

컴파일러 과제 1

2013011800 구장희

(스크린샷을 파일로 첨부하였습니다.)

compilation method and environment

- 환경 : 가상머신 환경(Ubuntu 14.04 LTS)에서 프로젝트 진행하였습니다.
- 컴파일 : default Makefile을 수정하여 컴파일하였습니다.
(스크린샷 첨부)

Explanation about how to implement and how to operate

- 구현

1. C-Scanner using C-code

pdf에 제시된 변경사항을 그대로 반영하였습니다. 변경사항은 다음과 같습니다.

- globals.h의 TokenType
- main.c의 NO_PARSE, EchoSource, TraceScan
- util.c의 printToken()
- scan.c의 stateType , reservedWord , getToken()

pdf에 제시되지 않은 변경사항은 다음과 같습니다.

- scan.c
 - 연산자(= , == , != , < , > , <= , >=)와 주석(/*, */)와
 - getToken() 함수 내에서 첫 토큰을 검사한 뒤 다음 토큰까지 검사하여 위 연산자와 주석을 구분하는 코드를 작성하였습니다.

2. C-Scanner using lex(flex) by Tiny.l modification

1번의 변경사항을 그대로 사용하였고, 변경사항은 다음과 같습니다.

- cminus.l
 - lex/tiny.l 코드를 수정하여 작성함
 - reserved word들을 추가하고
 - 주석(“/*”)의 경우 다음 토큰을 검사하는 부분을 추가하였습니다.

- 실행

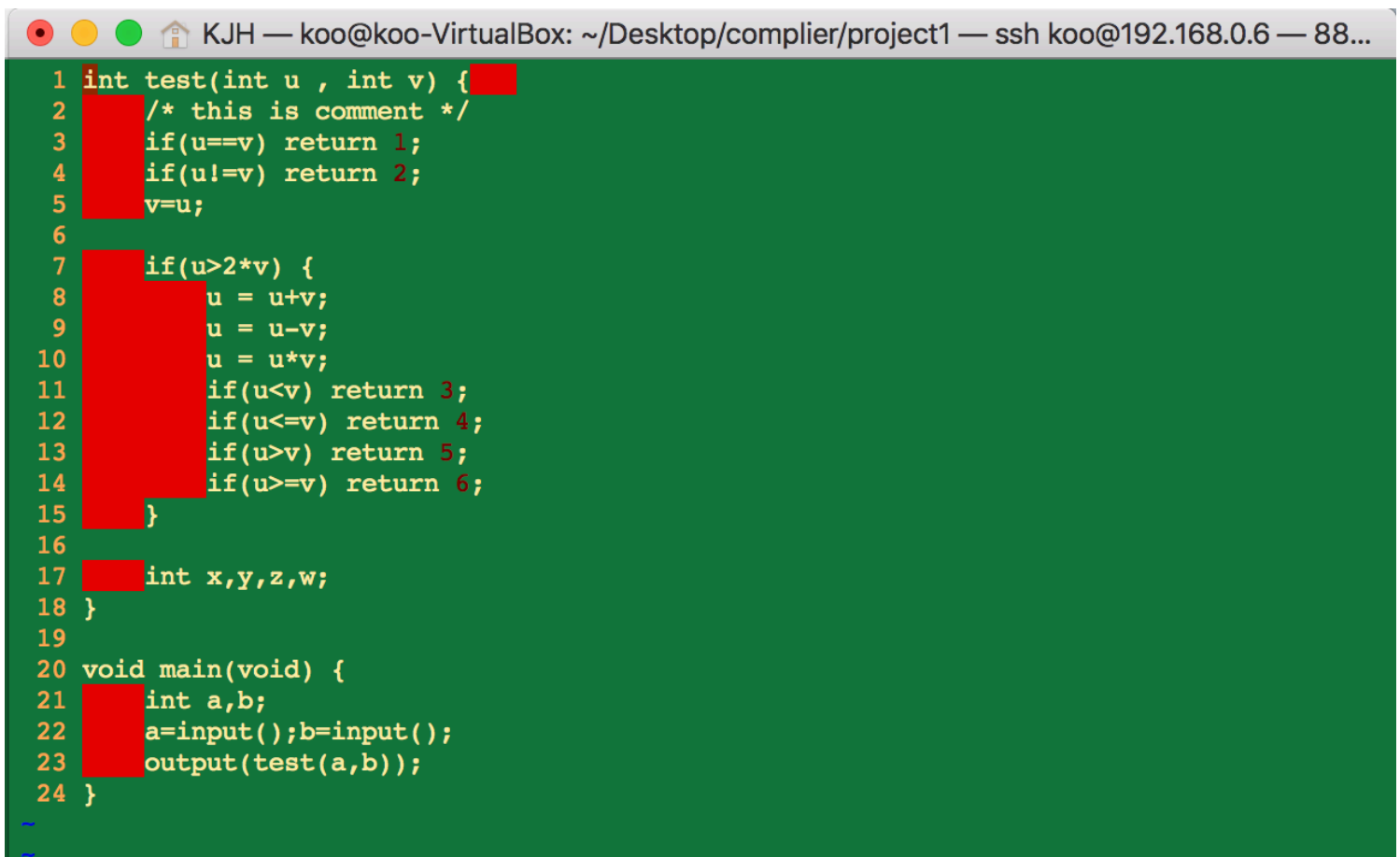
default Makefile에 각각 cminus와 cminus_flex라는 이름으로 실행할 수 있게끔 변경하였습니다.

