บทที่ 2
ทฤษฎีเบื้องต้น

2.1 ทฤษฎีเบื้องต้นของ PHP

PHP เป็นภาษาสคริปต์ที่ทำงานทางฝั่งเซิร์ฟเวอร์ (server-side scripting language) ซึ่งมีลักษณะเป็น embedded script นั่นคือเราสามารถฝังคำสั่ง PHP ไว้ในเว็บเพจร่วมกับคำสั่ง (tag) ของภาษา HTML ได้

2.1.1 ประวัติของ PHP

Rasmus lerdorf สร้างภาษา PHP ขึ้นมาในปี ค.ศ.1994 เนื่องจากเขาต้องการพัฒนาโปรแกรมเพื่อเก็บข้อมูลของผู้ใช้ที่แวะเวียนเข้ามาเยี่ยมชมโฮมเพจส่วนตัวของเขาเอง เขาเรียกโปรแกรมนี้ว่า PHP ซึ่งย่อมาจาก Personal Home Page Tools

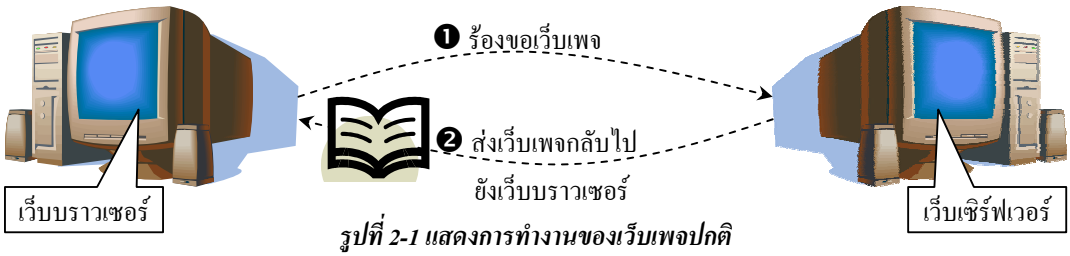
ในเวอร์ชั่นแรกสุดนั้น PHP ยังไม่มีความสามารถอะไรมากนัก โดยประกอบด้วยกลไกการแปลภาษาอย่างง่าย และชุดคำสั่ง/มาโคร ที่เป็นประโยชน์ต่อการสร้างสมุดเยี่ยม (guest book) และตัวนับจำนวนผู้เข้าชมเว็บ (counter) เท่านั้น

พอกลางปี ค.ศ.1995 เขาก็ได้พัฒนาตัวแปลภาษา PHP ขึ้นมาใหม่โดยใช้ชื่อว่า PHP/FI version 2 ซึ่งได้เพิ่มความสามารถในการรับข้อมูลที่ส่งมาจากฟอร์มของ HTML (จึงมีชื่อว่า FI หรือ Form Interpreter) นอกจากนั้นยังเพิ่มความสามารถในการติดต่อกับฐานข้อมูล mySQL อีกด้วย จึงทำให้ผู้คนเริ่มหันมาสนใจ PHP มากขึ้น

ในปี 1997 มีผู้ร่วมพัฒนา PHP เพิ่มอีก 2 คนคือ Zeev Suraski และ Andi Gutmans โดยได้แก้ข้อบกพร่องต่างๆ และเพิ่มเครื่องมือให้มากขึ้น กลายเป็น PHP version 3 และพัฒนาต่อมาจนถึง version 4 (PHP4) ในปัจจุบัน

