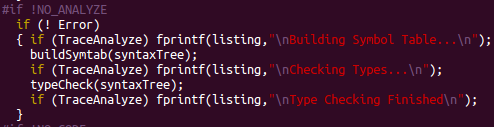
Project #3 Semantic Analysis

2013012148 이재일

# Main.c

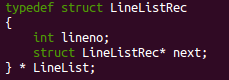
우선 analyze를 출력하기 위해서 define문을 변경시켜준다.

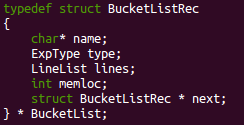


# Symtab.h

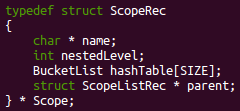
LineList를 구현해서 몇번 line에 호출이 되었는지 알 수 있도록 한다.



Bucket List를 구현해서 각 scope의 함수, 변수를 저장한다.



Scope를 구현한다(static Scope형태로 구현을 한다)

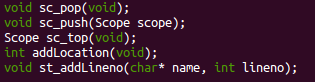


가장 상위 scope를 header file에 선언한다.



analyze.c에서도 사용할 함수들을 header에 선언해준다.

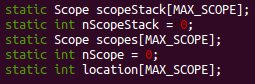




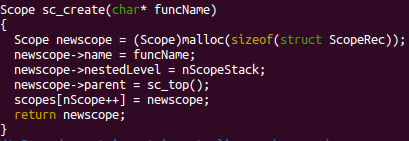
# Symtab.c

Static Scope형식으로 구현을 한다.

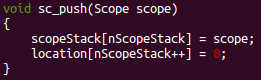
nScope 전체 Scope의 수를, nScopeStack은 stack에 쌓인 Scope의 수를 의미한다. scopes 배열에는 모든 scope가 들어간다. scopeStack배열에는 stack에 있는 Scope가 들어간다.



scope 생성 함수이다. 함수 이름을 scope 이름으로 한다(가장 상위의 scope의 이름은 global로 한다). parent정보와 nested level, name, 그리고 bucket들의 배열 scopes를 원소로 가진다.



scope를 stack에 넣고, 빼는 함수이다. 해당 scope의 함수, 변수들에 대한 작업이 다 끝나면 stack에서 빼준다.



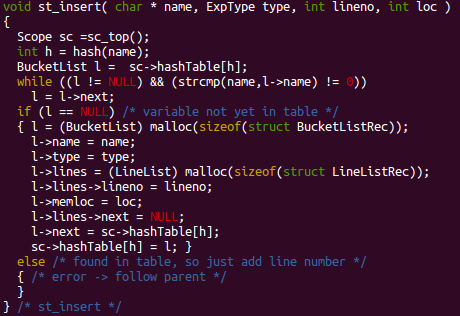




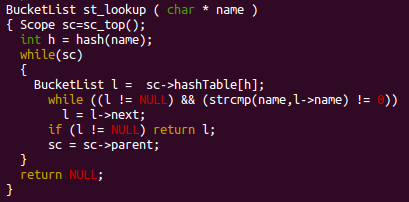
depth를 설정해주는 함수이다.



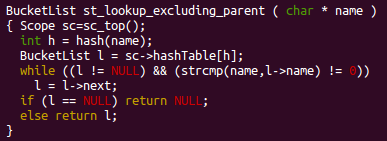
stack의 가장 바깥의 scope에 bucketList(변수,함수 저장)를 넣는 함수이다. bucket은 name, type, LineList, location, next bucket 정보를 가진다.



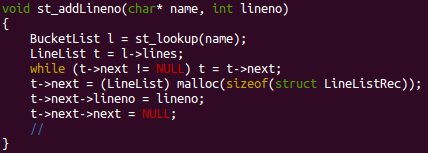
현재 scope부터 상위 scope 내에서 같은 이름인 함수, 변수를 찾는 함수이다.



현재 scope에서만 같은 이름인 함수, 변수를 찾는다.

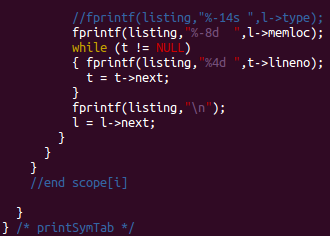


해당 함수, 변수가 나타난 line을 저장하는 함수이다.



symbol table을 출력하는 함수이다.

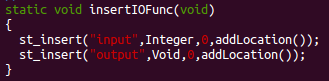




# analyze.c

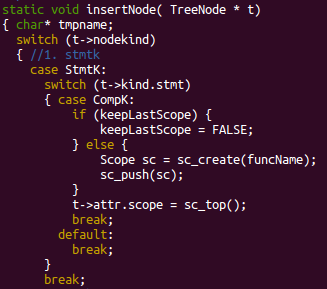
<build symtab>

insertIOFunc : building 함수인 input, output 함수를 global scope에 넣는 함수이다.

