SER502 Project Presentation
Team 4

Team Members
Zelin Bao
Yuan Cao
Yiru Hu
Lei Zhang

Outline

- Language Name
- Language Design
- Bytecode generate
- Runtime
- Virtual Machine

Language Name

The name of language is Luna, and the programming file suffix is *.lu, byte-code file suffix is *.luo.

Language Design - Lexical Tokens

Using tool - Lex

Key words -- In lowercase

Comment -- // to make a line of comment

/**/ to make a paragraph of comment

TreeNode struct -- makeNewNodes

create node for each token and initialize the node->line_no using the current line number.

Language Design - Grammar

- **Function define**
- **Statements**
  - -While statement
  - -If statement
  - -Assign statement
  - -Return statement
  - -Function call
- **Expression**
  - -Bool expression
  - -Math expression
  - -Term
  - -Function call

Language Design - Parsing AST

TreeNode:

line number

Node kind

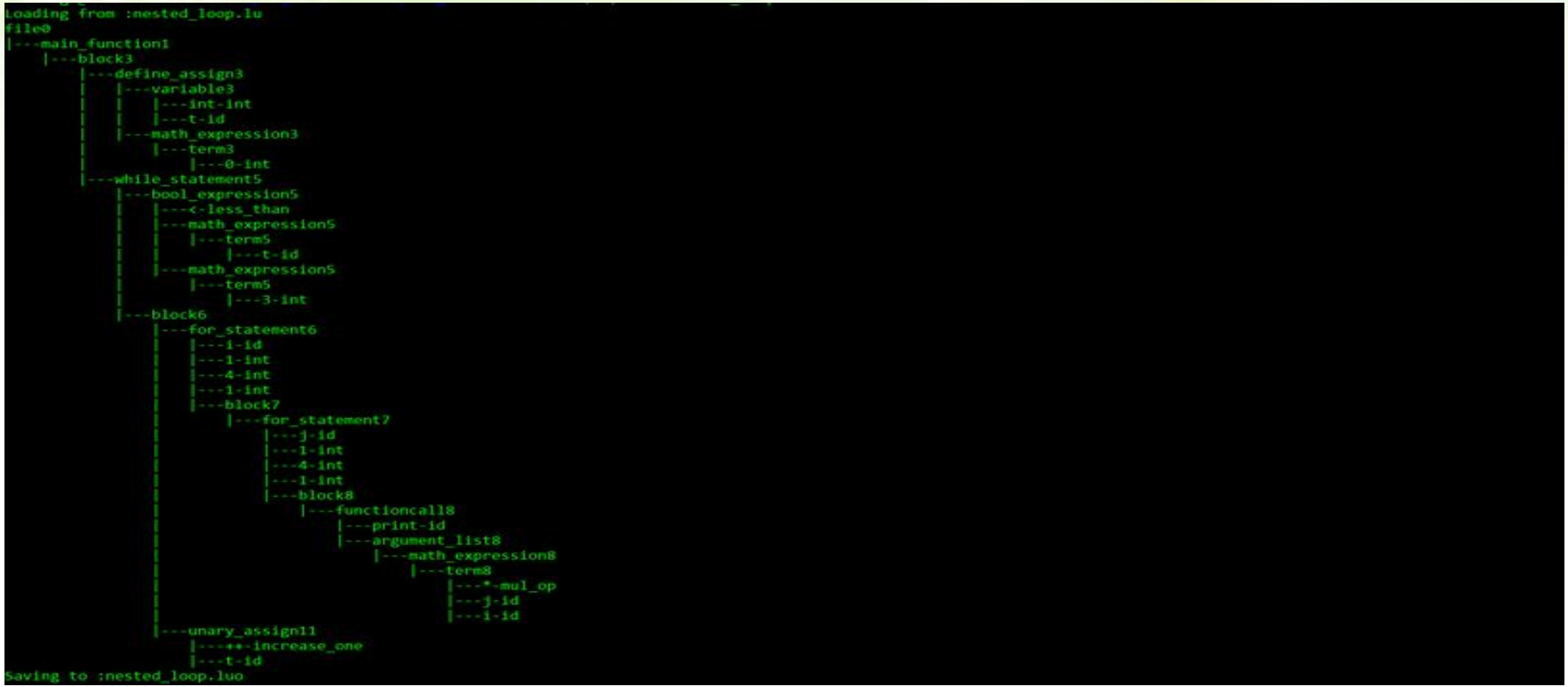
Child sibling pointer

start_ins_line & end_ins_line (used for generating jz and jmp instruction)

Literal (the content of treenode like the node, like for identifier node, the name of the identifier is the content of the node).

