**Project #3: Semantic Analysis**

2017029970 우원진

#1 Compile Method

텍스트이(가) 표시된 사진

자동 생성된 설명

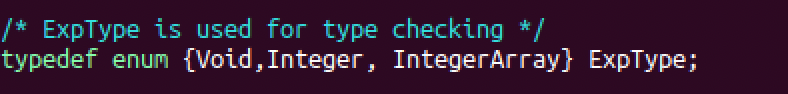
위 사진과 같이 CommandLine에 make를 입력하면 컴파일이 자동으로 된다.

#2 How to Operate



위와 같이 ./cminus target-file 명령어를 입력하면 실행이 된다.

#3 Code Explanation(Modified Code)



텍스트이(가) 표시된 사진

자동 생성된 설명

먼저 모든노드의 Type 을 정해 주기 위해 treeBode struct에 ExpType type;라는 변수를 추가했다.

텍스트이(가) 표시된 사진

자동 생성된 설명

BucketList 구조체에 이 Bucket(Variable) 이 어떤 Type을 가지는지 ExpType type , 이 Variable이 Function인지 알려주는 변수 int isFunc, 또 만약 함수라면 Parameter를 가져야하는 경우가 있기때문에 Parameter의 type을 저장하는 변수 ExpType params[10](파라미터의 갯수를 최대 10개로 지정했다. ), 저장된 Parameter의 갯수를 알려주는 int paramNumber 를 추가했다.

텍스트이(가) 표시된 사진

자동 생성된 설명

C-Minus는 tiny와 다르게 Scope를 가지고있기 때문에 struct ScopeListRec를 선언했다.

Scope 안에는 Scope 이름을 저장하는 char\* name, SymbolTable을 위해 BucketList를 저장하는 BucketList bucket[SIZE],  아래에 나올 printSymTab()함수를 위해 Scope들을 Linked list 형태로 순회하기 위한 struct ScopeListRec\* next, Scope들은 계층적 구조를 가지고 있기때문에 이를 계층적으로 표현하기위한 struct ScopeListRec\* parent, 이 스코프 내에서의 현재 memory 위치를 표현하는 location 변수들이 저장되어있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

위의 사진은 SymbolTable을 만들기 위해 필요한 함수들이다.

create\_scope 함수는 Scope의 이름으로 지정할 String 을 받아서 새로운 Scope를 만들어서 반환해주는 함수이다.

find\_scope 함수는 Scope의 이름을 받아서 해당 스코프를 찾아서 반환해주는 함수이다.

add\_line 함수는 추가할 bucket과 lineno를 받아서, 해당 bucket에 bucket이 사용된 line을 추가해주는 함수이다.

st\_lookup 함수는 Scope와, 찾으려는 bucket의 이름을 인자로 받아서, 해당 Scope에서 계층적인 구조로 올라가면서 해당 bucket을 찾으면 bucket을 반환해주고 찾지못하면 NULL을 반환하는 함수이다.

st\_lookat 함수는 Scope 와, 찾으려는 bucket의 이름을 인자로 받아서, 해당 Scope에서만 해당 bucket을 찾고 있으면 반환해주고 없으면 NULL을 반환하는 함수이다.

insertFuncParam은 함수를 선언할때, Paramete가 존재하면 그 Parameter들의 Type을 순서대로 해당 function bucket의 params에 넣어주는 함수이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이는 이번 과제를 하기위해 선언한 변수들이다.

Static char\* funcName 은 함수를 선언하거나 체크할때를 위해 해당 함수의 이름을 담는 변수이다.

static int isForFunc 는 CompK가 나왔을때 이 CompK가 Function Declare를 위한 CompK인지를 알기 위해 선언한 것이다.

Static char\* tempName은 처음에 switch-case 문법에서 case안에 변수 선언이 안되길래 사용했는데 후에는 Compound를 이용해 해결했다. Static BucketList tempBucket도 마찬가지이다.

extern ScopeList currentScope, extern ScopeList globalScope는 analyze.c에 선언되었지만, symtab.c 에서도 사용해야하고, main에서도 사용해야하기 때문에 extern으로 선언했다. globalScope는 말그래도 처음생성된 Scope은 GloblaScope를 가리키는 포인터이고, currentScope는 계층적인 구조에서 현재 어떤 Scope를 보고있는지 가리키는 포인터이다.

Static int compNum은 이름이 없는 Compound statement에 이름을 붙이기 위해 사용되는 변수로, Compound Statement로 Scope가 생성되면 1씩 증가하게 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

buildSymtab함수를 보면 먼저 GlobalScope를 만들어주고, currentScope이 GlobalScope를 가리키도록 하고, 내장 함수인 output(), input()함수 Global Scope에 담아주는 함수를 실행했습니다.

텍스트이(가) 표시된 사진

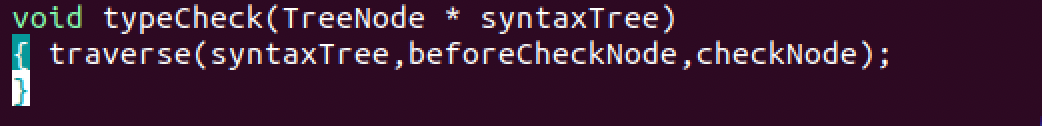
자동 생성된 설명

그리고 insertNode가 실행된다음 실행되는 함수, afterInsertNode를 traverse의 postproc함수에 넣었습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이 함수는 Compound Statement가 끝나면 currentScope가 그 Compound Statement의 parent로 바뀌어야 하기 때문에 위와같이 코드를 작성했습니다.



typeCheck함수에서는 반대로 CheckNode 함수를 실행하기전에 beforeCheckNode함수가 실행되도록 traverse함수의 preProc에 beforeCheckNode함수를 넣었습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이것도 단순히 Node가 함수를 선언하는 것이면 해당 함수의 이름을 이전에 선언했던 funcName 변수에 담았고, Compound Statement를 만났다면 해당 Compound Statement의 이름을통해 Scope를 찾은 후 해당 Scope로 currentScope가 가리키도록 했습니다.

insertNode함수와 CheckNode함수도 설명하고 싶지만 너무 양이 많을것 같네여..

#4 실행결과

Input(1)

텍스트이(가) 표시된 사진

자동 생성된 설명

Output(1)

텍스트이(가) 표시된 사진

자동 생성된 설명

Input(2)

텍스트이(가) 표시된 사진

자동 생성된 설명

Output(2)

텍스트이(가) 표시된 사진

자동 생성된 설명

이메일을 통한 질문에도 빠르게 답장해주시고, 질문도 친절하게 잘 받아주셔서 감사합니다.!!

한학기 동안 정말 감사했습니다!!