# LẬP TRÌNH ĐA LUỒNG (MULTI THREAD)

## Mục tiêu

Tìm hiểu về luồng (thread) trên Linux, và cơ chế lập trình ứng dụng đa luồng (multithreads).

## Nội dung thực hành

1. **Cơ bản về tạo luồng**

Sử dụng hàm **pthread\_create** để chạy một luồng trong một tiến trình đang chạy.Luồng vừa được tạo sẽ bắt đầu gọi hàm ***start\_routine();*** arg là các tham số được truyền vào hàm này.

**#include <pthread.h>**

**int pthread\_create(pthread\_t \*thread, const pthread\_attr\_t \*attr,**

**void \*(\*start\_routine) (void \*), void \*arg);**

* Ví dụ:

Chương trình tạo một luồng chính liên tục in ký tự ‘o’, và luồng tạo ra liên tục in ký tự ‘x’, kết quả các ký tự o, x xen kẽ phụ thuộc vào các luồng được hệ điều hành lập lịch cho chạy.

**#include <stdio.h>**

**#include <pthread.h>**

**void\* printx(void\* unused)**

**{**

**while(1)**

**{**

**fputc(‘x’, stdout); //Hàm xử lý của luồng in ký tự ‘x’**

**}**

**return NULL;**

**}**

**int main(int argc, char\*\* argv)**

**{**

**pthread\_t thread\_id**

**pthread\_create(&thread\_id, NULL, &printx, NULL); //Tao luong**

**while(1)**

**{**

**fputc(‘o’, stdout); //Luồng chính liên tục in ra ký tự ‘o’**

**}**

**return 0;**

**}**

**b.Truyền dữ liệu cho luồng**

Hàm xử lý của luồng có thể nhận tham số đầu vào dưới dạng *void\** .Trong trường hợp cần truyền nhiều thành phần dữ liệu cho luồng cần tạo một cấu trúc (**struct**), và truyền biến cấu trúc này vào hàm xử lý.

Ví dụ bên dưới tạo ra 2 luồng cùng sử dụng 1 hàm xử lý in một số lượng ký tự ra màn hình, với tham số truyền vào là ký tự và số lượng muốn in.

//VD2: tạo 2 luồng và truyền tham số vào hàm xử lý

**#include <pthread.h>**

**#include <stdio.h>**

**//parameters to print function**

**struct char\_print\_parms**

**{**

**char character; //character to print**

**int count; //times to print**

**};**

**//Ham xu ly cua thread**

**void\* char\_print(void\* params)**

**{**

**Struct char\_print\_parms\* p = struct char\_print\_parms\*) params;**

**int I;**

**int n = p->count;**

**char c = p->character;**

**for(i=0;i<n;i++)**

**fputc(c,stdout);**

**return NULL;**

**}**

**int main(int argc, char\*\* argv)**

**{**

**pthread\_t thread1\_id, thread2\_id;**

**struct char\_print\_parms p1, p2;**

**p1.character = ‘x’;**

**p1.count = 3000;**

**pthread\_create(&thread1\_id, NULL, &char\_print, &p1);**

**p2.character = ‘o’;**

**p2.count = 2000;**

**pthread\_create(&thread2\_id, NULL, &char\_print, &p2);**

**pthread\_join(thread1\_id, NULL);**

**pthread\_join(thread2\_id, NULL);**

**return 0;**

**}**

## Bài tập thực hành

Áp dụng cơ chế đa luồng, viết chương trình ứng dụng trên KIT thực hiện hiệu ứng LED nhấp nháy, sử dụng nút bấm để thay đổi thời gian chớp tắt của LED

Chương trình gồm một luồng chính (master thread) thực hiện hiệu ứng điều khiển LED ứng với thời gian delay thay đổi. Luồng thứ 2 phát hiện sự kiện nhấn nút và thay đổi giá trị thời gian delay chớp tắt.

* Delay time thay đổi từ 1000-3000ms với step là 100ms
* Biên dịch chương trình: gcc -pthread