

## Hướng dẫn Lab 5.4 – Hàng đợi thông điệp

Trong khi các đường ống gửi các luồng byte không có cấu trúc, các thông điệp được gửi dưới dạng các kí tự riêng biệt. Khi một thông báo được truy xuất từ hàng đợi, tiến trình sẽ nhận được toàn bộ một thông báo; việc gửi các cấu trúc dữ liệu phức tạp cũng khó thực hiện được.

Với phương pháp Vùng nhớ chia sẻ, một đối tượng được lưu trữ vào một vị trí trong vùng nhớ và nhiều (từ 2 đến n) tiến trình có thể cùng đọc / ghi trên vùng nhớ tạo thành cơ chế liên lạc phù hợp có tính chất “gián tiếp”. Tuy vậy, các tiến trình cần nắm rõ các vị trí cần đọc / ghi, cũng như các đối tượng được lưu trữ có độ ưu tiên như nhau.

Hàng đợi thông điệp có siêu dữ liệu được liên kết cho phép các tiến trình chỉ định thứ tự nhận tin nhắn. Có nghĩa là, hàng đợi thông điệp không yêu cầu hoặc không đảm bảo thứ tự xuất trước, nhập trước. Các hàng đợi thông điệp tồn tại ở mức hạt nhân và sử dụng các lời gọi đặc biệt để loại bỏ chúng. Việc kết thúc tiến trình sẽ không loại bỏ hàng đợi thông điệp.

Mục tiêu	Lý thuyết liên quan	Tài nguyên
Hàng đợi thông điệp	Ch3: Process	<a href="https://github.com/Trantin84/LAB_IntroOS">https://github.com/Trantin84/LAB_IntroOS</a> (mã nguồn ví dụ). Sử dụng image Ubuntu 16 / 18 Sử dụng Windows 10
Bài toán Producer - Consumer	Ch3.6: Message Passing system	
MQ trong System-V		

**Yêu cầu sinh viên:** Hiểu lý thuyết về liên lạc giữa các tiến trình. Hiểu và thực thi các đoạn mã đã cung cấp. Áp dụng cho các bài toán liên lạc giữa các tiến trình bằng Hàng đợi thông điệp. Phân tích và so sánh Hàng đợi thông điệp với các phương pháp liên lạc khác.

**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 3, Chapter 6.*

[2] Greg Gagne , [2019], GitHub OS-BOOK OSC10e, Westminster College, United States

Access <https://github.com/greggagne/osc10e> in September 2019.

[3] Compare System-V and POSIX.

Access <https://stackoverflow.com/questions/4582968/system-v-ipc-vs-posix-ipc>

[4] Michael S. Kirkpatrick, [2021], OpenCFS - Chapter 3.6. Message Passing With Message Queues,

Access <https://www.opencsf.org/Books/csf/html/MQueues.html>

[5] Linux manual page, [2021], on `mq_open(3)`

Access [https://man7.org/linux/man-pages/man3/mq\\_open.3.html](https://man7.org/linux/man-pages/man3/mq_open.3.html)

**Yêu cầu 1:** Tiến trình con đọc vào 2 số nguyên từ đối số truyền, ghi vào SM, tiến trình cha thực hiện tính tổng và ghi lại vào SM. Tiến trình con đọc kết quả và xuất ra màn hình

**Hướng dẫn:** Tải về tập tin `LAB_IntroOS/LAB_5/task4_1sender.c` và `/task4_1reader.c`, bao gồm cả file header `/task4_1queue.h` rồi biên dịch và thực thi trên 2 terminal khác nhau.

```
>gcc -o send task4_1sender.c -lrt
>gcc -o read task4_1reader.c -lrt
>./send
>./read
```

### a. Giải thích mã nguồn SENDER

- Dòng 8 để bao gồm tập tin .h chứa mô tả gói tin và khoá cho liên lạc.
- Dòng 10 là thư viện MQ.
- Dòng 25 mở ra kênh liên lạc mới cho phép tiến trình này GHI.
- Dòng 33 đến 40 là các câu cần gửi đi với độ ưu tiên chỉ định tại dòng 41 đến 48.
- Dòng 55 là lời gọi gửi thông điệp (lặp lại cho mỗi câu).
- Dòng 57 sẽ đóng Hàng đợi, lưu ý rằng tiến trình sender kết thúc nhưng Hàng đợi vẫn tồn tại ở mức nhân.

8	#include "task4_1queue.h"
---	---------------------------

10	#include <mqueue.h>
25	mqd_t mqd = mq_open( key, O_CREAT   O_WRONLY, 0666, NULL);
33	char * payloads[] = {
40	};
41	int priority[] = {
48	}; /* each must be >= 0 */
55	mq_send(mqd, (const char * ) & msg, sizeof(msg), msg.priority);
57	mq_close(mqd);

### b. Giải thích mã nguồn READER

- Dòng 28 là mở Hàng đợi để ĐỌC, với khoá cung cấp.
- Dòng 42 là nhận một thông điệp, việc này lặp lại đủ số lần bằng số lần gửi.
- Dòng 50 là đóng Hàng đợi thông điệp.

28	mqd_t mqd = mq_open( key, O_RDONLY);
42	mq_receive(mqd, (char * ) & msg, attr.mq_msgsize, & priority)
50	mq_close(mqd);

## Bài tập lập trình.

1. Tiến trình cha đọc hai số nguyên và một thao tác +, -, \*, / và chuyển tất cả cho tiến trình con bằng Hàng đợi thông điệp. Tiến trình con tính toán kết quả và trả về cho tiến trình cha để in ra màn hình.

```
>./baitap3A.out 4 6 +
4 + 6 = 10
```