

ข้อสอบวัดประสิทธิภาพของการโปรแกรมของนักศึกษาระดับปริญญาตรี (Exit Exam)

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ ครั้งที่ 1 ประจำปีการศึกษา 2566

วันเสาร์ที่ 9 กันยายน พ.ศ. 2566

เวลาสอบ 13.30 – 16.30 น

ข้อที่ 1

อาจารย์โมอยากสร้าง Kernel ของ Operating System อันใหม่ที่มีชื่อว่า “Monix” ซึ่งเป็น Kernel ขนาดเล็กที่ใช้ในอุปกรณ์ IoT หรือ Smart Phone โดยที่อาจารย์ไม่มีความต้องการดังนี้

ความต้องการของโจทย์ (Requirements)

- เขียนโปรแกรมโดยใช้แนวคิด MVC (Model-View-Controller) Design Pattern หรือแนวคิดอื่น ๆ ที่คล้ายคลึง โดยใช้ภาษาโปรแกรมใดก็ได้
- Kernel ของ OS “Monix” มีหน้าที่ในการเขียนข้อมูลเข้าและแสดงผลสรุปของข้อมูลเป็นสำคัญ โดยมีรายละเอียดดังนี้
 - รับข้อมูลจากฝั่งผู้ใช้ โดยข้อมูลที่รับจะเป็นลักษณะ [เวลา] [จำนวนเต็ม] เท่านั้น โดยเวลาจะเป็นลักษณะ ชั่วโมง:นาฬิกา:วินาที และจะเว้นช่องว่าง 1 ช่องระหว่างเวลาและจำนวนเต็ม เช่น 11:12:13 10000 และเก็บข้อมูลไว้ (สามารถใช้งานได้ทั้งฐานข้อมูลแบบ Relational หรือ NOSQL หรือ JSON File หรือ CSV File หรือวิธีการอื่น ๆ ขึ้นอยู่กับความถนัด)
 - แสดงผลสรุปของข้อมูลที่เป็นจำนวนเต็มย้อนหลังไม่เกิน 1 ชั่วโมงนับตั้งแต่เวลาปัจจุบันที่รับเข้ามา โดยการหาค่าเฉลี่ย ค่า Percentile ที่ 50, 90, และ 95 ของข้อมูลเหล่านั้นตามลำดับ (ดูตัวอย่างการหา Percentile ได้ที่ <https://www.opendurian.com/learn/percentile/>)
- Kernel นี้สามารถรองรับ User Interface ได้ทั้งหมดสองรูปแบบตามอุปกรณ์ทั้งสองแบบ คือ IoT และ Smart Phone ซึ่งจะมีการแสดงผลที่แตกต่างกัน

4) การแสดงผลในรูปแบบของ IoT จะเน้นแค่ข้อความเป็นหลัก โดยจะสามารถรับค่าและแสดงผลสรุปดังนี้

```
Average: 7456.39
50th Percentile: 8000
90th Percentile: 10000
95th Percentile: 11000
```

5) การแสดงผลในรูปแบบของ Smart Phone จะเน้นการใช้ภาพเป็นหลัก โดยจะสามารถรับค่าและแสดงผลสรุปดังนี้ (นักศึกษาไม่จำเป็นต้องแสดงเป็นข้อความแบบนี้ก็ได้ ขอแค่มีลักษณะที่คล้าย ๆ กับด้านล่างนี้)

```

                                90th: 10000
                                v
                                v 95th: 11000
                                v v
00000000000X000000X0X0
      ^
    50th: 8000 / Avg.: 7456.39
```

การส่งคำตอบสำหรับข้อนี้

- 1) ส่ง Source Code ของโปรแกรมหุ้นกล่าวทั้งหมด
- 2) ส่ง Script สำหรับการสร้างฐานข้อมูล (หากมี)
- 3) ข้อความอธิบายการทำข้อสอบแต่ละส่วน ในคอมเมนต์ของ Google Classroom ซึ่งประกอบด้วย ไฟล์ที่ส่งมาไฟล์ใดทำหน้าที่อะไรใน MVC และทำงานร่วมกันอย่างไร และทำข้อสอบข้อที่เท่าไร

อาจารย์โมต้องการสร้างภาษาเพื่อใช้รับส่งข้อมูลระหว่างหน้าบ้านและหลังบ้านของระบบ PEARLS ชื่อว่าภาษา MOML โดยเลียนแบบจากภาษาที่มีอยู่แล้ว แค่เปลี่ยนชื่อเพื่อความเท่ เพียงแต่ยังไม่แน่ใจว่าควรตั้งไวยากรณ์ของภาษาไว้อย่างไร

ความต้องการของโจทย์ (Requirements)