Turn on the debug option in macros.h to output the syntax tree.

Example of AST of example code nested_loop.lu



Bytecode generate - If statement

Structure:

IF LPAREN bool_expression RPAREN block else-statements END.

Exmaple:

If (x > 0)

x--

end

Bytecode:

Lines:	instructions	
0	LDV	X
1	LDC	0
2	GT	
3	JZ	-1
4	LDV	X
5	LDC	-1
6	ADD	
7	ASN	X
8		

Bytecode generate - For loop

Structure:

FOR int_num COMMA int_num COMMA int_num DO block END

Example:

```
for i = 0, 5, 1 do
    print(i)
    end
```

Bytecode generate - While loop

Structure:

WHILE LPAREN bool_expression RPAREN DO block END

Example:

```
i = 0
    while (i < 5)
    do
    print(i)
    i++
    end
```

Bytecode generate - Function define

```
            |---error-id
            |---argument_list9
Saving to :error.luo
error line 1: dulplicate definition of x
error line 4: unresolved reference z
error line 5: dulplicate definition of y
➜  data git:(master) ✗ cat error.lu
function int error(int x, int x)
  int y=0
  y++
  z=y
  int y=z
end

main()
  error()
end
```

(1) In the bytecode, although we have the output of function name, but the function name doesn't occupy the line number.

(2) Our function has the parameter table. we need to check whether the parameter name is repeat.

Bytecode generate - Function call

```
27 ASN a
28 LDC 64
29 ASN b
30 LDV b
31 LDV a
32 CALL gcd 2
33 CALL print 1
HALT
➜  data git:(master) ✗ cat gcd.lu
function int gcd(int x, int y)
  if (x==y)
    return x
  else
    if (x>y)
      int z=x-y
      return gcd(y,z)
    else
      z=y-x
      return gcd(x,z)
    end
  end
end

main()
  int a = 128
  int b = 64
  print(gcd(a,b))
end
```

(1) Call is a bytecode instruction which is used to call a function. The format of function call is: call f n. F is name of function, n is number of parameter.

(2) Take f(int x, int y) as an example. We LDV y first, then LDV x. so x in the top of stack and y in the second of stack.

(3) Call f n, we pop value from the stack, and n can tell us how many values we need to pop.

Bytecode generate - DefineAssign

```
            |---error-id
            |---argument_list9
Saving to :error.luo
error line 1: dulplicate definition of x
error line 4: unresolved reference z
error line 5: dulplicate definition of y
➜  data git:(master) ✗ cat error.lu
function int error(int x, int x)
  int y=0
  y++
  z=y
  int y=z
end

main()
  error()
end
```

This function is used to define a statement . for assigning an expression to a identifier.

(1) Check the duplication. If identifier has been defined report error.

(2) If it is not duplicate, add identifier into the identifier table

(3) We will generate the code of ex, it will put in the top of stack,Then assign the value from the stack to identifier.

.

Bytecode generate - IdentifierAssign

```
            |---error-id
            |---argument_list9
Saving to :error.luo
error line 1: dulplicate definition of x
error line 4: unresolved reference z
error line 5: dulplicate definition of y
➜  data git:(master) ✗ cat error.lu
function int error(int x, int x)
  int y=0
  y++
  z=y
  int y=z
end

main()
  error()
end
```

This function is used to assign a value to an exsisted identifier.

(1) check whether identifier exist, if it doesn't exist, unresolved error

(2) We will get the value of ex, it will push in the top of stack,Then assign the value from stack to identifier.

Runtime

Byte-Code Design:

```
-------------------------------------------------------------------------------
1. Assignment Instrctions
-------------------------------------------------------------------------------
LDV a          | Operand: 1        | Load the variable into stack using given name
LDC 10         | Operand: 1        | Load the constant into stack
ASN x          | Operand: 1        | Assign top element of stack to given variable name
DUP            | Operand: 0        | Duplicate the top element of stack
-------------------------------------------------------------------------
2. Arithmetic Instrctions (Pop two elements from stack, add them, push result back)
-------------------------------------------------------------------------
ADD            | Operand: 0        | Addition.
SUB            | Operand: 0        | Substraction.
MUL            | Operand: 0        | Multiplication.
DIV            | Operand: 0        | Division.
-------------------------------------------------------------------------
```

Virtual Machine