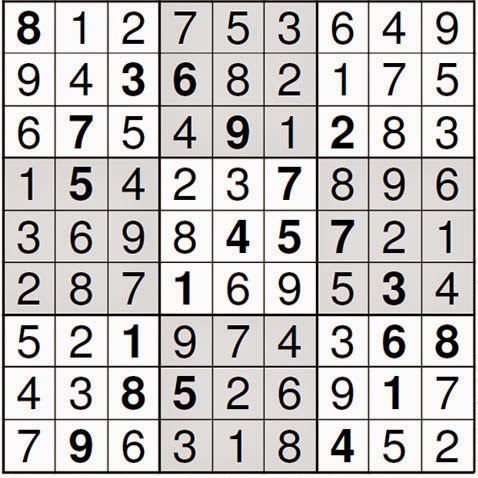
作業系統- 多執行緒數獨驗證程式

Deadline: 2018/06/12 23:55 PM 不可遲交

04052474陳昱叡

* **問題說明：**

利用pthread library撰寫多執行緒程式用來判斷9x9格子的數獨謎題的解答是否有效。



作法要求如下：

* + 由一個執行緒負責檢查每一列是否包含1~9的數字
  + 由一個執行緒負責檢查每一行是否包含1~9的數字
  + 由9個執行緒負責檢查每一個3x3的小方格是否包含1~9的數字

總共使用11個執行緒來驗證數獨謎題的解答是否有效。

* + 父執行緒將產生工作執行緒，並將所要檢查的位置傳給工作執行緒。要將多個資訊傳給執行緒，可利用以下結構型態來傳遞所需參數：

typedef struct

{

int thread\_number;

int x;

int y;

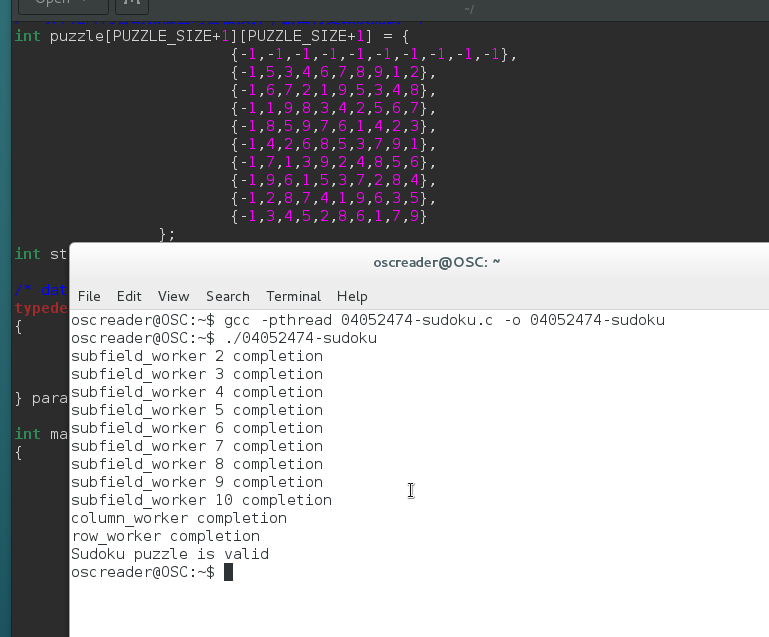
} parameters;

* + 每一工作執行緒結束時，將驗證結果傳回父執行緒。可利用一個全域整數陣列來記錄每個工作執行緒傳回的結果。把陣列第i個索引值位置對應到第i個工作執行緒。該位置的值若為1，表示所驗證的區域是正確的。若為0，表示所驗證的區域是錯的。因此，當所有工作執行緒完成時，父執行緒僅需檢查該陣列的值是否都是1，即可知道該數獨解答是否完全正確。
  + 編譯命令須連結pthread函式庫:

gcc -pthread sudoku.c -o sudoku

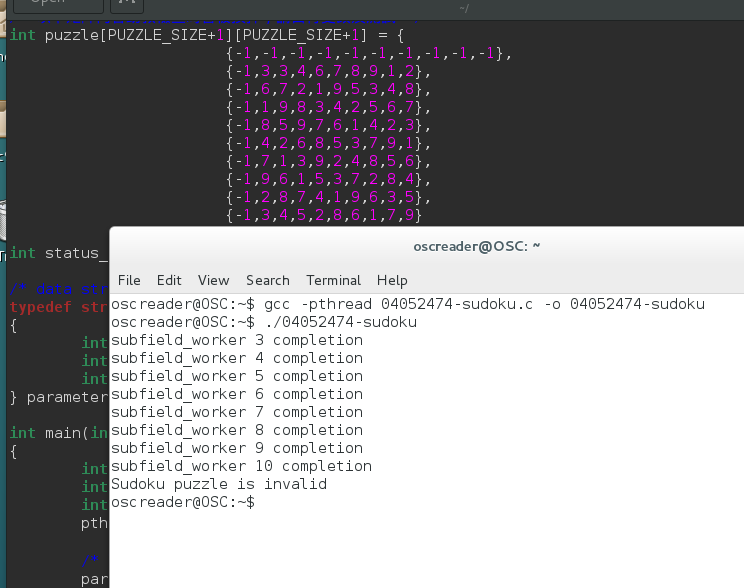
這份作業可能有點瑕疵? 因為我的執行緒並不會亂跳(隨機選擇執行哪一個9x9檢查)，不確定有沒有達到老師的要求，我認為問題在9x9方格那邊?然後就是9x9那邊有點攏長，或許該用選取基礎點的方式以每個方格的左上角方格為基準，做檢查，用迴圈方式來寫，我這份作業是以全域變數方式來作答(附上.c檔 程式部分理解以註解方式寫在程式碼中) <https://github.com/04052474/Multithreaded-Programming-20180612> (附上github因為不知道為什麼我在linux用gedit再載到原本桌面用dev c++開後註解文字會不見 可以在github看)

