

운영체제 project #2 report

- 학번: 2018044766
 - 학과: 소프트웨어학부
 - 이름: 안 요 한
 - 목차
-

1. 프로젝트 개요
2. 프로젝트 진행 과정 및 함수 설명
3. 실제 실행 모습
4. 느낀점 및 어려웠던점

1.프로젝트 개요

멀티 쓰레드를 이용하여 스도쿠의 각 열, 행, 3x3 subgrid 를 검사하는 프로그램을 작성한다.cpu는 해당 코어 개수 만큼의 작업을 실행할 수 있는데, 이때 쓰레드를 생성하여 병렬 처리를 진행하여 성능을 높인다. 하지만 cpu 개수보다 많은 쓰레드가 생성된다면 쓰레드 간의 context switching 이 발생하며 이 작업에 시간이 늘어 날수록 멀티스레딩의 성능은 떨어지게 된다.

하지만 이번 과제에선 성능보단 쓰레드의 생성과 join, 그리고 쓰레드가 프로세스의 자원을 공유하며 실행되는 과정에 집중하여 과제를 진행하도록 한다.

2.프로젝트 진행과정 및 함수 설명

- 구성 함수

1. `void *check_row(*arg)`

```
int i,j,k,tmp;
```

```
//해당 행을 검사하기 위한 check 배열
```

```
int check[9];
```

```
//valid 배열 초기화
```

```

for(i = 0;i<9;i++){
    valid[0][i] = 1;
}
//각행을 검사할때마다 check 배열의 값을 각 행의 sudoku 값으로 초기화
for(i = 0;i<9;i++){
    for(j=0;j<9;j++){
        check[j] = sudoku[i][j];
    }

    // check 배열 정렬
    for(k = 0;k<9;k++){
        for(j = 0;j<9;j++){
            if(check[j] > check[j+1]){
                tmp = check[j];
                check[j] = check[j+1];
                check[j+1] = tmp;
            }
        }
    }

    // check 배열이 정렬 되었기 때문에 1~9 까지 숫자와 i 를 비교
    for(k=0;k<9;k++){
        if(check[k] != k+1){
            valid[0][i] = 0;
            break;
        }
    }
}
pthread_exit(0); //쓰레드 종료
...

```

해당 함수의 경우 9x9 스도쿠의 각 행이 유효 한지 판단하는 함수이다. check 배열을 선언하고 valid[0] 의 값 들을 1로 초기화 하여 준다. 그 다음 check[] 배열에 항 을 돌때 마다 값을 넣어주고 그 배열을 오름차순 정렬 하여 배열안의 값들을 비교 해주면 1~9 까지 값들이 잘 들어 있는지 확인 할 수 있다.

2. void *check_columns(void *arg)

```

int i,j,k,tmp;
//해당 열을 검사하기 위한 check 배열
int check[9];

//valid 배열 초기화
for(i = 0;i<9;i++){
    valid[1][i] = 1;
}
//각 열을 검사할때마다 check 배열의 값을 각 열의 sudoku 값으로 초기화
for(i = 0;i<9;i++){

```

```

    for(j=0;j<9;j++){
        check[j] = sudoku[j][i];
    }

    // check 배열 정렬
    for(k = 0;k<9;k++){
        for(j = 0;j<9;j++){
            if(check[j] > check[j+1]){
                tmp = check[j];
                check[j] = check[j+1];
                check[j+1] = tmp;
            }
        }
    }

    // check 배열이 정렬 되었기 때문에 1~9 까지 숫자와 i 를 비교
    for(k=0;k<9;k++){
        if(check[k] != k+1){
            valid[1][i] = 0;
            break;
        }
    }
}
pthread_exit(0); //쓰레드 종료

```

행을 검사하는 함수와 구조는 동일하나, check[] 배열에 넣어줄때 열을 검사해야 하므로 `sudoku[i][j]` 대신 `sudoku[j][i]` 를 넣어 초기화 해준다.

