานดีเทรสผู้ใช้ต้องมีความเข้าใจในหลายด้าน ทั้งระบบปฏิบัติการยูนิกส์และดีเทรส เพื่อให้การศึกษาและการใช้งานดีเทรสง่ายขึ้น ผู้ใช้สามารถศึกษาด้วยตนเองได้ ในงานวิจัยนี้ได้ผนวกชุดเครื่องมือดีเทรส ซึ่งเป็นการรวบรวมส่วนประกอบต่างๆได้แก่ ปลั๊กอินเน็ตบีนส์ดีเทรสจียูไอ และพัฒนาเอกเซอร์ไซเซอร์ รวมเป็นชุดเครื่องมือชื่อว่าดีทีเอ็กซ์ (DTEx – DTrace Exerciser) โดยพัฒนาโปรแกรมต่างๆ ให้เป็นชุดโปรแกรมปลั๊กอิน สามารถนำไปติดตั้งบนเน็ตบีนส์ได้

สำหรับการทดสอบนั้น ได้ทำการทดสอบเปรียบเทียบการใช้งานดีเทรสด้วยมือ (Manual) คือ ใช้งานดีเทรสตามปกติทั่วไป กับใช้งานดีเทรสด้วยชุดโปรแกรม โดยการใช้งานดีเทรสด้วยมือผู้ใช้ต้องทราบถึงคำสั่งต่างๆ ทั้งของระบบปฏิบัติการยูนิกซ์และคำสั่งของดีเทรส รวมถึงต้องเข้าใจและสร้างดีสคริปต์ด้วยภาษา “ดี” เมื่อใช้คำสั่งใช้งานดีเทรสผ่านเทอร์มินอลเพื่อเรียกดีสคริปต์ ผลลัพธ์ที่ได้ดูค่อนข้างยาก แต่ในการใช้งานดีเทรสด้วยชุดเครื่องมือ ผู้ใช้สามารถเลือกดีสคริปต์ที่ถูกจัดไว้เป็นหมวดหมู่พร้อมคำอธิบาย และใช้งานได้เพียงกดปุ่มเล่น โดยผลลัพธ์ที่ได้แสดงผลที่เข้าใจง่าย เป็นกราฟิกแยกสีเข้าใจง่าย และโปรแกรมนำไปใช้งานจริงได้ด้วย

นอกจากนี้ทำการทดลองด้วยการใช้เอกเซอร์ไซเซอร์กับระบบย่อย 4 ระบบ ได้แก่ หน่วยประมวลผล หน่วยความจำหลัก พีเอชพี และไพธอน โดยทำการเลือกเอกเซอร์ไซเซอร์ ให้เอกเซอร์ไซเซอร์จำลองเหตุการณ์ต่อระบบ เช่น กระตุ้น ทำให้ระบบทำงานหนัก เป็นต้น จากนั้นใช้ดีเทรสในการตรวจวัดหาสาเหตุถึงส่วนที่เอกเซอร์ไซเซอร์ทำให้ระบบทำงานหนัก ผู้ใช้สามารถเห็นถึงจุดที่ทำให้ระบบทำงานหนักหรือจุดที่มีปัญหา สำหรับชุดเครื่องมือดีทีเอ็กซ์นี้พัฒนาเพื่อใช้ในการศึกษาด้วยตนเอง เป็นสื่อการเรียนการสอน และสามารถนำไปใช้ในตรวจติดตามสมรรถนะของระบบงานจริงได้ด้วย

5.2 ปัญหาและข้อจำกัด

ในการศึกษาดีเทรสนั้น เนื่องจากดีเทรสมีจำนวนตัวตรวจวัด (Probe) เป็นจำนวนมาก อีกทั้งดีสคริปต์สามารถสร้างได้หลายรูปแบบ จึงไม่สามารถนำมาใช้ได้ในทุกๆ สคริปต์ หากแต่ในงานวิจัยได้คัดเลือกดีสคริปต์ที่นิยมใช้และเป็นประโยชน์นำมารวมไว้

