Design and Development of Compiler

for C- Language

(설계 프로젝트 수행 결과)

과목 명 : [CSE4120] 기초 컴파일러 구성

담당 교수 : 정성원 교수님

개발자 : 20141602 황기덕

개발 기간 : 2019/03/21~2019/03/27

프로젝트 제목 : Design and Development of Compiler for C-Language:

Phase 1: Design and Implementation of Lexical Analyzer

제출일 : 2019년 3월 26일

개발자 : 황기덕

1. 개발 목표

* 현재 기본 코드에는 틀만 잡혀있다. C- 문법이 제대로 들어온다고 가정하고, Lexical Analyzer에 관한 구현을 실시한다.

1. 개발 범위 및 내용
   1. 개발 범위

* lexeme file

lexeme들이 어떤 토큰인지 확인하기 위해, 각각에 맞는 규칙을 만들어 주어야 한다.

* token

token을 정의해줘 각각의 lexeme이 알맞은 토큰과 매칭되어 출력되게끔 해준다.

* 1. 개발 내용
* lexeme file

tiny.l 파일에 기본적으로 rule들이 정의되어있는데, C-에서 사용하지만 정의되지 않은 부분들을 추가하여 C- language에서 사용하는 모든 토큰들에 대해 대응될 수 있도록 해준다.

* token

globals.h 헤더파일에 각각의 토큰을 서로 구별할 수 있도록 enum 구조체를 만들고, 각각에 대응되게끔 변수를 알맞게 설정한다. 그 후 util.c 파일에서 어떠한 토큰이 들어왔을 때 어떻게 처리할지 정해준다.

1. 추진 일정 및 개발 방법
   1. 추진 일정

03/21 : 요구사항 분석

03/22 ~ 03/25 : 프로젝트 개발

03/26 ~ 03/27 : 보고서 작성

* 1. 개발 방법
* lexeme / token

현재 기본 코드는 일부의 lexeme과 token만 반응하고, 그것을 분석할 수 있다. C-에서 사용하는 모든 문법, 토큰 등에 대응하여 line number, token, lexeme을 출력할 수 있도록 코드를 변경해준다. 그러기 위해서 globals.h 의 enum타입에서 각각 맞는 변수를 생성해 주었고, tiny.l 에 각각 변수를 대응시켜 주었고. util.c 에서 각 변수, lexeme들이 들어오면 그것에 대응하는 것을 출력해주었다.

* 1. 추가 코드
* Makefile

실행 파일과 lex.yy.c 등의 파일을 만들고, 오브젝트 코드등을 만들고 컴파일을 하기 쉽게끔 추가해주었다.

* readme.txt

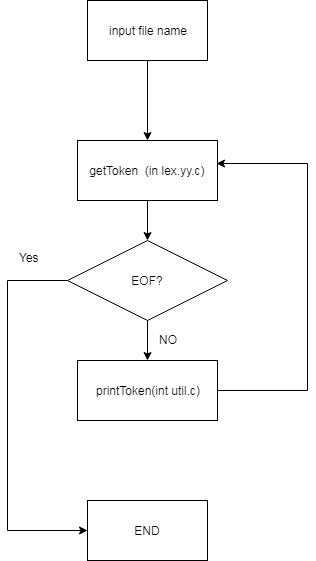
간단한 사용방법을 적어두었다.

* scan.h

예시 코드 중, 컴파일을 할 때의 dependency 때문에 추가해 두었다. 딱히 큰 내용은 들어가있지 않다.

1. 연구 결과
   1. 합성 내용

* 전체 구조



tiny.l 파일을 만든 후 flex를 통해 lex.yy.c를 만든다. main.c에서 분석할 파일 이름을 넘겨주고 lex.yy.c에서 lexeme들이 어느 token인지 분석하고, util.c에서 출력 형식에 맞게끔 출력해준다.