3. `void *check_subgrid(void *arg)`

```

int i,j,k,tmp;
k = 0;
int *args = (int*)arg;
int check[9];

//0~8번의 subgrid 시작 좌표를 args 를 통해 선언
int start_x = (*args/3) * 3;
int start_y = (*args%3) * 3;

//args 로 받은 해당 subgrid 의 valid 값 초기화
valid[2][*args] = 1;

//서브그리드에 들어있는 숫자들을 check 배열에 저장 및 초기화
for(i = start_x;i<start_x+3;i++){
    for(j = start_y;j<start_y+3;j++){
        check[k++] = sudoku[i][j];
    }
}

```

```

}

//check 배열 정렬
for(i = 0;i<9;i++){
    for(j = 0;j<9;j++){
        if(check[j] > check[j+1]){
            tmp = check[j];
            check[j] = check[j+1];
            check[j+1] = tmp;
        }
    }
}

// check_row,check_col 과 동일
for(i = 0;i<9;i++){
    if(check[i] != i+1){
        valid[2][*args] = 0;
    }
}

pthread_exit(0);

```

해당 함수는 3X3 subgrid 판별 함수로, 열 과 행을 검사하는 방법과 같다. 우선 각 스레드가 실행할 subgrid 번호를 args 로 넘겨 주면 해당함수에서 args 번호를 기반으로 subgrid 를 검사한다. check[] 배열에 해당 subgrid에 숫자를 넣어주고 위 와 같이 오름차순 정렬하여 마지막에 검사한다.

4. void check_sudoku(void)

```

int i, j;
int gid[9];
pthread_t rowtid, coltid, subgridtid[9];
/*
 * 검증하기 전에 먼저 스도쿠 퍼즐의 값을 출력한다.
 */
for (i = 0; i < 9; ++i) {
    for (j = 0; j < 9; ++j)
        printf("%2d", sudoku[i][j]);
    printf("\n");
}
printf("---\n");
/*
 * 스레드를 생성하여 각 행을 검사하는 check_rows() 함수를 실행한다.
 */
if(pthread_create(&rowtid,NULL,check_rows,NULL) != 0){
    fprintf(stderr, "pthread_create error: check_row \n");
    exit(-1);
}

```

```

/*
 * 스레드를 생성하여 각 열을 검사하는 check_columns() 함수를 실행한다.
 */
if(pthread_create(&coltid,NULL,check_columns,NULL) != 0){
    fprintf(stderr, "pthread_create error: check_col \n");
    exit(-1);
}

/*
 * 9개의 스레드를 생성하여 각 3x3 서브그리드를 검사하는 check_subgrid() 함수를 실행한다.
 * 3x3 서브그리드의 위치를 식별할 수 있는 값을 함수의 인자로 넘긴다.
 */
for(int i = 0;i<9;i++){
    // 공통된 변수를 공유하게 되면 순서의 무작위성을 해결할 수 없음.
    gid[i] = i;
    if(pthread_create(subgridtid+i,NULL,check_subgrid,&gid[i]) != 0){
        fprintf(stderr, "pthread_create error: check_subgrid \n");
        exit(-1);
    }
}
// 여기를 완성하세요
/*
 * 11개의 스레드가 종료할 때까지 기다린다.
 */
pthread_join(rowtid,NULL); // 열 함수 스레드 대기
pthread_join(coltid,NULL); // 행 함수 스레드 대기
for(i = 0;i<9;i++){
    pthread_join(subgridtid[i],NULL);
} // subgrid 함수 스레드 9개 대기

```

위의 함수들이 스도쿠 배열 자체를 검사하는 함수였다면, 이 함수는 스레드를 생성하여 위에서 정의한 각각의 함수들을 실행시켜 전체 스도쿠를 검사하는 함수이다. 행,열 검사 스레드의 경우 일반적으로 선언하듯이 해주었지만 subgrid 같은경우 9개의 스레드를 생성하는데 각각의 스레드가 공유자원을 사용하기 때문에 값이 무작위 적으로 변하고 사용될 수 있어 주어진 `gid[]`를 이용하여 각각의 스레드가 정해진 자원을 사용할 수 있도록 해주었다.

