

## CAAL Milestone 2

Ikhlas-27096

Ramail-26924

Musab-26923

Bilal-27151

Chosen Algorithm → **Sudoku Solver**

- Have attached the C code file in the drive link.

### **Underlying Theory for choosing this algorithm:**

Sudoku solver is a backtracking algorithm with a combination of recursive search and constraint satisfaction.

**Recursive search:** it recursively explores all possible solutions to place a digit at a position in the sudoku board by filling digits between 1-9 until a correct solution is found which does not violate any constraints.

## **Constraint Satisfaction:**

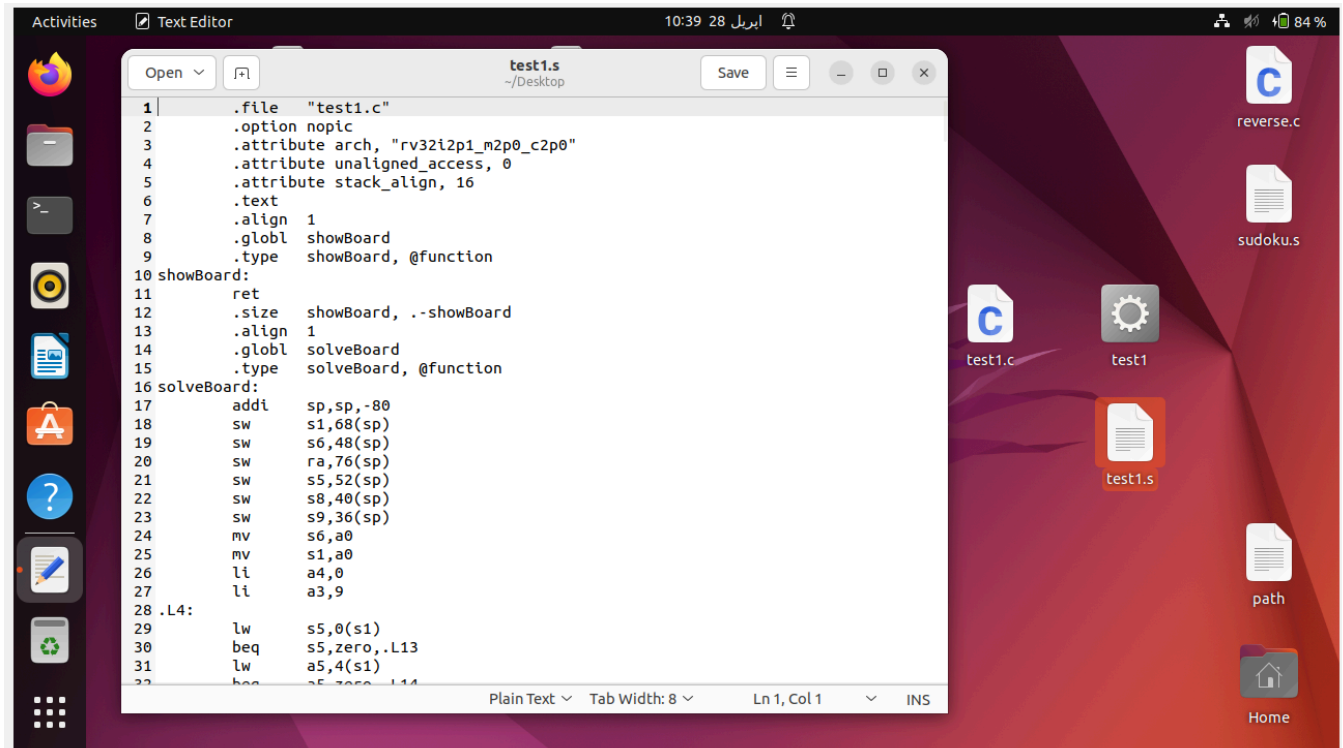
For this problem we have the following 3 constraints,

- 1) A number which is to be placed must not get repeated in its respective row.
- 2) A number which is to be placed must not get repeated in its respective column.
- 3) A number which is to be placed must not get repeated in its respective 3x3 subgrid.

**DFS(Backtracking):** The algorithm repeatedly tries all possible solutions at a place before moving on so if it places a wrong digit which violates any constraint it prunes/removes that placed digit and explores other branches of the search space.

