

Hệ thống Quản lý Hàng đợi Tác vụ và Nhật ký Hoạt động

Thực hành triển khai và sử dụng cả danh sách liên kết đơn (**singly linked list**) và danh sách liên kết đôi (**doubly linked list**) để giải quyết hai bài toán khác nhau trong cùng một ứng dụng nhúng. Qua đó, bạn sẽ hiểu rõ khi nào nên dùng mỗi loại và ưu nhược điểm giữa chúng.

Hãy tưởng tượng bạn đang lập trình cho một bộ điều khiển trung tâm. Hệ thống này cần thực hiện các tác vụ (ví dụ: đọc cảm biến, điều khiển động cơ) theo thứ tự "vào trước - ra trước" (FIFO). Đồng thời, nó cũng cần ghi lại nhật ký các hoạt động đã thực hiện và cho phép người dùng xem lại lịch sử đó, di chuyển tới lui trong các danh sách.

- Hàng đợi tác vụ (Task Queue) sẽ sử dụng danh sách liên kết đơn.
- Nhật ký hoạt động (Activity Log) sẽ sử dụng danh sách liên kết đôi.

1. Task Queue - Danh sách Liên kết Đơn

Trong phần này, bạn sẽ xây dựng một hàng đợi FIFO đơn giản. Các tác vụ mới sẽ được thêm vào cuối hàng đợi (tail) và tác vụ cần xử lý sẽ được lấy ra từ đầu hàng đợi (head). Danh sách liên kết đơn là lựa chọn hoàn hảo vì chúng ta chỉ cần duyệt theo một chiều.

1.1. Định nghĩa cấu trúc Node

Tạo cấu trúc TaskNode cho danh sách liên kết đơn.

```
typedef struct TaskNode {  
    char task_description[50];  
    struct TaskNode* next;  
} TaskNode_t;
```

1.2. Triển khai các hàm quản lý hàng đợi

1. void queue_add_task(const char* description);
 - Hàm này tạo một TaskNode mới.
 - Thêm node mới vào cuối danh sách.
2. TaskNode_t* queue_get_next_task(void);
 - Lấy node ở đầu danh sách để xử lý.

- Xóa node đó khỏi danh sách và trả về con trỏ tới nó (để có thể giải phóng bộ nhớ sau khi xử lý). Nếu hàng đợi rỗng, trả về NULL.
3. void print_task_queue(void);
 - In tất cả các tác vụ đang chờ trong hàng đợi.
 4. Bổ sung thêm các hàm cần thiết ...

2. Activity Log - Danh sách Liên kết Đôi

Sau khi một tác vụ được "thực thi" (lấy ra từ hàng đợi ở Phần 1), một mục nhật ký sẽ được tạo ra. Người dùng cần có khả năng xem lại các mục nhật ký này, di chuyển tới mục trước đó hoặc kế tiếp. Do đó, danh sách liên kết đôi là bắt buộc.

2.1. Định nghĩa cấu trúc Node

Tạo cấu trúc HistoryNode cho danh sách liên kết đôi.

```
typedef struct HistoryNode {
    char log_entry[50];
    struct HistoryNode* next;
    struct HistoryNode* prev; // Con trỏ tới node trước đó
} HistoryNode_t;
```

2.2. Triển khai các hàm quản lý nhật ký

1. void history_log_activity(const char* entry);
 - Hàm này tạo một HistoryNode mới và thêm nó vào đầu danh sách nhật ký (các hoạt động mới nhất sẽ ở đầu).
2. void history_navigate(void);
 - Đây là hàm tương tác chính. Nó cho phép người dùng nhập lệnh từ console (ví dụ: n cho "next", p cho "previous", q cho "quit").
 - Bạn cần một con trỏ currentNode để theo dõi vị trí hiện tại trong danh sách nhật ký.
 - Khi người dùng nhập n, di chuyển currentNode tới currentNode->next.
 - Khi người dùng nhập p, di chuyển currentNode tới currentNode->prev.
 - Sau mỗi lần di chuyển, in ra nội dung của currentNode. Phải xử lý trường hợp khi đang ở đầu hoặc cuối danh sách.

3. Yêu cầu triển khai và Tích hợp

1.1. Viết chương trình main để mô phỏng

1. Chương trình cho phép người dùng nhập lệnh, ví dụ:
 - o add <mô tả công việc>: Gọi hàm queue_add_task().
 - o run: Gọi queue_get_next_task(). Khi một tác vụ được lấy ra, hãy in ra là nó đang được thực thi và ngay lập tức gọi history_log_activity() để ghi lại sự kiện này. Đừng quên free() bộ nhớ của node tác vụ.
 - o history: Gọi history_navigate() để bắt đầu duyệt nhật ký.
 - o list: Gọi print_task_queue().
2. Quản lý bộ nhớ: Đảm bảo rằng bạn luôn giải phóng bộ nhớ (free()) cho các node đã được xử lý hoặc xóa để tránh rò rỉ bộ nhớ.

4. Câu hỏi phân tích

Sau khi hoàn thành bài tập, hãy tự trả lời các câu hỏi sau:

1. Tại sao danh sách liên kết đơn là đủ và hiệu quả cho hàng đợi tác vụ FIFO?
2. Điều gì sẽ xảy ra nếu bạn cố gắng triển khai chức năng maps_history_backward() (di chuyển về phía trước) trong Phần 2 chỉ bằng danh sách liên kết đơn? Thao tác đó sẽ phức tạp và tốn kém về hiệu năng như thế nào?
3. Sự đánh đổi chính về mặt bộ nhớ khi sử dụng danh sách liên kết đôi so với danh sách liên kết đơn là gì?