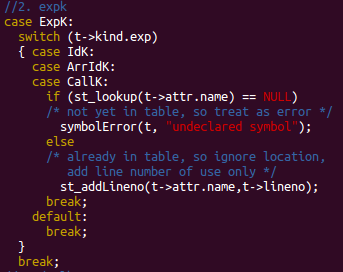


insertNode : treeNode를 symbol table에 넣는 함수이다.

treeNode의 type이 stmtK이고, kind.stmt이 CompK일 경우 keepLastScope 변수를 확인한다. keepLastScope가 TRUE일 경우 새로운 함수의 몸통부분이 derivation 되는 중이므로 기존 scope를 사용하고 해당 변수는 FALSE로 변환해준다, FALSE일 경우 새로운 scope를 생성해서 stack에 넣어준다. 그리고 새로운 treeNode를 새로 생성한 scope에 넣는다.

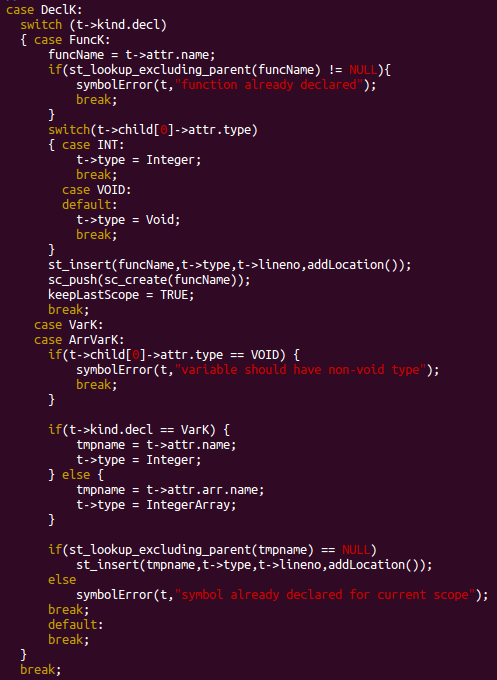


treeNode의 type이 ExpK인 경우, kind.exp가 CallK인 경우 해당 함수가 선언되어 있지 않다면 symbolError를 발생시킨다. 선언이 되어 있는 경우 호출된 위치의 line을 저장해준다.

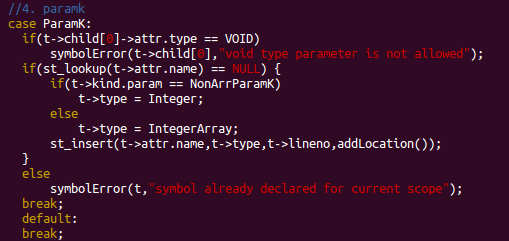


type이 DeclK일 때, kind.decl이 FuncK인 경우 같은 scope에서 같은 이름으로 선언된 함수가 있으면 error를 발생시킨다. 해당 treeNode의 첫번째 자식(return형이 저장되어 있다)의 type을 treeNode의 type으로 setting해준다. 그 후 함수명으로 한 새로운 scope를 생성해서 stack에 넣어준다. keepLastScope변수를 TRUE로 assign해서 compK type의 treenode가 insert되는 경우 기존의 scope를 사용하도록 해준다.

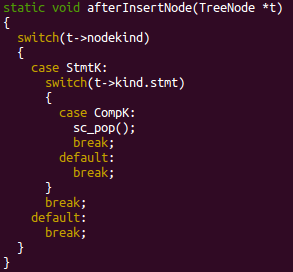
kind.decl이 VarK와 ArrVarK인 경우 첫번째 child(자료형이 저장되어 있다)기 Void로 선언되어 있거나 같은 scope 내에 같은 이름을 가진 함수(변수)가 선언되어 있으면 error를 발생시킨다. 그렇지 않은 경우 해당 kinde.decl에 따라서 Integer 아니면 IntegerArray로 kind를 setting해준다.



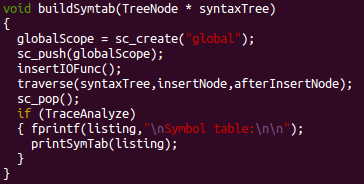
type이 ParamK인 경우, 자료형이 Void이거나 이전에 선언되지 않은 변수가 매개변수로 전달될 경우 error를 발생시킨다. 그렇지 않은 경우 현재 scope에 treeNode를 넣어준다.



treeNode를 scope에 삽입 후 kind.stmt이 CompK인 경우 stack에서 pop 연산을 수행해준다.



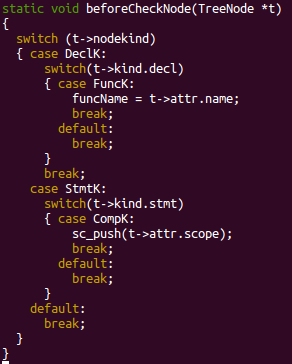
symtab를 만드는 함수이다. syntaxTree를 traverse하면서 적절한 treeNode들을 symbol table에 넣어준다.