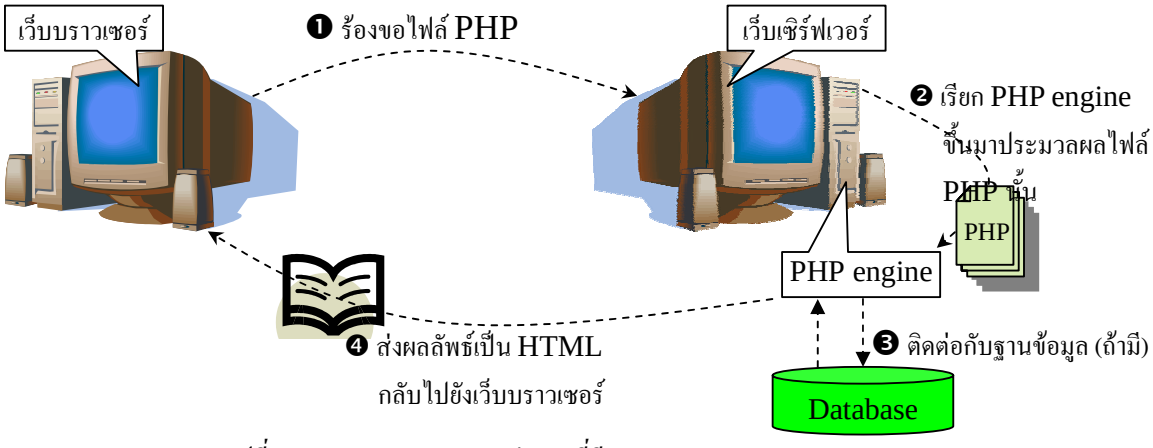
2.1.2 กลไกการทำงานของเว็บเพจกับ PHP

สำหรับเว็บเพจธรรมดาที่โดยปกติมีนามสกุลของไฟล์เป็น htm หรือ html นั้นเมื่อเราใช้ เว็บบราวเซอร์เปิดดูเว็บเพจใด เว็บเซิร์ฟเวอร์ก็จะส่งเว็บเพจนั้นกลับมายังเว็บบราวเซอร์ จากนั้นเว็บบราวเซอร์จะแสดงผลไปตามคำสั่งภาษา HTML ที่อยู่ในไฟล์ ซึ่งการทำงานในลักษณะนี้ทำให้ได้เว็บเพจที่เป็นแบบ static ซึ่งผู้ใช้ไม่สามารถเปลี่ยนแปลงหรือใช้งานในลักษณะที่แตกต่างได้



รูปที่ 2-1 แสดงการทำงานของเว็บเพจปกติ

PHP เป็นภาษาที่ทำงานในฝั่งเซิร์ฟเวอร์ การใช้ PHP ในการสร้างเว็บเพจที่เป็นแบบ dynamic ทำโดยการฝังสคริปต์ภาษา PHP ไว้ในเว็บเพจนั้น(และมีนามสกุลเป็น .PHP) เมื่อเว็บบราวเซอร์ร้องขอไฟล์ PHP ไฟล์ใด เว็บเซิร์ฟเวอร์จะเรียก PHP engine ขึ้นมาแปล (interpret) และประมวลผลคำสั่งที่อยู่ในไฟล์ PHP นั้น โดยอาจมีการดึงข้อมูลจากฐานข้อมูล หรือเขียนข้อมูลลงไปยังฐานข้อมูลด้วย หลังจากนั้นผลลัพธ์ในรูปแบบ HTML (และสคริปต์ที่ทำงานฝั่งบราวเซอร์ เช่น client-side JavaScript) จะถูกส่งกลับไปยังบราวเซอร์ บราวเซอร์ก็จะแสดงผลตามคำสั่ง HTML ที่ได้รับมา ซึ่งย่อมไม่มีคำสั่ง PHP ใดๆ หลงเหลืออยู่ เนื่องจากถูกแปลและประมวลผลโดย PHP engine ที่ฝั่งเซิร์ฟเวอร์ไปหมดแล้ว

