**Author** : Sri Mounika Puvvada

**Subject** : Advanced programming Languages

**Project** : 2

**File Name : UsersManual.docx**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Setup and Compilation**

1. Download and unzip the submission from eLearning on a Linux box in the multi-platform lab.

2. The submission includes:

· main.c

· error.c

· emitter.c

· symbol.c

· lexer.c

· parser.c

· codegen.c

· makefile

· execute.sh

· test1 – gcd program

· test2

· test3

· test4

· test5

· test6

· test7

· test8

· UsersManual.docx (this file)

· FunctionalDecomposition.docx

· Grammar.docx

 3. Environment: This program has been tested in the multi-platform lab and will run there.

 4. Compiling.

· This program includes a Makefile.

· At the command line in Linux, type *make*.

· Results a executable file named final .

 5. Running the program.

· This program includes a executable.sh file.

· Issue the command *sh executable.sh .*

· No command line arguments are required or checked.

· But , user wants to execute the test file from command prompt , type ./final <testfilename>.

· User input: no user interaction with the program is required.

**6.**  Output

· There are two types of outputs

1. If the file is legal or not goes to the console.
2. If the file is legal then the intermediate code goes to .obj file

· This project is done by not using the syntax tree generation.

· So, when a branch statement occur in the code then the output is generated in a such a way that

IF R0 goto L1 - implies that if R0==0 then take branch

* Negate the value in the Register

· Then for an IF statement it appears like

IF (cond){}ELSE{}

Code Generated:

Condition

IF R0 then goto L1

IF Block statements

Goto L2

L1:

ELSE Block statements

L2:

Next set of statements

· For a while statement WHILE(condition){}

Code Generated:

L1:

Condition

IF R0 goto L2

WHILE block statements

Goto L1

L2:

Next set of statements

· In this intermediate code first the code for IF block is generated and later for the else block

· Output will be similar to this:

  a. Legal – If the given input file is parsed with out any errors

b. Error messages- If the given input file has errors

1. If any parsing error occurred – Syntax error

2. ( - Missing open parenthesis

3. ) – Missing close parenthesis

4. } – Missing close parenthesis

5. ; - Missing delimiter

6. abc\_ - Identifier ending with underscore

7. a - b - Expression syntax error

8. void main() - Expecting void

-Expecting main

9. /\*\*/ -Improper comment termination

10. abc\_\_ -Identifier has consecutive underscore

11. file related -file did not open

12. command line error - wrong command : No filename

13. IDENTIFIER not Declared

14. Re-declaration