## **Achievements Milestone 2:**

- We were able to interpret and understand that when we run a c file on gcc there are 4 steps in it, preprocessing, compiling, assembling, linking. Where we can stop the process at any stage.
- Since we were to compile the non vectorized version on VeeR-ISS simulator so we stopped the stage at compiling by using the command; riscv32-unknown-elf-gcc -mabi=ilp32 -march=rv32imc -static -O3 -nostdlib -S test1 test1.c

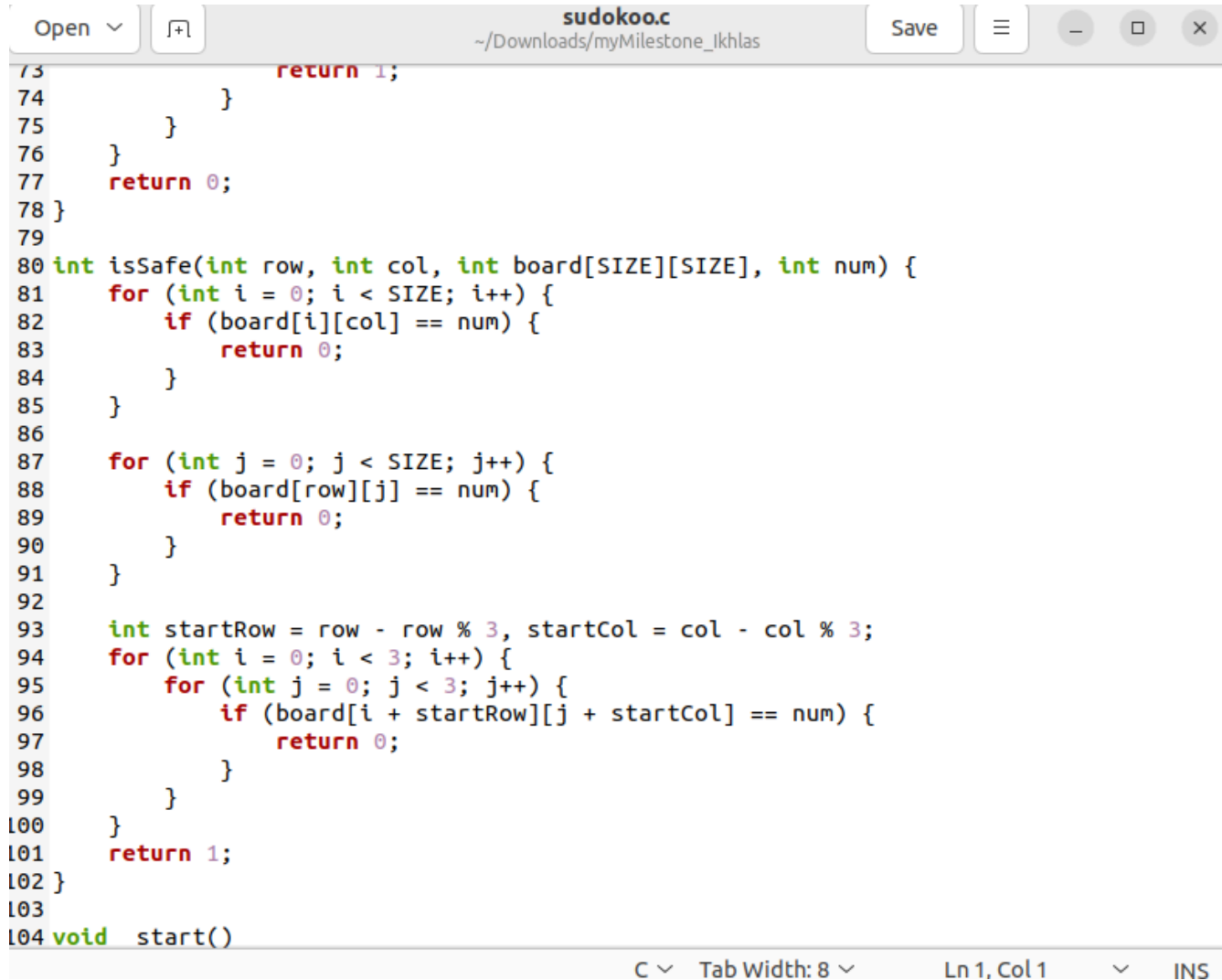


- This is where we got the assembled code in risc-V format (.S file) which could be runned using the veer simulator. The screenshot of the respective (.s) file for our sudoku code is given below which we successfully converted.
- We were successfully able to run the our c code of sudoku in VeeR simulator by using the same commands that sir salman used in his makefile which ran the code first by converting it to riscV assembly.