Example and Result Screenshot

- 테스트 코드

test1 : tiny C compiler에서 기본으로 제공된 test.cm 파일

test2 : 추가 symbol에 대한 코드



```
KJH — koo@koo-VirtualBox: ~/Desktop/complier/project1 — ssh koo@192.168.0.6 — 88...  
1 int test(int u , int v) {  
2     /* this is comment */  
3     if(u==v) return 1;  
4     if(u!=v) return 2;  
5     v=u;  
6  
7     if(u>2*v) {  
8         u = u+v;  
9         u = u-v;  
10        u = u*v;  
11        if(u<v) return 3;  
12        if(u<=v) return 4;  
13        if(u>v) return 5;  
14        if(u>=v) return 6;  
15    }  
16  
17    int x,y,z,w;  
18 }  
19  
20 void main(void) {  
21     int a,b;  
22     a=input();b=input();  
23     output(test(a,b));  
24 }
```

- 컴파일 & 실행결과(using C-code)

```
koo@koo-VirtualBox:~/Desktop/complier/project1$ make
make: `cminus' is up to date.
koo@koo-VirtualBox:~/Desktop/complier/project1$ ./cminus test.cm
```

```
TINY COMPILATION: test.cm
1: int gcd(int u , int v) {
  1: reserved word: int
  1: ID, name= gcd
  1: {
  1: reserved word: int
  1: ID, name= u
  1: ,
  1: reserved word: int
  1: ID, name= v
  1: }
  1: {
2: if(v==0) return u;
2: reserved word: if
2: {
2: ID, name= v
2: ==
2: NUM, val= 0
2: }
2: reserved word: return
2: ID, name= u
2: ;
3: else return gcd(v,u-u/v*v);
3: reserved word: else
3: reserved word: return
3: ID, name= gcd
3: {
3: ID, name= v
3: ,
3: ID, name= u
3: -
3: ID, name= u
3: /
3: ID, name= v
3: *
3: ID, name= v
3: )
3: ;
4: }
4: }
5:
```

```
koo@koo-VirtualBox:~/Desktop/complier/project1$ ./cminus test_symbol.cm
```

```
TINY COMPILATION: test_symbol.cm
1: int test(int u , int v) {
  1: reserved word: int
  1: ID, name= test
  1: {
  1: reserved word: int
  1: ID, name= u
  1: ,
  1: reserved word: int
  1: ID, name= v
  1: }
  1: {
2: /* this is comment */
3: if(u==v) return 1;
3: reserved word: if
3: {
3: ID, name= u
3: ==
3: ID, name= v
3: }
3: reserved word: return
3: NUM, val= 1
3: ;
4: if(u!=v) return 2;
4: reserved word: if
4: {
4: ID, name= u
4: !=
4: ID, name= v
4: }
4: reserved word: return
4: NUM, val= 2
4: ;
5: v=u;
5: ID, name= v
5: =
5: ID, name= u
5: ;
6:
7: if(u>2*v) {
7: reserved word: if
7: {
7: ID, name= u
7: >
7: NUM, val= 2
7: *
7: ID, name= v
7: )
7: {
8: u = u+v;
8: ID, name= u
8: =
8: ID, name= u
8: +
8: ID, name= v
8: ;
9: u = u-v;
9: ID, name= u
9: =
9: ID, name= u
```

