

BÀI TẬP THỰC HÀNH 2

Mô hình MultiThread

Lý Thuyết: Thread C#

Sử dụng thư viện: `System.Threading`

Tạo mới Thread xử lý phương thức **P** truyền tham số param

```
Thread t = new Thread(P);
t.Start(param);
```

Phương thức xử lý cho thread

```
void P(object param)
{...}
```

Đợi thread con xử lý xong mới xử lý dòng lệnh tiếp theo trong Process cha.

```
t.Join()
```

Project 1: Cho ma trận *Sudoku* có kích thước 9x9 như sau, gồm 9 ma trận con có kích thước 3x3. Trong đó mỗi phần tử trong ma trận con có giá trị từ 1 đến 9 không được phép trùng, các dòng và cột trong *Sudoku* có giá trị từ 1 đến 9 không được phép trùng:

6	2	4	5	3	9	1	8	7
5	1	9	7	2	8	6	3	4
8	3	7	6	1	4	2	9	5
1	4	3	8	6	5	7	2	9
9	5	8	2	4	7	3	6	1
7	6	2	3	9	1	4	5	8
3	7	1	9	5	6	8	4	2
4	9	6	1	8	2	5	7	3
2	8	5	4	7	3	9	1	6

- Viết chương trình xây dựng *Sudoku Solution Validator* để kiểm tra Sudoku là hợp lệ, sử dụng nhiều Thread chạy song song để kiểm tra.
- **Gợi ý:** sử dụng
 - 9 Thread kiểm tra không được phép trùng trên các dòng trong **Sudoku**.
 - 9 Thread kiểm tra không được phép trùng trên các cột trong **Sudoku**.
 - 9 Thread kiểm tra không được phép trùng trên các ma trận con 3x3 trong **Sudoku**.
- **Cấu trúc dữ liệu:**

```
//27 Theards kiem tra Sudoku
static Thread[] arrThread = new Thread[27];
static int[] validC = new int[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };
static int[] validR = new int[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };
static int[,] valid3x3 = new int[3,3] {
    { 0, 0, 0 },
    { 0, 0, 0 },
    { 0, 0, 0 }
};

//Cau truc paramaters truyen trong theard lu tru thong tin ma tran con 3x3
//la dia chi cua o dau tien
11 references
struct paramaters
{
    public int row;
    public int column;
};

//Ma tran Sudoku
static int[,] sudoku =new int[9,9] {
    {6, 2, 4, 5, 3, 9, 1, 8, 7},
    {5, 1, 9, 7, 2, 8, 6, 3, 4},
    {8, 3, 7, 6, 1, 4, 2, 9, 5},
    {1, 4, 3, 8, 6, 5, 7, 2, 9},
    {9, 5, 8, 2, 4, 7, 3, 6, 1},
    {7, 6, 2, 3, 9, 1, 4, 5, 8},
    {3, 7, 1, 9, 5, 6, 8, 4, 2},
    {4, 9, 6, 1, 8, 2, 5, 7, 3},
    {2, 8, 5, 4, 7, 3, 9, 1, 6}
};

static void Main(string[] args)
{
    int i, j,z, index=0; //index thu tu thread
    paramaters p = new paramaters();
    //9 Thread kiem tra dong
    //9 Thread kiem tra cot
    //9 Thread kiem tra 9 ma tran => chon duoc toa do ma tran kiem tra (selectedMaTrix)
    //Doi 27 ket thuc
    //Kiem tra 9 dong hop le => kiem tra validR
    //Kiem tra 9 cot hop le => kiem tra validC
    //Kiem tra 9 ma tran hop le => kiem tra valid3x3
    Console.WriteLine("Sudoku Valid!!");
    Console.ReadLine();
}

//Phuong thuc kiem tra cot hop le
static void isColumnValid(object param)
{
    paramaters p = (paramaters)param;

    //Kiem tra Tu 1-9 chi xuất hiện 1 lần trên cot p. column, nếu xuất hiện 2 lần trở lên
    // ket thuc thread
    int[] validityArray = new int[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };
    ...
}

//Phuong thuc kiem tra dong hop le
static void isRowValid(object param)
```

```
{
    paramaters p = (paramaters)param;
    //Kiem tra Tu 1-9 chi xuất hiện 1 lần trên cột p. row, nếu xuất hiện 2 lần trở lên
    // kết thúc thread
    int[] validityArray = new int[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };
    ...
}
//Phương thức kiểm tra ma trận còn hợp lệ
static void is3x3Valid(object param)
{
    //Kiem tra Tu 1-9 chỉ xuất hiện 1 lần trên ma trận bắt đầu p.row, p.column, nếu xuất
    //hiện 2 lần trở lên kết thúc thread
    paramaters p = (paramaters)param;
    int[] validityArray = new int[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };
    ...
    valid3x3[p.row/3, p.column/3] = 1;
}

//Phương thức chọn ma trận thu z trong 9 ma trận :0->8 -> trả về tọa độ phần tử đầu tiên
static paramaters selectedMaTrix(int z)
{
    ...
}
```

