

LẬP TRÌNH WINDOWS

BÀI 10: MULTITHREADING TRONG C#

Giảng viên: Lý Anh Tuấn

Email: tuanla@wru.vn

Lý do sử dụng thread

- Xét bài toán:
 - Viết một chương trình tìm tất cả các số nguyên tố, các số chính phương, các số hoàn chỉnh, các số amstrong, các số lẻ, các số chẵn nhỏ hơn 1 tỷ
- Thực hiện:
 - Có thể tìm chúng cùng 1 lúc trong 1 vòng lặp
 - Hoặc lần lượt tìm từng loại số

Lý do sử dụng thread

- Tuy nhiên, làm như vậy dễ thấy máy bị chậm hoặc chương trình bị treo mặc dù máy tính của bạn có đa nhân (2 nhân: Dual Core, 4 nhân: Quad Core, 6 nhân: Hexa Core, 8 nhân: Octo Core, 10 nhân: Deca Core)
- Để tận dụng được hết hiệu năng của máy tính, cần chia nhỏ chương trình ra thành nhiều phần, mỗi phần được xử lý trên một nhân

=> Thread (luồng/tiểu trình)

Cách sử dụng thread

- Chia nhỏ bài toán lớn thành các bài toán con
- Mỗi bài toán nhỏ là một hàm riêng biệt
- Tạo các thread
- Gán cho mỗi thread thực thi một nhiệm vụ

Khai báo và khởi tạo thread

- Khai báo thư viện:

`using System.Threading;`

- Khai báo thread:

`Thread <tên thread>;` //tạo ra một thread rỗng

- Khởi tạo thread:

`<tên thread> = new Thread(new ThreadStart(<tên
hàm thực thi nhiệm vụ>));`

Khai báo và khởi tạo thread

- Một thread mới được khai báo là một đối tượng rỗng
- Hàm khởi tạo thread là một hàm cần truyền vào một delegate **ThreadStart** tham chiếu đến hàm có kiểu void và không có tham số

Ví dụ

//định nghĩa các hàm tìm số nguyên tố, tìm số chính phương

```
static void TimSCP()  
static void TimSNT()
```

```
static void Main(string[] args)  
{  
    Thread timSCP = new Thread(new ThreadStart(TimSCP));  
    Thread timSNT = new Thread(new ThreadStart(TimSNT));  
    timSCP.Start();//gọi thread chạy  
    timSNT.Start();//gọi thread chạy  
    Console.ReadLine();  
}
```

- Lưu ý:
 - Khi một thể hiện của lớp Thread được tạo, nó bắt đầu ở trạng thái chưa khởi động
 - Sử dụng lệnh **Start()** để khởi động thread

Vòng đời của thread

- Vòng đời của một thread được bắt đầu từ khi thực hiện nhiệm vụ đến khi kết thúc nhiệm vụ
- Thuộc tính ThreadState: Lấy/thiết lập các trạng thái trong vòng đời của thread
 - Unstarted: trạng thái chưa khởi động
 - Running: trạng thái đang chạy
 - Suspended: Trạng thái tạm dừng
 - Stopped: Trạng thái dừng hẳn

Một số thuộc tính của thread

- IsBackground – get, set
 - Quy trình không kết thúc cho đến khi các luồng “Foreground” kết thúc
 - Các luồng “Background” được chấm dứt khi ứng dụng kết thúc
- CurrentThread – get, static
 - Trả về luồng tham chiếu để gọi luồng
- IsAlive – get
 - Luồng đã bắt đầu nhưng chưa kết thúc
- Priority – get, set
 - Highest, AboveNormal, Normal, BelowNormal, Lowest

Một số phương thức của thread

- Start(): khởi động thread (cho thread chạy)
- Suspend(): đình chỉ tạm thời thread để bắt đầu một thread khác
- Resume(): gọi phục hồi thread
- Sleep(): cho thread ngủ một khoảng thời gian
- Abort(): huỷ thread
- Join(): Cho phép một thread kết nối vào chuỗi thread đang hoạt động, dừng những thread đang chạy để thực hiện thread mới kết nối

Bài tập



- Viết hai chương trình (1 chương trình dùng thread và 1 chương trình không dùng thread) thực hiện gọi các hàm:
 - Xuất ra màn hình các số nguyên tố < 1 triệu
 - Xuất ra màn hình các số chính phương < 1 triệu
 - Xuất ra màn hình các số chẵn < 1 triệu
 - Xuất ra màn hình các số lẻ < 1 triệu
- Chạy 2 chương trình và so sánh CPU được dùng cho 2 chương trình đó

Kết quả

Chương trình sử dụng thread

Chương trình không sử dụng thread

Name	St...	90% CPU
> ThreadExample (32 bit) (2)		46.7%
> ThreadExample (32 bit) (2)		0%
> Task Manager		2.4%
> Snipping Tool		1.0%
> Microsoft Visual Studio 2017 (32 bit...		6.8%
> Microsoft PowerPoint (32 bit) (2)		0%
> Microsoft PowerPoint (32 bit)		0%
> Microsoft Excel (32 bit) (2)		0%
> KhongThread (32 bit) (2)		17.7%

Sử dụng thread trong winform

- Khi chạy ứng dụng winform, một thread được tự động tạo ra và là thread chính
- Khi sử dụng một thread khác trong đó có mã lệnh liên quan đến xử lý giao diện thì cần chuyển quyền thực hiện đoạn mã lệnh này cho thread chính theo cú pháp:

```
this.Invoke(new Action(()=> {  
    // mã lệnh liên quan đến xử lý giao diện  
}));
```

- Lưu ý: **this** là form hiện tại, có thể được thay thế bằng control bất kỳ

Bài tập



- Tạo một form cho phép nhập vào một số n nguyên dương
- Sử dụng thread thực hiện tìm các số nguyên tố, số chính phương, số hoàn chỉnh nhỏ hơn số n vừa nhập
- Tìm được số nào thì hiển thị số đó lên ô tương ứng
- Nếu ấn vào nút tạm dừng thì dừng việc tìm số.
- Nếu ấn vào nút tiếp tục thì thực hiện tiếp các thread đang tạm dừng.
- Nếu ấn vào nút dừng thì stop tất cả các thread đang chạy

Bài tập (tiếp)

Tim các số chính phương, nguyên tố, hoàn chỉnh

Nhập số n

Các số chính phương < n:

Các số nguyên tố < n:

Các số hoàn chỉnh < n:

- Yêu cầu:
 - Trong lúc các thread tìm số vẫn thực hiện thì các số tìm được sẽ hiển thị lên màn hình ở các ô tương ứng
 - Trong mỗi thread tìm số cần phải sử dụng lệnh chuyển cho thread chính hiển thị số vừa tìm được