

[Compiler design Report]

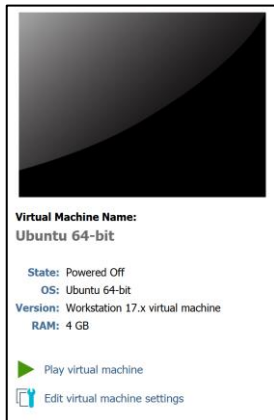
한양대학교 컴퓨터소프트웨어학부

2021088304 박현준

1. 과제 설명

- C-Minus Scanner를 C_minus와 Lex 두 가지 방법을 이용하여 구현하는 과제
- 6개의 예약어와 19개의 token, 공백과 comment 등을 Scanner가 구분할 수 있도록 구현해야 한다.
- Scanner가 소스코드에서 읽어 들인 token들의 시작점과 끝점을 파악할 수 있도록 구현해야 한다.

2. 개발 환경



: Ubuntu 64-bit – Workstation 17.x virtual machine

3. 코드 설명 (수정 사항 위주)

1) main.c

: 과제 명세에 맞게 NO_PARSE, TraceScan을 True로 두고 EchoSource를 False로 설정했습니다.

2) globals.h

: 예약어가 IF, ELSE, WHILE, RETURN, INT, VOID 6개 이므로 MAXRESERVED를 6으로 설정하고, 예약어와 관련된 코드 (enum 내부, switch문 등)를 전부 바꿔주었습니다. 또한 Symbol들도 C-minus에서 사용할 token들만 남기고 나머지는 삭제하거나 수정해 주었습니다.

3) utils.c

: utils.c에서도 header 파일에서 설정해준 조건에 맞게 PrintToken 함수 내에 있는 switch문의 case들을 수정하여 주었습니다.

4) scan.c

▶ ungetNextChar 함수

- ungetNextChar 함수의 경우 다음에 읽어들이 token을 마치 읽지 않은 것처럼 처리를 하는 함수입니다. 예를 들어 소스코드에서 >를 읽었을 때, INGT 상태에서 =를 읽어서 이게 GT인지 GE인지 구분하는 시점에서 =를 읽지 않은 것처럼 처리를 하여 Scanner가 >만 읽은 것으로 처리하여 이게 >인지 >=인지 구분하기 위하여 존재하는 함수입니다.

▶ getToken함수

: getToken 함수는 Scanner가 token을 읽어서 state를 구분하기 위한 함수 입니다.

- START case에서 =,== & <,<= & >,>= 등과 같이 Scanner가 처음에 읽은 token에 의해 ambiguous state가 발생할 수 있기 때문에 이러한 문제를 해결하기 위해 Scanner가 처음으로 읽은 token에 대응하는 state를 미리 바꿔서 (INEQ, INLT, INGT, INNE) 한 번 더 읽었을 때의 token에 의해 state가 구분될 수 있게 해줍니다.

INEQ

: 처음에 =를 읽으면 INEQ 상태로 변경한 뒤, 한 번 더 =를 읽으면 비교 연산자 (==) 처리를 하고, 그렇지 않으면 대입 연산자 (=) 처리를 합니다.

INNE

: 처음에 !을 읽으면 INNE로 상태를 변경한 뒤, ! 다음에 =를 읽는다면 NE로 바꾸고, 그렇지 않으면 ERROR로 상태를 변경합니다.

INLT

: 처음에 <를 읽으면 INLT로 상태를 변경한 뒤, < 다음에 =를 읽었다면 LE로 바꾸고, 그렇지 않으면 LT로 상태를 변경합니다.

INGT

: 처음에 >를 읽으면 INLT로 상태를 변경한 뒤, > 다음에 =를 읽었다면 GE로 바꾸고, 그렇지 않으면 GT로 상태를 변경합니다.

