운영체제론 실습

- 스레드를 이용한 Sudoku 답 검증기

목 차

예제 코드 다운로드 경로

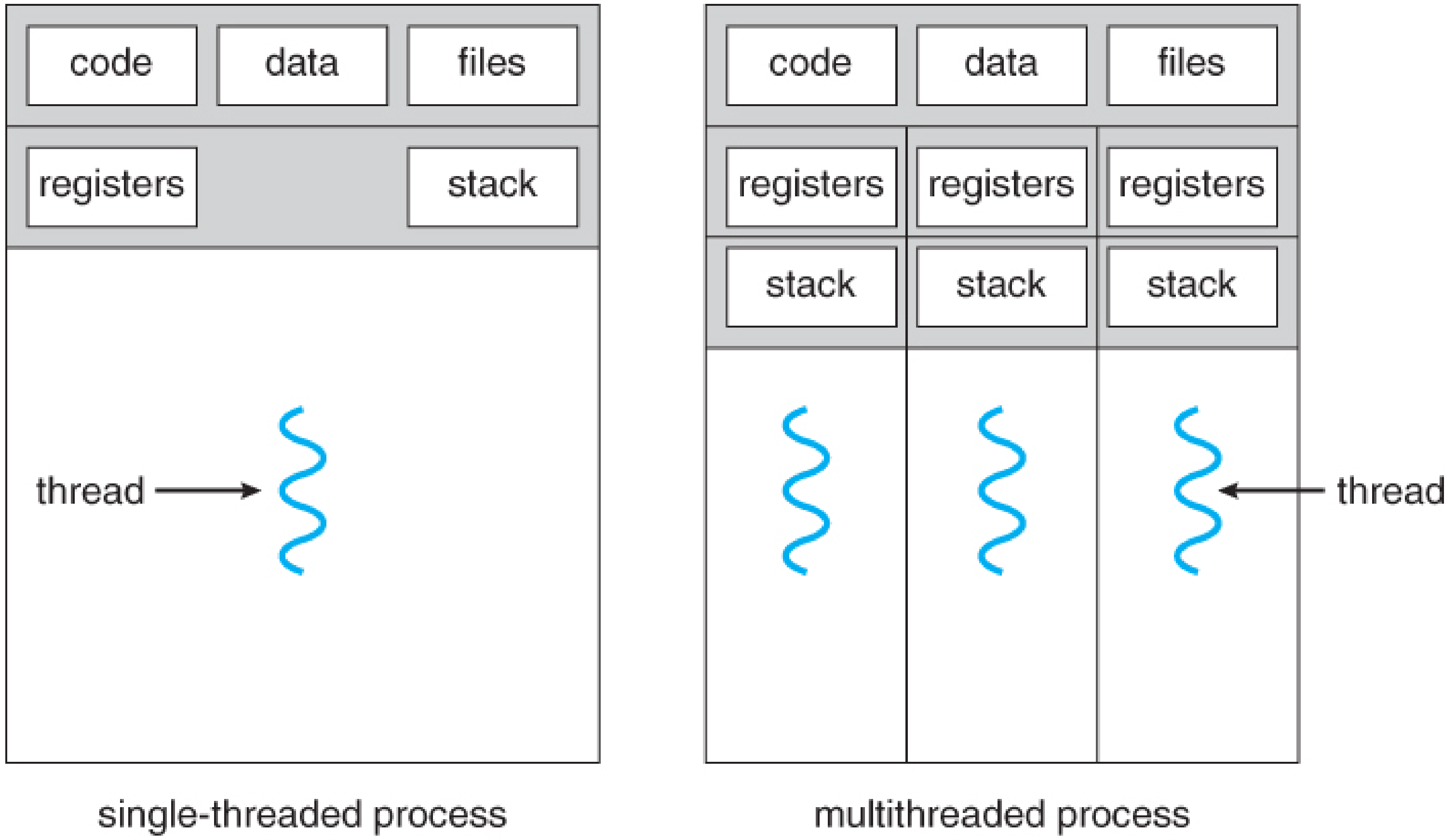
아래 명령어를 linux 환경에서 치면 다운받을 수 있음.

```
$ wget http://ce.hanyang.ac.kr/week6.zip
$ unzip week6.zip
```

