Report

Project1. Scanner

2016025196

김동규

**프로젝트 목표**

Text로 구성된 c 코드를 parsing하는 scanner를 구현한다.

**컴파일 method와 환경**

컴파일은 makefile을 사용해서 했으며, Windows의 WSL – Ubuntu 20.04LTS를 사용했다.

**구현한 Token**

ENDFILE, ERROR

Reserved words : IF, ELSE, WHILE, RETURN, INT, VOID

Multicharactor token : ID, NUM

Special token : ASSGIN(=), EQ(==), NE(!=), LT(<), LE(<=), GT(>), GE(>=), PLUS(+), MINUS(-), TIMES(\*), OVER(/), LPAREN( ( ), RPAREN( ) ), LBRACE( [ ), RBACE( ] ), LCURLY( { ), RCULY( } ), SEMI( ; ), COMMA( , )

**구현한 State**

START, INEQ, INSKIP, SKIP, ENDSKIP, INNUM, INID, DONE

State 동작 알고리즘

getToken() 함수는 현재 읽는 파일의 지점부터 1개의 token을 찾을 때까지 탐색하는 함수이다. 항상 start state에서 시작하며, 읽은 토큰의 type에 따라 적절한 state로 전이하며 확실한 토큰이 나올 경우 done state로 전이한 뒤 반환한다.

이 과정에서 토큰의 형태를 확정하기 위한 state로 INEQ, INSKIP, ENDSKIP, INNUM, INID등의 state가 선언되었다.

해당 state는 token들의 겹쳐지는 character에 따라 분기한다.

ASSIGN( = ), EQ( == ), GE( >= ), LE( <= )가 =의 유무에 따라 분기되며,

OVER( / ), 주석처리( /\* ~ \*/)가 /다음 \*의 여부에 따라 분기된다.

따라서 1번 조건의 경우 INEQ state로 넘어가 currentToken에 따라 적절한 알고리즘에 따라 state와 currentToken을 결정하며,

2번 조건의 경우 INSKIP의 state로 넘어가 주석처리 여부를 결정한다.

주석 처리의 경우, INSKIP – SKIP – ENDSKIP의 state로 전이하며 주석을 처리한다.

SKIP의 경우 “/\*”가 확정될 경우 넘어오는 state며, 이후 \*가 나올 때까지 character를 저장하지 않고 넘어가며, \*가 나올 시 다음 character에 따라 /이 나올 시 ENDSKIP으로, 아닐 시 SKIP state로 돌아간다.

이렇게 하여, “/\*“이후 부분은 “\*/”가 아니라면 어떤 char이 와도 skip할 수 있다.

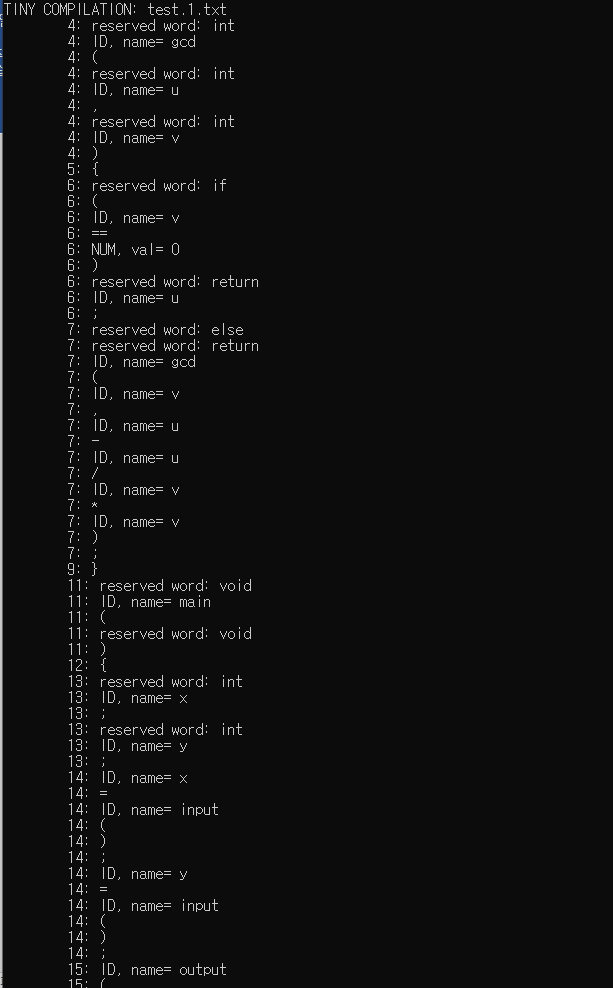
**기존 C 컴파일러 조건과의 차이점**

1. \*=, +=, -=등의 instruction을 고려하지 않았다.
2. ! instructor를 구현하지 않았다.
3. 변수형으로 INT와 VOID만 구현했다.
4. 반복문은 WHILE만 구현하였다.

**Example code와 Result**

test.1.txt와 test.2.txt, ex\_input.txt를 사용했다.

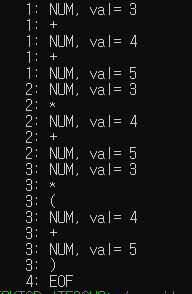
Test.1.txt



Test.2.txt

텍스트이(가) 표시된 사진

자동 생성된 설명

Ex\_input.txt  


Test1과 test2는 결과화면을 다 찍지 못했지만, 세 종류의 테스트 모두 정상 작동 하는 것을 확인했다.