INCOMMENT & INCOMMENT

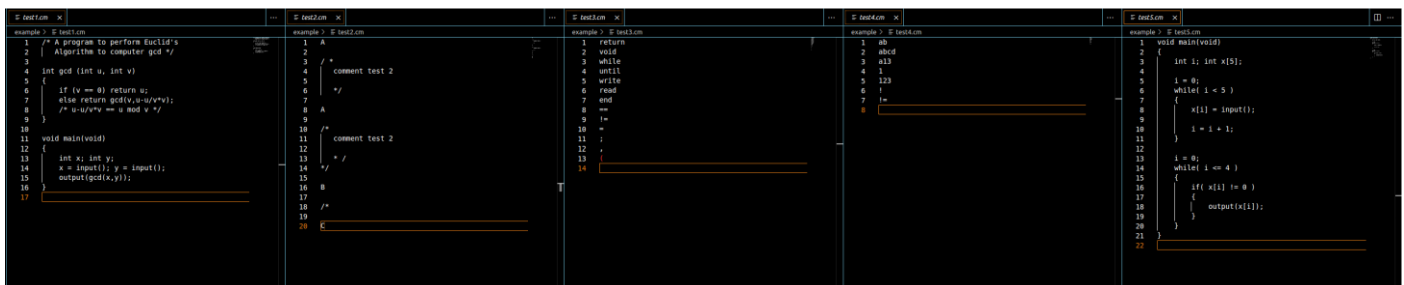
: 처음에 /를 읽었을 때 / (INOVER)과 /* (주석)을 구분하기 위해 존재하는 case 입니다.

5) cminus.l

: cminus.l 파일도 rule 영역에서 사용할 token 들만 적어 놓고 나머지는 수정, 삭제 하였습니다. 또한 definition 부분의 identifier를 {letter} 에서 {letter}({digit}){letter}*로 고쳐서 문자 뒤에 숫자나 문자가 연속해서 왔을 때 읽어들이 수 있도록 하였습니다. 이로 인해 cminus_lex 코드가 test4.cm 에서 작동을 하지 않던 것을 해결할 수 있었습니다.

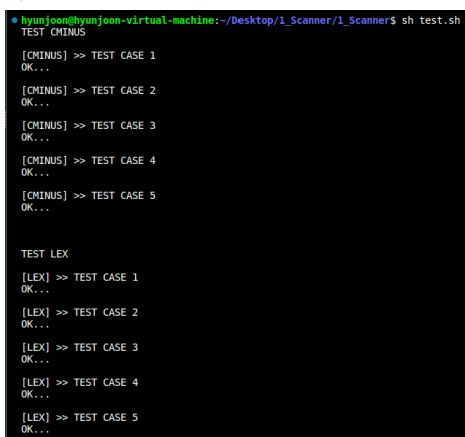
4. 결과

1) test cases



→ 왼쪽부터 test1 ~ test5

2) test.sh 파일을 실행시켰을 때



→ 5가지 test case를 모두 통과한 모습

3) test case들을 컴파일 했을 때의 결과

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
# hyunjong@hyunjong-virtual-machine:~/Desktop/1_Scanner/1_Scanners/.cmimus_cimpl ./example/test1.cm
C-MINUS COMPILATION: ./example/test1.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: ;
7: ID, name= u
7: /
7: ID, name= v
7: *
7: ID, name= v
7: }
7: ;
9: ;
11: reserved word: void
11: ID, name= main
11: {
11: reserved word: void
11: )
12: {
13: reserved word: int
13: ID, name= x
13: ;
13: reserved word: int
13: ID, name= y
13: ;
14: ID, name= x
14: =
14: ID, name= input
14: {
14: ;
14: ;
14: ID, name= y
14: =
14: ID, name= input
14: {
14: ;
14: ;
14: ID, name= output
15: {
15: ID, name= gcd
15: {
15: ID, name= x
15: ;
15: ID, name= y
15: ;
15: ;
15: ;
15: ;
16: ;
17: EOF