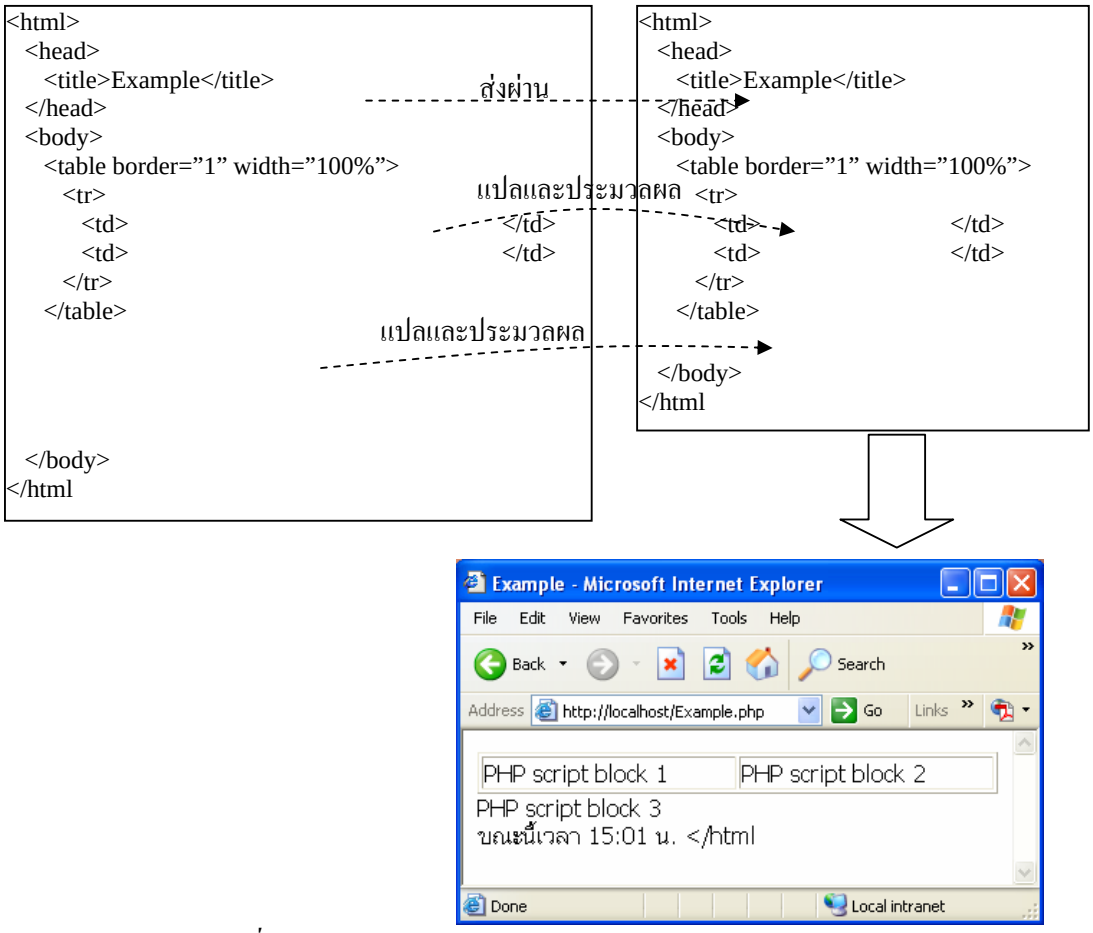


รูปที่ 2-2 แสดงการทำงานของเว็บเพจที่มี PHP script

จะเห็นว่าการทำงานของเว็บบราวเซอร์ในกรณีที่ร้องขอไฟล์ PHP ไม่แตกต่างจากกรณีที่ร้องขอไฟล์เว็บเพจธรรมดาเลย เพราะสิ่งที่บราวเซอร์ต้องกระทำก็คือการร้องขอไฟล์จากเว็บเซิร์ฟเวอร์ จากนั้นก็รอรับผลลัพธ์กลับมาแสดงผล ความแตกต่างจริงๆ อยู่ที่การทำงานฝั่งเซิร์ฟเวอร์ ซึ่งกรณีที่มีการเรียกไฟล์ PHP จะผ่านการประมวลผลก่อน แทนที่จะถูกส่งไปยังบราวเซอร์เลยทันที

2.1.3 การประมวลผลไฟล์ PHP

PHP engine จะแปลและประมวลผลเฉพาะคำสั่งที่อยู่ภายในแท็กของ PHP เท่านั้น การทำงานที่เกิดขึ้นคือ หลังจาก PHP engine ถูกเว็บเซิร์ฟเวอร์เรียกขึ้นมาประมวลผลไฟล์ PHP แล้ว มันจะส่งผ่านเนื้อหาของไฟล์ไปยังบราวเซอร์โดยไม่ทำอะไรกับเนื้อหานั้น ยกเว้นเมื่อพบกับสัญลักษณ์ (tag) ที่ระบุจุดเริ่มต้นของบล็อกคำสั่ง PHP มันก็จะแปลและประมวลผลคำสั่งต่างๆ ไปตามลำดับ (ภายในบล็อค PHP นี้ การส่งผลลัพธ์ให้แก่บราวเซอร์ จะต้องเรียกใช้ คำสั่ง/ฟังก์ชั่น ของ PHP เช่น echo หรือ print) โดยเมื่อพบสัญลักษณ์ปิดท้ายบล็อกคำสั่ง PHP PHP engine ก็จะหันกลับมาส่งผ่านเนื้อหาของไฟล์ต่อไปเช่นเดิมจนกว่าจะพบสัญลักษณ์ระบุจุดเริ่มต้นของบล็อคคำสั่ง PHP อีก และเป็นอย่างนี้เรื่อยไปจนจบไฟล์



