บทที่ 6
การประมวลผลภาษาสคริปต์ที่ใช้ในการควบคุม

PHP Lite Script เป็นภาษาฝังตัว (Embedded Script) ที่ถูกพัฒนาขึ้น เพื่อใช้ในการควบคุมการทำงานของ เว็บเซิร์ฟเวอร์แบบฝังตัว โดยบางส่วนของโครงสร้างคำสั่งถูกดัดแปลงมาจาก PHP (Personal Home Page) อันเป็นภาษาสคริปต์ที่นิยมใช้งานแพร่หลาย ในการสร้างโปรแกรมประยุกต์ทางด้านเว็บ

6.1. ลักษณะทั่วไปของ PHP Lite Script

PHP Lite Script ประกอบด้วย ตัวแปร (Variables) ชุดคำสั่ง (Statements) และประโยคควบคุม (Control Statements) ทั้งนี้จุดมุ่งหมายหลักในการพัฒนา PHP Lite Script คือ เพื่อให้ผู้ใช้สามารถควบคุมการทำงานของ ระบบเว็บเซิร์ฟเวอร์แบบฝังตัวได้ในขณะทำงาน (Runtime Control) ซึ่งวิธีการใช้งานนั้น สคริปต์จะถูกฝังเป็นส่วนหนึ่งของ HTML Code เพื่อให้สามารถสร้างผลลัพธ์ได้อย่างมีประสิทธิภาพ ดังนั้นจึงต้องมีการ Escape หรือใส่รหัสหลีกเพื่อแยกระหว่าง Code ส่วนที่เป็น HTML Code และ PHP Code โดยทำการแทรกสคริปต์ลงระหว่างกรอบ Escape เพื่อช่วยในการสร้างผลลัพธ์แบบพลวัต (Dynamic)

6.2. โครงสร้างของภาษา PHP Lite Script

การ Escape ระหว่าง HTML Code และ PHP Code ตามมาตรฐานของ PHP ในปัจจุบันนั้น สามารถกระทำได้ทั้งสิ้น 4 วิธี คือการใช้ เครื่องหมาย "<? ?>", "<?php ?>", "<% %>" และ การกำหนดด้วย tag SCRIPT ของ HTML แต่บน PHP Lite Script ในที่นี้ เลือกใช้เพียงวิธีเดียวคือ การใช้เครื่องหมาย "<%" และ "%>" ทั้งนี้เนื่องจากเป็นวิธีการที่ใช้กันบนหลายระบบ เช่น ASP (Active Server Page ของ Microsoft) และ PHP (ดังตัวอย่างในภาพที่ 29)

```
<HTML><HEAD><TITLE>
<% echo ("TEST OF PHP LITE SCRIPT"); %>
</TITLE></HEAD>
<BODY>
<% echo ("PHP Lite Script Body"); %>
</BODY></HTML>
```

ภาพที่ 29 ตัวอย่างการแทรก Code ของ PHP Lite Script ลงใน HTML Code

6.2.1. ประเภทของค่าตัวแปรและค่าคงที่ (Variable and Constant)

ค่าตัวแปร (Variable) ภายใน PHP Lite Script นั้น ประกอบด้วยประเภทข้อมูล (Type) 2 ประเภท คือ จำนวนเต็ม (Integer) และสายอักษร (String)การอ้างอิงชื่อของตัวแปลทำได้โดยการใช้ชื่อซึ่งเป็นตัวอักษรความยาวไม่เกิน 8 ตัวอักษร และขึ้นต้นด้วย $ เท่านั้น ส่วนค่าคงที่นั้น ในกรณีของจำนวนเต็ม สามารถอ้างอิงได้ดังนี้

1. จำนวนเต็ม แทนด้วยตัวเลขฐานสิบ เช่น 100

2. อ้างอิงด้วยเลขฐาน 16 เช่น 0x4000,0x2000 เป็นต้น

ทั้งนี้ค่าต่างๆ จะอ้างอิงได้ไม่เกิน 16 bit

ส่วนกรณีของค่าคงที่ประเภทสายอักษรนั้น จะอ้างอิงด้วยการใช้เครื่องหมาย " (Double Quote) เช่น "Name" เป็นต้น