# hyunjong@hyunjong-virtual-machine:~/Desktop/1_Scanner/1_Scanners/.cmimus_lex ./example/test1.cm
C-MINUS COMPILATION: ./example/test1.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: ;
7: ID, name= u
7: /
7: ID, name= v
7: *
7: ID, name= v
7: }
7: ;
9: ;
11: reserved word: void
11: ID, name= main
11: {
11: reserved word: void
11: )
12: {
13: reserved word: int
13: ID, name= x
13: ;
13: reserved word: int
13: ID, name= y
13: ;
14: ID, name= x
14: =
14: ID, name= input
14: {
14: ;
14: ;
14: ID, name= y
14: =
14: ID, name= input
14: {
14: ;
14: ;
14: ID, name= output
15: {
15: ID, name= gcd
15: ID, name= x
15: ;
15: ID, name= y
15: ;
15: ;
15: ;
16: ;
17: EOF

```

→ test1 입니다. (왼쪽 : cimpl, 오른쪽 : lex)

```

hynjose@hynjose-virtual-machine:~/Desktop/1_Scanner/1_Scanner$ ./cinmus_cimpl ./example/testt2.cn
C-MINUS COMPILATION: ./example/testt2.cn
1: ID, name= A
2: ID, name= B
3: ID, name= comment
4: ID, name= test
5: NPM, val= 2
6: NPM, val= 0
7: reserved word: while
8: ID, name= A
9: ID, name= B
10: ID, name= B
21: EOF

hynjose@hynjose-virtual-machine:~/Desktop/1_Scanner/1_Scanner$ ./cinmus_lex ./example/testt2.cn
C-MINUS COMPILATION: ./example/testt2.cn
1: ID, name= A
2: ID, name= B
3: ID, name= comment
4: ID, name= test
5: NPM, val= 2
6: NPM, val= 0
7: reserved word: while
8: ID, name= A
9: ID, name= B
10: ID, name= B
21: EOF

hynjose@hynjose-virtual-machine:~/Desktop/1_Scanner/1_Scanner$ ./cinmus_cimpl ./example/testt3.cn
C-MINUS COMPILATION: ./example/testt3.cn
1: reserved word: return
2: reserved word: void
3: reserved word: while
4: ID, name= until
5: ID, name= write
6: ID, name= read
7: ID, name= end
8: NPM, val= 0
9: NPM, val= 0
10: NPM, val= 0
11: NPM, val= 0
12: NPM, val= 0
13: NPM, val= 0
14: EOF

hynjose@hynjose-virtual-machine:~/Desktop/1_Scanner/1_Scanner$ ./cinmus_lex ./example/testt3.cn
C-MINUS COMPILATION: ./example/testt3.cn
1: reserved word: return
2: reserved word: void
3: reserved word: while
4: ID, name= until
5: ID, name= write
6: ID, name= read
7: ID, name= end
8: NPM, val= 0
9: NPM, val= 0
10: NPM, val= 0
11: NPM, val= 0
12: NPM, val= 0
13: NPM, val= 0
14: EOF

hynjose@hynjose-virtual-machine:~/Desktop/1_Scanner/1_Scanner$ ./cinmus_cimpl ./example/testt4.cn
C-MINUS COMPILATION: ./example/testt4.cn
1: ID, name= ab
2: ID, name= abcd
3: ID, name= a13
4: NPM, val= 1
5: NPM, val= 123
6: EPMOS: 1
7: EPMOS: 1
8: EOF

hynjose@hynjose-virtual-machine:~/Desktop/1_Scanner/1_Scanner$ ./cinmus_lex ./example/testt4.cn
C-MINUS COMPILATION: ./example/testt4.cn
1: ID, name= ab
2: ID, name= abcd
3: ID, name= a13
4: NPM, val= 1
5: NPM, val= 123
6: EPMOS: 1
7: EPMOS: 1
8: EOF