การใช้งานดีสคริปต์ในบางดีสคริปต์ของชุดเครื่องมือดีเทรสนั้น บางสคริปต์ไม่สามารถให้เอกเซอร์ไซเซอร์จำลองสถานการณ์ได้เนื่องจากต้องใช้อุปกรณ์จำนวนมาก เช่น ใช้ดีเทรสกับระบบเครือข่าย เป็นต้น และในบางสคริปต์เป็นการใช้งานที่ลึกเกินไปที่จะทำความเข้าใจได้

สำหรับการสร้างเอกเซอร์ไซเซอร์ที่เป็นโปรแกรมภาษาอื่นนอกเหนือจากภาษาจาวา เช่น พีเอชพี ไพธอน รูบี้ เพิร์ล เป็นต้น โดยเมื่อทำการติดตั้งระบบปฏิบัติการโซลาริสเมื่อเริ่มแรกไม่มีรันไทม์ (Run time) ของภาษาเหล่านี้ ผู้ใช้จึงต้องทำการติดตั้งในส่วนของรันไทม์เพิ่ม นอกจากนี้ในการติดตั้งชุดเครื่องมือ ต้องมีการตั้งค่าสิทธิ์ในการใช้งาน (Privilege) ให้กับโปรแกรม เพื่อสามารถเรียกใช้งานดีเทรสได้

5.3 ข้อเสนอแนะ

ในงานวิจัยนี้ได้นำเสนอชุดเครื่องมือดีทีเอ็กซ์ที่ช่วยทำให้การใช้งานดีเทรสเป็นไปได้ง่ายเข้าใจง่ายขึ้น แต่ผู้ใช้งานไม่ใช่ว่าจะต้องไม่ศึกษาหรือไม่มีพื้นฐานด้านระบบปฏิบัติการยูนิกซ์และดีเทรสเลย แต่ผู้ใช้งานควรมีความรู้พื้นฐานของระบบปฏิบัติการยูนิกซ์และดีเทรสในระดับต้นได้ เพราะในช่วงการติดตั้งและการดูผลลัพธ์ที่ได้ยังใช้ความรู้ด้านนี้ หากแต่ชุดเครื่องมือนี้ช่วยทำให้ผู้ใช้ศึกษาดีเทรสได้เข้าใจง่ายขึ้นและใช้งานได้สะดวกขึ้น

งานวิจัยเพิ่มเติมในอนาคต ทำการปรับปรุงชุดเครื่องมือให้มีเอกเซอร์ไซเซอร์ในด้านอื่นเพิ่มขึ้น เช่น โปรแกรมภาษาเพิลล์ โมดูลผู้ใช้ (User) โมดูลโซน (Zone) เป็นต้น เพื่อให้เข้ากับหมวดหมู่ในดีสคริปต์มากขึ้น นำชุดเครื่องมือไปติดตั้งลงบนระบบปฏิบัติการอื่นๆ เช่น ฟรีบีเอสดี (FreeBSD) ลินุกซ์ (Linux) เป็นต้น นำชุดเครื่องมือทดลองให้งานจริงกับผู้ใช้ โดยทำการทดสอบใช้งานรวมทั้งประเมินความพอใจในการใช้งานด้วยแบบสอบถามก่อนและหลังใช้งานชุดเครื่องมือ

รายการอ้างอิง

[1] Cantrill, B. M., Shapiro, M. W., and Leventhal, A. H. Dynamic instrumentation of production systems, Proceedings of the annual conference on USENIX Annual Technical Conference, USEIX Association, Berkeley, CA, USA, (2004): 15-28

[2] Gregg, B. DTrace. [online]. (2009). Available from: http://www.dtrace.org/ [2011, April 24]

[3] The Cacti Group, Inc. Cacti. [online]. (2004). Available from: http://www.cacti.net/ [2012, Jan 29]

[4] Nagios Enterprises. Nagios. [online]. (2009). Available from: http://www.nagios.org/ [2012, Jan 29]

[5] The Gnome Project. Gnome System Monitor. [online]. (2005). Available from: http://www.gnome.org/ [2012, Jan 29]

[6] Cecil, R. The Power of DTrace. Sun Micosystems, Inc, 2007.

