Rocio Krebs

Programming Languages

Project 2

User's Manual

Setup and Compilation

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

- lexer.h
- parser.h
- table.h
- lexer.c
- parser.c
- table.c
- main.c
- Makefile
- run

- a1.in
- a2.in
- a3.in
- a4.in
- a5.in
- a6.in
- a7.in
- a8.in
- UsersManual.pdf (this file)

3. Environment: The program has been tested in the multi-platform lab and will run there.
4. Compilation: This program includes a `Makefile`. At the command line in Linux, type make. The program produces an executable entitled `main`.

Running the program: Be sure the files `a1.in, a2.in, a3.in, t4.in, t5.in, t6.in, t7.in,` and `t8.in` are in the same directory as the executables `main and run`. Issue the command `./run` No command-line arguments are required or checked.

User Input: User input is not required to run this program.

Output: The output will go into a file with the file name of the input file and a file extension .out. So, if the input file were named `legal1.in`, the generated code would go to a file named `legal1.out`. The terminal will show which files were legal and if an error occur the error will be display.

The process of running the program looks like the following screenshoot:

```
[[rk50@cs-ssh Project2]$ ls
a1.in  a3.in  a5.in  a7.in  lexer.c  main.c    parser.c  run       table.h
a2.in  a4.in  a6.in  a8.in  lexer.h  Makefile  parser.h  table.c
[[rk50@cs-ssh Project2]$ make
gcc -g -Wall -c lexer.c
gcc -g -Wall -c parser.c
gcc -g -Wall -c table.c
gcc -g -Wall -c main.c
gcc -g -Wall lexer.o parser.o table.o main.o -o main
[[rk50@cs-ssh Project2]$ ./run

Compiling a1.in
legal

Compiling a2.in
Line: 2 Identifier cannot end with underscore

Compiling a3.in
legal

Compiling a4.in
Line: 3 variable 'b' already defined

Compiling a5.in
legal

Compiling a6.in
Line: 7 syntax error expected '='

Compiling a7.in
legal

Compiling a8.in
Never found start of program
[[rk50@cs-ssh Project2]$ ls
a1.in    a3.in    a5.in    a7.in    lexer.c  main     Makefile  parser.o  table.h
a1.out   a3.out   a5.out   a7.out   lexer.h  main.c   parser.c  run       table.o
a2.in    a4.in    a6.in    a8.in    lexer.o  main.o   parser.h  table.c
[[rk50@cs-ssh Project2]$ cat a1.out
R0 = b
a = R0
*****[b]*****
R0 = b
R1 = 2
R0 = R0 + R1
a = R0
*****[b,2,+]*****
[rk50@cs-ssh Project2]$ 
```