6.2.2. ตัวปฏิบัติการเกี่ยวกับตัวแปร (Operator)

ตัวปฏิบัติการนั้น จะทำหน้าที่คำนวณและประมวลค่าเกี่ยวกับตัวแปรต่างๆ เช่นการบวกลบ หรือต่อสายอักษรซึ่งสามารถจำแนกเป็นกลุ่มตามตัวแปรได้ดังนี้

6.2.2.1. ตัวปฏิบัติการบนตัวแปรประเภทจำนวนเต็ม (Number Operator) ประกอบด้วย "+" เพื่อทำการบวกค่าจำนวนต่าง และ "-" เพื่อทำการลบจำนวนเต็ม

6.2.2.2. ตัวปฏิบัติการบนตัวแปรประเภทสายอักษร (String Operator) ประกอบด้วย "." เพื่อใช้ทำการต่อสายอักษร (Concatenate) เช่น $A = "My ". "Name"; จะมีความหมายเหมือนกับ $A = "My Name"; เป็นต้น

6.2.3. นิพจน์ (Expression)

เป็นการกระทำเพื่อ ประมวลผลลัพธ์ที่เกิดการการกระทำของตัวปฏิบัติการ บนค่าคงที่และตัวแปลต่างๆ เช่น "1 + 2 + 0x2000" โดยผลลัพธ์ที่ได้จากนิพจน์ จะให้ค่าเป็นคำตอบของผลลัพธ์ ทั้งนี้ขึ้นอยู่กับว่านิพจน์ดังกล่าวอาศัยตัวปฏิบัติการใด ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 1

1+2-3+0x2000

ผลลัพธ์ที่ได้จะเป็น เลขจำนวนเต็มมีค่าเป็น 0x2000 ใน เลขฐาน 16 หรือ 8192 ในเลขฐาน 10 นั้นเอง

ตัวอย่างที่ 2

"User [" . $Name . "] is Authenticated"

หากค่าที่กำหนดในตัวแปร $Name เป็น "Krerk" ผลลัพธ์ที่ได้จะเป็น "User [Krerk] is Authenticated" เป็นต้น

ไวยากรณ์ของนิพจน์

Expression -> Term

Term -> Term Operator Expression

Term -> Variable

Term -> Constant

Term -> Null

6.2.4. ประโยค (Statements)

ประโยคหรือ Statements คือสิ่งที่กำหนดการทำงานของ PHP Lite Script เช่น การประมวลผล หรือ การกำหนดค่า เป็นต้น Statement สามารถจำแนกเป็นกลุ่มตามลักษณะการทำงานได้ดังนี้

1. ประโยคกำหนดค่า (Assignment Statement)
2. ประโยคควบคุม (Control Statement)
3. ประโยคประยุกต์ของระบบ (System Statement)

ไวยากรณ์ของ Statements

Statements -> Statements

Statements -> Statement

Statement -> Assignment Statement

Statement -> Control Statement

Statement -> System Statement

6.2.5. ประโยคกำหนดค่า (Assignment Statement)

ใช้เพื่อการกำหนดค่าให้กับตัวแปรต่างๆ โดยการใช้เครื่องหมาย = เช่น ต้องการให้ตัวแปร $a มีค่าเป็น 5 จะกระทำโดย $a=5; เป็นต้น โครงสร้างของ Assignment จะประกอบด้วย ตัวแปรตามด้วยเครื่องหมาย "=" และ นิพจน์ตามลำดับ

ไวยากรณ์ของประโยคกำหนดค่า

Assignment -> Variable = Expression;

6.2.6. ประโยคควบคุม (Control Statement)

ประโยคควบคุมจะทำหน้าที่ในการกำหนดทิศทางของการทำงานของระบบ เช่น การทำงานวนซ้ำเป็นต้น ในต้นแบบของงานวิจัยนี้ ได้กำหนดประโยคควบคุมขึ้นมา 1 ประเภทคือ While loop เพื่อใช้ในการทำงานที่ต้องวนซ้ำเป็นรอบ โดย While Loop จะทำงานซ้ำไปเรื่อยตราบเท่าที่เงื่อนไขของ Expression ยังเป็นจริง (True) หรือ ไม่เท่ากับศูนย์