Thread

Thread

- 프로세스 내에서 실행되는 여러 흐름의 단위
- 프로세스 1개당 최소 1개의 쓰레드가 존재



pthread

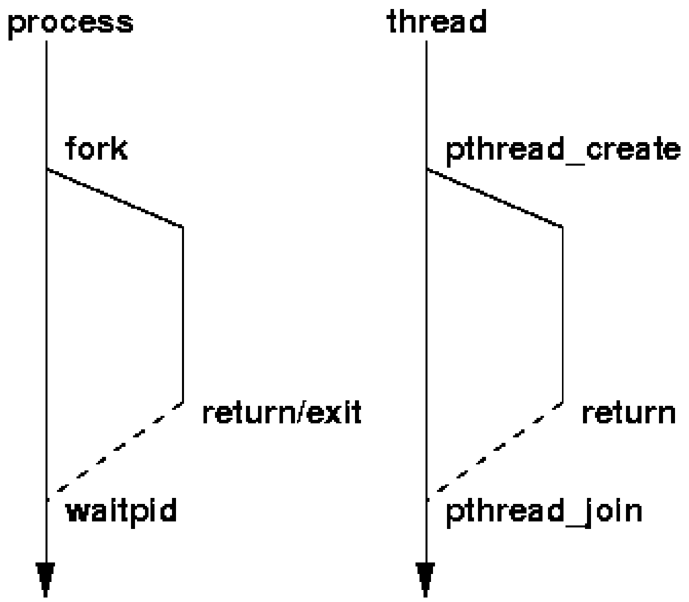
- POSIX thread, POSIX에서 표준으로 제안한 thread 함수 모음

- 기본 pthread 함수

  pthread_create() : thread를 생성한다.

  pthread_exit() : thread를 종료한다.

  pthread_join() : thread 종료를 기다린다.



pthread_create

- **int pthread_create**

  (pthread_t *thread, const pthread_attr_t *attr, void *(*start_routine)(void *), void *arg)

  : 새로운 쓰레드를 생성한다.

  1) pthread_t *thread : 쓰레드가 성공적으로 생성되었을 때 생성된 쓰레드를 식별하기 위해

  사용되는 쓰레드 식별자

  2) pthread_attr_t *attr : 쓰레드 특성을 지정, default로 지정할 경우 NULL로 설정

  3) void *(*start_routine)(void *) : thread가 실행할 함수

  4) void *arg : start_routine으로 넘어갈 매개변수

  5) 리턴 0 : 정상종료 , 0이아닌값은 에러

pthread_exit

- **void pthread_exit(void *retval)** : thread를 종료한다.

  1) retval : 쓰레드 종료 시 자원을 반납해야 하는 등의 처리작업이 필요할 때

     pthread_cleanup_push()를 사용해 콜백함수를 등록해 이 인자를 사용

     사용하지 않아도 되는경우 NULL

pthread_join

- **int pthread_join(pthread_t th, void **thread_return**)

  : th번호를 가지는 thread의 종료를 기다린다.

  1) pthread_t th

  pthread_t th를 식별번호로 가지는 쓰레드가 종료될 때까지 기다린다.

  2) void **thread_return

  th의 리턴값을 NULL이 아니면 thread_return에 전달.

