Design and Development of Compiler  
for C- Language

**Phase III**

**Design and Implementation**

**of Semantic Analyzer**

**Project Result**

과목명: CSE4120 기초 컴파일러 구성

담당 교수: 서강대학교 컴퓨터공학과 정성원

개발자: 1조 20141500 권태국, 20141570 이태승, 20141589 최광희(팀장)

제출일: 2017. 6. 02.

1. **개발 목표**

전체 C- Compiler 설계 프로젝트 중 Phase III의 개발 목표는 Semantic Analysis of C- Language로, C- Language로 쓰인 소스 코드의 Semantic Analysis를 할 수 있도록 한다. 구체적으로, Symbol Table을 만들기 위하여 Variable과 Function의 Name equivalence를 이용한 Type checking을 하는 것을 목표로 한다.

1. **개발 범위 및 내용**

**1. 개발 범위**

Phase III는 Phase II에서 구현한 C- parser를 이용하여, semantic analyzer를 구현하는데, 결과적으로 Symbol table을 출력할 수 있어야 한다. 각각의 Table은 개별 Scope를 대표하며, 각 Scope에 있는 Variable과 Parameter, Function에 대한 정보를 출력해주어야 한다. 구체적으로, 각 AST Node의 이름, Scope, Stack pointer로 부터 Location Offset (Loc), Node의 종류(Variable, Parameter, Function), Array인지의 여부, Array인 경우 Size, Type(int, void), Line Numbers(등장하는 코드상의 위치) 등을 출력해야 한다.

**2.개발 내용**

C- 언어는 C 언어의 Subset으로써, 단순한 문법 구조를 갖고 있다. 이번 Phase에서는 먼저, 해당 언어의 분석을 통하여 BNF Grammar을 만족하지만 Semantic하게는 오류가 생기는 코드를 확인할 수 있어야 한다. Phase II에서 개발한 LALR Parser가 산출하는 Abstract Syntax Tree를 가지고 Semantic Analyzer를 만들고, 결과로 Symbol Table을 출력한다. Symbol Table는 각 Symbol의 Scope를 고려할 수 있게 작성한다. 구체적인 출력 내용은 위 1. 개발 범위에서 다룬 바와 같다.

전체 Semantic Analyzer의 개발은 크게 2단계로 수행된다. 첫 번째 단계는 Symbol Table을 작성하고 Scoping error를 Check 하는 단계이고, 두 번째 단계는 Name equivalence를 이용한 Type Check와 그 외 Semantic error를 Check하는 단계이다.

구체적으로, 다음과 같은 오류 체크 사항이 있다.

* 선언되지 않은 변수나 함수는 사용 불가.
* 변수나 함수, 함수 파라미터 선언 시 중복된 이름 검사.
* 변수나 Parameter 선언 시 void type으로 선언할 수 없음.
* 변수나 Parameter 선언 시 Array 변수인지 확인해야 함.
* 변수에 값을 assign 하는 경우 type에 대한 검사.
* Array 변수의 경우 Array index가 int가 맞는지 확인.
* array가 아닌 변수를 array처럼 사용하려 하는 지 검사.
* 함수 호출 시 Parameter의 개수와 type이 일치하는지 검사.
* 함수 호출 시 호출된 것이 함수가 맞는지 (즉 변수 등이 아니었는지) 확인.
* 함수의 return 값과 return type이 일치하는지 검사
* 함수의 return type이 void일 경우 return이 있는지 확인. (없어야 함)
* main함수는 가장 마지막에 선언되어야 함.
* main함수는 반드시 void로 선언되어야 함.
* main함수는 반드시 Parameter가 없어야 함.
* loop문의 반복 조건을 표시하는 부분에 오는 statement는 int값을 가져야 한다.

그 과정에서 필요한 C- 언어로 이뤄진 예제 코드 역시 작성한다. 해당 코드는 에러가 없어야 하며, 존재하는 모든 Keyword가 포함되어야 하고, Nested scope가 3-Level 이상, 각 Scope마다 Local 변수가 선언되어야 한다.

1. **추진 일정 및 개발 방법**

**1. 추진 일정**

5/21 ~ 5/22 : 팀원간 역할 분배 및 이전 Project의 개발 내역을 합성

5/22 ~ 5/25 : Symbol Table 구현

5/26 ~ 5/29 : Name equivalence를 이용한 Type Check와 그 외 Semantic Error의 Check 구현

5/30 ~ 6/1 : 테스트 및 코드 품질 향상.

6/2 : 보고서 작성 및 제출.