รูปที่ 2-3 รูปแสดงผลบนบราวเซอร์จากการทำงานของสคริปต์ PHP

2.1.4 ความสามารถของ PHP

PHP ได้รับการพัฒนาความสามารถขึ้นมาเรื่อยๆ อย่างต่อเนื่อง ทั้งนี้เป็นเพราะมีการเปิดเผยซอร์สโค้ดของ PHP สู่สาธารณะในลักษณะของ open source ทำให้มีหน่วยงานและองค์กรต่างๆ เข้ามาช่วยกันพัฒนา ในที่นี้จะขอกล่าวถึงความสามารถหลักของ PHP เท่านั้น ดังนี้

- ความสามารถในการจัดการกับตัวแปรหลายๆ ประเภท เช่น เลขจำนวนเต็ม(integer), เลขทศนิยม(float), สตริง(string) และอาร์เรย์(array) เป็นต้น
- ความสามารถในการรับข้อมูลจากฟอร์มของ HTML
- ความสามารถในการรับ-ส่ง Cookies
- ความสามารถเกี่ยวกับ Session (ตั้งแต่ PHP เวอร์ชั่น 4 ขึ้นไป)
- ความสามารถทางด้าน OOP (Object Oriented Programming) ซึ่งรองรับการเขียนโปรแกรมเชิงวัตถุ
- ความสามารถในการเรียกใช้ COM component
- ความสามารถในการติดต่อและจัดการฐานข้อมูล
- ความสามารถในการสร้างภาพกราฟิก

2.2 ระบบปฏิบัติการซิมเบียน

ระบบปฏิบัติการซิมเบียนเป็นระบบปฏิบัติการที่ออกแบบมาเพื่อรองรับเทคโนโลยีการสื่อสารแบบไร้สาย ถูกสร้างขึ้นมาจากความต้องการหลากหลาย โดยต้องการที่จะทำงานได้โดยไม่ขึ้นต่อฮาร์ดแวร์ของโทรศัพท์เคลื่อนที่ซึ่งมีความแตกต่างกัน และสามารถจัดการกับการพัฒนาอย่างต่อเนื่องในอนาคตได้ ช่วยในการส่งข้อมูลของโทรศัพท์เคลื่อนที่ เป็นระบบที่ใช้งานได้ง่าย มีความปลอดภัยสูง ช่วยประหยัดพลังงาน และใช้หน่วยความจำที่มีขนาดเล็ก เพื่อรองรับกับโทรศัพท์มือถือทั้งในปัจจุบันและอนาคต

ทั้งนี้ซิมเบียน นั้นไม่ได้เป็นเพียงแค่ระบบปฏิบัติการเท่านั้น แต่ซิมเบียน ยังเป็นระบบเปิดที่ให้ผู้อื่น สามารถพัฒนาซอฟท์แวร์ต่างๆ บนระบบปฏิบัติการซิมเบียน ได้ เรียกได้ว่าในอนาคตจะมีแอพพลิเคชั่นมากมายที่ถูกสร้างขึ้นบนระบบปฏิบัติการซิมเบียน และยังส่งผลให้เป็นตลาดผลิตภัณฑ์ซอฟท์แวร์ที่ยิ่งใหญ่ในอนาคตอีกด้วย

2.2.1 ความเป็นมาและการพัฒนาของระบบปฏิบัติการซิมเบียน

ซิมเบียน คือ บริษัทผลิตซอฟท์แวร์ที่เป็นผู้นำในด้านการผลิตซอฟท์แวร์ที่รองรับเทคโนโลยีการสื่อสารแบบไร้สาย (Wireless) ซึ่งผลิตภัณฑ์ที่ได้รับการยอมรับจากบริษัทคู่ค้า (Supplier) ก็คือ ระบบปฏิบัติการซิมเบียน (Symbian Operating System)