### Commands used;

- riscv32-unknown-elf-objcopy -O verilog sudokoo program.hex  
Which created a program.hex file for our machine code.
- riscv32-unknown-elf-objdump -S sudokoo > sudokoo.dis  
Which created a .dis file which contained the assembly code parallel with the original c code as well.

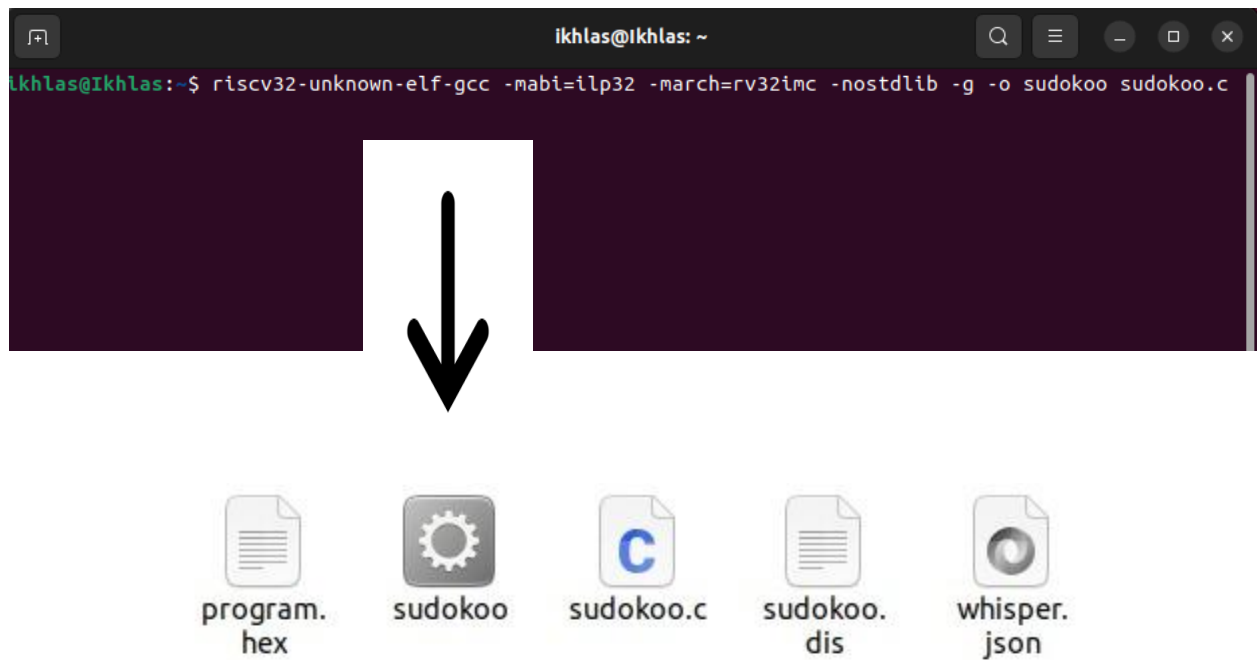
**FOR REFERENCE I HAVE ATTACHED ALL THE PRODUCED FILES IN THE DRIVE LINK.**



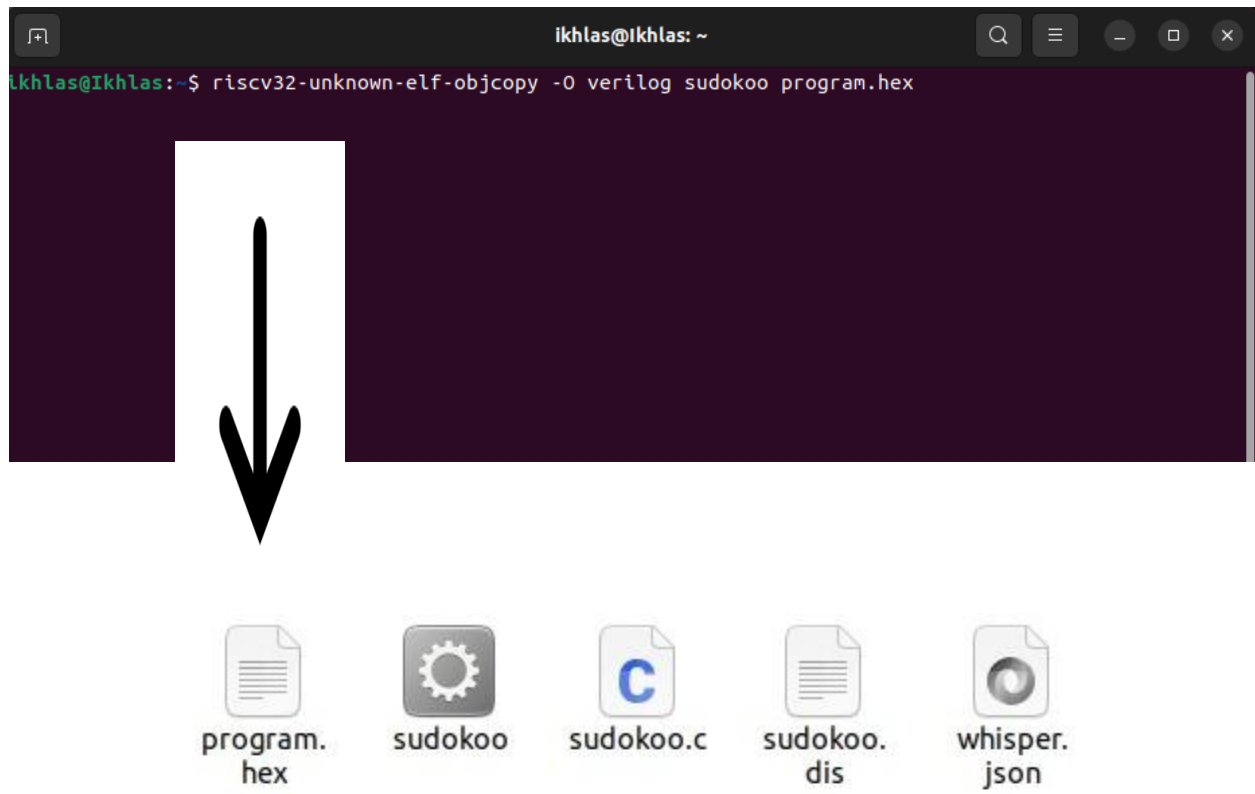
```
73         return 1;
74     }
75 }
76 }
77 return 0;
78 }
79
80 int isSafe(int row, int col, int board[SIZE][SIZE], int num) {
81     for (int i = 0; i < SIZE; i++) {
82         if (board[i][col] == num) {
83             return 0;
84         }
85     }
86
87     for (int j = 0; j < SIZE; j++) {
88         if (board[row][j] == num) {
89             return 0;
90         }
91     }
92
93     int startRow = row - row % 3, startCol = col - col % 3;
94     for (int i = 0; i < 3; i++) {
95         for (int j = 0; j < 3; j++) {
96             if (board[i + startRow][j + startCol] == num) {
97                 return 0;
98             }
99         }
100     }
101     return 1;
102 }
103
104 void start()
```


## Screenshots for running on Veer-Simulator:





1)



2)