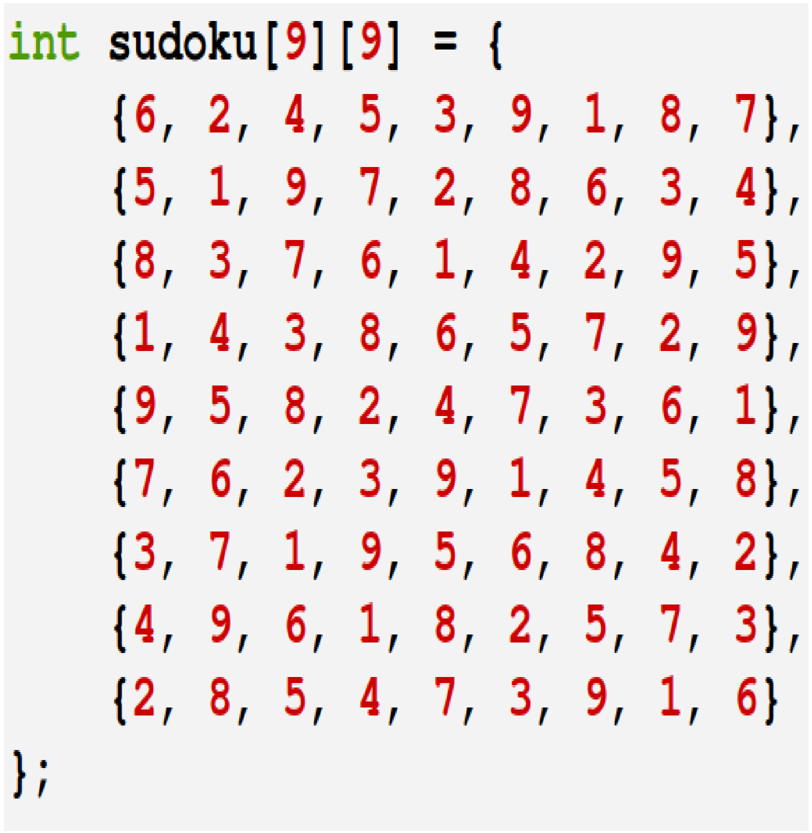
  3) 성공시 0을 리턴, 0이아닌 경우 에러

프로젝트:

Sudoku 답 검증기

Sudoku 검증기 설명

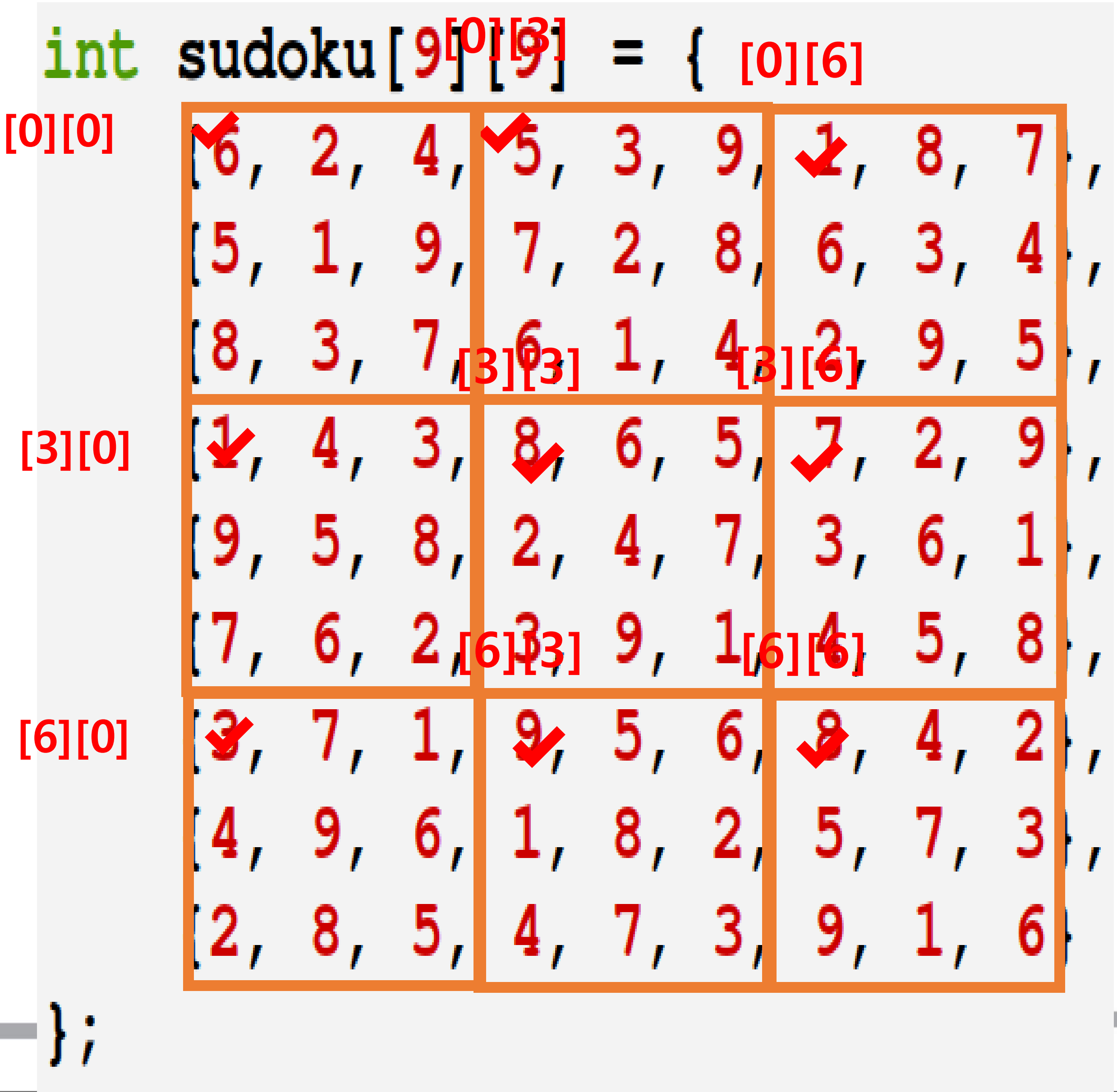
- 2차원 배열로 만들어진 스도쿠의 유효성을 판단
- 아래 조건별로 숫자 1~9를 갖는지 쓰레드를 생성해 검사
    1. 3x3 matrix
    2. Column
    3. Row



```
int sudoku[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}
};
```