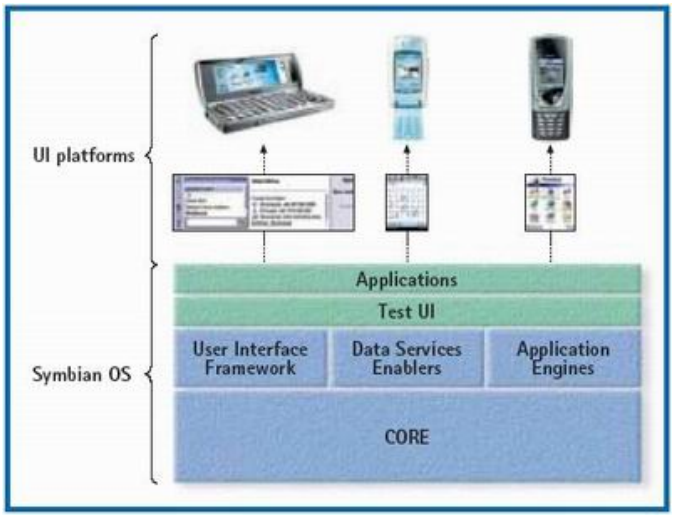
ระบบปฏิบัติการซิมเบียนเกิดขึ้นในเดือนมิถุนายน ปี ค.ศ.1998 ซึ่งในตอนนั้นมีพันธมิตรร่วมกัน 4 ราย คือ Ericsson, Motorola, Nokia และ PSION มีสำนักงานใหญ่อยู่ที่ประเทศอังกฤษ

ถัดมาในปี ค.ศ. 1999 ซิมเบียนก็ได้พันธมิตรรายใหม่นั่นคือ Matsushita (Panasonic) ในปี ค.ศ. 2000 พันธมิตรของซิมเบียน ก็มากขึ้นอีกโดยมีการจับมือกับ Sony, Sanyo และ Kenwood รวมถึงการเปิดตัวโทรศัพท์ที่ใช้ระบบปฏิบัติการซิมเบียน เครื่องแรกในรูปแบบของ Smart Phone นั่นก็คือ Ericsson R380s ซึ่งมีคุณสมบัติการใช้ที่มากมาย และเป็น Touch Screen Display ด้วย

ในปี 2001 Nokia ที่เคยใช้ระบบปฏิบัติการ Geos ต้องเปลี่ยนมาใช้ระบบปฏิบัติการที่เป็น ซิมเบียน บ้าง ทาง Nokia จึงได้พัฒนา Communicator รุ่นใหม่ออกมา คือ Nokia 9210 ซึ่งระบบปฏิบัติการนั้นแตกต่างจาก Ericsson R380s ก็คือ ระบบปฏิบัติการซิมเบียน ของ Nokia 9210 นี้เป็นระบบเปิด (Open Symbian OS Phone) คือสามารถที่จะนำโปรแกรมอื่นที่รองรับซิมเบียน มาลงเพิ่มในเครื่องได้ และในปี 2002 ทางซิมเบียนมีการพัฒนาระบบปฏิบัติการรุ่นใหม่ที่เรียกว่า Symbian OS V 7.0 และทาง Sony Ericsson ก็เข้ามาเป็นพันธมิตรและเป็นหุ้นส่วนรายใหญ่ รวมทั้งทาง Sony Ericsson ก็เปิดตัว Smart Phone รุ่นล่าสุด คือ Sony Ericsson P800 และ Nokia ก็ออกผลิตภัณฑ์รุ่นใหม่ที่ใช้ " Series 60 Platform" เช่นกัน โดยใช้ชื่อรุ่นว่า Nokia 3650 ซึ่งมีประสิทธิภาพที่เหนือกว่า Nokia 7650

2.2.2 โครงสร้างของระบบปฏิบัติการซิมเบียน

- Symbian OS core เป็นแกนหลักในการทำงานของอุปกรณ์ เช่น kernel, file server, memory management และ device driver ซึ่งส่วนประกอบต่างๆนั้นสามารถที่เพิ่มหรือลดได้ขึ้นอยู่กับอุปกรณ์แต่ละชนิด
- System Layer เป็นส่วนที่จัดการเกี่ยวกับการคำนวณ การติดต่อสื่อสาร เช่น TCP/IP IMAP4, SMS และ การจัดการฐานข้อมูล
- Application engine ซึ่งอยู่บน System Layer ทำให้นักพัฒนาโปรแกรมสามารถสร้าง User interface ติดต่อกับข้อมูล
- User Interface Software โดยผู้ผลิตแต่ล่ะรายสามารถที่จะสร้างขึ้นได้ เช่น Nokia Series 60 platform ของบริษัท Nokia
- Application ซึ่งอยู่ส่วนบนของ User interface



