Đồng bộ hóa tiến trình
Nền tảng

- Truy nhập đồng thời đến dữ liệu chia sẻ có thể gây ra sự không nhất quán về dữ liệu → Cần các kĩ thuật để việc thực thi tuần tự của các tiến trình cộng tác
- Vấn đề cộng tác tiến trình (phần1)
+ Bài toán "Sản xuất – Tiêu dùng"
+ Bộ nhớ chia sẻ (có giới hạn)
+ Biến count lưu số các items trong buffer

Mã nguồn cho Producer và Consumer

• Producer

```
while (true)
{
//produce an item
while (count == BUFFER_SIZE);
// do nothing
buffer [in] = nextProduced;
in = (in + 1) % BUFFER_SIZE;
count++;
}
```

• Consumer

```
while (1)
{
while (count == 0)
; // do nothing
nextConsumed= buffer[out];
out = (out + 1) %
BUFFER_SIZE;
count--;
/*consume theitem in
}
```

Chạy đua (Race condition)

count++ có thể được thực hiện như sau:

```
register1 = count
register1 = register1 + 1
count = register1
```

count - - có thể được thực hiện như sau:

```
register2 = count
register2 = register2 - 1
count = register2
```

Giả sử ban đầu "count = 9":

```
S0: producer thực thi
    register1 = count          {register1 = 9}
S1: producer thực thi
    register1 = register1 + 1 {register1 = 10}
S2: consumer thực thi
    register2 = count          {register2 = 9}
S3: consumer thực thi
    register2 = register2 - 1 {register2 = 8}
S4: producer thực thi
    count = register1          {count = 10}
S5: consumer thực thi
    count = register2          {count = 8}
```

Producer thì tưởng count = 10 nhưng consumer thì thấy count = 8;

- Tình huống nhiều tiến trình cùng truy cập và xử lý dữ liệu dùng chung đồng thời. Giá trị cuối cùng của dữ liệu phụ thuộc vào tiến trình sẽ kết thúc cuối.
Để tránh việc chạy đua, các tiến trình đồng thời cần phải được đồng bộ hóa

Vấn đề "Đoạn tới hạn" (Critical section)

- n tiến trình cùng muốn truy cập đến dữ liệu chia sẻ
- Mỗi tiến trình có một đoạn mã gọi là "đoạn tới hạn" (miền găng), trong "đoạn tới hạn" dữ liệu chia sẻ mới được truy cập tới.
- Vấn đề - đảm bảo rằng khi một tiến trình đang trong đoạn tới hạn của nó, các tiến trình khác không được phép thực thi đoạn tới hạn của chúng.

Giải pháp cho vấn đề "đoạn tới hạn"

1. Loại trừ lẫn nhau (độc quyền truy xuất). Nếu tiến trình P_i đang trong đoạn tới hạn, không tiến trình nào khác được phép ở trong đoạn tới hạn.
2. Phát triển. Nếu không tiến trình nào ở trong đoạn tới hạn và có một số tiến trình muốn vào đoạn tới hạn, việc lựa chọn một tiến trình vào đoạn tới hạn sẽ không bị trì hoãn mãi.
3. Chờ đợi có cận. Phải có một cận về số lần mà các tiến trình khác được phép vào đoạn tới hạn sau khi một tiến trình đã yêu cầu vào đoạn tới hạn và trước khi yêu cầu đó được đáp ứng.
+ Giả thiết rằng mỗi tiến trình thực thi với tốc độ khác không
+ Không có giả thiết về mối tương quan tốc độ của n tiến trình

Giải pháp của Peterson

- Giải pháp đồng bộ cho hai tiến trình
- Giả sử hai lệnh LOAD và STORE là các lệnh nguyên tử (thao tác không thể phân chia)
- Các tiến trình chia sẻ các biến sau:
+ int turn;
+ Boolean flag[2];
- Biến turn cho biết tiến trình nào được vào đoạn tới hạn.

- Mảng flag chỉ xem liệu một tiến trình có sẵn sàng vào đoạn tới hạn hay không.
+ flag[i] = true là tiến trình P_i sẵn sàng vào đoạn tới hạn

Đồng bộ nhờ phần cứng

- Nhiều hệ thống cung cấp sự hỗ trợ của phần cứng cho vấn đề đoạn tới hạn
- Các hệ đơn xử lý – có thể bỏ chức năng ngắt
+ Mã được thực thi mà không có sự chiếm đoạt
+ Thông thường không hiệu quả trên các hệ thống đa xử lý: Các hệ điều hành với chiến lược này thường không khả cỡ

- Các máy tính hiện đại cung cấp các lệnh phần cứng nguyên tử (Nguyên tử = không phân chia)
+ Kiểm tra từ nhớ và thiết lập giá trị
+ Tráo đổi nội dung của hai từ nhớ

Semaphores

3 bài toán đồng bộ điển hình

Monitors