[7] McDougall, R. Mauro, J. and Gregg, B. Solaris Performance and Tools, Prentice Hall, 2006.

[8] Haslam, J. DTrace Documentation. [online]. (2010). Available from: https://wikis. oracle.com/ [2012, Jan 29]

[9] Rajadurai, A. OSCON Hands-on Lab. [online]. (2010). Available from: http://www. dtracehol.com/ [2012, Jan 29]

[10] Oracle Corporation. DTrace Community Group. [online]. (2009). Available from: http://hub.opensolaris.org/ [2012, Jan 29]

[11] Nouri, N. Netbeans DTrace GUI Plug-in. [online]. (2008). Available from: http://netbeans.org/kb/docs/ide/NetBeans_DTrace_GUI_Plugin_0_4.html [2012, Jan 29]

[12] B. Cantrill, "Hidden in Plain Sight," *Queue* 4, Feb, 2006.

[13] Gregg, B. DTraceToolkit. [online]. (2004). Available from: http://www. brendangregg.com/ dtrace.html [2012, Jan 29]

[14] Gousios, G., and Spinellis, D. Java Performance Evaluation Using External Instrumentation, IEEE Proceedings of the 2008 Panahellenic Conference on Infromatics (PCI'08), pp.173-177. Washington, DC, USA:IEEE, 2008.

[15] Oracle Corporation. Netbeans, [online]. (2012). Available from: http://www. netbeans.org/ [2012, Jan 29]

[16] Ni, Q., Sun, W., and Liang, X. Developing Solaris GUI Application with GTK+, IEEE International Conference on Information Science and Engineering (ICISE'09), pp.3283-3286. Washington, DC, USA: IEEE,2009.

[17] Gregg, B. DTraceTazTool. [online]. (2004). Available from: http://www. brendangregg.com/dtrace.html [2012, Jan 29]

[18] Rice, A. DLight. [online]. (2010). Available from: http://developers.sun.com/ sunstudio/documentation/tutorials/dlight/ [2012, Jan 29]

[19] Oracle Corporation. Netbeans Plug-in Tutorial. [online]. (2012). Available from: http://netbeans.org/kb/trails/ platform.html (2012, Jan 29)

[20] Oracle Corporation. VirtualBox. [online]. (2009). Available from: https://www.virtualbox.org/ (2012, Jan 29)

[21] Oracle Corporation. Oracle Solaris 11. [online]. (2009). Available from: http://www.oracle.com/us/products/servers-storage/solaris/solaris11/overview/ index.html (2012, Jan 29)

[22] Second Know. วิธีเขียนโปรแกรมวน Loop แล้ว CPU ขึ้นไม่ถึง 100%. [online]. (2009). Available from: http://www.secondknow.com/2009/194/break-loop-using-cpu-100/ (2012, May 18)

[23] Oracle Corporation. How to Use Sliders. [online]. (2012). Available from: http://docs.oracle.com/javase/tutorial/uiswing/components/slider.html (2012, May 18)

[24] Oracle Corporation. How to Use Swing Timers. [online]. (2012). Available from: http://docs.oracle.com/javase/tutorial/uiswing/misc/timer.html (2012, May 18)

[25] Back, E. Java Memory Leaks. [online]. (2012). Available from: http://elliottback.com/wp/java-memory-leaks-w-finalize-examples/ (2012, May18)

ประวัติผู้เขียนวิทยานิพนธ์

นาย วรวุฒิ ไพเราะ เกิดเมื่อวันที่ 8 มกราคม พ.ศ. 2527 ที่จังหวัดขอนแก่น สำเร็จการศึกษาในระดับประถมศึกษาจากโรงเรียนสนามบิน จังหวัดขอนแก่น สำเร็จการศึกษาในระดับมัธยมศึกษาจากโรงเรียนขอนแก่นวิทยายน จังหวัดขอนแก่น สำเร็จการศึกษาในระดับปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ จากคณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น ในปีการศึกษา 2550 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2552