Activity 7: Process Synchronization

วัตถุประสงค์

1. เพื่อให้นิสิตเข้าใจหลักการของ process synchronization

2. เพื่อให้นิสิตสามารถเขียนโปรแกรมใช้งาน semaphore ได้

เตรียมตัว

1. ศึกษาหลักการ semaphore ในบทที่ 6 Process Synchronization

2. ศึกษา Linux POSIX named semaphore

ความรู้พื้นฐาน

Process Synchronization เป็นองค์ประกอบที่สำคัญในการทำงานร่วมกันของ Process หรือ Thread ซึ่งเครื่องมือใน Linux จะรองรับทั้งการทำ Semaphore และ Shared Memory

Semaphore เป็นตัวแปรประเภท counter ที่แสดงถึงสถานะของทรัพยากร โดยที่ counter แบบ semaphore จะมีลักษณะพิเศษคือ Operating System จะทำการดูแลไม่ให้เกิด race condition กล่าวคือ ผู้ใช้งานสามารถมั่นใจได้ว่า ณ เวลาใดเวลาหนึ่ง ค่าใน counter จะถูกแก้ไขได้โดยเพียง Process หรือ Thread เดียวเท่านั้น

Semaphore มี 2 ประเภทคือ Named Semaphore สำหรับรองรับการทำงาน ระหว่างหลายๆ Process (สร้างโดยคำสั่ง **sem_open**) และ Unnamed Semaphore สำหรับรองรับการทำงานของหลาย Thread ภายใน Process เดียวกัน (สร้างโดยคำสั่ง **sem_init**)

การทำงานของ Semaphore จะขึ้นอยู่กับค่าของ counter โดย Process หรือ Thread สามารถทำการลดค่า (ทำการ Lock) ด้วยคำสั่ง **sem_wait** หรือเพิ่มค่า (ทำการ Unlock) ด้วยคำสั่ง **sem_post** ของ Semaphore ได้ โดยที่ถ้า Process หรือ Thread พยายามจะลดค่าของ Semaphore ในขณะที่มีค่าเป็นศูนย์ Process หรือ Thread นั้นจะถูก block และจะต้องรอจนกว่าค่า Semaphore จะถูกเพิ่มจนมีค่ามากกว่าศูนย์ ซึ่งจะเกิดขึ้นได้ก็ต่อเมื่อมี Process หรือ Thread อื่นมาทำการเพิ่มค่าหรือ Unlock Semaphore นี้สำหรับ Binary Semaphore จะเป็น Semaphore ที่มีค่าอยู่ระหว่าง 0 กับ 1 ซึ่งจะมีชื่อเรียกเป็นพิเศษว่า Mutex โดยจะถูกใช้สำหรับการ Lock/Unlock Critical Section

รายละเอียดของ Semaphore ศึกษาเพิ่มเติมได้จาก

https://linux.die.net/man/7/sem_overview

กิจกรรม PABX Simulation

ในกิจกรรมนี้จะให้นิสิตทำการปรับปรุง source code ของโปรแกรม PABX Simulator โดยให้นิสิตดาวน์โหลดไฟล์ act-7.zip จาก Course Material “Activity 7: Synchronization (act-7.zip)” ใน CourseVille ภายหลังจากการทำ unzip จะพบไฟล์ 4 ไฟล์ใน semaphore คือ

- Makefile – สำหรับการใช้คำสั่ง make ในการ compile
- pabx_server.c – เป็นโปรแกรมในส่วนของ server ที่จะทำการจำลองระบบโทรศัพท์จำนวน n คู่สาย (n จะเป็นค่าที่ส่งผ่านทาง command line ไปยังตัวโปรแกรมเช่น ถ้า run ด้วยค าสั่ง pabx_server 3 หมายถึงให้ทำการจำลองระบบโทรศัพท์จำนวน 3 คู่สาย) โดยทำการสร้าง Named Semaphore “pabx” พร้อมทั้งระบุค่าตั้งต้นของ Semaphore เป็น n
- pabx_caller.c – เป็นโปรแกรมที่จำลองผู้ใช้งานโทรศัพท์ที่พยายามจะโทรออกผ่าน pabx_server โดยผู้โทรจะทำการติดต่อไปยัง server เพื่อร้องขอคู่สาย (โดยการใช้ Named Semaphore ชื่อ “pabx”) เมื่อได้รับคู่สาย จะทำการโทรออกเป็นระยะเวลาสุ่มระหว่าง 1-5 วินาที หลังจากนั้นจะวางหู แล้วทำการรอเป็นระยะเวลาสุ่มระหว่าง 1-3 วินาที ก่อนที่จะทำการโทรในครั้งต่อไป
- pabx_rm.c – เป็นโปรแกรมที่ทำการยกเลิก Semaphore ที่ใช้ใน pabx ใน source code ของ pabx_server.c และ pabx_caller.c ที่ได้รับจะมีรายละเอียดไม่ครบถ้วน กล่าวคือในส่วนคำสั่งที่เกี่ยวข้องกับ Semaphore ได้ถูกแทนที่ด้วย Comment ตัวอย่างเช่น ในไฟล์ pabx_caller.c บรรทัดที่ 16-18 จะมีข้อความ

  ```
  //
  // OS -- OPEN NAMED SEMAPHORE HERE
  //
  ```

  เป็นการระบุว่า ให้นำคำสั่งเกี่ยวกับการเปิด named semaphore มาแทนที่ comment นี้

<u>สิ่งที่ต้องทำ</u>

- ปรับปรุง source code โดยการเพิ่มคำสั่งเกี่ยวกับ Semaphore ที่ เหมาะสม
- ใช้คำสั่ง make เพื่อคอมไพล์โปรแกรม ซึ่งจะได้ผลลัพธ์เป็นโปรแกรมสองโปรแกรมชื่อ pabx_server และ pabx_caller
- ทำการทดสอบด้วยการรันโปรแกรม pabx_server โดยมี argument เป็นจำนวนคู่สาย และรันโปรแกรม pabx_caller หลาย ๆ ครั้ง ให้มีจำนวนโปรเซสมากกว่าจำนวนคู่สาย (แต่ละ process อยู่คนละหน้าต่าง terminal กัน)
- ตัวอย่างดังต่อไปนี้เป็นการให้ pabx_server สร้างคู่สายจำนวน 1 คู่สาย และมี pabx_caller จำนวน 2 process ถ้าทำได้อย่างถูกต้อง ควรจะได้ผลลัพธ์ในลักษณะดังนี้

```
$ ./pabx_server 1
Starting PABX with 1 phone lines.
There are 1 phone lines available.
There are 1 phone lines available.
There are 1 phone lines available.
There are 1 phone lines available.
```

```
There are 0 phone lines available.
There are 1 phone lines available.
There are 0 phone lines available.
There are 0 phone lines available.
There are 0 phone lines available.
...
```

```
$ ./pabx_caller
Starting caller
Wait for 2 seconds
Access outside phone line
Get a phone line after waiting for 0 seconds. I will use for 2
seconds.
Hang up the phone.
Wait for 1 seconds
Access outside phone line
Get a phone line after waiting for 1 seconds. I will use for 1
seconds.
Hang up the phone.
Wait for 3 seconds
Access outside phone line
Get a phone line after waiting for 1 seconds. I will use for 1
seconds.
Hang up the phone.
Wait for 2 seconds
...
```

ให้ capture หน้าจอผลลัพธ์เก็บไว้

สิ่งที่ต้องส่งใน courseville

1) souce code ที่ได้แก้แล้ว
2) ภาพหน้าจอผลลัพธ์