**컴파일러 설계**

**프로젝트 2: 파서**

**2013012278 하경모**

**0. compilation environment**

- Ubuntu 18.04 LTS

- gcc 7.3.0

- report는 Microsoft word로 작성

**1. Parser**

**1.1 목적**

- bison을 이용한 c-minus parser 구현

**1.2 디자인**

- 기존 파일의 parse.c를 이용하지 않고 cminus.y파일과 bison을 이용해 parser 구현

- c-minus 문법에 맞게 cminus.y 파일 등을 작성 혹은 수정

- 그에 따른 Makefile 수정

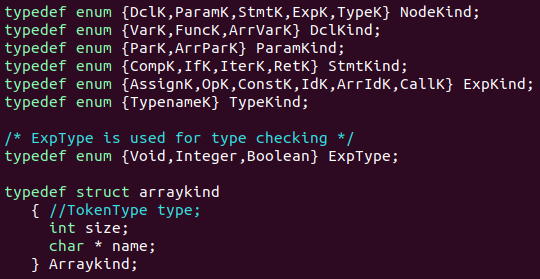
**1.3 구현**

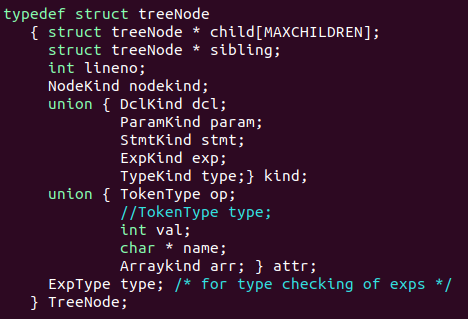
- 기존 globals.h를 bison의 globals.h로 치환 후 수정

- 늘어난 노드의 종류에 맞게 추가로 선언

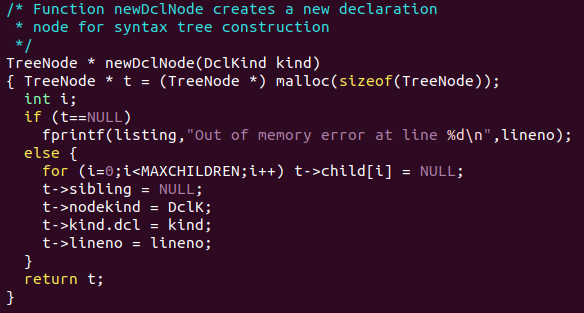
- DclKind, ParamKind, TypeK

- 특히, 배열을 위해 구조체 arraykind를 추가로 선언



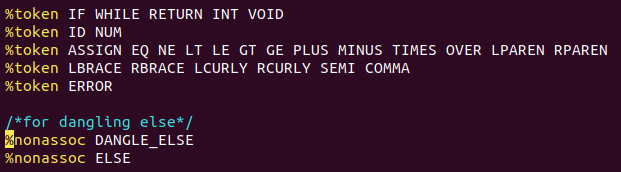
 - 수정된 노드의 종류에 맞게 treeNode 구조체 수정

- 추가된 노드들의 종류에 맞게 util.c에 노드 생성 함수 추가

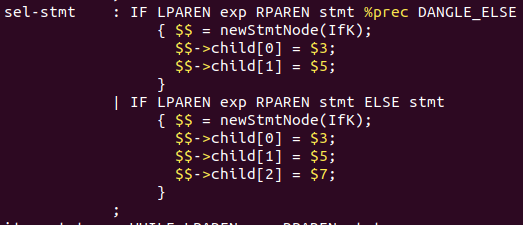
 - newDclNode(), newParamNode(), newTypeNode()