<typecheck>

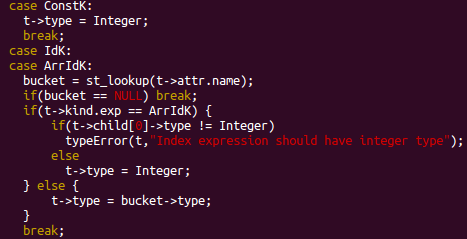
type check 전, treeNode가 DeclK type의 kind.decl이 FuncK인 경우 전역변수 funcName을 해당 노드의 attr.name으로 설정해준다(후에 return type check 시 사용을 위해서 저장).

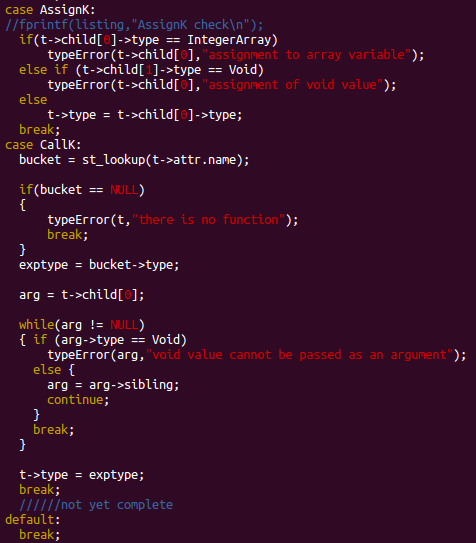
treeNode의 StmtK type의 kind.stmt이 CompK인 경우 해당 노드 scope를 stack에 넣어준다(type check를 위해서).

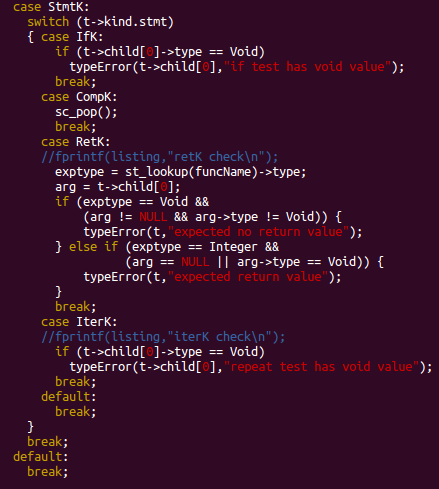


treeNode의 type check하는 함수이다.



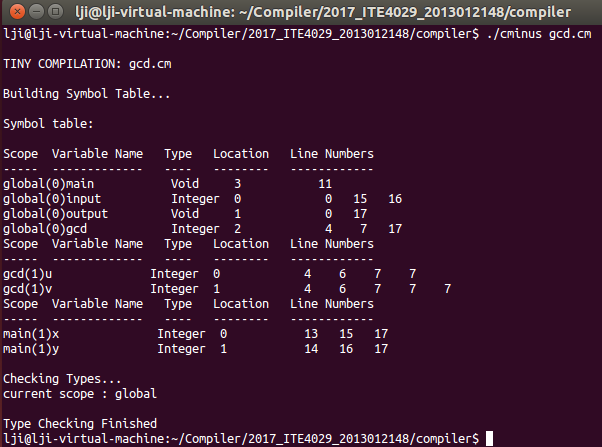




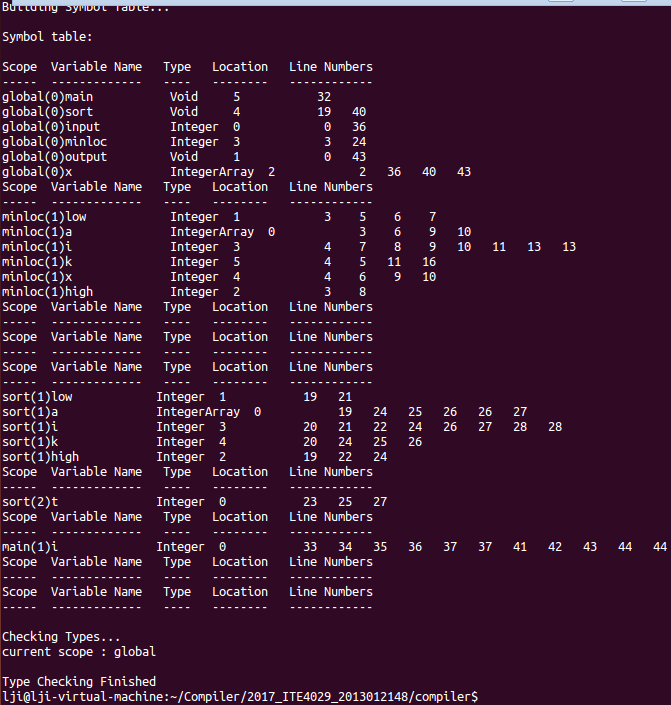


# 실행결과

gcd.cm 실행결과



sort.cm 실행결과



type check 실행결과