3.컴파일 &실제 실행 모습

- 컴파일 과정

```
an-yohan@an-yohan-ui-MacBookPro:~/Desktop/Desktop/os:c/수업자료/쓰레드
yohan ~/Desktop/Desktop/os:c/수업자료/쓰레드 master ± gcc -v sudoku.skeleton.c -lpthread
Apple clang version 13.0.0 (clang-1300.0.29.3)
Target: arm64-apple-darwin21.1.0
Thread model: posix
InstalledDir: /Library/Developer/CommandLineTools/usr/bin
"/Library/Developer/CommandLineTools/usr/bin/clang" -cc1 -triple arm64-apple-macosx12.0.0 -Wundef-prefix=TARGET_OS_ -Wdeprecated-objc-isa-usage -Werror=deprecated-objc-isa-usage -Werror=implicit-function-declaration -emit-obj -mrelax-all --mrelax-relocations -disable-free -disable-llvm-verifier -discard-value-names -main-file-name sudoku.skeleton.c -mrelocation-model pic -pic-level 2 -mframe-pointer=non-leaf -fno-strict-return -fno-rounding-math -munwind-tables -target-sdk-version=12.0 -fvisibility-inlines-hidden-static-local-var -target-cpu apple-m1 -target-feature +v8.5a -target-feature +fp-armv8 -target-feature +neon -target-feature +crc -target-feature +crypto -target-feature +dotprod -target-feature +fp16fml -target-feature +ras -target-feature +lse -target-feature +rdm -target-feature +rcpc -target-feature +zcm -target-feature +zcz -target-feature +fullfp16 -target-feature +sm4 -target-feature +sha3 -target-feature +sha2 -target-feature +aes -target-abi darwinpcs -fallow-half-arguments-and-returns -debugger-tuning=lldb -target-linker-version 711 -v -resource-dir /Library/Developer/CommandLineTools/usr/lib/clang/13.0.0 -isysroot /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk -I/usr/local/include -internal-isystem /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/local/include -internal-isystem /Library/Developer/CommandLineTools/usr/lib/clang/13.0.0/include -internal-externc-isystem /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include -internal-externc-isystem /Library/Developer/CommandLineTools/usr/include -Wno-reorder-init-list -Wno-implicit-int-float-conversion -Wno-c99-designator -Wno-final-dtor-non-final-class -Wno-extra-semi-stmt -Wno-misleading-indentation -Wno-quoted-include-in-framework-header -Wno-implicit-fallthrough -Wno-enum-enum-conversion -Wno-enum-float-conversion -Wno-elaborated-enum-base -fdebug-compilation-dir /Users/an-yohan/Desktop/Desktop/os:c/수업자료/쓰레드 -ferror-limit 19 -stack-protector 1 -fstack-check -mdarwin-stkchk-strong-link -fblocks -fencode-extended-block-signature -fregister-global-dtors-with-atexit -fgnuc-version=4.2.1 -fmax-type-align=16 -fcommon -fcolor-diagnostics -clang-vendor-feature=+nullptrToBoolConversion -clang-vendor-feature=+messageToSelfInClassMethodIdReturnType -clang-vendor-feature=+disableInferNewAvailabilityFromInit -clang-vendor-feature=+disableNeonImmediateRangeCheck -clang-vendor-feature=+disableNonDependentMemberExprInCurrentInstantiation -fno-odr-hash-protocols -clang-vendor-feature=+revert09abecef7bbf -mllvm -disable-aligned-allocation-awareness=1 -mllvm -enable-dse-memoryssa=0 -o /var/folders/vk/046dgn896m7fm2d60xxtycfw0000gn/T/sudoku-9825dd.o -x c sudoku.skeleton.c
clang -cc1 version 13.0.0 (clang-1300.0.29.3) default target arm64-apple-darwin21.1.0
ignoring nonexistent directory "/usr/local/include"
ignoring nonexistent directory "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/local/include"
ignoring nonexistent directory "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/Library/Frameworks"
#include "..." search starts here:
#include <...> search starts here:
/Library/Developer/CommandLineTools/usr/lib/clang/13.0.0/include
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include
/Library/Developer/CommandLineTools/usr/include
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/System/Library/Frameworks (framework directory)
End of search list.
"/Library/Developer/CommandLineTools/usr/bin/ld" -demangle -lto_library /Library/Developer/CommandLineTools/usr/lib/libLT0.dylib -no_deduplicate -dynamic -arch arm64 -platform_version macos 12.0.0 12.0 -syslibroot /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk -o a.out -L/usr/local/lib /var/folders/vk/046dgn896m7fm2d60xxtycfw0000gn/T/sudoku-9825dd.o -lpthread -lSystem /Library/Developer/CommandLineTools/usr/lib/clang/13.0.0/lib/darwin/libclang_rt.osx.a
yohan ~/Desktop/Desktop/os:c/수업자료/쓰레드 master ±
```

