

ยานอวกาศ Mars Pathfinder กับ Priority inversion problem

- Mars Pathfinder มีภารกิจในการรวบรวมข้อมูลด้านอุตุนิยมวิทยาของดาวอังคาร
- Mars Pathfinder ใช้ VxWorks ซึ่งเป็นระบบปฏิบัติการแบบ real-time ที่ใช้สำหรับ embedded system
- VxWorks ประกอบด้วย threads ของ task ต่างๆ ซึ่งแต่ละ thread ได้รับการกำหนด priority ตามความสำคัญของแต่ละ task นอกจากนี้ VxWorks ยังใช้ preemptive scheduling ในการเลือก thread เพื่อประมวลผล
- Mars Pathfinder มี Information bus ที่เปรียบได้กับ shared memory ในที่นี้ใช้เป็นช่องทางในการขนถ่ายข้อมูลระหว่างแต่ละส่วนในยานอวกาศ การครอบครอง information bus ของ threads ต่างๆ กระทำผ่าน mutex lock นั่นคือ มีเพียง thread เดียวเท่านั้นที่ครอบครอง information bus ณ ขณะใดขณะหนึ่ง (อยู่ใน critical section)
- Threads ในภารกิจการรวบรวมข้อมูลอุตุนิยมวิทยาของ Pathfinder มีอยู่ด้วยกัน 3 กลุ่ม ซึ่งมี priority ที่แตกต่างกัน คือ
 - Meteorological task thread ที่ทำหน้าที่รวบรวมข้อมูลอุตุนิยมวิทยา และเขียนข้อมูลดังกล่าวลงใน information bus โดย thread นี้มีความถี่ในการประมวลผลน้อย และมี priority ระดับ low
 - Data bus thread ที่ทำหน้าที่อ่านข้อมูลจาก information bus เพื่อนำไปใช้งาน โดย thread นี้มีความถี่ในการประมวลผลสูง และมี priority ระดับ high
 - Communication task thread ที่ทำหน้าที่อื่นๆ ที่ไม่เกี่ยวข้องกับการใช้ information bus แต่อาจทำงานในระยะเวลาที่ค่อนข้างนาน โดย thread นี้มี priority ระดับ medium
- ในกรณีที่มีเพียง Meteorological task thread กับ Data bus thread อาจมีบางครั้งที่ Data bus thread ต้อง block รอ Meteorological task thread เนื่องจาก Meteorological task thread ได้ down mutex ก่อน และได้ครอบครอง information bus ซึ่งในที่สุดแล้ว Meteorological task thread จะเสร็จจากการทำงานกับ Information bus และ up mutex ซึ่งจะส่งผลให้ Data bus thread ได้รับการ unblock และทำงานกับ Information bus ตามที่ต้องการได้
- นอกจากนี้อาจมีกรณีที่ threads ทั้งสามทำงานในจังหวะที่ใกล้เคียงกัน ซึ่งจะส่งผลให้ Communication task thread ซึ่งมี priority สูงกว่า Meteorological task thread ทำการ preempt CPU ไปจาก Meteorological task thread ที่อยู่ระหว่างการครอบครอง Information bus และมี Data bus thread block คอยอยู่ในจังหวะเดียวกัน ในที่นี้จะทำให้ Data bus thread ซึ่งมี priority สูงกว่า Communication task thread ต้อง block นานขึ้น อันเนื่องมาจาก Meteorological task thread ต้องรอ CPU จนกว่า Communication task thread จะทำงานแล้วเสร็จ
- สถานการณ์ข้างต้นนี้ เรียกว่า **priority inversion problem** อันเนื่องมาจาก high priority task ต้องรอ low priority task ซึ่งขัดแย้งกับการทำงานในแบบ preemptive scheduling ผลกระทบที่เกิดจาก priority inversion problem อาจแตกต่างกันไปในแต่ละปัญหา
- ในส่วนของ Mars Pathfinder ระบบปฏิบัติการ VxWorks มี watchdog timer สำหรับกำหนดระยะเวลาในการทำงานขององค์ประกอบที่มีความสำคัญ ในกรณีที่ watchdog timer เติบโตจนครบระยะเวลา แสดงว่าอาจมีสิ่งผิดปกติเกิดขึ้น
- ในสถานการณ์ของ priority inversion ที่เกิดขึ้นนี้ ทำให้ watchdog timer ของ VxWorks เติบโตจนครบระยะเวลา และตรวจสอบพบว่า Data bus task ไม่ได้ทำงานเป็นระยะเวลาหนึ่ง จึงสันนิษฐานว่ามีสิ่งผิดปกติเกิดขึ้น ส่งผลให้มีการ reset system โดยอัตโนมัติ ซึ่งเป็นปัญหาที่พบในการทำงานของ VxWorks ของ Mars Pathfinder จนเป็นที่มาของบทความนี้
- การแก้ปัญหา priority inversion problem อาจทำได้โดยการทำให้ **priority inheritance** นั่นคือ low priority task จะได้รับ priority ให้เท่ากับ high priority task เป็นการชั่วคราว เพื่อให้ได้สิทธิในการครอบครอง CPU จนกว่าจะทำงานใน critical section เสร็จ แล้วจึงกลับสู่ระดับ priority เดิม ซึ่งส่งผลให้ high priority task ใช้เวลารอเท่าที่จำเป็นเท่านั้น