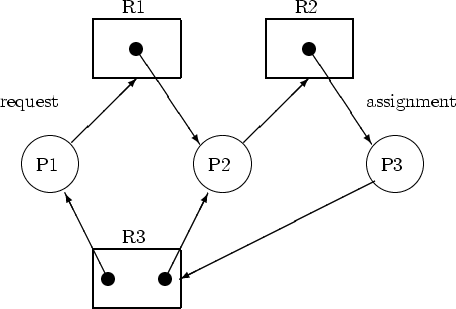
|  |  |
| --- | --- |
|  | **แบบฝึกหัด วงจรอับ (Deadlock)** |
| **คำสั่ง ตอบคำถามต่อไปนี้** | |



**จากรูป** จงวิเคราะห์ปัญหาการเกิดวงจรอับ

ตอบ. P1 ต้องการร้องขอใช้ R1 แต่ P2 กำลังครอบครองทรัพยากร R1 อยู่

P2 ต้องการร้องขอใช้ R2 แต่ P3 กำลังครอบครองทรัพยากร R2 อยู่ P3 ต้องการร้องขอใช้ R3 แต่ทั้ง P1 และ P2 กำลังครอบครองทรัพยากรพร้อมกันทั้งคู่

สาเหตุของการเกิดวงจรอับ

ตอบ.

1. ทรัพยากรไม่สามารถใช้ร่วมกันได้หลายๆโปรเซสพร้อมกัน
2. แย่งชิงการใช้ทรัพยากรของระบบ มีการถือครองและรอคอยกัน
3. การทำงานในระบบไม่มีการแทรกกลางครัน ไม่สามารถบังคับให้มีการปลดปล่อยทรัพยากรได้

การแก้ไขปัญหาวงจรอับ

ตอบ.

1. พยายามอย่าให้เหตุการณ์ทั้ง 3 เกิดขึ้นพร้อมกัน
2. หากเกิดขึ้น ให้ใช้วิธีการแก้คือ ให้โปรเซสใดโปรเซสหนึ่งปลดปล่อยทรัพยากรโปรเซสนั้นครอบครองอยู่

จากภาพ P2 ต้องปลดปล่อยทรัพยากรที่ครอบครองอยู่ ให้ P1 ครอบครองและทำงานจนเสร็จ และ P3 ต้องปลดปล่อยทรัพยากรที่ครอบครองอยู่ ให้ P2 ครอบครองและทำงานจนเสร็จ และ P1 กับ P2 ต้องปลดปล่อยทรัพยากรที่ครอบครองอยู่ ให้ P3 ครอบครองและทำงานจนเสร็จ

วิธีการป้องกันปัญหาวงจรอับ

ตอบ.

1. ให้ใช้ทรัพยากรร่วมกันได้(Mutual Exclusion Prevention) : OS จะต้องจัดสรรให้โปรเซสในระบบสามารถใช้ทรัพยกรเหล่านั้นร่วมกันได้ เช่น การเข้าถึงแฟ้มข้อมูลยอมให้ทุกโปรเซสเข้าใช้งานโดยอ่านได้อย่างเดียวและยอมให้มีการเขียนกระบวนการเดียวเท่านั้น
2. ป้องกันการถือครองและรอคอย(Hold and wait prevention) : กำหนดให้แต่ละโปรเซสต้องร้องขอทรัพยากรทั้งหมด และจะอนุญาตเมื่อได้ทรัพยากรครบ
3. ยอมให้มีการแทรกกลางคัน(Preemptable) : ระบบจะไม่ยอมให้โปรเซสร้องขอทรัพยากรจนกว่าจะปลดปล่อยทรัพยากรของตัวเอง ถ้าโปรเซสต้องการเพิ่ม ต้องปลดปล่อยทรัพยากรแล้วทำการขอใหม่ และระบบสามารถแทรกได้กลางคัน ใช้ได้กับระบบที่มีสิทธิและลำดับความสำคัญ (Priority)
4. ป้องกันการเกิดวงจรรอคอย(Circular wait protection) : โดยการกำหนดลำดับของทรัพยากรทั้งหมดในระบบโปรเซสจะร้องขอได้เฉพาะทรัพยากรที่อยู่ในลำดับสูงกว่าเท่านั้น