- 실행 결과

```
an-yohan@an-yohan-ui-MacBookPro:~/Desktop/Desktop/os:c/수업자료/쓰레드
yohan ~/Desktop/Desktop/os:c/수업자료/쓰레드 master ± ./a.out
***** BASIC TEST *****
6 3 9 8 4 1 2 7 5
7 2 4 9 5 3 1 6 8
1 8 5 7 2 6 3 9 4
2 5 6 1 3 7 4 8 9
4 9 1 5 8 2 6 3 7
8 7 3 4 6 9 5 2 1
5 4 2 3 9 8 7 1 6
3 1 8 6 7 5 9 4 2
9 6 7 2 1 4 8 5 3
---
ROWS: (0,YES) (1,YES) (2,YES) (3,YES) (4,YES) (5,YES) (6,YES) (7,YES) (8,YES)
COLS: (0,YES) (1,YES) (2,YES) (3,YES) (4,YES) (5,YES) (6,YES) (7,YES) (8,YES)
GRID: (0,YES) (1,YES) (2,YES) (3,YES) (4,YES) (5,YES) (6,YES) (7,YES) (8,YES)
---
6 3 9 8 4 1 2 7 5
7 2 4 9 5 3 1 6 8
1 8 5 7 2 6 3 9 4
2 5 6 1 3 7 4 8 9
4 9 1 5 8 2 6 3 7
8 7 3 2 6 9 5 2 1
5 4 4 3 9 8 7 1 6
3 1 8 6 7 5 9 4 2
9 6 7 2 1 4 8 5 3
---
ROWS: (0,YES) (1,YES) (2,YES) (3,YES) (4,YES) (5,NO) (6,NO) (7,YES) (8,YES)
COLS: (0,YES) (1,YES) (2,NO) (3,NO) (4,YES) (5,YES) (6,YES) (7,YES) (8,YES)
GRID: (0,YES) (1,YES) (2,YES) (3,YES) (4,NO) (5,YES) (6,NO) (7,YES) (8,YES)
---
***** RANDOM TEST *****
6 3 9 8 4 1 2 7 5
7 2 4 9 5 3 1 6 8
1 8 5 7 2 6 3 9 4
2 5 6 1 3 7 4 8 9
4 9 1 5 8 2 6 3 7
8 7 3 4 6 9 5 2 1
5 4 2 3 9 8 7 1 6
3 1 8 6 7 5 9 4 2
9 6 7 2 1 4 8 1 3
---
ROWS: (0,NO) (1,NO) (2,NO) (3,NO) (4,NO) (5,NO) (6,NO) (7,NO) (8,NO)
COLS: (0,NO) (1,NO) (2,NO) (3,NO) (4,NO) (5,NO) (6,NO) (7,NO) (8,NO)
GRID: (0,YES) (1,YES) (2,YES) (3,YES) (4,YES) (5,YES) (6,YES) (7,YES) (8,YES)
---
```

```
an-yohan@an-yohan-ui-MacBookPro:~/Desktop/Desktop/os:c/수업자료/쓰레드
---
6 2 8 6 5 8 7 9 1
3 8 5 5 4 1 7 3 8
4 7 6 2 7 4 2 5 1
4 9 7 6 1 9 8 6 7
2 8 5 7 3 6 8 3 9
6 1 3 4 9 5 1 2 6
8 3 1 4 3 7 7 9 9
5 2 7 9 1 6 7 4 9
4 8 5 1 2 4 4 5 9
---
ROWS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
COLS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
GRID: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,YES)(6,YES)(7,YES)(8,YES)
---
8 3 7 5 4 7 9 8 5
1 7 9 1 2 3 5 1 3
2 4 5 9 8 4 3 8 2
6 7 8 7 1 6 1 6 8
9 8 7 5 2 8 1 5 9
1 6 2 3 7 9 3 8 5
8 4 6 3 4 8 1 8 7
4 5 9 2 6 9 2 9 5
7 2 9 7 1 8 3 6 4
---
ROWS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
COLS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
GRID: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,YES)(6,YES)(7,YES)(8,NO)
---
2 6 4 3 7 2 7 9 4
7 9 4 6 9 3 6 8 7
8 1 7 1 8 5 2 3 5
1 5 6 2 8 3 9 7 4
7 6 1 9 5 6 2 1 3
8 9 2 2 1 9 8 6 5
6 9 1 7 1 5 4 3 2
4 3 9 1 7 3 7 5 6
1 9 3 4 7 5 2 9 3
---
ROWS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
COLS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
GRID: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,YES)(6,YES)(7,YES)(8,YES)
---
```



```
an-yohan@an-yohan-ui-MacBookPro:~/Desktop/Desktop/os:c/수업자료/쓰레드
2 6 4 3 7 2 7 9 4
7 9 4 6 9 3 6 8 7
8 1 7 1 8 5 2 3 5
1 5 6 2 8 3 9 7 4
7 6 1 9 5 6 2 1 3
8 9 2 2 1 9 8 6 5
6 9 1 7 1 5 4 3 2
4 3 9 1 7 3 7 5 6
1 9 3 4 7 5 2 9 3
---
ROWS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
COLS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
GRID: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,YES)(6,YES)(7,YES)(8,YES)
---
7 6 4 1 9 4 3 8 2
8 7 1 8 9 6 4 2 5
6 3 2 7 5 3 1 7 8
7 6 1 8 2 5 4 2 9
4 7 8 9 7 8 7 4 5
9 2 4 3 2 1 1 3 8
4 1 5 3 7 2 1 4 3
3 6 7 6 4 9 6 9 5
6 8 1 8 9 1 2 3 8
---
ROWS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
COLS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
GRID: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,YES)(6,YES)(7,YES)(8,YES)
---
3 4 1 8 1 6 8 6 1
5 2 6 7 4 5 4 9 7
9 8 7 3 9 2 3 2 5
4 3 9 6 4 7 9 3 4
8 5 7 2 5 1 2 7 1
6 1 2 8 9 3 8 6 5
8 4 9 4 6 3 3 5 6
1 3 5 9 8 5 8 2 4
6 7 2 2 1 7 7 9 1
---
ROWS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
COLS: (0,NO)(1,NO)(2,NO)(3,NO)(4,NO)(5,NO)(6,NO)(7,NO)(8,NO)
GRID: (0,YES)(1,YES)(2,YES)(3,YES)(4,YES)(5,YES)(6,YES)(7,YES)(8,YES)
---
yohan ~/Desktop/Desktop/os:c/수업자료/쓰레드 master
```

