

Bài tập lập trình hệ thống

Người lập: **Phạm Công Phan – PM1 - ESOF**

Bài tập 1: Viết chương trình, có tên là “tail”, chương trình sẽ nhận đối số input trên dòng lệnh là 1 tên file (text file), và in ra màn hình **n** dòng cuối cùng của file đó. Cho giả thiết là mỗi dòng trong file input không quá 256 kí tự.

*Lưu ý: bài tập cần phát triển theo tư tưởng thiết kế hướng đối tượng, sử dụng lại được.

Bài tập 1 mở rộng:

Làm bài tập 1 với giả thiết bỏ giới hạn tối 256 kí tự mỗi dòng của file input.

Hướng dẫn cách làm: Sử dụng các kỹ thuật như file mapping hoặc random access file.

Bài tập 2: Viết chương trình sắp xếp 1 dãy số nguyên theo thứ tự tăng dần và ghi kết quả ra 1 file text có tên là **output.txt**, với các yêu cầu:

- Dãy số nguyên đầu vào được cho trong 1 file text có tên là **input.txt**, mỗi số nguyên phân cách nhau bởi kí tự backspace hoặc kí tự xuống dòng.
- Chương trình sử dụng **n** thread để sắp xếp đồng thời dãy số nguyên đó (vì số lượng phần tử của dãy số nguyên có thể là rất lớn, sử dụng multithreading để tối ưu tốc độ trên các CPU hỗ trợ xử lý multithreading), **n** sẽ được đưa vào trong tham số dòng lệnh.
- Gợi ý giải thuật: Chia dãy số input thành **n** dãy con, mỗi dãy con sau đó sẽ được sắp xếp bởi 1 thread tương ứng. Sau khi **n** thread đã sắp xếp xong **n** dãy con ta thu được **n** dãy con đã có thứ tự, khi đó ta sẽ dễ dàng trộn **n** dãy con đó thành 1 dãy hoàn chỉnh có thứ tự tăng dần.

*Lưu ý:

- Các tham số cấu hình phải để ngoài file config, có thể cấu hình lại được (ko fix code).
- Lập tài liệu statistic, benchmark: khi sử dụng các tham số test khác nhau (ví dụ config 1 thread, 2 thread, 4 thread ..., dữ liệu 1M, 2M, 10M ...)

Bài tập 3: Viết chương trình giải quyết bài toán producer & consumer với các yêu cầu sau:

- Có một message queue chứa các message, cấu trúc message là tùy chọn. Message queue có size giới hạn.
- Một thread đóng vai trò producer: producer định kỳ sẽ tạo ra một message và đưa vào message queue, nếu msgq đã full thì thread sẽ phải đợi cho tới khi msgq không ở trong trạng thái full và tiếp tục tạo message mới đưa vào queue.
- Một thread đóng vai trò consumer: mỗi lần sẽ lấy ra 1 message từ message queue và in message ra màn hình, nếu message queue đang empty thì sẽ phải đợi cho tới khi có message trong queue để xử lý.

***Lưu ý:** cần thể hiện được kết quả và ý tưởng, ý nghĩa bài toán qua logs trên màn hình test, sao cho dễ hiểu, thể hiện đúng ý đồ bài toán nêu ra.

Bài tập 4: Thread Synchronization

Xây dựng chương trình với các yêu cầu sau: Chương trình sẽ sắp xếp 1 mảng số nguyên cho trước, dữ liệu input sẽ được đọc ra từ file input.txt, các dữ liệu cần tìm kiếm được đọc vào từ file search.txt.

Chương trình có 3 thread A, B, C chạy đồng thời (concurrently running). Thread B sẽ thực hiện sắp xếp mảng input, Thread C thực hiện tìm kiếm trên mảng output đã được sắp xếp và Thread A hiển thị kết quả đã tìm kiếm.