ไวยากรณ์ของประโยคควบคุม

Control Statement -> While Loop

While Loop -> while (Expression) {Statements}

6.2.7. ประโยคประยุกต์ของระบบ (System Statement)

ประโยคประยุกต์ของระบบ เป็นประโยคที่ระบบกำหนดให้เพื่อให้ผู้ใช้สามารถทำการควบคุมการทำงานที่เกี่ยวข้องกับระบบได้ เช่น echo เพื่อแสดงค่าของ Expression เป็นต้น ในงานวิจัยนี้มีการพัฒนาประโยคประยุกต์ขึ้นมาทั้งสิ้น 4 ประโยคดังนี้

1. Echo Statement
2. Outport Statement
3. Inport Statement
4. String Convert Statement

ไวยากรณ์ของประโยคประยุกต์ของระบบ

System Statement -> Echo Statement

System Statement -> Outport Statement

System Statement -> Inport Statement

System Statement -> String Convert Statement

6.2.7.1. Echo Statement

เป็น Statement เพื่อใช้สำหรับแสดงค่าของ Expression โดยโครงสร้างของประโยคสามารถอธิบายได้ดังนี้

โครงสร้างของประโยค echo

echo (String Expression);

ไวยากรณ์ของประโยค echo

Echo Statement -> echo (Expression);

ประโยคดังกล่าวจะทำการประมวลนิพจน์จากนั้น จะแสดงค่าผลลัพธ์ที่ได้ของมายังหน้าเว็บ ทั้งนี้ ค่าของนิพจน์ที่ได้ควรมีผลลัพธ์เป็น ข้อมูลประเภทสายอักษร เพื่อให้สามารถแสดงผลได้ถูกต้อง

6.2.7.2. Outport Statement

เป็นประโยคเพื่อใช้สำหรับแสดงค่าที่ต้องการออกไปยัง Memory Mapped I/O ที่ระบุ โดยผู้ใช้สามารถระบุค่าของพอร์ตและข้อมูลที่ต้องการผ่านทางนิพจน์ จุดประสงค์ของประโยคดังกล่าวเพื่อใช้ในการกำหนดค่าให้กับอุปกรณ์ (Device) ที่ต้องการ

โครงสร้างของประโยค Outport

outp (Port , Data);

ไวยากรณ์ของประโยค Outport

Outport Statement -> outp (Expression , Expression);

ประโยค outp จะนำผลลัพธ์ที่ได้จากนิพจน์ที่ 1 มากำหนดค่าพอร์ตหรือตำแหน่งของหน่วยความจำ จากนั้นจะนำผลลัพธ์ที่จากการประมวลนิพจน์ที่ 2 แสดงเป็นผลลัพธ์ไปเก็บยังหน่วยความจำในตำแหน่งของพอร์ตที่กำหนดข้างต้น

6.2.7.3. Inport Statement

ประโยค inp จะทำหน้าที่อ่านค่าข้อมูลในหน่วยความจำ หรือ Memory Mapped I/O ที่ระบุโดยนิพจน์มาเก็บยังตัวแปรที่ต้องการ วัตถุประสงค์ของประโยคคือการอ่านค่าและสถานะจากอุปกรณ์ที่ต่อพ่วงอยู่กับระบบ

โครงสร้างของประโยค Inport

inp (Port , Variable);

ไวยากรณ์ของประโยค Inport

Inport Statement -> inp (Expression , Variable);

ผลลัพธ์ที่ได้จากนิพจน์ตัวแรกจะใช้กำหนดพอร์ตหรือตำแหน่งของหน่วยความจะที่ต้องการอ่านข้อมูล โดยประโยคดังกล่าวจะทำการอ่านค่าจากพอร์ต มาเก็บยังตัวแปรที่ระบุ

6.2.7.4. String Convert Statement

เป็นประโยคประยุกต์เพื่อทำหน้าที่แปลข้อมูลประเภทจำนวนเต็มที่เก็บอยู่ในตัวแปร ให้เป็นข้อมูลประเภทสายอักษร

โครงสร้างของประโยค String Convert

str (Variable);

