**BÁO CÁO THỰC TẬP**

Nội dung thực tập: Làm quen với Linux Programming.

Vấn đề đặt ra : Tìm hiểu luồng trong C (pthread), xây dụng các luồng để lấy được thời gian hiện tại của hệ thống (đơn vị nanoseconds) , lưu trữ kết quả ra file txt cũng như tính toán thời gian chênh lệch giữa hai lần lấy mẫu. Thời gian lấy mẫu được lấy từ file txt.

Giải quyết vấn đề: Tạo ra 3 luồng chính là INPUT (chức năng là đọc thời gian lấy mẫu ) , luồng SAMPLE dùng để lấy ra thời gian hiện tại (nanoseconds) và luồng LOGGING (tính offset giữa hai lần lấy mẫu, xuất kết quả ra file txt).

Chi tiết:

* Sử dụng thư viện time.h và hàm clock\_gettime() để lấy thời gian hiện tại (nanoseconds).
* Sử dụng khóa Mutex ( Để kiểm tra xem chu kỳ X có thay đổi không, nếu có cập nhật lại chu kỳ X cho thread SAMPLE).
* Sử dụng hàm clock\_nanosleep() để thực hiện delay cho mỗi lần lấy mẫu (100ns đến 1000000ns ).

Tiến độ công việc: Hoàn thành

Kết quả đạt được:

* Thu được kết quả tính offset sau mỗi chu kỳ như sau:
  + Với chu kỳ 1000000ns :
    - Thời gian trung bình: 0.001028457(s).
    - Số mẫu thu được trong 60s: 58337 (mẫu).
  + Với chu kỳ 100000ns :
    - Thời gian trung bình: 0.000108783(s).
    - Số mẫu thu được trong 60s: 551514 (mẫu).
  + Với chu kỳ 10000ns :
    - Thời gian trung bình: 0.0000126163(s).
    - Số mẫu thu được trong 60s: Khoảng 1048575 (mẫu).
  + Với chu kỳ 1000ns :
    - Thời gian trung bình: 0. 0000125251(s).
    - Số mẫu thu được trong 60s: Khoảng 1048575(mẫu).
  + Với chu kỳ 100ns :
    - Thời gian trung bình: 0. 0000125703(s).
    - Số mẫu thu được trong 60s: Khoảng 1048575 (mẫu).

Kết luận: Đối với chu kỳ lấy mẫu trong khoảng 1000000ns đến 10000ns, kết quả offset tương đối gần với chu kì lấy mẫu. Đối với chu kỳ 1000ns và 100ns, kết quả đã không còn được chính xác ( chỉnh dừng lại ở khoảng 0,0000125s ).

Khó khăn:

* Chưa quen thuộc với Linux Programming, shell script.
* Kết quả thu được chưa đạt được yêu cầu mong muốn ( xuất hiện mẫu có sai lệch lên đến 0.003s mặc dùng chu kì lấy mẫu là 1000000ns).
* Thuật toán chưa được tối ưu hoàn toàn.

Các lỗi đã gặp phải: Chưa tối ưu thuật toán, dẫn đến sai số thời gian lớn ( Khởi tạo liên tục 3 thread trong mỗi vòng lặp), sử dụng hàm nanosleep() để tạo chu kì gấy ra tốn kém tài nguyên của CPU, làm delay chương trình.

Khắc phục: Sử dụng khóa Mutex để thực hiện các thread một cách hiệu quả, thay thế hàm nanosleep() bằng hàm clock\_nanosleep() (hàm clock\_nanosleep() không làm delay hệ thống, trong lúc thực hiện hàm này, Thread khác có thể sử dụng tài nguyên hệ thống để tính toán offset giữa 2 thời điểm lấy mẫu khác nhau).