Project 2: Viết chương trình (Multithreaded Sorting Application) cho phép xử lý sắp xếp các phần tử tăng dần sử dụng lập trình song song MultiThreads sử dụng thuật toán:

- Quicksort
- Mergesort

Theo mô hình sau:

- Dãy số được chia làm 2 phần phân cho 2 Thread xử lý song song sắp xếp tăng dần trên các dãy con.
- Sau khi thực hiện xong thực hiện trộn (merge) 2 dãy con lại để có dãy số tăng dần

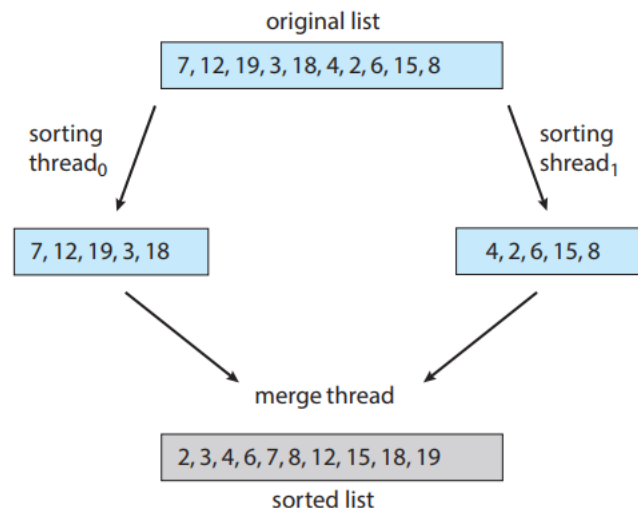


Figure 4.27 Multithreaded sorting.

❖ Hướng dẫn giải

```

//Cấu trúc dữ liệu
static int[] arrayNumber;
static int arrayLength;
10 references
struct parameters
{
    public int l;
    public int r;
}
static void Main(string[] args)
{
    //Doc file
    //Start 2 Thread su dung merge sort tren cac doan
    // Doi cac Thread ket thuc
    // Merge cac doan con
    //In ket qua
}
//Phuong thuc doc so luong phan tu tu file => arrayLength => de khoi tao mang
static void readLength(string path)
{
    ...
}
//Phuong thuc doc cac phan tu vao mang arrayNumber tu file
static void readFile(string path)
{

```

```
...
}
//Thuat toan merge sort
static void merge(int[] arr, int left, int middle, int right)
{
}
static void merge_sort(object param)
{}
```

Project 3: Viết chương trình nhân hai ma trận $A_{n \times m} \times B_{m \times z} = C_{n \times z}$, xử lý song song. Tạo các Thread cho phép nhân dòng với cột xử lý song song. Sau khi xử lý xong tạo một Thread để in ma trận C.