Sudoku 검증기 설명

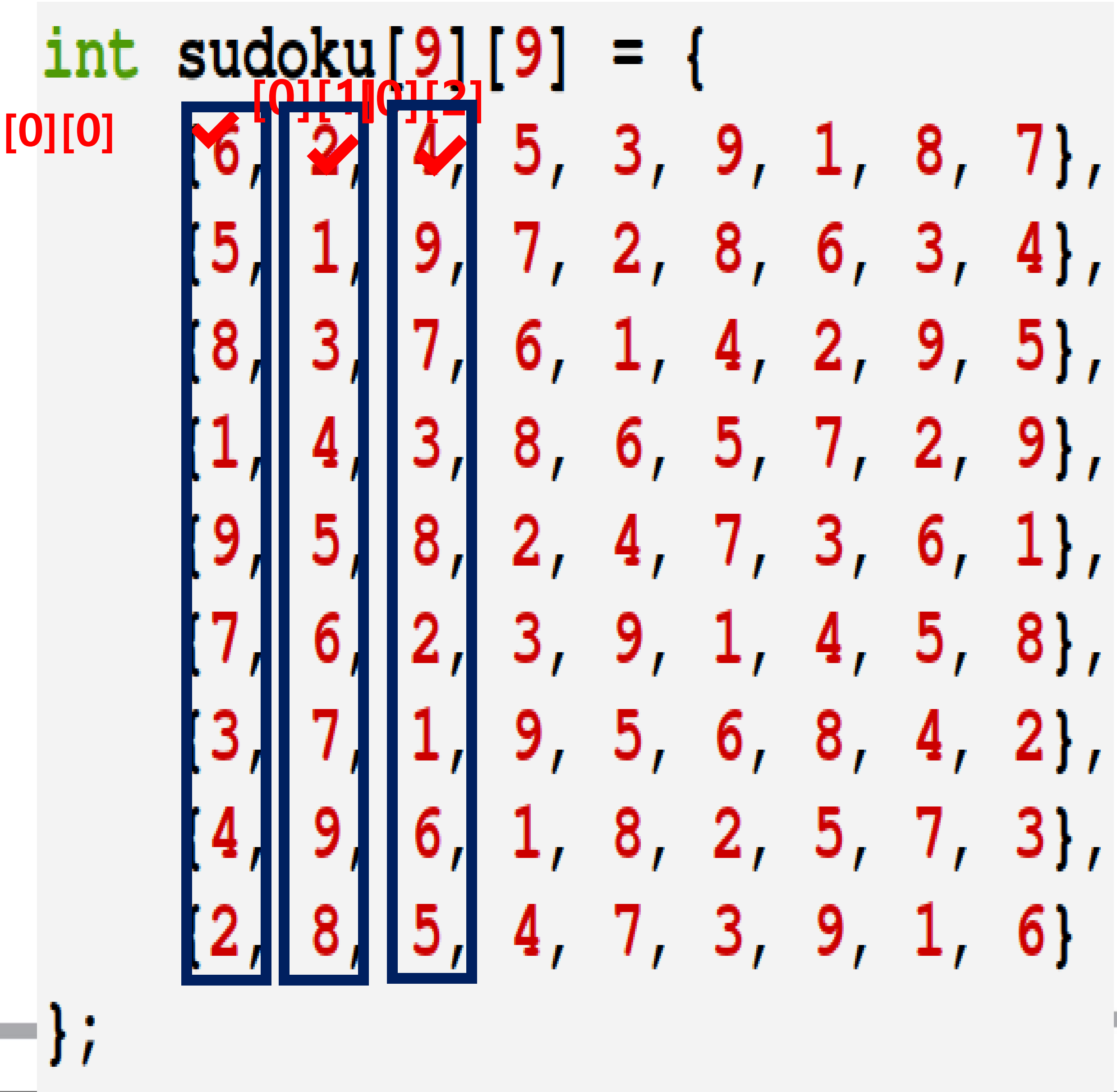
① 3x3 크기 마다 1-9의 숫자를 모두 가지고 있는지 검사



```
int sudoku[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}
};
```

