

# BÁO CÁO THỰC TẬP

## Linux Programming Assignment

Họ và tên: Nguyễn Mạnh Hùng

1. Viết chương trình C trên Linux chạy 3 thread SAMPLE, LOGGING, INPUT

- Thread SAMPLE thực hiện vô hạn lần nhiệm vụ sau với chu kỳ X ns. Nhiệm vụ là đọc thời gian hệ thống hiện tại (chính xác đến đơn vị ns) vào biến T.

```
void *getTime(void *args )
{
    while (1)
    {
        //unsigned long x = *((unsigned long*)args);
        if (clock_nanosleep(CLOCK_REALTIME,0,&t1,&t2) == 0)
        {
            clock_gettime(CLOCK_REALTIME,&tmp1);
        }
    }
}
```

Em sử dụng hàm `clock_nanosleep` thay cho hàm `nanosleep` vì `nanosleep` làm hệ thống ngủ khi chạy đến lệnh đó còn `clock_nanosleep` thì không.

Hàm `clock_gettime` để lưu thời gian thực của hệ thống vào biến `tmp1`.

- Thread INPUT kiểm tra file “freq.txt” để xác định chu kỳ X (của thread SAMPLE) có bị thay đổi không? Nếu có thay đổi thì cập nhật lại chu kỳ X. Người dùng có thể echo giá trị chu kỳ X mong muốn vào file “freq.txt” để thread INPUT cập nhật lại X.

- Thread LOGGING chờ khi biến T được cập nhật mới, thì ghi giá trị biến T và giá trị interval (offset giữa biến T hiện tại và biến T của lần ghi trước) ra file có tên “time\_and\_interval.txt”.

**Code của phần này đã được em up lên github ạ**

2. Viết shell script để thay đổi lại giá trị chu kỳ X trong file “freq.txt” sau mỗi 1 phút. Các giá trị X lần lượt được ghi như sau: 1000000 ns, 100000 ns, 10000 ns, 1000 ns, 100ns.

```
#!/bin/sh

echo "start"
echo "1000000">freq.txt
timeout 60s ./bai1
echo "100000">freq.txt
timeout 60s ./bai1
echo "10000">freq.txt
timeout 60s ./bai1
echo "1000">freq.txt
timeout 60s ./bai1
echo "100">freq.txt
timeout 60s ./bai1
echo "end"
```

3. Chạy shell script + chương trình C trong vòng 5 phút, sau đó dừng chương trình C.
4. Thực hiện khảo sát file "time\_and\_interval.txt": Vẽ đồ thị giá trị interval đối với mỗi giá trị chu kỳ X và đánh giá.

**Em đã thực hiện khảo sát và đánh giá ở file "Khảo sát.pdf"**

#### **Khó khăn:**

- Chưa quen với Linux Programming Assignment
- Kết quả thu được chưa được như kì vọng, giá trị offset còn sai số nhiều với chu kỳ X bằng 10000 ns, 1000 ns và sai hẳn với  $X = 100\text{ns}$
- Chưa tối ưu được thuật toán