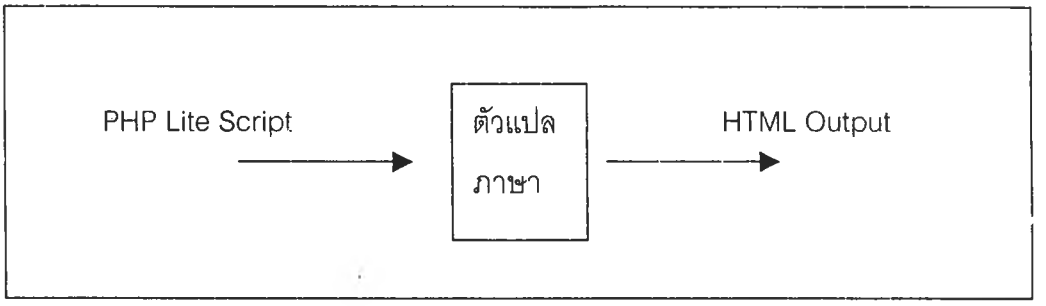
ไวยากรณ์ของประโยค String Convert

String Convert Statement -> str (Variable);

ผลลัพธ์ที่ได้จากประโยคนี้คือ การเปลี่ยนค่าคงที่ประเภทจำนวนเต็มในตัวแปรที่ระบุ มาเป็นค่าคงที่ประเภทสายอักษร เช่น $A มีค่าเป็น 0x0001 เมื่อผ่านการทำงานของประโยค str($A); จะได้ผลลัพธ์ค่า $A เป็น "0001" เป็นต้น

6.3. การสร้างตัวแปลภาษา PHP Lite Script

การแปลภาษานั้น คือการสร้างผลลัพธ์ที่ได้จากการประมวล PHP Lite Script รวมถึง การจัดการสภาพแวดล้อม (Environment) ในการทำงานเพื่อสนับสนุนการทำงาน ของภาษาด้วย ทั้งนี้การแปลจะใช้การ Interpret เพื่อเป็นการสร้างผลลัพธ์แบบขณะทำงาน และสามารถปรับเปลี่ยนตัว Script ได้ง่าย ลักษณะการทำงานของ PHP Lite Script นั้น เป็นการแปลภาษาแบบ Interpret ซึ่งสามารถอธิบายโครงสร้างการทำงานได้ดังภาพที่ 30



ภาพที่ 30 การทำงานของตัวแปลภาษา PHP Lite Script

การทำงานภายในของตัวแปลภาษานั้น ยังแบ่งออกได้อีก 2 ขั้นตอนย่อย คือ การวิเคราะห์คำศัพท์ (Lexical Analysis) ซึ่งจะทำหน้าที่ในการตรวจสอบและรู้จำคำต่างๆ จากนั้นผลลัพธ์ที่ได้จากการวิเคราะห์คำศัพท์ หรือ TOKEN จะถูกส่งผ่านไปทำการวิเคราะห์ความหมาย (Semantic Analysis) เพื่อทำการตีความและสร้างผลลัพธ์ต่อไป

6.3.1. การเตรียมสภาวะแวดล้อม และ การรับค่าผ่านฟอร์มใน HTML

สภาวะแวดล้อมในการทำงานของ PHP Lite Script นั้น ประกอบด้วยค่าต่างๆ ที่ส่งผ่านให้ตัวแปลภาษาเรียกใช้งาน เช่น ค่าที่ได้รับผ่านเว็บในรูปแบบต่างๆ สภาวะของระบบในขณะนั้น ผู้ใช้งาน Script สามารถอ้างถึงตัวแปรต่างๆ ที่ได้จากการ Submit ผ่านฟอร์มบน HTML ได้โดยตรงภายใน PHP Lite Script

จากตัวอย่างในภาพประกอบ 31 จะพบว่า ตัวแปร username จะอ้างอิงด้วย $username เมื่ออยู่ภายใน PHP Lite Script อย่างไรก็ตาม เบื้องต้นนั้นระบบจะสามารถรับค่าได้จาก METHOD GET เท่านั้น

```
<FORM METHOD=GET>
<INPUT TYPE="text" name="username">
<INPUT TYPE="SUBMIT">
</FORM>
<%
echo ($username);

%>
```

ภาพที่ 31 การอ้างอิงค่าจากฟอร์มบน HTML

6.3.2. การวิเคราะห์คำศัพท์ (Lexical Analysis)