當我數獨是正確的時候



當我數獨是錯誤的時候(我將數獨[1][1]的位置設為3)

可以清楚看到我的第一個9x9方格是沒有顯示completion的，且列跟行的檢查也是沒有completion，代表有成功?



* **輸出範例：**

subfield\_worker 5 completion

subfield\_worker 6 completion

subfield\_worker 7 completion

subfield\_worker 8 completion

subfield\_worker 9 completion

subfield\_worker 10 completion

subfield\_worker 4 completion

subfield\_worker 3 completion

subfield\_worker 2 completion

column\_worker completion

row\_worker completion

若驗證成功輸出

Sudoku puzzle is valid.

若驗證失敗輸出

Sudoku puzzle is invalid.

* **繳交規定**
  + 程式檔(檔名請用「學號-sudoku.c」)於截止期限前上傳至Moodle
  + 程式檔最前面請加以下之註解:
    - Author：作者姓名（含學號）
    - Purpose：Multithreaded Programming
    - Date：撰寫日期（如2017/06/07）

#include <pthread.h>

#include <stdio.h>

#include <stdlib.h>

#define NUMBER\_OF\_THREADS 11

#define PUZZLE\_SIZE 9

void \*column\_worker(void \*param); /\* 檢查一整行的執行緒 \*/

void \*row\_worker(void \*param); /\* 檢查一整列的執行緒 \*/

void \*subfield\_worker(void \*param); /\* 檢查一個3x3大小的執行緒 \*/

/\* 以下矩陣內容助教檢查時會被換掉，請自行更改及測試 \*/

int puzzle[PUZZLE\_SIZE+1][PUZZLE\_SIZE+1] = {

{-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},

{-1,5,3,4,6,7,8,9,1,2},

{-1,6,7,2,1,9,5,3,4,8},

{-1,1,9,8,3,4,2,5,6,7},

{-1,8,5,9,7,6,1,4,2,3},

{-1,4,2,6,8,5,3,7,9,1},

{-1,7,1,3,9,2,4,8,5,6},

{-1,9,6,1,5,3,7,2,8,4},

{-1,2,8,7,4,1,9,6,3,5},

{-1,3,4,5,2,8,6,1,7,9}

};

int status\_map[NUMBER\_OF\_THREADS] = {0}; //全域整數陣列來記錄每個工作執行緒傳回的結果

/\* data structure for passing data to threads \*/

typedef struct

{

int thread\_number;

int x;

int y;

} parameters;

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

{

int i,j;

int count = 0;

int rv = 1;

pthread\_t workers[NUMBER\_OF\_THREADS];

/\* 建立一個執行緒負責檢查每一行是否包含1~9的數字 \*/

parameters \*data = (parameters \*) malloc(sizeof(parameters));

data->thread\_number = count;

data->x = 1;

data->y = 1;

pthread\_create(&workers[data->thread\_number], 0, row\_worker, data);

/\* 建立一個執行緒負責檢查每一列是否包含1~9的數字 \*/

count++;

data = (parameters \*) malloc(sizeof(parameters));

data->thread\_number = count;

data->x = 1;

data->y = 1;

pthread\_create(&workers[data->thread\_number], 0, column\_worker, data);

count++;

/\* 撰寫程式碼建立9個執行緒負責檢查每一個3x3的小方格是否包含1~9的數字 \*/

/\* 撰寫程式碼等待所有工作執行緒結束 \*/

/\* 撰寫程式碼檢查status\_map所放的值，確認該解答是否正確 \*/

return 0;

}

void \*row\_worker(void \*params)

{

}

void \*column\_worker(void \*params)

{

int i,j, k, status;

int map[PUZZLE\_SIZE + 1] = {0};

parameters\* p = (parameters \*)params;

for (i = 1; i < PUZZLE\_SIZE + 1; i++) {

for (j = 1; j < PUZZLE\_SIZE + 1; j++) {

map[puzzle[j][i]] = 1;

}

/\* make sure the digits 1 .. 9 all appear \*/

status = 1;

for (k = 1; k < PUZZLE\_SIZE + 1; k++) {

if (map[k] == 0) {

status = 0;

break;

}

}

if (status == 0)

break;

else {

/\* reinitialize the map \*/

for (i = 0; i < PUZZLE\_SIZE + 1; i++)

map[i] = 0;

}

}

if (status)

status\_map[p->thread\_number] = 1;

pthread\_exit(0);

}

void \*subfield\_worker(void \*params)

{

}