

2018 Compiler Project #2. Parser

소프트웨어전공 2016024793 김유진

1. Compilation method and environment

- Compilation : "make" for cminus
- Environment: Ubuntu 18.04.1 LTS

2. Implementation of C-Minus Parser using Yacc

- Syntax Tree 구조

배열 변수 선언 시, 그 크기를 자식노드에 저장하는 것이 아니라 그 노드의 attr 에 저장해주었다. 이를 제외한 나머지 구조는 명세서의 구조를 그대로 구현하였다.

■ globals.h

NodeKind 에 DeclK, ParamK 를 추가해 주고, 각 종류의 노드의 종류를 수정 및 추가하였다.

```
65 + typedef enum {StmtK, ExpK, DeclK, ParamK} NodeKind;
66 + typedef enum {IfK, CompK, IterK, RetK} StmtKind;
67 + typedef enum {OpK, ConstK, IdK, ArrIdK, AssignK, CallK, TypeK} ExpKind;
68 + typedef enum {FuncK, VarK, VarArrayK} DeclKind;
69 + typedef enum {ArrParamK, SingleParamK} ParamKind;
```

변수, 배열변수, 함수의 선언에서 저장해야 할 정보들을 추가할 구조체들(VarAttr, ArrayAttr, FuncAttr)을 선언하고, 이를 treeNode 구조체의 attr 에 추가하였다.

■ util.c, util.h

globals.h 에 추가한 DeclK 와 ParamK 노드를 생성해주는 newDeclNode(), newParamNode() 함수를 추가하였다. 그 밖에 생성된 Syntax Tree 를 알맞게 출력해주는 printTree() 함수를 수정하였다.

■ main.c

Syntax Tree 만을 출력하기 위해 flag 값들을 수정하였다.

■ Makefile

scanner.c 와 parser.c 파일을 제외하고, lex 와 yacc 로 생성된 scanner 와 parser 를 사용하도록 수정하였다. 또한 globals.h 에서 수정한 사항들이 analyze.c 과 cgen.c 를 컴파일 할 때 문제를 발생시켜 이를 제외하고 컴파일하도록 했다.

- cminus.y

기존 tiny.y 파일을 수정하여 구현하였다.

ID 와 NUM 을 그때 그때 저장해주어야 했기 때문에, 따로 states (saveName, saveNum)를 만들어주어 별도로 처리하게 하고 곧바로 생성된 노드에 값들을 저장해주었다.

```
54 + saveName      : ID
55 +               { savedName = copyString(tokenString);
56 +               savedLineNo = lineno;
57 +               }
58 +               ;
59 + saveNum       : NUM
60 +               { savedNumber = atoi(tokenString);
61 +               savedLineNo = lineno;
62 +               }
63 +               ;
```

```
65 + var_decl      : type_spec saveName
66 +               { $$ = newDeclNode(VarK);
67 +               $$->attr.var.name = savedName;
68 +               $$->lineno = lineno;
69 +               $$->attr.var.type = $1->attr.type;
70 +               }
71 +               SEMI { $$ = $3; }
72 +               | type_spec saveName
73 +               { $$ = newDeclNode(VarArrayK);
74 +               $$->lineno = lineno;
75 +               $$->attr.arr.name = savedName;
76 +               }
77 +               LBRACE saveNum RBRACE SEMI
78 +               { $$ = $3;
79 +               $$->attr.arr.size = savedNumber;
80 +               $$->attr.arr.type = $1->attr.type;
81 +               }
82 +               ;
```

3. Example and result screenshot

- test.cm (명세서 예시와 동일한 코드) 파싱 결과

```
clare@ubuntu:~/Desktop/cmp$ ./cminus test.cm

CMINUS COMPILATION: test.cm

Syntax tree:
Function declaration, name: gcd, return type: int
  Single Parameter, name: u, type: int
  Single Parameter, name: v, type: int
  Compound Statement :
    If : (condition) (body) (else)
      Op : ==
      Id : v
      Const : 0
      Return :
        Id : u
      Return :
        Call, name : gcd with arguments below
          Id : v
          Op : -
          Id : u
          Op : *
          Op : /
          Id : u
          Id : v
          Id : v
Function declaration, name: main, return type: void
  Single Parameter, name: (null), type: void
  Compound Statement :
    Variable declaration, name: x, type: int
    Variable declaration, name: y, type: int
    Assign : (destination) (source)
      Id : x
      Call, name : input with arguments below
    Assign : (destination) (source)
      Id : y
      Call, name : input with arguments below
    Call, name : output with arguments below
    Call, name : gcd with arguments below
      Id : x
      Id : y
```

- test2.cm 파싱 결과

```
clare@ubuntu:~/Desktop/cmp$ cat test2.cm
int aaa[1234];
int function(int a, int b, int c[], int d) { }
clare@ubuntu:~/Desktop/cmp$ ./cminus test2.cm

CMINUS COMPILATION: test2.cm

Syntax tree:
  Array declaration, name: aaa, size: 1234, type: int
  Function declaration, name: function, return type: int
    Single Parameter, name: a, type: int
    Single Parameter, name: b, type: int
    Array Parameter, name c, type: int
    Single Parameter, name: d, type: int
    Compound Statement :
```

- test3.cm 파싱 결과

```
clare@ubuntu:~/Desktop/cmp$ cat test3.cm
/* example code */

void main(void)
{
  int i; int x[6];
  i = 0;
  while(i<=4)
  {
    i = i + 1;
  }
}
clare@ubuntu:~/Desktop/cmp$ ./cminus test3.cm

CMINUS COMPILATION: test3.cm

Syntax tree:
  Function declaration, name: main, return type: void
    Single Parameter, name: (null), type: void
    Compound Statement :
      Variable declaration, name: i, type: int
      Array declaration, name: x, size: 6, type: int
      Assign : (destination) (source)
        Id : i
        Const : 0
      While : (condition) (body)
        Op : <=
        Id : i
        Const : 4
        Compound Statement :
          Assign : (destination) (source)
            Id : i
            Op : +
            Id : i
            Const : 1
```