Sudoku 검증기 설명

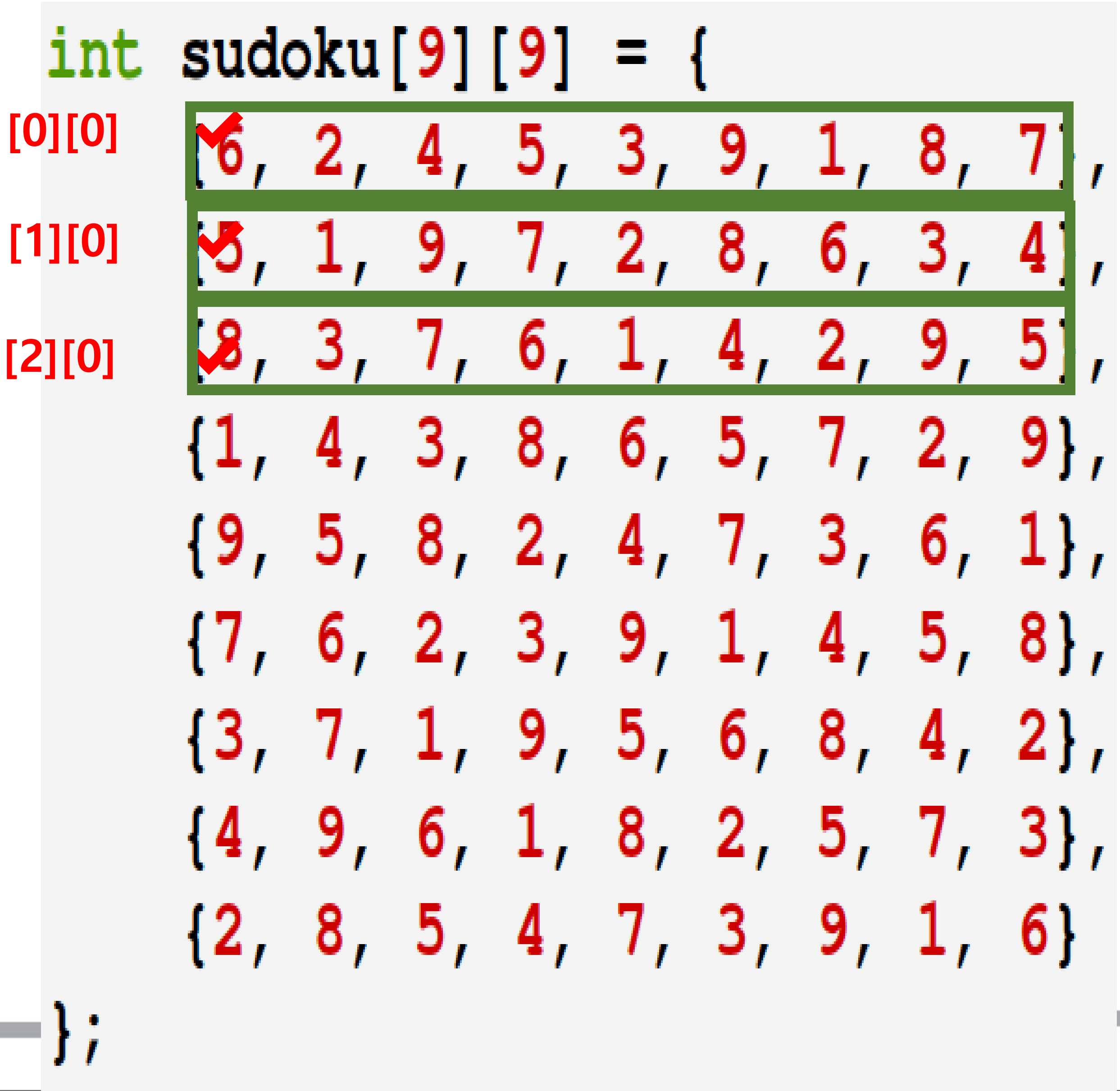
① 세로(Column)가 1-9의 숫자를 모두 가지고 있는지 검사



```
int sudoku[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}
};
```