해당 스도쿠는 처음에 valid 한 상태이기 때문에 행,열,subgrid 모두 yes 라는 값을 보여준다. 2,3 번째 결과는 main 함수에서 `tmp = sudoku[5][3]; sudoku[5][3] = sudoku[6][2]; sudoku[6][2] = tmp;` 로 바꾸어 주었기 때문에 basic_test 에선 5,6행 2,3열 4,6번 subgrid 값이 valid 하지 않게 된다. 이제 랜덤으로 스도쿠 퍼즐을 섞는데, 이 와중에 검증을 시행하기 때문에 검증이 올바르게 않을수 있다. 마지막 랜덤 셔플이 끝나고 나면 올바른 valid 값을 보여주게 된다.

4.느낀점 및 어려웠던점

- 느낀점

이번과제는 simple_shell 과제보단 수월하게 진행 할 수 있었다. 저번 과제를 하면서 어느정도 c 프로그래밍에대한 공부도 병행 하였고, 이번과제는 교수님이 예제들을 제공 해주셔서 도움이 되었던 것 같다. 사실 check_sudoku 함수에서 9개의 스레드를 생성하는 과정의 경우 스레드가 자원을 공유하는 것은 수업시간에 들어 알고있었지만, `gid[]` 배열을 통해 처리 하는 생각은 뼈대코드가 없었다면 조금 어려울수도 있었을것 같다. 하지만 이번 과제를 통해 스레드의 생성과 스레드의 구동 방식 들을 이해할수 있었고, 병렬처리의 중요성도 어느정도 깨달을수 있어서 의미가 큰 과제였다고 생각한다.