

README สำหรับปฏิบัติการที่ 8

ศิระกร ลำไย (5910500023) และกรวิชญ์ ชัยกังวาล (5910501909)

สถานะการส่งงาน

เสร็จสมบูรณ์

คำถามก่อนคอมไพล์และรัน

- เมื่อ simulate ทำงานเสร็จสิ้นแล้ว balance ที่เหลือให้แต่ละ account มีค่าเท่ากับ 800,000,000
- ถ้าเราเปลี่ยนค่า INIT_BALANCE ที่ account.h เป็นค่า 0 เมื่อ simulate ทำงานเสร็จสิ้นแล้ว balance ที่เหลือในแต่ละ account มีค่าเท่ากับ 400,000,000
- ถ้าเราเปลี่ยนค่า INIT_BALANCE ที่ account.h เป็นค่า 0 แล้วสลับบรรทัด 55 lender((void *)i); กับบรรทัดที่ 56 borrower((void *)i); เมื่อ simulate ทำงานเสร็จสิ้นแล้ว balance ที่เหลือในแต่ละ account มีค่าเท่ากับ 800,000,000

ข้อที่ 1

ผลการใช้คำสั่ง time ของ single thread และ multi thread เป็นดังแสดงด้านล่าง

```
size of account_info is 40; size of int is 4; size of void pointer is 8
Account: 0; name: test_account0; balance: 400000000
Account: 1; name: test_account1; balance: 400000000
Account: 2; name: test_account2; balance: 400000000
Account: 3; name: test_account3; balance: 400000000

I am lender # 0
I am borrower # 0
I am lender # 1
I am borrower # 1
I am lender # 2
I am borrower # 2
I am lender # 3
I am borrower # 3

Account: 0; name: test_account0; balance: 800000000
Account: 1; name: test_account1; balance: 800000000
Account: 2; name: test_account2; balance: 800000000
Account: 3; name: test_account3; balance: 800000000
```

```
./simulate 20.85s user 0.06s system 47% cpu 43.850 total
```

```
size of account_info is 72; size of int is 4; size of void pointer is 8
Account: 0; name: test_account0; balance: 400000000
Account: 1; name: test_account1; balance: 400000000
Account: 2; name: test_account2; balance: 400000000
Account: 3; name: test_account3; balance: 400000000

I am lender # 0
I am borrower # 0
I am borrower # 3
I am lender # 1
I am borrower # 2
I am lender # 3
I am borrower # 1
I am lender # 2

Account: 0; name: test_account0; balance: 800000000
Account: 1; name: test_account1; balance: 800000000
Account: 2; name: test_account2; balance: 800000000
Account: 3; name: test_account3; balance: 800000000
./simulate_multithread 100.77s user 18.00s system 276% cpu 42.965
total
```

จะเห็นได้ว่าการเขียน lock (ซึ่งในที่นี้เขียนเป็น fine-grain locking ที่ละเอียดเกินไป) จะทำงานได้ช้ากว่า

ข้อที่ 2

```
size of account_info is 40; size of int is 4; size of void pointer is 8
Account: 0; name: test_account0; balance: 0
Account: 1; name: test_account1; balance: 0
Account: 2; name: test_account2; balance: 0
Account: 3; name: test_account3; balance: 0

I am lender # 0
I am borrower # 0
I am lender # 1
I am borrower # 1
I am lender # 2
I am borrower # 2
I am lender # 3
I am borrower # 3
```

```
Account: 0; name: test_account0; balance: 4000000000
Account: 1; name: test_account1; balance: 4000000000
Account: 2; name: test_account2; balance: 4000000000
Account: 3; name: test_account3; balance: 4000000000
./simulate 13.38s user 0.00s system 98% cpu 13.597 total
```

```
size of account_info is 72; size of int is 4; size of void pointer is
8
Account: 0; name: test_account0; balance: 0
Account: 1; name: test_account1; balance: 0
Account: 2; name: test_account2; balance: 0
Account: 3; name: test_account3; balance: 0

I am lender # 0
I am borrower # 0
I am borrower # 1
I am lender # 2
I am lender # 1
I am borrower # 3
I am lender # 3
I am borrower # 2

Account: 0; name: test_account0; balance: 4000000000
Account: 1; name: test_account1; balance: 4000000000
Account: 2; name: test_account2; balance: 4000000000
Account: 3; name: test_account3; balance: 4000000000
./simulate_multithread 105.02s user 21.86s system 306% cpu 41.461
total
```

ซึ่งให้ผลลัพธ์ถูกต้อง