## TRUÒNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN, ĐHQG-HCM

## BÀI TẬP ÔN TẬP

Môn học: HỆ ĐIỀU HÀNH

Thời gian: 75 phút

Ngày: 24/11/2021

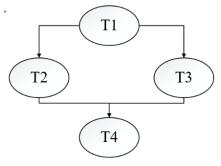
GROUP INFO	
Điểm	NAMEs: Hồ Hồng Hà
	IDs: 20520480

Bài tập 1. Xét hai tiến trình sau: B

process A {while (TRUE) na = na +1;}
process B {while (TRUE) nb = nb +1;}

- a. Đồng bộ hóa xử lý của 2 tiến trình trên, sử dụng 2 semaphore tổng quát, sao cho tại bất kỳ thời điểm nào cũng có nb  $\leq$ = na  $\leq$ = nb +10.
- b. Nếu giảm điều kiện chỉ còn là na <= nb +10, cần sửa chữa giải pháp trên như thế nào?
- c. Giải pháp trên còn đúng nếu có nhiều tiến trình loại A và B cùng thực hiện?

**Bài tập 2.** Xét một hệ thống có 4 tiểu trình T1, T2, T3, T4. Quan hệ giữa các tiểu trình này được biểu diễn như sơ đồ bên dưới, với mũi tên từ tiểu trình (Tx) sang tiểu trình (Ty) có nghĩa là tiểu trình (Tx) phải kết thúc quá trình hoạt động của nó trước khi tiểu trình (Ty) bắt đầu thực thi. Giả sử tất cả các tiểu trình đã được khởi tạo và sẵn sàng để thực thi. Hãy sử dụng semaphore để đồng bộ hoạt động của các tiểu trình sao cho đúng với sơ đồ đã cho.



Bài tập 3. Một biến X được chia sẻ bởi 2 tiến trình cùng thực hiện đoạn code sau:

Bắt đầu với giá trị X=0, chứng tỏ rằng giá tri X có thể vượt quá 20. Cần sửa chữa đoạn chương trình trên như thế nào để đảm bảo X không vượt quá 20?

## Bài tập 4. Xét 2 tiến trình xử lý đoạn chương trình sau:

```
process P1 {A1; A2 }
process P2 {B1; B2 }
```

Đồng bộ hóa hoạt động của 2 tiến trình này sao cho cả A1 và B1 đều hoàn tất trước khi A2 và B2 bắt đầu.

```
Bài tập 1:
a)
S_1 = 0;
S_2 = 10;
Process A{
while (1)
wait (S_2);
na = na + 1;
signal (S_1);
Process B
while (1)
wait (S_1);
nb = nb + 1;
signal (S_2);
}
b)
s = 10;
Process A{
while (1)
wait (s);
```

na = na + 1;

```
}
Process B{
while (1)
nb = nb + 1;
signal(s);
}
c) Đúng vì nếu có nhiều tiến trình loại A và B cùng thực hiện nhưng chỉ tác động đến 2 biến
semaphore toàn cục ở câu a và chỉ 1 semaphore ở câu b.
Bài tập 2:
synch.value = 0;
processP1{
T1;
signal(synch);
processP2{
wait(synch);
T2;
}
processP3{
wait(synch);
T3;
}
Bài tập 3:
Do X được chia sẻ chung ở cả hai tiến trình và chỉ bi reset khi X == 20 nên X có thể vượt quá
20 khi: Có một lý do trong máy đột ngột khiến tiến trình 2 dừng lại. Sau đó tại tiến trình 1, khi
X = 19 thì tiến trình 2 được release đúng ngay đoạn X = X + 1, và cộng dồn với X = 20 sau
lênh
X = X + 1 của tiến trình 1 dẫn tới X vượt quá 20 và còn bị cộng tới vô cùng.
Cách sửa:
Tạo một Semaphore Mutex, khởi tạo giá trị Value bằng 1
while (TRUE)
wait(Mutex);
```

```
X = X + 1;
if (X == 20) X = 0;
signal(Mutex);
}
Bài tập 4:
semaphore s1.v = s2.v = 0
processP1{
       A1;
    signal(s1);,
    wait(s2);
    A2;
}
processP2{
    B1
    signal(s2);
    wait(s1);
    B2;
}
```