

ARM Assembly Programming Assignment

Tic-Tac-Toe Engine

1. Objectives

Write an ARM assembly procedure of an AI (artificial intelligence) engine for the famous Tic-Tac-Toe game. The procedure is supposed to return the next reasonable movement.

2. Details

A. You are given a skeleton assembly file for the AI engine along with the required C files for testing the engine. Unzipping "tictactoe.tgz" will yield a directory named "tictactoe", which contains the assembly skeleton, C files and make file. Invoking 'make' command in the directory will produce an executable binary file "ttd".

B. Your goal is to write an intelligent game engine for the Tic-Tac-Toe game. The game engine function prototype in C is as follows:

```
int t20189999(board *current_bd);
```

The file name "20189999" must be replaced with your student ID. Never use other names for your engine.

C. The assembly file that contains your engine must have the same name with your engine. For example, if your student ID is 2018999, your assembly file must be named "t2018999.s". Create your own assembly file with reference to the sample assembly file, "dummy.s". (Hint: Replace the sample procedure names, "dummy", in "dummy.s" with "t"+your student ID, and then rename the file to "t"+ your student ID.)

D. The input data type "board" is defined in the header file "board.h". Each tile on the board can have one of the three values; 0, 1 or 2. "0" means that the tile is currently empty and can be occupied by the next movement. "1" means that the first player took the tile. Naturally, "2" means that the tile is taken by the second player.

E. The output is an integer between 0 and 8 that indicates the next movement of the player. Each number is mapped to a tile on the board as follows:

0	1	2
3	4	5
6	7	8

F. The first player is called "O player" while the second is called "X player". If you want to use your engine for O player, initiate the command "make OP=t2018999". "make XP=t2018999" will place your engine for X player. You can set both players' engines at the same time by using the command "make OP=t2018999 XP=t2018888". Note that your engine must be able to detect which side it is being used for. O player engine name and X player engine name must be different.

- G. Whenever you want to change the game engine for a player, you must clean all object and binary files before recompilation by using “make clean”.
- H. In addition to the cross assembler, you may need to install the cross C compiler for the ARM architecture. In that case, use “apt-get install gcc-arm-linux-gnueabi” command to install the compiler.
- I. You are supposed to complete the skeleton assembly code so that they work correctly as written in the problem description. Code with fewer instructions will earn better score.
- J. Any reverse engineering (from C code) is strictly prohibited. Think and write only in assembly. Also leave comments or notes in your assembly code that describe what each code segment does.
- K. The skeleton files can be downloaded at
<http://csl.skku.edu/uploads/SWE3005F19/tictactoe.tar.gz>
- L. The description for the Tic-Tac-Toe game is here:
<http://en.wikipedia.org/wiki/Tic-tac-toe>
- M. By default, the “ttt” program will use a random engine for both players. You can test your engine against the random engine.
- O. Prepare a separate document in PDF format, which explains the design and implementation of your code in detail. The document file name should be “studentid.pdf” (e.g. 2018310123.pdf)

3. Logistics

- A. Submit the assembly source file (OO.s) and the documentation file (OO.pdf) through the i-Campus assignment section. Compress them in a single file and submit the compressed file.
- B. Only the assignments submitted before the deadline will receive the full credit. 25% of the credit will be deducted for every single day delay.