#include <pthread.h>

#include <stdio.h>

#include <stdlib.h>

#define MAX\_THREADS 32

void \*child\_fn(void \*arg)

{

    int idx = (int) arg;

    printf("I'm thread %d\n", idx);

    fflush(stdout);

    return NULL;

}

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

{

    int nthreads, i;

    pthread\_t child[MAX\_THREADS];

    /\* Syntax \*/

    if (argc < 2) {

        fprintf(stderr, "syntax: test-threads <nthreads>\n");

        exit(1);

    }

    /\* Get number of threads \*/

    nthreads = atoi(argv[1]);

    if (nthreads < 1 || nthreads > MAX\_THREADS) {

        fprintf(stderr, "error: nthreads must be > 0 and < %d\n", MAX\_THREADS);

        exit(1);

    }

    /\* Run \*/

    for (i = 1; i < nthreads; i++)

        pthread\_create(&child[i], NULL, child\_fn, (int \*) i);

    child\_fn((int \*) 0);

    for (i = 1; i < nthreads; i++)

        pthread\_join(child[i], NULL);

    return 0;

}

* Khai báo thư viện :

pthread.h: Thư viện để làm việc với luồng (thread).

stdio.h: Thư viện để sử dụng các hàm nhập/xuất chuẩn.

stdlib.h: Thư viện để sử dụng các hàm hỗ trợ bộ nhớ động và chức năng

* Định nghĩa hằng : số luồng max mà chương trình có thể tạo
* Hàm thực thi luồng con : Hàm này sẽ được thực thi bởi mỗi luồng con. Nó in ra thông điệp chứa số hiệu của luồng và sau đó kết thúc.

Hàm main() : Hàm chính của chương trình

* Xử lý tham số dòng lệnh
* Lấy số lượng luồng từ tham số dòng lệnh
* Kiểm tra giới hạn số luồng
* Tạo và chạy luồng
* Chạy hàm child\_fn trên luồng chính
* Đợi tất cả các luồng con kết thúc
* Kết thúc chương trình

File : report