Open program.hex  
~/Downloads/myMilestone\_ikhlas

Save 

sudokoo.c ×program.hex ×

1 @00010094

2 01 11 06 CE 22 CC 00 10 23 26 A4 FE 23 24 B4 FE

3 C5 67 13 85 87 3C 8D 20 AA 87 89 C7 C5 67 13 85

4 87 3C 39 20 81 47 3E 85 F2 40 62 44 05 61 82 80

5 79 71 22 D6 00 18 23 2E A4 FC 23 26 04 FE 1D A0

6 23 24 04 FE 31 A0 83 27 84 FE 85 07 23 24 F4 FE

7 03 27 84 FE A1 47 E3 D8 E7 FE 83 27 C4 FE 85 07

8 23 26 F4 FE 03 27 C4 FE A1 47 E3 DB E7 FC 01 00

9 01 00 32 54 45 61 82 80 79 71 06 D6 22 D4 00 18

10 23 2E A4 FC 23 26 04 FE F1 A0 23 24 04 FE 4D A8

11 03 27 C4 FE BA 87 8E 07 BA 97 8A 07 3E 87 83 27

12 C4 FD 3E 97 83 27 84 FE 8A 07 BA 97 9C 43 C1 E7

13 85 47 23 22 F4 FE 8D A8 83 26 44 FE 03 26 C4 FD

14 83 25 84 FE 03 25 C4 FE 39 22 AA 87 A9 CB 03 27

15 C4 FE BA 87 8E 07 BA 97 8A 07 3E 87 83 27 C4 FD

16 3E 97 83 27 84 FE 8A 07 BA 97 03 27 44 FE 98 C3

17 03 25 C4 FD 51 37 AA 87 99 C3 85 47 95 A0 03 27

18 C4 FE BA 87 8E 07 BA 97 8A 07 3E 87 83 27 C4 FD

19 3E 97 83 27 84 FE 8A 07 BA 97 23 A0 07 00 83 27

20 44 FE 85 07 23 22 F4 FE 03 27 44 FE A5 47 E3 D5

21 E7 F8 81 47 35 A0 83 27 84 FE 85 07 23 24 F4 FE

22 03 27 84 FE A1 47 E3 D5 E7 F4 83 27 C4 FE 85 07

23 23 26 F4 FE 03 27 C4 FE A1 47 E3 D8 E7 F2 85 47

24 3E 85 B2 50 22 54 45 61 82 80 79 71 22 D6 00 18

25 23 2E A4 FC 23 26 04 FE A9 A0 23 24 04 FE 05 A8

26 03 27 C4 FE BA 87 8E 07 BA 97 8A 07 3E 87 83 27

27 C4 FD 3E 97 83 27 84 FE 8A 07 BA 97 9C 43 99 E3

28 85 47 35 A0 83 27 84 FE 85 07 23 24 F4 FE 03 27

29 84 FE A1 47 E3 D6 E7 FC 83 27 C4 FE 85 07 23 26

30 F4 FE 03 27 C4 FE A1 47 E3 D8 E7 F2 85 47

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

```
ikhlas@ikhlas: ~  
ikhlas@ikhlas:~$ riscv32-unknown-elf-objdump -S sudokoo > sudokoo.dis
```



  
program.  
hex

  
sudokoo

  
sudokoo.c

  
sudokoo.  
dis

  
whisper.  
json



Open

+

sudokoo.dis

Save

≡

—

□

×

~/Downloads/myMilestone\_1khl

codesfor\_milestone2

×

sudokoo.c

×

sudokoo.dis

×

```

37 100d8: 4781          ll    a5,0
38 }
39 100ba: 853e          mv    a0,a5
40 100bc: 40f2          lw    ra,28(sp)
41 100be: 4462          lw    s0,24(sp)
42 100c0: 6105          addi  sp,sp,32
43 100c2: 8082          ret
44
45 000100c4 <showBoard>:
46
47 void showBoard(int board[SIZE][SIZE]) {
48 100c4: 7179          addi  sp,sp,-48
49 100c6: d622          sw    s0,44(sp)
50 100c8: 1800          addi  s0,sp,48
51 100ca: fca42e23      sw    a0,-36(s0)
52   for (int i = 0; i < SIZE; i++) {
53 100ce: fe042623      sw    zero,-20(s0)
54 100d2: a01d          j     100f8 <showBoard+0x34>
55   for (int j = 0; j < SIZE; j++) {
56 100d4: fe042423      sw    zero,-24(s0)
57 100d8: a031          j     100e4 <showBoard+0x20>
58 100da: fe842783      lw    a5,-24(s0)
59 100de: 0785          addi  a5,a5,1
60 100e0: fef42423      sw    a5,-24(s0)
61 100e4: fe842703      lw    a4,-24(s0)
62 100e8: 47a1          li    a5,8
63 100ea: fee7d8e3      bge   a5,a4,100da <showBoard+0x16>
64   for (int i = 0; i < SIZE; i++) {
65 100ee: fec42783      lw    a5,-20(s0)
66 100f2: 0785          addi  a5,a5,1

```

Plain Text

Tab Width: 8

Ln 1, Col 1

INS

## **References that helped in milestone 2:**

<https://medium.com/@laura.derohan/compiling-c-files-with-gcc-step-by-step-8e78318052>

<https://medium.com/swlh/the-gnu-toolchain-explained-4bf14666bc0>

3