hynjose@hynjose-virtual-machine:~/Desktop/1_Scanner/1_Scanner$ ./cinmus_cimpl ./example/testt5.cn
C-MINUS COMPILATION: ./example/testt5.cn
1: reserved word: void
2: ID, name= main
3: ID, name= main
4: ID, name= main
5: reserved word: void
6: ID, name= main
7: ID, name= main
8: reserved word: int
9: ID, name= i
10: ID, name= i
11: ID, name= i
12: reserved word: int
13: ID, name= x
14: ID, name= x
15: NPM, val= 5
16: NPM, val= 5
17: ID, name= i
18: ID, name= i
19: NPM, val= 0
20: reserved word: while
21: ID, name= i
22: ID, name= i
23: NPM, val= 5
24: NPM, val= 5
25: ID, name= i
26: NPM, val= 0
27: reserved word: while
28: ID, name= i
29: ID, name= i
30: NPM, val= 5
31: NPM, val= 5
32: ID, name= i
33: ID, name= i
34: ID, name= input
35: ID, name= input
36: ID, name= input
37: ID, name= i
38: ID, name= i
39: ID, name= i
40: NPM, val= 1
41: NPM, val= 1
42: ID, name= i
43: NPM, val= 1
44: NPM, val= 1
45: ID, name= i
46: ID, name= i
47: NPM, val= 0
48: reserved word: while
49: ID, name= i
50: NPM, val= 4
51: NPM, val= 4
52: reserved word: if
53: ID, name= x
54: ID, name= i
55: ID, name= i
56: ID, name= i
57: NPM, val= 0
58: ID, name= output
59: ID, name= x
60: ID, name= i
61: ID, name= i
62: ID, name= i
63: ID, name= i
64: ID, name= i
65: ID, name= i
66: ID, name= i
67: ID, name= i
68: ID, name= i
69: ID, name= i
70: ID, name= i
71: ID, name= i
72: ID, name= i
73: ID, name= i
74: ID, name= i
75: ID, name= i
76: ID, name= i
77: ID, name= i
78: ID, name= i
79: ID, name= i
80: ID, name= i
81: ID, name= i
82: ID, name= i
83: ID, name= i
84: ID, name= i
85: ID, name= i
86: ID, name= i
87: ID, name= i
88: ID, name= i
89: ID, name= i
90: ID, name= i
91: ID, name= i
92: ID, name= i
93: ID, name= i
94: ID, name= i
95: ID, name= i
96: ID, name= i
97: ID, name= i
98: ID, name= i
99: ID, name= i
100: ID, name= i
101: ID, name= i
102: ID, name= i
103: ID, name= i
104: ID, name= i
105: ID, name= i
106: ID, name= i
107: ID, name= i
108: ID, name= i
109: ID, name= i
110: ID, name= i
111: ID, name= i
112: ID, name= i
113: ID, name= i
114: ID, name= i
115: ID, name= i
116: ID, name= i
117: ID, name= i
118: ID, name= i
119: ID, name= i
120: ID, name= i
121: ID, name= i
122: ID, name= i
123: ID, name= i
124: ID, name= i
125: ID, name= i
126: ID, name= i
127: ID, name= i
128: ID, name= i
129: ID, name= i
130: ID, name= i
131: ID, name= i
132: ID, name= i
133: ID, name= i
134: ID, name= i
135: ID, name= i
136: ID, name= i
137: ID, name= i
138: ID, name= i
139: ID, name= i
140: ID, name= i
141: ID, name= i
142: ID, name= i
143: ID, name= i
144: ID, name= i
145: ID, name= i
146: ID, name= i
147: ID, name= i
148: ID, name= i
149: ID, name= i
150: ID, name= i
151: ID, name= i
152: ID, name= i
153: ID, name= i
154: ID, name= i
155: ID, name= i
156: ID, name= i
157: ID, name= i
158: ID, name= i
159: ID, name= i
160: ID, name= i