- 1) เขียนโปรแกรมโดยใช้แนวคิด MVC (Model-View-Controller) Design Pattern หรือแนวคิดอื่น ๆ ที่คล้ายคลึง โดยใช้ภาษาโปรแกรมใดก็ได้
 - 2) พัฒนาส่วนเชื่อมประสานกับผู้ใช้ (User Interface) เพื่อรับ Code ของภาษา MOML โดยมีลักษณะดังนี้
 - **DESC** เป็นข้อความที่เป็นข้อมูลที่สำคัญ
 - **TAG** เป็นการกำหนดประเภทของข้อความ โดย TAG มีสองลักษณะคือ
 - **แบบที่ 1 TAG ที่มีการเปิดและปิด** จะประกอบด้วย <ชื่อประเภท>DESC</ชื่อประเภท> เช่น <MVC>Please let me pass</MVC>
 - **แบบที่ 2 TAG ที่ไม่มีการปิด** จะประกอบด้วย <ชื่อประเภท/>DESC เช่น <MVC/>Please let me pass
 - 3) ข้อความ DESC ที่ไม่มี TAG กำกับ จะเรียกว่า UNTAGGED หรือเป็นข้อความที่ไม่มีประเภท
 - 4) ไวยากรณ์แบบแรกที่อาจารย์โมต้องการ คือการส่งข้อความที่ให้ความสำคัญกับ TAG เป็นสำคัญ ดังนั้นข้อความที่แปลผลได้จะขึ้นอยู่กับ TAG
 - หากมี TAG แบบที่ 1 ข้อความที่อยู่ระหว่าง TAG เปิดและปิดจะเป็น DESC ทั้งหมด (ไม่สนใจว่าจะมีเครื่องหมาย <> หรือไม่ ดังนั้นไม่ต้องกังวลเรื่อง Hierarchical Data หรือข้อมูลที่มีลำดับชั้น เหมือนภาษาที่คล้าย ๆ กันอื่น ๆ)
 - หากมี TAG แบบที่ 2 ข้อความที่อยู่หลังจากนั้นทั้งหมดจะเป็น DESC จนกว่าจะเจอ TAG ที่เข้าลักษณะเป็น TAG ปิดต่อไป
 - หากไม่เข้าเงื่อนไขสองข้อด้านบน ให้กำหนดข้อความเป็น UNTAGGED
- โดยให้แสดงผลลัพธ์ออกมาตามตัวอย่างด้านล่าง
- 5) ไวยากรณ์แบบที่สองที่อาจารย์โมต้องการ คือการส่งข้อความที่ให้ความสำคัญกับบรรทัดเป็นสำคัญ ดังนั้นข้อความที่แปลผลได้จะขึ้นอยู่กับบรรทัดที่ส่งมา

- หากในบรรทัดนั้นมี TAG แบบที่ 1 อยู่ในส่วนใดของบรรทัด ข้อความที่อยู่ระหว่าง TAG เปิดและปิดจะเป็น DESC ทั้งหมด (ไม่สนใจว่าจะมีเครื่องหมาย <> หรือไม่ ดังนั้นไม่ต้องกังวลเรื่อง Hierarchical Data หรือข้อมูลที่มีลำดับชั้น เหมือนภาษาที่คล้าย ๆ กันอื่น ๆ) หรือจนกว่าจะจบบรรทัด
 - หากมี TAG แบบที่ 2 ข้อความที่อยู่หลังจากนั้นทั้งหมดจะเป็น DESC จนกว่าจะจบบรรทัดหรือเจอ TAG ที่เข้าลักษณะเป็น TAG เปิดต่อไป
 - หากไม่เข้าเงื่อนไขสองข้อด้านบน ให้กำหนดข้อความเป็น UNTAGGED
- อาจารย์ต้องการให้นักศึกษาสร้างอีก Model หนึ่ง เพื่อรองรับการทำงานที่เปลี่ยนไป และให้แสดงผลลัพธ์ออกมาตามตัวอย่างด้านล่างเมื่อเลือกการแบ่งอีกวิธีหนึ่ง

ตัวอย่างการทำงาน

ตัวอย่าง Source Code ภาษา MOML	คำตอบของข้อ 4)	คำตอบของข้อ 5)
TEST 1 <ABC>TEST 2<NONTAG></ABC> TEST 3 <DEF/>TEST 4 TEST 5<HI>TEST 6 </HI>	[UNTAGGED]{TEST 1} [TAG= "ABC"]{TEST 2<NONTAG>} [UNTAGGED]{TEST 3} [TAG= "DEF"]{TEST 4 TEST 5} [TAG= "HI"]{TEST 6}	[UNTAGGED]{TEST 1} [TAG= "ABC"]{TEST 2<NONTAG>} [UNTAGGED]{TEST 3} [TAG= "DEF"]{TEST 4} [UNTAGGED]{TEST 5} [TAG= "HI"]{TEST 6} [UNTAGGED]{</HI>}

การส่งคำตอบสำหรับข้อนี้

- 1) ส่ง Source Code ของโปรแกรมดังกล่าวทั้งหมด
- 2) ส่ง Script สำหรับการสร้างฐานข้อมูล (หากมี)
- 3) ข้อความอธิบายการทำข้อสอบแต่ละส่วน ในคอมเมนต์ของ Google Classroom ซึ่งประกอบด้วย ไฟล์ที่ส่งมา ไฟล์ใดทำหน้าที่อะไรใน MVC และทำงานร่วมกันอย่างไร และทำข้อสอบข้อที่เท่าไร