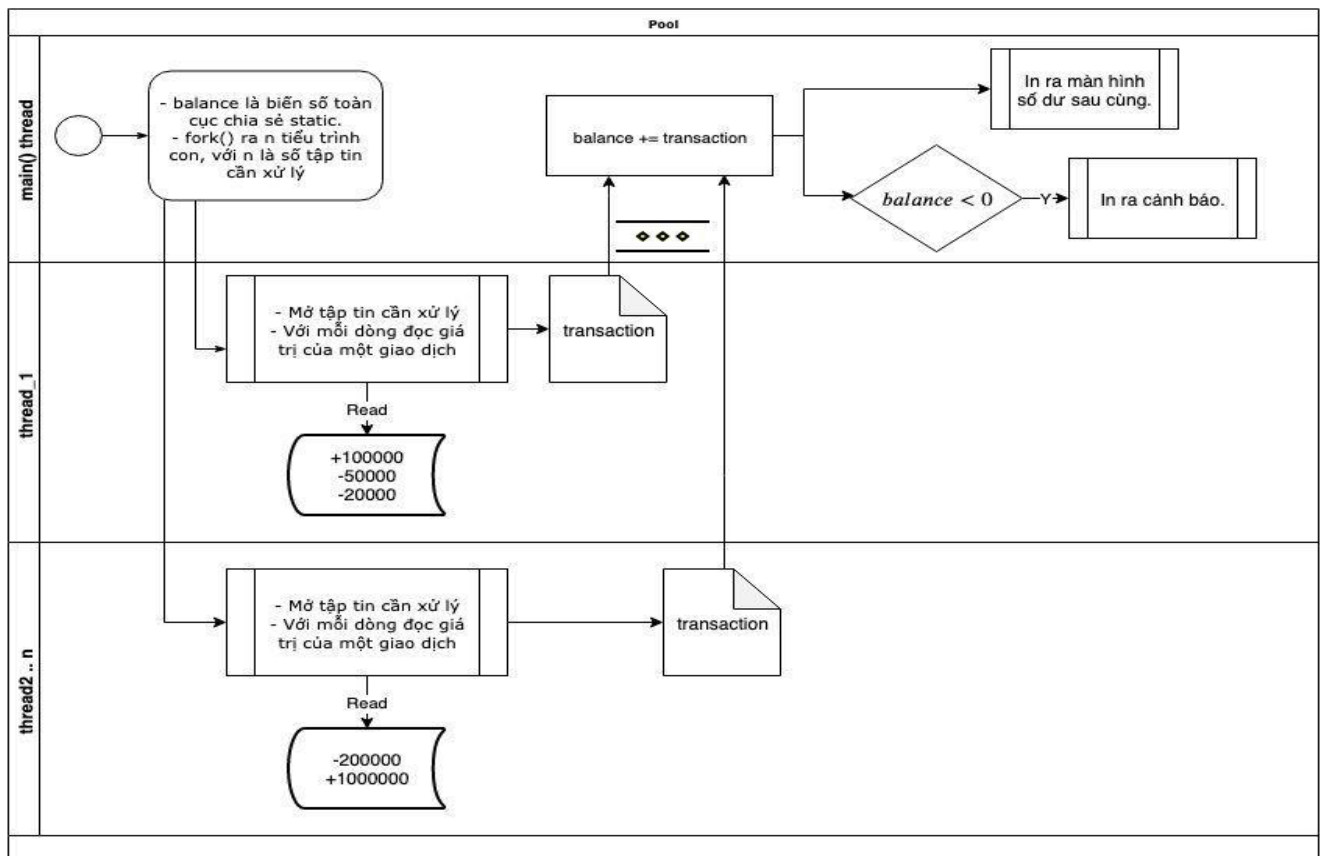


HỆ ĐIỀU HÀNH

SEMAPHORE

Bài 1:

Tình trạng cạnh tranh (race condition) có thể xuất hiện trong nhiều hệ thống máy tính. Hãy xem xét một hệ thống ngân hàng duy trì số dư tài khoản với hai hàm thực thi: deposit(số tiền) và withdraw(số tiền). Hai hàm này được truyền vào số tiền sẽ được gửi hoặc rút từ số dư tài khoản ngân hàng. Giả sử rằng người chồng và người vợ chia sẻ một tài khoản ngân hàng. Một cách đồng thời, người chồng gọi hàm withdraw() và người vợ gọi hàm deposit(). Mô tả làm thế nào một tình trạng cạnh tranh có thể xảy đến và làm cách nào để ngăn chặn tình trạng cạnh tranh này xảy ra.



Nội dung tập tin a.txt

+10 + thể hiện gửi tiền vào
-5 - thể hiện rút tiền ra
+20

Nội dung tập tin b.txt

+20 + thể hiện gửi tiền vào
-8 - thể hiện rút tiền ra

-10

Nội dung tập tin c.txt

-20

+ thể hiện gửi tiền vào

+50

- thể hiện rút tiền ra

Khi thực hiện:

```
>./bank.out a.txt b.txt
```

```
Final Balance = 27
```

```
>./bank c.txt
```

```
Warning: Balance under 0.00
```

```
Final Balance = 30
```

Bài 2:

Tình trạng cạnh tranh (race condition) có thể xuất hiện trong nhiều hệ thống máy tính. Hãy xem xét một hệ thống đấu giá trực tuyến trong đó giá đấu cao nhất hiện thời cho mỗi mặt hàng phải được duy trì. Một người muốn đặt giá đấu cho một mặt hàng sẽ gọi hàm `bid(số_tiền)`, hàm này sẽ so sánh số tiền đang được đặt giá đấu với giá đấu cao nhất hiện tại. Nếu số tiền vượt quá giá đấu cao nhất hiện thời, giá đấu cao nhất sẽ được đặt thành số tiền mới. Điều này được minh họa dưới đây:

Giả sử rằng có nhiều tập tin mà mỗi tập tin (đại diện cho một người) chứa các số tiền của mỗi lần đặt lệnh. Chương trình cần tạo ra nhiều tiểu trình, mỗi tiểu trình sẽ đọc một tập tin và các tiểu trình đồng thời đặt lệnh đấu giá, có các tình huống sau có thể xảy ra:

- Giá đấu cao hơn giá hiện tại (giá hiện tại là biến số chia sẻ và khởi tạo là 0): giá hiện tại sẽ được cập nhật và tên người thắng cũng sẽ được cập nhật (là tên của tập tin chứa giá đấu đang được xử lý)

- Giá đấu thấp hơn hay bằng giá hiện tại: không cập nhật gì cả.

- Giá đấu cao hơn giá hiện tại nhưng nếu giá hiện tại cao nhất cũng là của người đấu này thì tiểu trình này bị chặn lại cho đến khi người thắng được cập nhật. (Không ai tự bỏ giá cao hơn giá vừa bỏ liền trước đó).

Nội dung tập tin andy.txt

10

Giá đấu lần đầu

50

Giá đấu tiếp theo

60

Giả sử rằng giá đấu luôn dương và tăng dần.

100

Hàng cuối cùng thể hiện giá cao nhất mà người này có thể đấu.

Nội dung tập tin ben.txt

15

40

Lời gọi:

```
>./bid.out andy.txt ben.txt
Value on bid:
Andy 10
Ben 15
Andy 50
Ben 40
Andy 60
The winner is Andy with value 60
```