- 컴파일 & 실행결과(using flex)

```
koo@koo-VirtualBox:~/Desktop/complier/project1$ clear

koo@koo-VirtualBox:~/Desktop/complier/project1$ make cminus_flex
flex cminus.l
gcc -Wall -g -c lex.yy.c -lfl
lex.yy.c:1326:17: warning: 'yyunput' defined but not used [-Wunused-function]
    static void yyunput (int c, register char * yy_bp )
                  ^
gcc -Wall -g main.o util.o parse.o syntab.o analyze.o code.o cgen.o lex.yy.o -o cminus_flex -lfl
koo@koo-VirtualBox:~/Desktop/complier/project1$ ./cminus_flex test.cm

TINY COMPILATION: test.cm
1: reserved word: int
1: ID, name= gcd
1: (
1: reserved word: int
1: ID, name= u
1: ,
1: reserved word: int
1: ID, name= v
1: )
1: [
2: reserved word: if
2: (
2: ID, name= v
2: ==
2: NUM, val= 0
2: )
2: reserved word: return
2: ID, name= u
2: ;
3: reserved word: else
3: reserved word: return
3: ID, name= gcd
3: (
3: ID, name= v
3: ,
3: ID, name= u
3: -
3: ID, name= u
3: /
3: ID, name= v
3: *
3: ID, name= v
3: )
3: ;
4: ]
6: reserved word: void
6: ID, name= main
6: (
6: reserved word: void
6: )
6: [
7: reserved word: int
7: ID, name= x
7: ,
7: ID, name= y
7: ;
8: ID, name= x
8: =
8: ID, name= input
8: (
```

```
[koo@koo-VirtualBox:~/Desktop/complier/project1$ ./cminus_flex test_symbol.cm
```

```
TINY COMPILATION: test_symbol.cm
```

```
1: reserved word: int
1: ID, name= test
1: (
1: reserved word: int
1: ID, name= u
1: ,
1: reserved word: int
1: ID, name= v
1: )
1: [
3: reserved word: if
3: (
3: ID, name= u
3: ==
3: ID, name= v
3: )
3: reserved word: return
3: NUM, val= 1
3: ;
4: reserved word: if
4: (
4: ID, name= u
4: !=
4: ID, name= v
4: )
4: reserved word: return
4: NUM, val= 2
4: ;
5: ID, name= v
5: =
5: ID, name= u
5: ;
7: reserved word: if
7: (
7: ID, name= u
7: >
7: NUM, val= 2
7: *
7: ID, name= v
7: )
7: [
8: ID, name= u
8: =
8: ID, name= u
8: +
8: ID, name= v
8: ;
9: ID, name= u
9: =
9: ID, name= u
9: -
9: ID, name= v
9: ;
10: ID, name= u
10: =
10: ID, name= u
10: *
10: ID, name= v
10: ;
```