รูปที่ 2-4 โครงสร้างโดยรวมของ Symbian OS

2.2.3 โครงสร้างของไคลเอนต์-เซิร์ฟเวอร์

การทำงานแบบ ไคลเอนต์-เซิร์ฟเวอร์ เป็นที่รู้จักกันดีในการติดต่อสื่อสารของซอร์ฟแวร์ ในระบบปฏิบัติการซิมเบียน ส่วนของโปรแกรมไคลเอนต์ จะมี user interface ที่ติดต่อกับโปรแกรมเซิร์ฟเวอร์ โดยสามารถติดต่อได้ทาง interface ที่โปรแกรมเซิร์ฟเวอร์ ได้สร้างไว้ให้ หลักของการทำงานก็คือ ไคลเอนต์ จะร้องขอบริการจากเซิร์ฟเวอร์ ซึ่งเซิร์ฟเวอร์ จะตอบสนองตามที่ร้องขอ

2.2.4 การจัดการกับเหตุการณ์ (Event management)

การจัดการกับเหตุการณ์ เป็นส่วนสำคัญที่จะต้องพิจารณา ในระบบปฏิบัติการซิมเบียน ถูกสร้างขึ้นจาก event-base time sharing ใน single thread แต่อย่างไรก็ตาม ก็มีวิธีการที่ทำให้เป็น multi-thread application Object oriented design เนื่องจากระบบปฏิบัติการซิมเบียน สร้างขึ้นในลักษณะ Object-oriented ทำให้สามารถทำการแก้ไขได้ง่ายสำหรับชนิดของฮาร์ดแวร์ที่ต่างกันและอนุญาตให้ผู้ผลิตต่างๆสามารถที่จะเพิ่มหรือลดคอมโพเนนท์ ต่างได้ทำให้เหมาะกับความต้องการของลูกค้ามากขึ้น ซึ่งการที่มีความยืดหยุ่นนี้ในการออกแบบทำให้สามารถออกแบบ user interface ในอุปกรณ์ชนิดต่างกัน ทำงานในระบบปฏิบัติการเหมือนกันได้ ระบบปฏิบัติการซิมเบียน เป็นแพลตฟอร์มที่เสถียรภาพสามารถที่รองรับเทคโนโลยีใหม่ในอนาคต เช่น SyncML, Bluetooth, Multimedia Messaging

2.2.5 การจัดการเกี่ยวกับพลังงาน (Power management)

การจัดการเกี่ยวกับพลังงานถูกสร้างขึ้นใน kernel ของ Symbian OS การออกแบบทำให้การใช้พลังงานของ processor และอุปกรณ์ต่างได้อย่างมีประสิทธิภาพ เมื่ออุปกรณ์ชนิดใดไม่ถูกใช้งานก็จะปิดโดยระบบ การใช้พลังงานที่น้อยนี้เองทำให้สามารถยืดอายุการใช้งานของแบตเตอร์รี่ขนาดเล็กได้

2.2.6 ความแข็งแรงและความน่าเชื่อถือ (Robust and dependable)

ระบบปฏิบัติการซิมเบียน ประสบความสำเร็จในด้านประสิทธิภาพในระบบโทรศัพท์เคลื่อนที่ โดยอุปกรณ์ไม่ควรที่จะสูญเสียข้อมูล ระบบล่ม หรือมีการรีบูต สิ่งที่ทำให้ซิมเบียน ประสบความสำเร็จ คือ

1. แต่ละ process ทำงานอยู่บน protect address space ทำให้แอพพลิเคชัน อื่นไม่สามารถเขียนข้อมูลทับใน address space ของแอพพลิเคชันอื่นได้
2. kernel ทำงานอยู่บน protect address space จึงทำให้ข้อผิดพลาดของโปรแกรมต่างๆไม่สามารถเขียนข้อมูลทับลงในส่วน stack หรือ heap ของ kernel ได้

2.2.7 การจัดการหน่วยความจำ (Memory management)

ในอุปกรณ์เคลื่อนที่ การจัดการหน่วยความจำมีความจำเป็นอย่างยิ่ง เนื่องจากหน่วยความจำที่มีขนาดจำกัด ปฏิบัติการซิมเบียน จะมองว่าหน่วยความจำมีขนาดจำกัดจะใช้หน่วยความจำให้น้อยที่สุดในแต่ละครั้ง การใช้หน่วยความจำน้อยนี้ทำให้การใช้พลังงานลดลงด้วย

