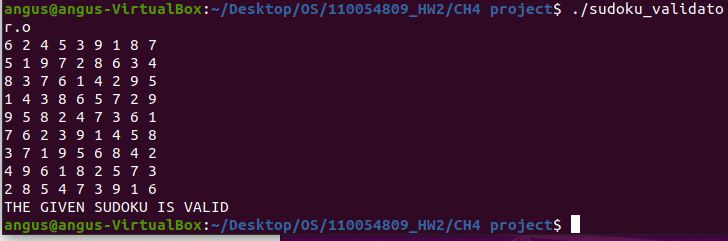
Group member：Threads for checking rows and column: 辛愷庭，Threads for checking each square: 江宥潔，(Main function)Creation of threads for checking row and column: 易頡，(Main function)Creation of threads for checking 9 square: 戴陽

Environment：Linux

Instructions：

1. make

2. run ./sudoku\_validator.o



Source code：

#include<pthread.h>

#include<stdlib.h>

#include<stdio.h>

//Example Of a Valid Sudoku Given in Book

int sud[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}

};

struct data{

int row;

int col;

};

//Threads for Checking Rows and Columns

void \*row\_col(void \*rc){

int p,count=0;

int dirc=(int)rc; //row or col check

for(int i=0;i<9;i++){

int cell=1;

while(cell<10){

for(int j=0;j<9;j++){

if(dirc==0 && sud[i][j]==cell){ //col

count++;

break;

}

else if(dirc==1 && sud[j][i]==cell){ //row

count++;

break;

}

}

cell++;

}

}

if(count==81){

p=1;

}else{

p=-1;

}

pthread\_exit((void\*)p);

}

//Threads for Checking each Square

void \*sq\_check(void \*sq){

struct data \*my\_data=(struct data \*)sq;

int i=(\*my\_data).row;

int j=(\*my\_data).col;

int n,m,p;

n=i+3;

m=j+3;

int count=0;

for(int cell=1;cell<10;cell++){ // cell -> number 1~9

for(i=0;i<n;i++){

for(j=0;j<m;j++){

if(sud[i][j]==cell){ // find cell number, count++

count++;

i=n;j=m; //exit loop, find next cell number

}

}

}

}

if(count==9){ // valid

p=1;

}else{

p=-1;

}

pthread\_exit((void\*)p);

}

int main(){

struct data \*sq;

sq=(struct data\*)malloc(sizeof(struct data));

pthread\_t thread[11];

int a,t\_num;

void \*valid;

for(t\_num=0;t\_num<2;t\_num++){

// Creation of Threads for Checking row and column

int rc = t\_num;

a=pthread\_create(&thread[t\_num],NULL,row\_col,(void\*)rc);

if(a){

printf("error");

}

}

// Creation of Threads for Checking 9 squares

for(int l=0;l<=6;l=l+3){

for(int k=0;k<=6;k=k+3){

(\*sq).row=l;

(\*sq).col=k;

a=pthread\_create(&thread[t\_num++],NULL,sq\_check,(void\*)sq);

if(a){

printf("error");

}

}

}

for (int i=0;i<9;i++){ // print the sudoku matrix

for (int j=0;j<9;j++){

printf("%d ", sud[i][j]);

}

printf("\n");

}

int check=0;

for(int i=0;i<11;i++){ // check 11 thread, if valid, check+1

pthread\_join(thread[i],&valid);

check=check+(int)valid;

}

if(check==11){ // all 11 thread check are valid

printf("THE GIVEN SUDOKU IS VALID\n");

}

else{

printf("THE GIVEN SUDOKU IS INVALID\n");

}

pthread\_exit(NULL);

}