***Lưu ý:** cần thể hiện được kết quả và ý tưởng, ý nghĩa bài toán qua logs trên màn hình test, sao cho dễ hiểu, thể hiện đúng ý đồ bài toán nêu ra.

Bài tập 5: Viết chương trình trao đổi file giữa Client – Server, client có thể send file lên server hoặc get file xuống từ server, yêu cầu cụ thể:

- Server có thể đáp ứng được nhiều client đồng thời.
- Client có thể gửi request `sendFile(filename)` để gửi một file bất kỳ lên server:
 - Nếu trên server đã tồn tại 1 file trùng tên, server sẽ gửi response từ chối nhận file tới client
 - Nếu trên server không có file trùng tên server sẽ gửi response OK thông báo chấp nhận file xuống client và bắt đầu nhận dữ liệu file được truyền từ client và lưu trữ dữ liệu dưới tên file nhận được từ client (trong lệnh `sendFile` request) vào thư mục hiện thời.
- Client có thể gửi request `getFile(filename)` với 1 filename cụ thể để yêu cầu get dữ liệu file từ server, server sẽ kiểm tra tên file trong thư mục hiện thời:
 - Nếu trên server không tồn tại file với tên yêu cầu, server gửi response từ chối truyền file.
 - Nếu tồn tại file với tên được yêu cầu server sẽ gửi response OK và thực hiện truyền dữ liệu file xuống client.

*Lưu ý:

- Server cần đáp ứng được nhiều client cùng kết nối đồng thời.
- Các tham số có thể config được phải để riêng ra file config, ko fix cứng trong code
- Với bài tập này cần đầy đủ các tài liệu hệ thống: Tài liệu đặc tả yêu cầu, tài liệu đặc tả chức năng, tài liệu thiết kế module (thông qua các sequence diagram, biểu đồ lớp thiết kế).

Bài tập 6: Lập trình mạng

Xây dựng chương trình giám sát sự thay đổi và đồng bộ dữ liệu trên các máy client thông qua một server trung tâm, tạm gọi là **Content Monitoring & Synchronizing System (CMSS)**:

Các yêu cầu cho ứng dụng CMSS client, CMSS server: Ứng dụng CMSS client chạy trên các máy client.

- Client sẽ được cấu hình để giám sát nội dung của một thư mục trên máy client, khi có bất kì sự thay đổi nào liên quan đến các file trong thư mục đó (thay đổi nội dung file, thêm file mới, xóa một hoặc nhiều file, đổi tên file...), client sẽ thông báo lên server các thông tin về các file liên quan.
- Server khi nhận được thông báo thay đổi dữ liệu từ một client, sẽ thực hiện gửi lệnh RequestGetFile(filename) xuống client có thay đổi dữ liệu để yêu cầu Client đó truyền dữ liệu các file thay đổi về qua server, đồng thời khi Server nhận được dữ liệu đó sẽ gửi lệnh NewFileUpdate(filename) xuống các Client khác (các client mà chưa có file đó, hoặc đã có file đó nhưng nội dung file đã outOfdate) để đồng bộ dữ liệu giữa các Client.
- Mục đích của hệ thống **CMSS** là để đảm bảo dữ liệu trên các máy có cài đặt CMSS Client sẽ luôn được đồng bộ (*giống nhau*).

* Lưu ý:

- CMSS server phải có khả năng đáp ứng đồng thời được nhiều CMSS client kết nối tới.
- Hệ thống CMSS phải có cơ chế thực hiện xác nhận authentication khi các CMSS client kết nối đến server để phòng tránh các kết nối ko hợp lệ kết nối tới CMSS server.
- Các message (thông điệp) truyền giữa client – server phải là các message có cấu trúc.
- Các hoạt động diễn ra tại client và server phải được ghi log lại để giúp cho việc dễ dàng trace log lại khi hệ thống xảy ra lỗi.
- Các thông tin cấu hình như: tên thư mục mà Client giám sát, server ip, server port... phải được cấu hình trong file config riêng.