ในการแปลภาษาโดยทั่วไปนั้น ขั้นตอนแรกของการแปลคือการหาค่า TOKEN ของภาษา ซึ่ง TOKEN เหล่านี้คือข้อความที่เป็น KEYWORD ภายในภาษานั้น สำหรับกรณีของ PHP Lite Script นี้ TOKEN ของภาษาประกอบด้วย 18 TOKEN ดังนี้

1. STOP หรือเครื่องหมาย "}"
2. VAR คือตัวแปรทุกตัว ขึ้นต้นด้วย "$"
3. WHILE เป็นข้อความคำว่า "while" ซึ่งเป็น Control Statement
4. IF ข้อความ "if" เป็น Control Statement (มิได้ทำการทดสอบในงานวิจัยนี้) ซึ่งจะพัฒนาต่อไปในอนาคต
5. ECHO ข้อความ "echo"
6. INP ข้อความ "inp"
7. OUTP ข้อความ "outp"
8. COMMA หรือเครื่องหมาย ","
9. OB เครื่องหมาย "("
10. CB เครื่องหมาย ")"
11. SCOLON เครื่องหมาย ";" (Semicolon)
12. START หรือเครื่องหมาย "{"
13. PLUS ในที่นี้คือ Operator "+"
14. MINUS ในที่นี้คือ Operator "-"
15. DOT ในที่นี้คือ Operator "."

16. CONST ค่าคงที่ ซึ่งอาจเป็นสายอักษร จำนวนเต็ม หรือเลขฐาน 16 ก็ได้
17. EQ เครื่องหมาย "="
18. STRCONV ข้อความ "str"

การทำงานของการวิเคราะห์คำศัพท์นั้น เบื้องต้นจะเปรียบเทียบเครื่องหมายหรือสัญลักษณ์ที่เป็นอักษรเพียงตัวเดียวก่อน จากนั้นจึงจะเปรียบเทียบกับสัญสักษณ์อื่นๆ หากไม่ตรงกับสัญลักษณ์ใดๆ จะคืนค่าเป็น Error กรณีที่ TOKEN ที่ได้เป็น VAR จะคืนค่าเป็น VAR และ บอก VALUE เป็นชื่อตัวแปร ในทำนองเดียวกัน กรณีที่ TOKEN ที่ได้เป็น CONST ผลลัพธ์ที่ได้จะบอก VALUE เป็นค่าของCONST ที่อ่านได้

6.3.3. การวิเคราะห์ความหมาย (Semantic Analysis)

การทำงานของการวิเคราะห์ความหมาย จะตรวจสอบความถูกต้องของภาษา ประโยค และนิพจน์ พร้อมทั้งแสดงผลลัพธ์ที่ได้ออกมา โดยเราสามารถจัดกลุ่ม Semantic Analysis เป็นหมวดหมู่ตามหน้าที่การทำงานได้ตามไวยากรณ์ของภาษาดังกล่าวข้างต้น ดังนี้

1. นิพจน์ หรือ Expression จะทำหน้าที่ประมวลผล Expression
2. กลุ่มประโยคควบคุม Control Statement
3. กลุ่มประโยคประยุกต์ของระบบ
4. กลุ่มประโยคกำหนดค่า

กลุ่มนิพจน์จะประมวลค่าต่างๆ ตามตัวปฏิบัติการหรือ Operator ที่ได้รับ จากนั้นจะคืนค่าเป็นผลลัพธ์เพื่อให้กลุ่มอื่นใช้งานต่อไป

กลุ่มประโยคควบคุม จะตรวจสอบผลลัพธ์ที่ได้จากนิพจน์ให้เป็นไปตามเงื่อนไขที่กำหนด หากเป็นไปตามเงื่อนไขที่กำหนดจะทำงานในช่วง START และ STOP หากไม่เป็น ประโยคจะข้ามไปจนกว่าจะเจอ STOP

กลุ่มประโยคประยุกต์ของระบบจะทำงานตามแต่ความหมายของ TOKEN ที่ได้ เช่น echo จะทำการแสดงผลนิพจน์ เป็นต้น

กลุ่มประโยคกำหนดค่า จะนำค่าที่ได้จากนิพจน์มาเก็บยังค่าของชื่อตัวแปรที่กำหนด