161: ID, name= i
162: ID, name= i
163: ID, name= i
164: ID, name= i
165: ID, name= i
166: ID, name= i
167: ID, name= i
168: ID, name= i
169: ID, name= i
170: ID, name= i
171: ID, name= i
172: ID, name= i
173: ID, name= i
174: ID, name= i
175: ID, name= i
176: ID, name= i
177: ID, name= i
178: ID, name= i
179: ID, name= i
180: ID, name= i
181: ID, name= i
182: ID, name= i
183: ID, name= i
184: ID, name= i
185: ID, name= i
186: ID, name= i
187: ID, name= i
188: ID, name= i
189: ID, name= i
190: ID, name= i
191: ID, name= i
192: ID, name= i
193: ID, name= i
194: ID, name= i
195: ID, name= i
196: ID, name= i
197: ID, name= i
198: ID, name= i
199: ID, name= i
200: ID, name= i
201: ID, name= i
202: ID, name= i
203: ID, name= i
204: ID, name= i
205: ID, name= i
206: ID, name= i
207: ID, name= i
208: ID, name= i
209: ID, name= i
210: ID, name= i
211: ID, name= i
212: ID, name= i
213: ID, name= i
214: ID, name= i
215: ID, name= i
216: ID, name= i
217: ID, name= i
218: ID, name= i
219: ID, name= i
220: ID, name= i
221: ID, name= i
222: ID, name= i
223: ID, name= i
224: ID, name= i
225: ID, name= i
226: ID, name= i
227: ID, name= i
228: ID, name= i
229: ID, name= i
230: ID, name= i
231: ID, name= i
232: ID, name= i
233: ID, name= i
234: ID, name= i
235: ID, name= i
236: ID, name= i
237: ID, name= i
238: ID, name= i
239: ID, name= i
240: ID, name= i
241: ID, name= i
242: ID, name= i
243: ID, name= i
244: ID, name= i
245: ID, name= i
246: ID, name= i
247: ID, name= i
248: ID, name= i
249: ID, name= i
250: ID, name= i
251: ID, name= i
252: ID, name= i
253: ID, name= i
254: ID, name= i
255: ID, name= i
256: ID, name= i
257: ID, name= i
258: ID, name= i
259: ID, name= i
260: ID, name= i
261: ID, name= i
262: ID, name= i
263: ID, name= i
264: ID, name= i
265: ID, name= i
266: ID, name= i
267: ID, name= i
268: ID, name= i
269: ID, name= i
270: ID, name= i
271: ID, name= i
272: ID, name= i
273: ID, name= i
274: ID, name= i
275: ID, name= i
276: ID, name= i
277: ID, name= i
278: ID, name= i
279: ID, name= i
280: ID, name= i
281: ID, name= i
282: ID, name= i
283: ID, name= i
284: ID, name= i
285: ID, name= i
286: ID, name= i
287: ID, name= i
288: ID, name= i
289: ID, name= i
290: ID, name= i
291: ID, name= i
292: ID, name= i
293: ID, name= i
294: ID, name= i
295: ID, name= i
296: ID, name= i
297: ID, name= i
298: ID, name= i
299: ID, name= i
300: ID, name= i
301: ID, name= i
302: ID, name= i
303: ID, name= i
304: ID, name= i
305: ID, name= i
306: ID, name= i
307: ID, name= i
308: ID, name= i
309: ID, name= i
310: ID, name= i
311: ID, name= i
312: ID, name= i
313: ID, name= i
314: ID, name= i
315: ID, name= i
316: ID, name= i
317: ID, name= i
318: ID, name= i
319: ID, name= i
320: ID, name= i
321: ID, name= i
322: ID, name= i
323: ID, name= i
324: ID, name= i
325: ID
```

→ 왼쪽은 test2, test3, test4, 오른쪽 2개는 test5 입니다.