Middle East Technical University   Department of Computer Engineering

CENG 444

Language Processors

Spring 2023-2024

Project III - Target Code Generation

Due date: 9 June 2024, Sunday, 23.59

1 Problem Definition

In this assignment, you are required to extend the program you developed in the previous assignment in a way that it can produce and run x64 code when the conditions listed below are met:

- The source code is verified to be a valid TQL program.
- The source code has two statements only. The first one is a let statement and the second one is an evaluation statement.
- The let statement, which may have more than one declaration, declares all variables as numeric.
- There are no function calls in the expressions.

The code generated by your program must be placed in the process memory space that contains a runnable x64 code fragment which will be called through a function pointer from your main program. If the conditions mentioned above are met, the driver must return a valid function pointer to the code fragment which accepts no parameters and returns a number (double-precision floating point). If the conditions are not met, then the driver must return a nullptr to prevent a call. See the below sample code.

```
// This is the moment when the processing was complete,
// the IR and the IC were generated and reported.
// f is a pointer to any function that accepts
// no parameter and returns a double.

f=driver->getIR()->getFunctionPointer();
if (f!=nullptr)
   cout << f( );

// Disposal of the structures before completion
```

The driver is the parser you developed, getIR is the method that returns the pointer to the generated TQLIR, and the getFunctionPointer is the method that returns pointer to the generated code fragment.

2 Processor Output

This assignment is an extension of the previous one. So, the language processor you develop will generate an improved JSON file that delivers the three main components (the graphical IR, the linear IR, and the list of messages) given in the introduction to the problem in the previous assignment. It will generate a text file to present the linear IR in a more readable format. Additionally, it will echo the result produced by the code generated by your program if the conditions are met.

3 Regulations

- **Implementation:** You are required to use the supplemental code and apply modifications as you see fit. You are responsible for applying modifications and patches to the supplemental components when issued.
- **Attendance:** Many details will be clarified in the lab hours. Additional validity checks and improvements on IR may be asked as identified during the discussions. Attendance is crucial for successfully completing this assignment.
- **Submission:** You need to submit all relevant files you have implemented (.cpp, .h, .l, .y, etc.) as well as a README file with instructions on how to run, a shell script and a makefile to run in a single .zip file named `<studentID.zip>`. If you are using Eclipse IDE, please submit your project folder as a .zip file named `<studentID.zip>`
- This project is expected to be an individual effort, but discussions are always welcome. Please do not hesitate to use the mail group for your questions.
- In case you use Eclipse IDE, take care following the directions below, which are easy as these are compliant with defaults:
  - Place all of your source files in the project folder, not in any subfolder.
  - Develop your solution in a way that the input and output text files (.txt) will be placed in the project folder.
  - Do not make any changes to project options that designate executable target folder other than Debug. The evaluation process will use Debug build.
  - Make sure that the make utility succeeds in building when run (`make clean` followed by `make all`) in the Debug folder of the project.
  - Submit your project folder (not the workspace) as a .zip file in compliance with the naming conventions noted above. The project folder is the folder containing the .project folder.
- If you breach from the advice of using Eclipse C/C++ complying with the directions provided, your presence and participation in the evaluation may be required. In this case, please describe precisely the steps to build and run the program. Additionally, provide shell scripts for building and running with the support of a README file that contains directions for building and running, you may also provide a makefile if needed.