2.2.8 การทำงานแบบ Multitasking

แอพพลิเคชันต่างๆในระบบปฏิบัติการซิมเบียน ทำงานอยู่ต่าง Process กัน จึงสามารถทำให้หลายแอพพลิเคชัน ทำงานได้พร้อมกัน ตัวอย่างเช่น เมื่อผู้ใช้ต้องการตรวจสอบรายการประจำวันและรับสายโทรเข้า ระบบต้องสามารถที่จะสลับการทำงานระหว่างสองกระบวนการได้อย่างรวดเร็ว ผู้ใช้สามารถดูรายการประจำวันขณะที่มีการโทรได้ ซึ่งในปัจจุบันโทรศัพท์เป็นสิ่งที่ให้ข้อมูลอย่างมาก ดังนั้นความสามารถนี้จึงเป็นสิ่งที่สำคัญ

2.2.9 Application Framework บนระบบปฏิบัติการซิมเบียน

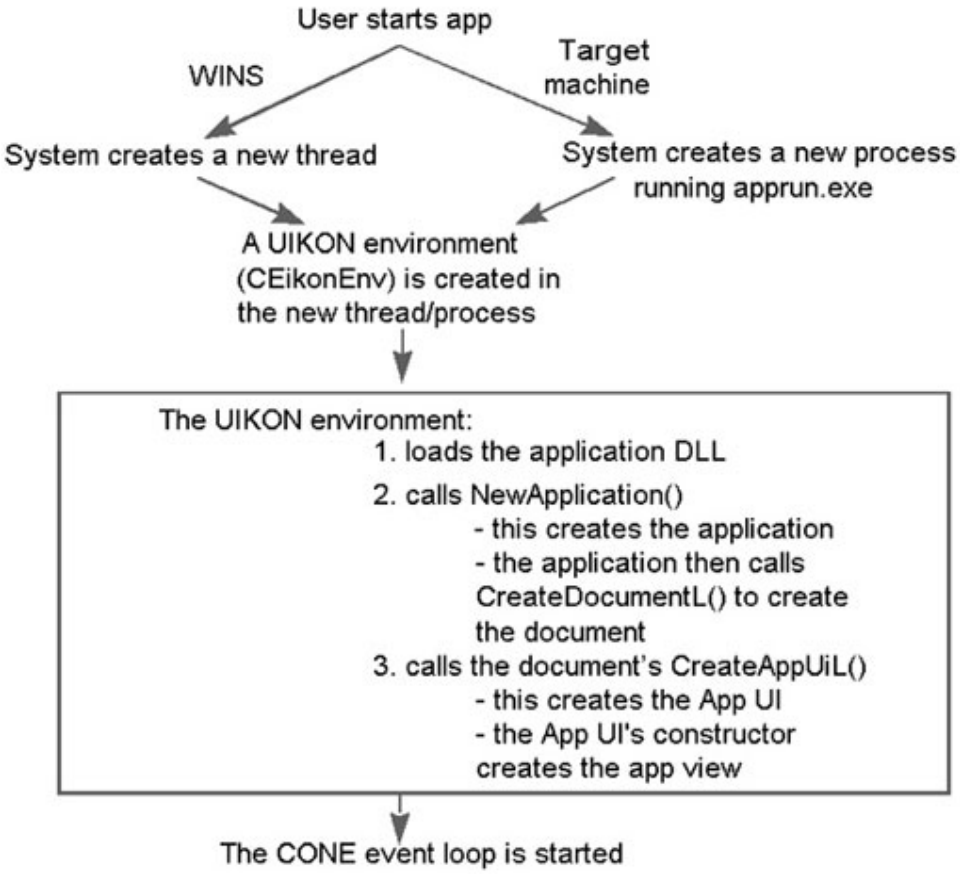
การทำงานของแอพลิเคชันที่ใช้ระบบปฏิบัติการซิมเบียน บนอีมูเลเตอร์ (Emutator) ที่อยู่บนระบบปฏิบัติการ Windows (ในทางเทคนิคเรียกว่า “WINS” ที่ทำงานแบบ Single Process) และอุปกรณ์จริงอย่างเช่น Nokia 3650 จะมีความแตกต่างกันอยู่เล็กน้อย ซึ่งความแตกต่างทำให้ผู้พัฒนาโปรแกรมนั้นมีปัญหากับความไม่สัมพันธ์กันระหว่างอีมูเลเตอร์กับอุปกรณ์จริง โดยเฉพาะผู้พัฒนาโปรแกรมที่เป็นโปรแกรมที่มีความซับซ้อนมาก จะเป็นห่วงเกี่ยวกับความแตกต่างระหว่าง WINS กับอุปกรณ์จริง