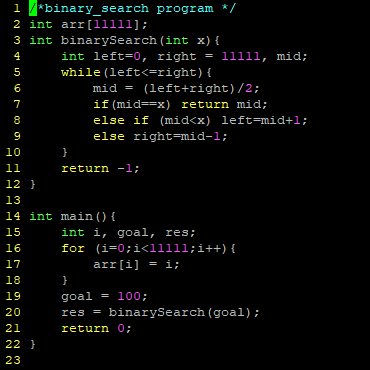
* 1. 분석 내용
* lexeme file, token

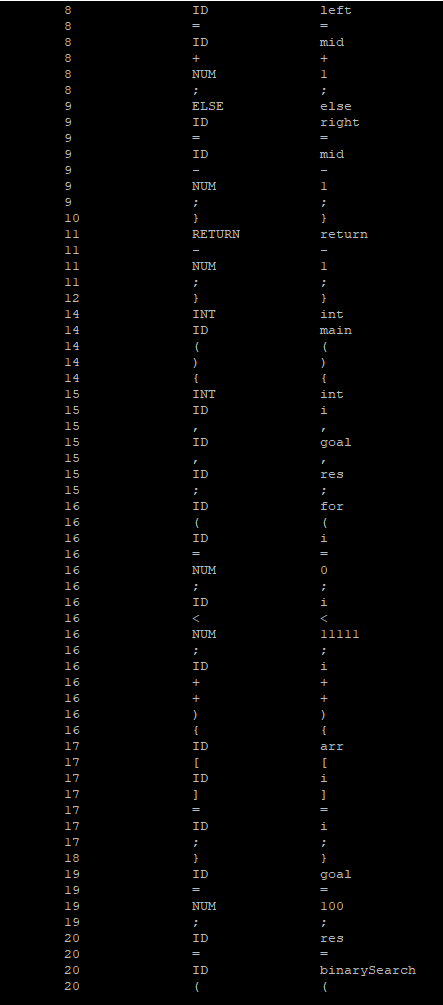
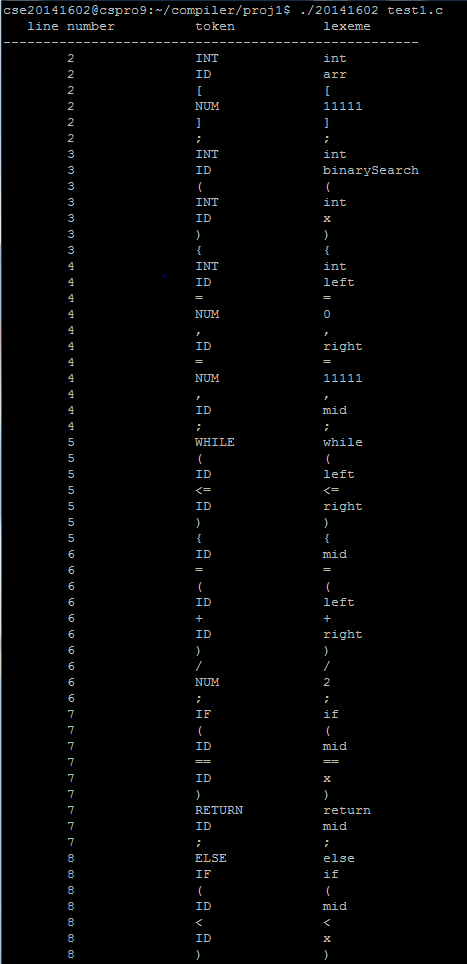
C- 언어로 작성되어 주어진 테스트 파일들 내에 있는 lexeme들을 각각에 맞는 토큰으로 구분해주어야 한다. 그러므로 lex.yy.c 파일을 새로 만들어야 하는데 이는 flex를 이용해 해결한다. 출력해주는 부분은 util.c/h를 수정해야 할 것이다. 각각에서 사용하는 변수들은 globals.h에 추가해 주었다.

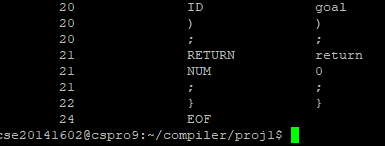
* 1. 제작 내용
* lexeme file , token

본 교재에 구현되어있는 tiny.l 을 참고하여 수정하였다. 본 교재에는 tny 파일에 있는 lexeme들을 구분하기 위하여 tiny.l 을 만들어 두었는데, 내용을 C-에 맞게끔 수정해 주었다. 그 후 flex를 사용하여 lex.yy.c 파일을 만들어 주었다. 출력 내용을 맞추기 위해 util.c/h 를 수정해 주었다. 각 파일에서 사용하는 변수들은 globals.h 에 추가해 주었다.

* 1. 시험
* ./20141602 test1.c

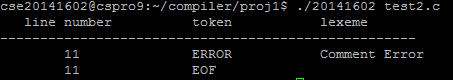






* ./20141602 test2.c





* 1. 평가
* 안정성, 신뢰성

위 시험 내용에서 예시로 든 두 개의 소스코드 말고 다른 C- 소스 코드를 넣어도 문제없이 제대로 동작하고, 반복 수행해도 결과값이 다르게 나오지 않는다. 이와 같은 이유로 안정성과 신뢰성이 잘 갖추어져있다고 볼 수 있다.

1. 기타
   1. 자체 평가

* 출력 예시와 비교해보고, 명세서에 나와있는 것들을 모두 구현했기 때문에 최초의 목표는 모두 달성했다고 본다.

* 1. 소감
* 지금까지 우리가 당연하게 생각했던 변수, 문법 등의 처리에 대해서 어떠한 방식으로 되고 있었는지 알게 되는 프로젝트였다.