

1. Compilation method and environment

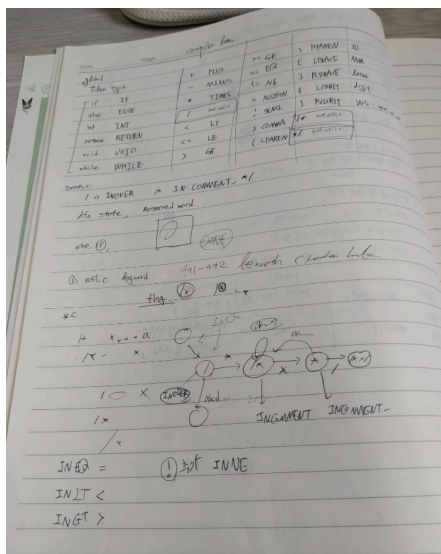
Environment

OS : Ubuntu 18.04.1 LTS x86_64
gcc: 7.3.0 (Ubuntu 7.3.0-16ubuntu3)

Compilation method

cminus : make cminus
cminus_flex : make cminus_flex
cminus, cminus_flex : make all
clean : make clean

2. How to Implement, and how to operate



2-1 global.h 에 있는 token들이 어떻게 어떤걸 의미하는지 정리

- reserved word 중요한 키워드들 이라고 생각
- ID, NUM 변수와 숫자
- special symbol 에러가 아닌 특별한 심볼들

2-2 util.c 에서 토큰에 따라 출력하는 방식이 다르다는 것을 보고 토큰만 리턴하도록 이해

2-3 scan.c 과제에 명시된 INEQ, INNE 등을 어떻게 쓰는지 생각하다 INCOMMENT만 INCOMMENT_

임을 보고 주석처리하는 부분을 dfa로 그려봤는데 IN이 prefix인 상태들은 dfa로 처리함을 알게됨

ex) /* 는 /가 나온상태에서 *이 만나오고 다른 char이 오면 OVER(/)가 되고 *이 나오면 INCOMMENT(/*) 새로 생긴 상태와 바뀐상태 (:= -> ==) 등을 처리

** 2-3 까지의 FLOW

글자를 하나하나 getToken을 할 때 글자하나하나마다 보고 내가 원하는 reserved word나 special symbol이 나 ID, NUM 을 확인한 후 리턴하고, 에러가 있다면 에러를 리턴한다

2-4 cminus_flex를 만들기 위해서 tiny.l을 copy 해 cminus.l을 만들고 과제에 명시된 word나 symbol이 나오면 토큰 이름을 리턴하게 해서 scan.c 의 역할을 대신한다.

3. Example and Result Screen shot

3-1 compile

Usage : make all

```
clucle@clucle:~/compiler/2018_ELE4029_2014005123$ make all
gcc -fno-builtin -c main.c
gcc -fno-builtin -c util.c
gcc -fno-builtin -c scan.c
gcc -fno-builtin -c parse.c
gcc -fno-builtin -c symtab.c
gcc -fno-builtin -c analyze.c
gcc -fno-builtin -c code.c
gcc -fno-builtin -c cgen.c
gcc -o cminus -fno-builtin main.o util.o scan.o parse.o symtab.o analyze.o code.o cgen.o
flex cminus.l
gcc -fno-builtin -c lex.yy.c -lfl
gcc -fno-builtin main.o util.o lex.yy.o -o cminus_flex -lfl
```

3 -2 cminus

Usage : ./cminus (cminus file name)

```
clucle@clucle:~/compiler/2018_ELE4029_2014005123$ cminus test.cm ^C
clucle@clucle:~/compiler/2018_ELE4029_2014005123$ ./cminus test.cm

TINY COMPILATION: test.cm
1: /* A program to perform Euclid's
2:    Algorithm to computer gcd */
3:
4: int gcd(int u, int v)
4: reserved word: int
4: ID, name= gcd
4: (
4: reserved word: int
4: ID, name= u
4: ,
4: reserved word: int
4: ID, name= v
4: )
5: {
5: {
6: if (v == 0) return u;
6: reserved word: if
6: (
6: ID, name= v
6: ==
6: NUM, val= 0
```

3 - 3 cminus_flex

Usage : ./cminus_flex (cminus file name)

```
clucle@clucle:~/compiler/2018_ELE4029_2014005123$ ./cminus_flex test.cm
TINY COMPILATION: test.cm
4: reserved word: int
4: ID, name= gcd
4: (
4: reserved word: int
4: ID, name= u
4: ,
4: reserved word: int
4: ID, name= v
4: )
5: {
6: reserved word: if
6: (
6: ID, name= v
6: ==
6: NUM, val= 0
6: )
6: reserved word: return
6: ID, name= u
6: ;
7: reserved word: else
7: reserved word: return
7: ID, name= gcd
7: (
7: ID, name= v
7: ,
7: ID, name= u
7: -
```