Bài thực hành số 5

Lập trình thread

Ghi chú:

- Sinh viên nộp bài tại trang web: `www.cse.hcmut.edu.vn/portal`
- File nộp bài đặt tên là: *`lab5.tar.bz2`* (sử dụng lệnh *make pack*)
- Hạn chót nộp bài: *`11:59PM 19/08/2010`*
- SV có thể chỉnh sửa source code mẫu nếu thấy cần thiết
- Mọi gian lận sẽ nhận điểm KHÔNG nếu bị phát hiện

1 Giới thiệu

1.1 Mục tiêu

Viết một chương trình C trên Linux để nhân hai vector sử dụng kỹ thuật lập trình multi-thread.

1.2 Kiến thức cần biết

- Lập trình C trên Linux
- Lập trình multi-thread
- Mutex
- Makefile

1.3 Mô tả chương trình

Đầu vào (Input) của chương trình:

- Hai vector chứa các số nguyên
- Số thread cần sinh ra

Đầu ra (Output) của chương trình:

- Kết quả của việc nhân 2 vector đó.

1.4 Ý tưởng hiện thực chương trình

- Giả sử có n thread
- Chia mỗi vector ra làm n phần bằng nhau
- Mỗi thread sẽ nhận nhiệm vụ tính toán một phần, sau đó từng thread cộng kết quả của mình với kết quả tổng của chương trình

1.5 Một số kỹ thuật lập trình

1.5.1 Xử lý thông số nhập vào từ chương trình

Một chương trình tốt thường cho phép người dùng thiết lập một vài thông số khi chạy chương trình, chẳng hạn

khi thực hiện lệnh:

```
$ ls -R
```

người dùng đã truyền vào thông số `-R` để liệt kê các file và thư mục không chỉ trong thư mục hiện hành mà còn cả những thư mục con của thư mục hiện hành nếu có.

Sau đây là đoạn chương trình mẫu, sử dụng hàm `getopt()` để xử lý thông số -R ở trên:

```
int opt;
extern char *optarg;

while ((opt = getopt(argc, argv, "R")) != EOF) {
    switch (opt) {
        case 'R':
            // Option -R occurs
            // Proccess that option here
            break;
        default:
            // Other options
            break;
    }
}
```

1.5.2 Xử lý file cấu hình

File cấu hình thường ở dạng text và có cấu trúc (đơn giản). Ta thường sử dụng các hàm sau để xử lý những dạng file này:

- fopen: mở một file
- fscanf: lấy dữ liệu với định dạng đã xác định trước
- fgets: lấy dữ liệu với định dạng chưa xác định
- fclose: đóng file đã mở

1.5.3 Kỹ thuật lập trình thread

Tham khảo slide trên lớp

2 Yêu cầu

Chương trình sau khi biên dịch có tên là `mulvector`, hỗ trợ các thông số sau:

- `-h`: Hiển thị thông tin hướng dẫn sử dụng chương trình.
- `-n` *nthreads*: Chọn số thread được sinh ra trong chương trình. Mặc định số thread được sinh ra là 1.
- *file1*: Chọn file input số 1, file này chứa dữ liệu của vector đầu tiên.
- *file2:* Chọn file input số 2, file này chứa dữ liệu của vector thứ hai.

Cú pháp chạy chương trình `mulvector`:

```
mulvector [-h] [-n nthreads] file1 file2
```

Kết quả của chương trình được in ra màn hình.

Lưu ý khi xử lý option:

- Nếu dòng nhập vào không đúng với cú pháp ở trên thì thì báo lỗi, thoát chương trình ngay lập tức.

- Thông tin hướng dẫn sử dụng chương trình chỉ được hiển thị khi người dùng nhập vào dòng lệnh `mulvector -h`

3 Định dạng file input

File input chứa các phần tử của vector tương ứng, tất cả các phần tử này phải cách nhau bởi một khoảng trắng.

Chương trình đọc file input để xác định xem có bao nhiêu phần tử trong vector đó.

Chương trình chỉ quan tâm đến hàng đầu tiên của file input (không xử lý dữ liệu ở các hàng kế tiếp nếu có).

```
11 25 36
```

Ví dụ file input ở trên biểu diễn một vector gồm 3 phần tử là: 11, 25, 36.