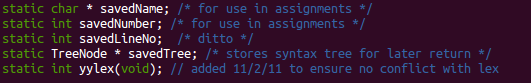
컴파일러 HW#2 Parser

2013012148 이재일

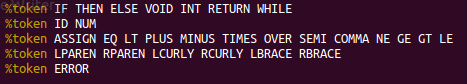
# 1) cminus.y

CFG grammar가 들어있는 파일이다. cminus.y파일을 input으로 집어넣으면 yyac가 c로 만들어진 parser를 생성한다(yacc -d cminus.y 명령어를 수행하면 y.tab.h와 y.tab.c 파일이 만들어진다).

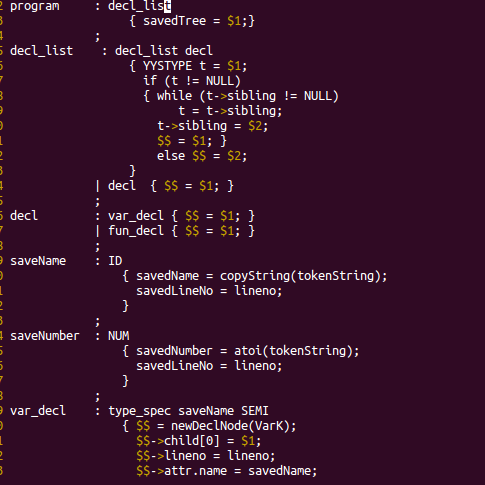
우선 ID(변수명이나 함수명)을 가져와야 하는 경우를 위해서 savedName과 savedNumber변수를 static으로 선언해서 ID를 가져오기 위한 별도의 처리를 했다(따로 저장을 하지 않으면 뒤의 토큰 처리시 해당 정보가 날아간다).



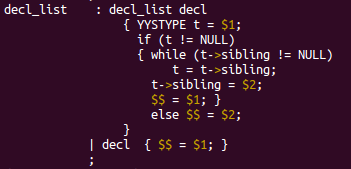
또한 지난 과제였던 cminus.l과 연동시키기 위해서 scanner에서 읽는 Token들로 변경해준다.



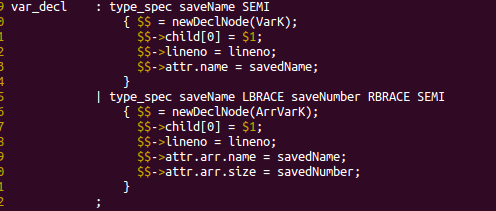
과제에 명시되어 있는(6page) BNF Grammer를 구현한다. BNF Grammer는 LHS->RHS 형태이다. 밑의 program이 LHS이고 RHS가 decl\_list이다. { } 사이에는 c코드가 들어간다. $변수를 통해서 token에 접근할 수 있다. $1이라는 뜻은 RHS의 첫번째 토큰을 의미한다.



YYSTYPE는 parsing stack의 value type을 definition하는 전처리 symbol이다. YYSTYPE t가 NULL인 경우는 decl\_list에 decl이(t->sibling)이 없다는 의미이므로 decl로 derivation해준다. NULL이 아닌 경우 t->sibling이 NULL이 아닐 때 까지 while문을 순회한다.



globals.h에 존재하는 treeNode 구조체의 원소 child 배열을 사용하여서 필요한 token들을 저장할 수 있다. 원소 lineno에 해당 line번호를 저장한다. 원소 union attr에 변수명/함수명/배열원소번호 등을 저장할 수 있다.



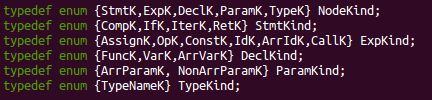
# 2) globals.h

yacc폴더에 있는 globals.h 파일을 기반으로 변형시켜서 사용한다.

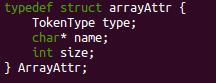
yacc -d cminus.y를 통해서 생성한 y.tab.h를 include해준다.



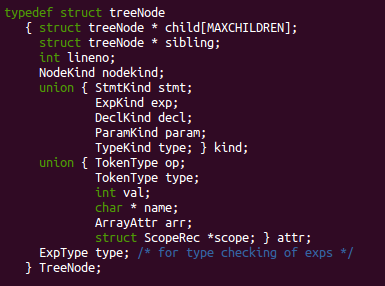
cminus.y의 Grammar들의 non-terminal들의 type에 대해 선언해준다.



attr이 배열일 경우를 위해서 따로 구조체를 만들어준다.

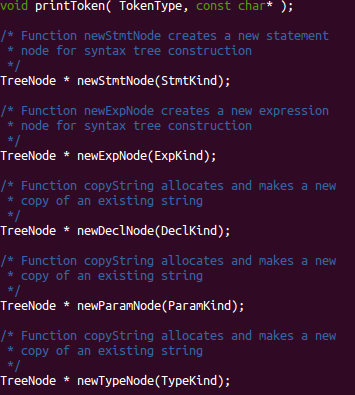


트리 노드 구조체에서 추가해준 nonterminal의 type을 입력해준다.



# 3) util.h

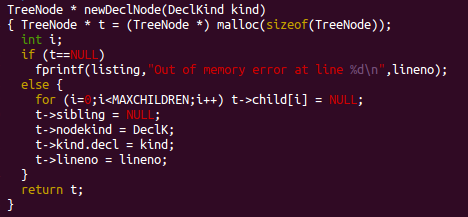
Nonterminal type별로 처리를 해서 treeNode를 반환해주는 함수들을 선언해준다.



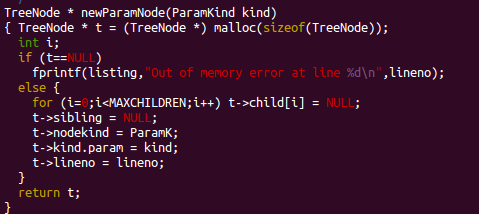
# 4) util.c

printToken함수로 derivation된 token들을 print해준다. 또한 Non Terminal type별로 treeNode를 반환해주는 함수들이 정의되어 있다.

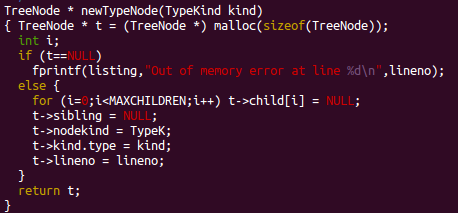
Declaration관련 NonTerminal들을 다루는 함수이다.



parameter관련 NonTerminal을 다루는 함수이다.



Type관련 Non Terminal들을 다루는 함수이다.

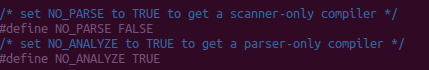


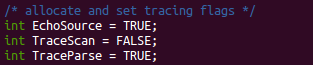
printToken함수이다.



# 5) main.c

main함수에서 parser처리를 위해 상수들을 수정해준다.





# 6) 실행결과