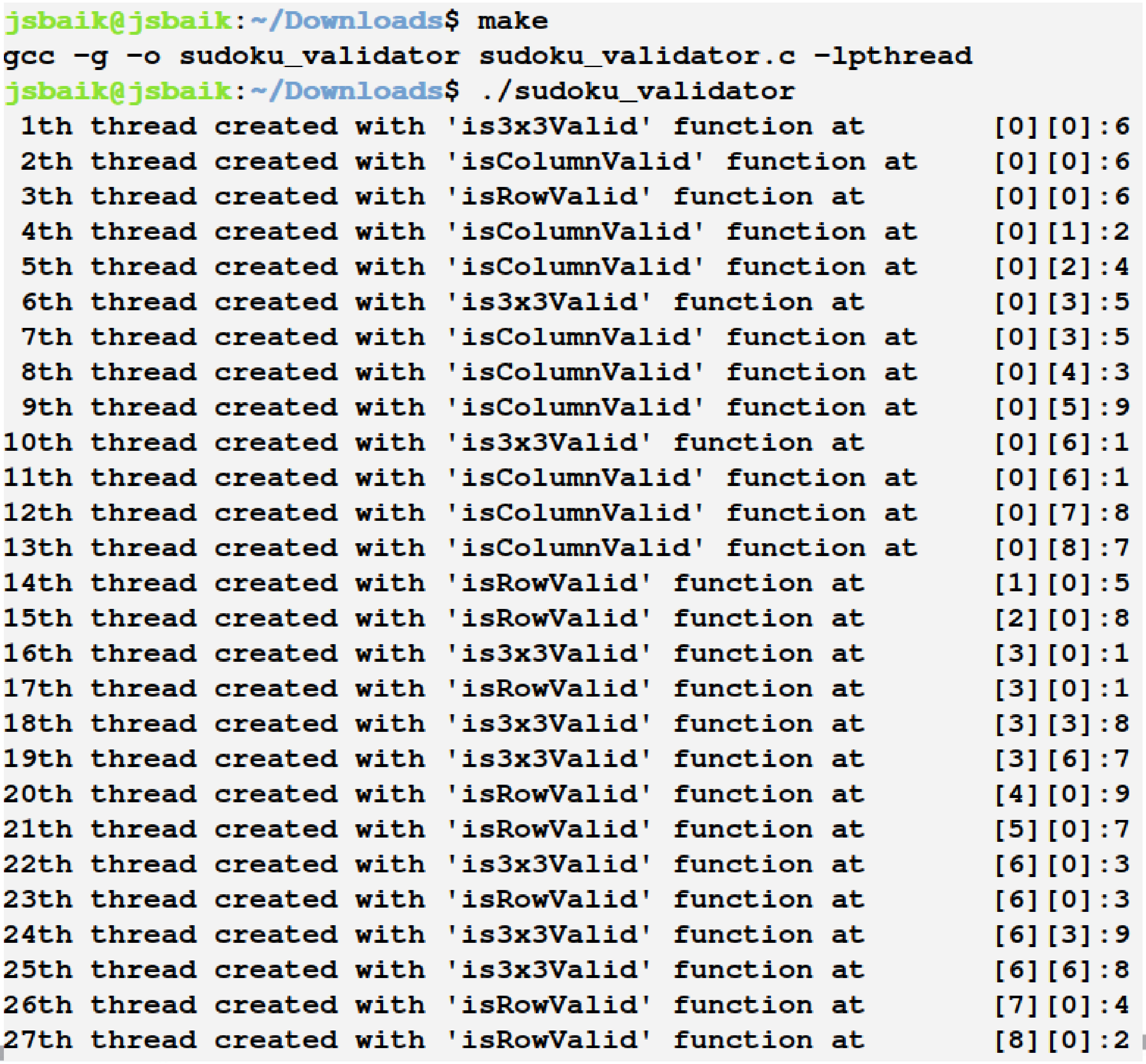
Sudoku 검증기 설명

① 가로(Row)가 1-9의 숫자를 모두 가지고 있는지 검사



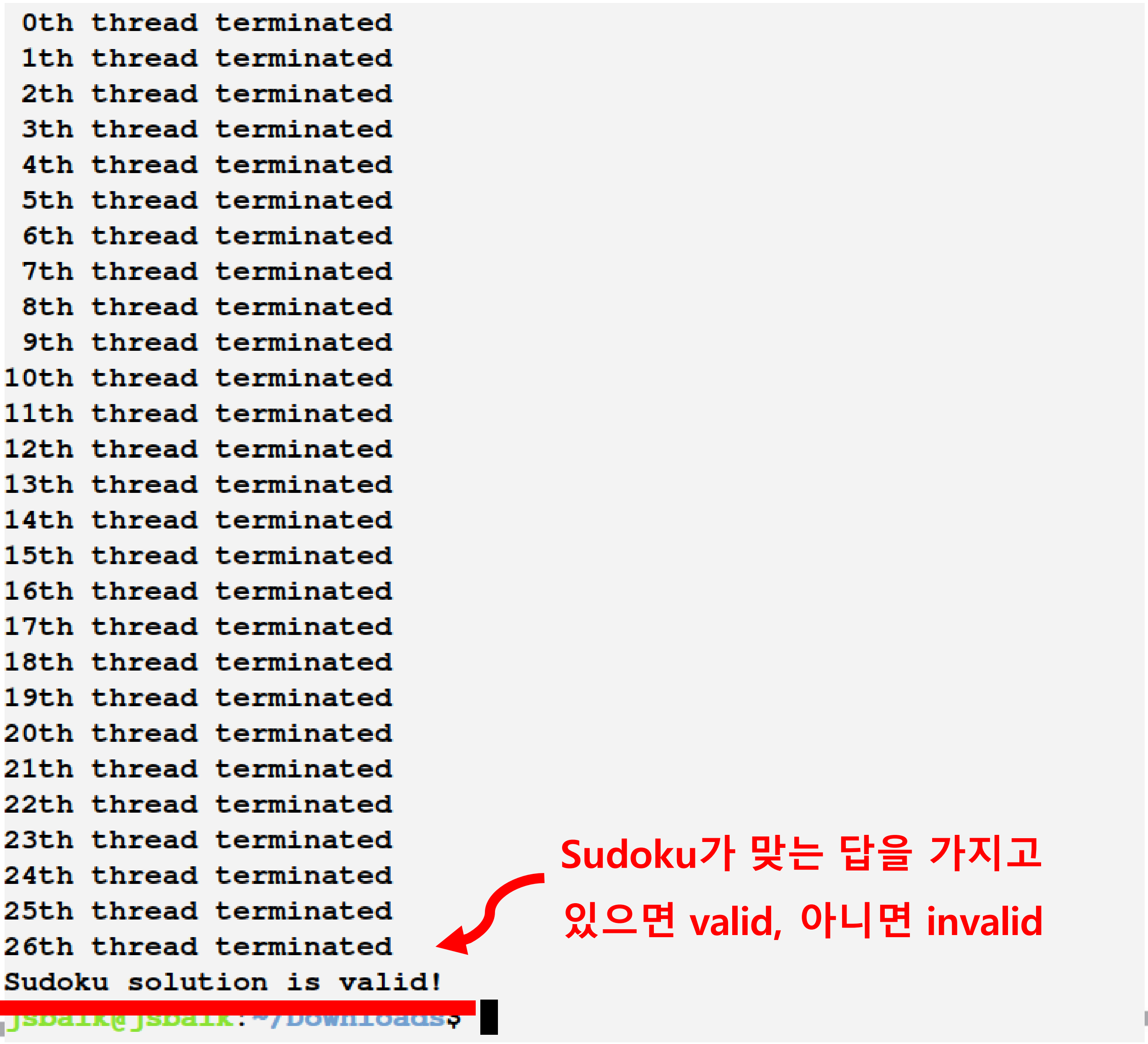
```
int sudoku[9][9] = {
        {6, 2, 4, 5, 3, 9, 1, 8, 7},
        {5, 1, 9, 7, 2, 8, 6, 3, 4},
        {8, 3, 7, 6, 1, 4, 2, 9, 5},
        {1, 4, 3, 8, 6, 5, 7, 2, 9},
        {9, 5, 8, 2, 4, 7, 3, 6, 1},
        {7, 6, 2, 3, 9, 1, 4, 5, 8},
        {3, 7, 1, 9, 5, 6, 8, 4, 2},
        {4, 9, 6, 1, 8, 2, 5, 7, 3},
        {2, 8, 5, 4, 7, 3, 9, 1, 6}
};
```

