0 1 2 3 4 5 6 7

0123456789012345678901234567890123456789012345678901234567890123456789012345

Issues

* Extern static storage
* Tag name
* Enum set
* LoadValueToRegister
* CheckRegister
* GenerateIntegralMultiply
* GenerateFloatingToIntegral
* LoadAddressToRegister
* PreviousTrack

Book

* Introduction
* Middle Code Generation
  + Main Scanner
    - SlashToChar
  + Main Parser and Middle Code Generator
    - Declaration
      * class Specificator
      * class Declarator
      * class Mask
    - Statement
      * class Statement
    - Expression
      * class Expression
    - Middle Code
      * Class MiddleCode
      * Class MiddleOperator
      * Class MiddleCodeGenerator
  + Constant Expression
  + Static Expression
    - Class StaticValue
    - Class StaticAddress
  + SymbolSymbol Table
    - enum Scope
    - enum Storage
  + Type
    - Enum Sort
    - TypeCast
  + Initalizers
    - class AutoInitializer
    - class StaticInitializer
    - Windows
    - Linux
* Middle Code Optimization
* Assmbly Code Generation
  + - Class AssemblyCode
    - Class AssemblyCodeGenerator
    - Class Track and TrackEntry
  + Register Allocatation
    - Enum Register
    - Class RegisterAllocater
  + Error Handling
    - class Assert
    - enum Message
* Advanced Target Code Generator
  + Linker
* Appendix 1: Preprocessor
* Appendix 2: Standard Library
* Appendix 3: Byte Code Generation
* Appendix 4: Abstract Data Types
  + Pair, Unordered Pair, and Triple
  + ListSet and ListMap
  + Graph
* Appendix 5: Lex and Yacc
  + The Caculator
* Flow
  + Parser
    - Input
      * Source Code
    - Output:
      * Middle Code List
  + Assembly Generator
    - Input
      * Middle Code List
    - Output
      * Assemebly Code List
      * Middle To Assemebly Target Map
  + Target Code Generator
    - Input
      * Assemebly Code List
      * Middle To Assemebly Target Map
    - Output
      * Assembly File
  + Target Code Generator
    - Input
      * Assembly Code List
      * Middle To Assemebly Target Map
    - Output
      * Target Code List
      * Call Map
      * Access Map
      * Return Set
  + Linker
    - Input
      * Target Code List
      * Call Map
      * Access Map
      * Return Set
    - Output
      * Executable Code
* Karin Linder
* Linda Edler
* Problems
  + Twin tracks
  + Static value
  + Check the code
  + Translate to Java
  + bx => si
  + Map Add []
  + GetOperand []
  + AddMiddleCode
  + Switch temporary expression
  + Linux/Windows
    - ConstantExpression: 1
      * value
    - MiddleCodeGenerator: 2
      * value
      * function
    - Type: 1
      * size
    - AssemblyCodeGenerator: 2
      * Exit
      * Init
    - Main: 3
  + Syscall
    - Exists
    - Signal
    - Raise
  + Initialization, a[2][2] = {1, 2, 3, 4};
    - Set type dimension
    - Check maximal dimension
    - Generate TypeMap
    - Fix the list
      * Iterate from 1 to (top dimension - 1)
      * Add to dimension map
  + Minor
    - IList -> List.AddRange
    - MainX -> Start
  + ListMap
    - Switch cases
    - Entry map struct/union
  + MainParser
    - Parse errors
      * Function optional\_specifier
      * Specifier stack
  + DeclarationGenerator
    - Old function parameter order twice defined
  + ExpressionGenerator
    - Index pointer/array type
* classes
  + GenerateIntegral
  + GenerateFloating
  + CallHeader
  + PostCall
  + m\_recordStack
  + m\_stackSize
  + DecreaseStack
  + CheckTrackMapFloatStack
  + Record 1 -> 0
    - GetReturnValue
    - Parameter
    - Call
  + Temporary symbol
  + Parameter symbol
  + Addressable
    - Init
      * Variable, Constant, Parameter
      * Not temporary, register, or bitfield
    - Index: &a[i], Deref: &\*(p + i)
  + Assignable
    - Init
      * Not array, function, constant or temporary
    - Index: a[i] = 2;
      * a not constant
    - Deref: \*(p + i) = 2;
      * p:s pointer type not constant
  + Preprocessor
  + -----------------------------
  + ObjectCodeGenerator
    - a \* b + c \* d
    - t1 = a \* b, ax
    - t2 = c \*d, ax
    - t3 = t1 + t2
    - mov ax, [a]
    - mul [b]
    - mov bx, ax
    - mov ax, [c]
    - mul [d]
    - add ax, bx
    - a << (b \*c)
    - t1 = b \* c, ax
    - t2 = a << t1, cl
    - \*(list + index)
    - t1 = index \* 2
    - t2 = list + t1
    - t3 = \*t2
    - mov ax, [index]
    - mul [2]
    - add ax, [list]
  + ObjectCode
    - ByteList

1. Behålla andelen
2. Begära ny utvärdering av Kavlanda, och därefter räkna eller dela rätt av.
3. Acceptera nuvarande utvärdering, och räkna.
4. Acceptera nuvarande utvärdering, och dela rätt av.