กุญแจหลักสำคัญของ Application Framework นี้คือ UIKON ซึ่ง UIKON ก็คือ มาตรฐานของ Framework ที่ทุกๆอุปกรณ์ซึ่งทำงานอยู่บนระบบระบบปฏิบัติการซิมเบียน ต่างมีเหมือนกัน UIKON ไม่เพียงแต่จะมี Framework ที่จัดเตรียมไว้สำหรับการทำงานของแอพพลิเคชันเท่านั้นแตยังมีส่วนที่สนับสนุนการทำงานของ control component ทั่วๆไปด้วย (เช่น dialog box, number editor, date editor) ซึ่งกลุ่มของแอพพลิเคชันจะสามารถใช้ประโยชน์ได้ขณะทำงาน

แอพพลิเคชันบนระบบระบบปฏิบัติการซิมเบียน ประกอบด้วย 4 คอมโพเนนท์ ซึ่งแต่และส่วนมีความเกี่ยวข้องกับคลาสใน UIKON Framework ได้แก่

1. Application shell ซึ่งได้รับการถ่ายทดคุณสมบัติมาจาก CEikApplication และคลาสนี้จะถูกสร้างขึ้นตอนแรกสุดโดย Framework ทันทีที่ถูกสร้างขึ้น มันจะมีหน้าที่รับผิดชอบต่อการกำหนดค่าเริ่มต้นต่างๆให้กับโค้ด ส่วนที่เหลืออยู่
2. Document ซึ่งได้รับการถ่ายทดคุณสมบัติมาจาก CEikDocument ส่วนนี้จะทำให้เข้าใจผิดเล็กน้อยในเรื่องที่ว่า ไม่ใช่ทุกแอพพลิเคชันต้องมี Document ที่จะต้องมาเกี่ยวข้องกับมัน เหมือนกับที่แอพพลิเคชันเหล่านี้จะต้องเกี่ยวข้องกับผู้ใช้ (User) ตัวอย่างเช่น โปรแกรม Notepad ที่เป็นโปรแกรมเกี่ยวข้องกับ Document โดยตรงอย่างชัดเจน ในขณะที่โปรแกรม เช่น นาฬิกา นั้นจะไม่ถูกกำหนดความสามารถในการจัดการ Document ทั้งการสร้าง, การเปิดเอกสาร, การแก้ไข document ใดๆ ในความเป็นจริงทุกๆแอพพลิเคชันจะต้องมีคลาส CEikDocument ที่ Framework จะต้องการสร้างขึ้นมา
3. App UI ซึ่งได้รับการถ่ายทดคุณสมบัติมาจาก CEikAppUi ซึ่งคลาสนี้จัดเตรียมหน้าที่การทำงานหลักสำหรับทุกๆแอพพลิเคชัน เช่น event handing, event control, การสร้าง control, การเข้าถึง system call ที่เป็นประโยชน์หลายๆตัว เป็นต้น คลาส CEikDocument-derived นั้นจะรับผิดชอบต่อการสร้างแอพพลิเคชันในส่วนสุดท้าย
4. View ส่วนนี้ คือ ส่วนที่ผู้ใช้สามารถมองเห็นได้จริงในหน้าจอของอุปกรณ์ มันสามารถใช้งานสำหรับการแสดงข้อมูลอย่างง่าย หรือรวบรวมข้อมูลจากผู้ใช้ในแอพพลิเคชันที่ซับซ้อนมากขึ้น (ตัวอย่างเช่น editor ในแอพพลิเคชันประเภท word ตัวหนังสือที่ถูกพมพ์นั้น ก็คือ control มาตรฐาน ซึ่งถูกจัดไว้โด UIKON โดยจัดเก็บอยู่ในส่วนของ View) ในขณะที่ทุกแอพพลิเคชันจะมีอยู่หนึ่ง View โดยปรกติ แต่แอพพลิเคชันที่มีความซับซ้อนมากสามารถมีได้หลาย View

การทำงานของ Framework สามารถแสดงได้ดังรูป



รูปที่ 2-5 การทำงานของแอพพลิเคชันบนระบบปฏิบัติการซิมเบียน