결과 화면 (1)



```
jsbaik@jsbaik:~/Downloads$ make
gcc -g -o sudoku_validator sudoku_validator.c -lpthread
jsbaik@jsbaik:~/Downloads$ ./sudoku_validator
 1th thread created with 'is3x3Valid' function at      [0][0]:6
 2th thread created with 'isColumnValid' function at   [0][0]:6
 3th thread created with 'isRowValid' function at      [0][0]:6
 4th thread created with 'isColumnValid' function at   [0][1]:2
 5th thread created with 'isColumnValid' function at   [0][2]:4
 6th thread created with 'is3x3Valid' function at      [0][3]:5
 7th thread created with 'isColumnValid' function at   [0][3]:5
 8th thread created with 'isColumnValid' function at   [0][4]:3
 9th thread created with 'isColumnValid' function at   [0][5]:9
10th thread created with 'is3x3Valid' function at      [0][6]:1
11th thread created with 'isColumnValid' function at   [0][6]:1
12th thread created with 'isColumnValid' function at   [0][7]:8
13th thread created with 'isColumnValid' function at   [0][8]:7
14th thread created with 'isRowValid' function at      [1][0]:5
15th thread created with 'isRowValid' function at      [2][0]:8
16th thread created with 'is3x3Valid' function at      [3][0]:1
17th thread created with 'isRowValid' function at      [3][0]:1
18th thread created with 'is3x3Valid' function at      [3][3]:8
19th thread created with 'is3x3Valid' function at      [3][6]:7
20th thread created with 'isRowValid' function at      [4][0]:9
21th thread created with 'isRowValid' function at      [5][0]:7
22th thread created with 'is3x3Valid' function at      [6][0]:3
23th thread created with 'isRowValid' function at      [6][0]:3
24th thread created with 'is3x3Valid' function at      [6][3]:9
25th thread created with 'is3x3Valid' function at      [6][6]:8
26th thread created with 'isRowValid' function at      [7][0]:4
27th thread created with 'isRowValid' function at      [8][0]:2
```

결과 화면 (2)



```
 0th thread terminated
 1th thread terminated
 2th thread terminated
 3th thread terminated
 4th thread terminated
 5th thread terminated
 6th thread terminated
 7th thread terminated
 8th thread terminated
 9th thread terminated
10th thread terminated
11th thread terminated
12th thread terminated
13th thread terminated
14th thread terminated
15th thread terminated
16th thread terminated
17th thread terminated
18th thread terminated
19th thread terminated
20th thread terminated
21th thread terminated
22th thread terminated
23th thread terminated
24th thread terminated
25th thread terminated
26th thread terminated
Sudoku solution is valid!
jsbaik@jsbaik:~/Downloads$
```

Sudoku가 맞는 답을 가지고 있으면 valid, 아니면 invalid

수고하셨습니다.

- 다음 시간: **휴 강**