

Hướng dẫn Lab 7.1 – Tính nhất quán của dữ liệu

Khi nhiều tiến trình hay tiểu trình cùng thực thi, rủi ro về tính đồng nhất dữ liệu bị đe dọa! Thứ nhất, khi những vùng nhớ chia sẻ hay biến số dùng chung hay các tập tin cùng được ghi bởi nhiều luồng xử lý, tình trạng cạnh tranh xuất hiện. Thứ hai, trong các thao tác tính toán lệ thuộc, trình tự thực thi công việc thật sự quan trọng, nếu như các luồng công việc không được đồng bộ, kết quả mong đợi sẽ không thể đến.

Trong LAB này, các tình huống sai sót được giới thiệu và thảo luận.

Mục tiêu	Lý thuyết liên quan	Tài nguyên
Bài toán Sản xuất –Tiêu Thụ	Ch6.1 Sync tools: background.	https://github.com/Trantin84/LAB_IntroOS (mã nguồn ví dụ).
Bài toán ước lượng giá trị số PI		Sử dụng image Ubuntu 16 / 18
Đồng bộ trong Python+		Sử dụng Windows 10 Python 3.9

Yêu cầu sinh viên: Hiểu lý thuyết về Điều kiện cạnh tranh và vùng nguy cơ, những rủi ro sai sót khi thao tác dữ liệu với nhiều luồng xử lý. Phân tích vùng nguy cơ và áp dụng các giải pháp để đồng bộ tiểu trình.

Đánh giá sinh viên: Hỏi đáp các vấn đề lý thuyết. Kỹ năng thực hành. Bài tập.

Yêu cầu nộp bài: các tập tin mã nguồn .c và tập tin khả thực thi .out của các Yêu cầu trong buổi thực hành và Bài tập cuối hướng dẫn trong thời gian cho phép của giảng viên.

Preferences

[1] Abraham Silberschatz, Peter B. Galvin, Greg Gagne, [2018], Operating System Concepts, 10th edition, John Wiley & Sons, New Jersey.

Programming Problems of Chapter 6.

[2] GeeksforGeeks, [2021], Multithreading in Python | Set 2 (Synchronization)

Access <https://www.geeksforgeeks.org/multithreading-in-python-set-2-synchronization/>

Bài toán Producer – Consumer.

Yêu cầu 1: Mô tả tính nhất quán dữ liệu bị vi phạm trong bài toán P-C.

Hướng dẫn: Tải về tập tin *LAB_IntroOS/LAB_7/task1_1_PC.c*, biên dịch và thực thi.

- Dòng 25 đến 28 là dữ liệu chia sẻ, đã trình bày tại Slide Ch6, slide 5, 6; và sách [1] p. 258.
- Dòng 56 đến 59 là thao tác ghi dữ liệu (ở đây là số nguyên giá trị 1) vào bộ nhớ đệm. Thao tác này thực hiện max lần, với max được truyền qua lời gọi thực thi.
- Dòng 66 đến 70 là thao tác đọc dữ liệu và cộng dồn vào biến receive.

25	<code>int count = 0;</code>
26	<code>int in = 0;</code>
27	<code>int out = 0;</code>
28	<code>int buffer[BUFFER_SIZE];</code>
56	<code>while (count == BUFFER_SIZE); //do nothing</code>
57	<code>buffer[in] = 1;</code>
59	<code>count++;</code>
66	<code>while (count == 0); //do nothing</code>
67	<code>receive += buffer[out];</code>
68	<code>count--;</code>
70	<code>out = (out + 1) % BUFFER_SIZE;</code>

- Hãy thực thi với các đối số: 10, 100, 10000, 100000 và ghi lại kết quả quan sát. SV kết luận gì từ giá trị nhận được?

```
$ gcc -o pc.out task1_1_PC.c -lpthread
$ ./pc.out 10
$ ./pc.out 100
$ ./pc.out 10000
$ ./pc.out 100000
```

Bài toán ước lượng số PI

Yêu cầu 2: Mô tả tính nhất quán dữ liệu bị vi phạm trong bài toán Ước lượng số PI.

Hướng dẫn: Tải về tập tin *LAB_IntroOS/LAB_7/task1_2_PI.c*, biên dịch và thực thi. Phần code này đã giải thích trong LAB 6.1

- SV chạy các lời gọi sau và nhận xét về độ chính xác của số PI.

```
$gcc -o pi.out task1_2_PI.c -lpthread
$./pi.out 1 100
$./pc.out 1 100000
$./pc.out 2 50000
$./pc.out 10 10000
$./pc.out 10 100000
```

Bài toán ước lượng số PI trên ngôn ngữ Python (SV tự học tại nhà)

Yêu cầu 3: Vấn đề đồng bộ có xảy ra trong các ngôn ngữ khác không?

Hướng dẫn: Tải về tập tin *LAB_IntroOS/LAB_7/task6_1_PI.py*, và chạy trên môi trường Python. Đánh giá độ sai số của số PI. Số tiểu trình mặc định là 2, số điểm sinh ra trong mỗi tiểu trình là giá trị biến seed.

Sau đó tải về tập tin *LAB_IntroOS/LAB_7/task6_1_PI_sync.py* chạy và so sánh.

Bài tập lập trình.

Sinh viên nộp các đoạn mã và chương trình đã biên dịch từ các Yêu cầu nêu trên.