- c-minus의 문법에 따라 cminus.y에 작성

- c-minus에 문법에 쓸 새로운 토큰들 추가

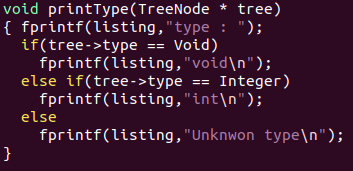


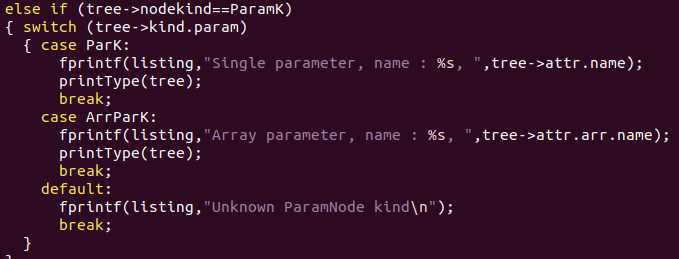
- dangling else 해결하기 위해 DANGLE\_ELSE 추가

 - Appendix에 따라 1 ~ 29까지 작성

- dangling else를 해결하기 위해 selection-statement 약간 수정

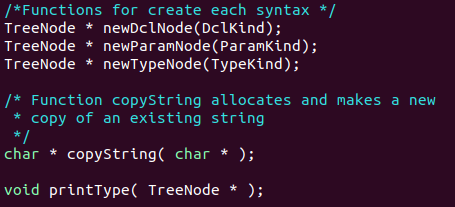
- 노드의 종류가 달라졌으므로 util.c의 printTree() 수정

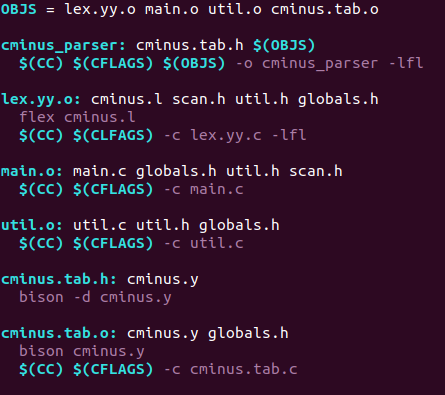
 - type-specifier에서 derive되는 type 터미널들 출력하기 위해 printType() 추가



- DclK, ParamK의 경우 위와 같이 출력, TypeK의 경우 break;

- type node의 type을 부모 노드에 반영해 부모 노드에서 출력하기 때문

 - util.c에 추가했던 함수들을 util.h에 추가

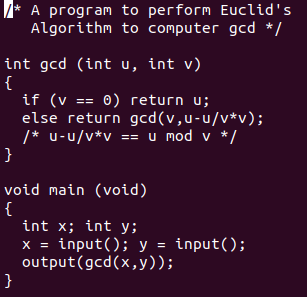
 - Makefile 수정

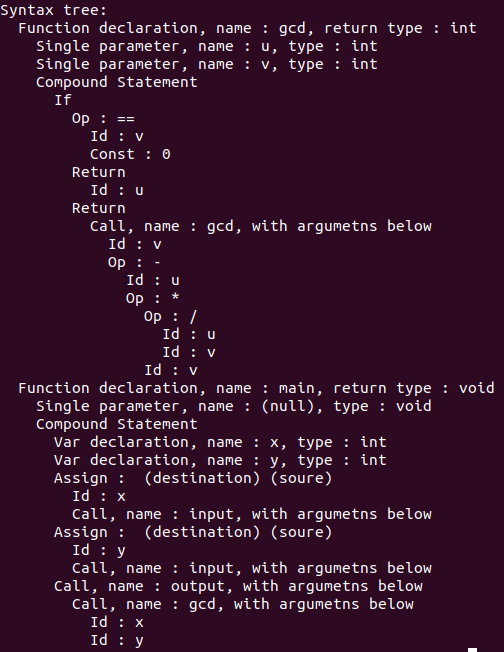
**1.4 결과**

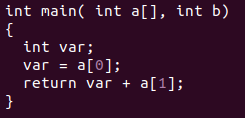
- 실행 방법

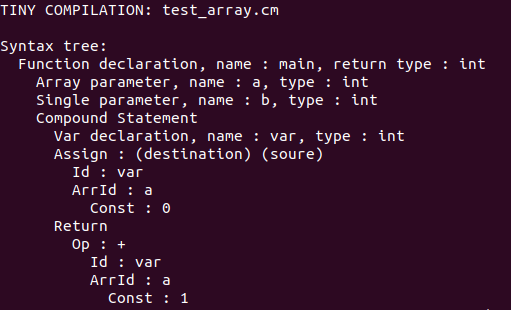
- make => cminus\_parser 실행파일 제작

- ./cminus\_parser test.cm

 - 예제 코드 (test.cm)

 - 실행 결과

 - 예제 코드 (test\_array.cm)

 - 실행 결과