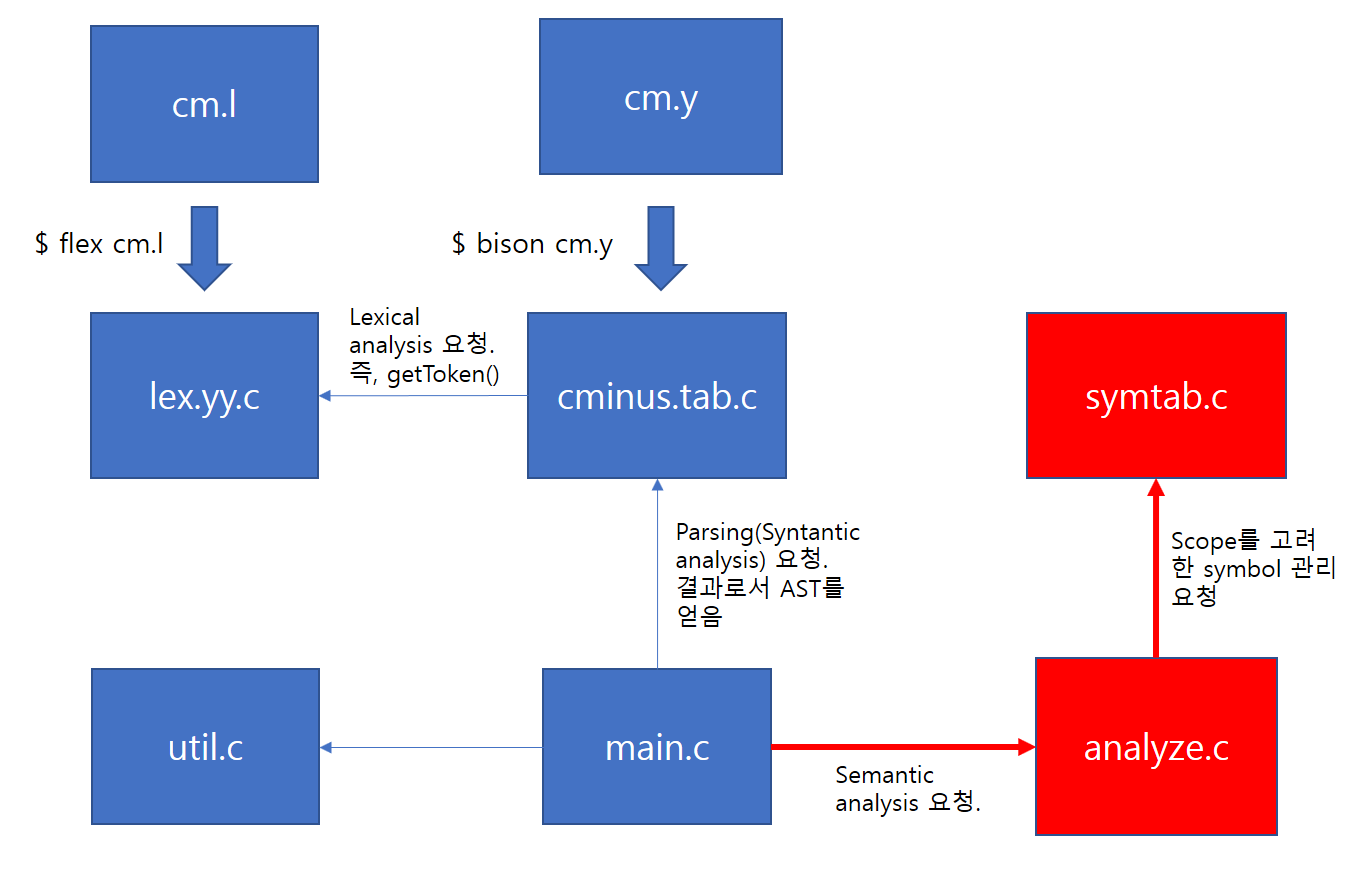
|  |
| --- |
| commit c1a6a27b7289ede1c022aafecc95fff639f4eca7  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Sun May 21 03:49:39 2017 -0700  Structualize folders for project.  commit 9a8d7a2d24a778fb7abde1a1a43b8f1182deed7e  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Sun May 21 03:53:00 2017 -0700  Structualize folders for merging taeguk's repository.  commit 54aa78d8628ad9f0661c850f3256b6fae308da3c  Merge: c1a6a27 9a8d7a2  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Sun May 21 03:54:21 2017 -0700  Merge remote-tracking branch 'taeguk-repo/master'  commit 585292508c739293a213bed771f3084df5b3e758  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Sun May 21 03:58:46 2017 -0700  Structualize folders for merging taeseung's repository  commit f0a4d0afe06d91e80b13d181efdf89cd605b94d8  Merge: 54aa78d 5852925  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Sun May 21 04:01:01 2017 -0700  Merge remote-tracking branch 'taeseung-repo/master'  commit fe06ce8d18c18ce6bfeb0e9bc19a96e728d91342  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Sun May 21 04:07:07 2017 -0700  Add kwanghee's codes for project 1 and 2.  commit f9e25a2c40d5c6fc08160e1fb5e3320a7ad35717  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Sun May 21 04:11:12 2017 -0700  Modify my mistake for merging taeseung's repo.  commit b567eb0a8ae9845668dd80593b6e087bda911e19  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Sun May 21 04:25:17 2017 -0700  Change line endings from CRLF to LF in taeguk and taeseung's project 1,2 codes.  commit c4f977922eb4281b1808a3b0019b55989e1bb4b8  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Sun May 21 06:28:11 2017 -0700  [Project 3] Add given files for symbol table.  commit ed6156f6de8c46010442f00faa33088312cfc187  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Sun May 21 06:55:19 2017 -0700  [Project 3] Implement scope in symbol table (#2). (But, not tested.)  commit 674c79c5946bfac2ee0efe506c827ffa1e52ac46  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Tue May 23 05:57:38 2017 -0700  [Project 3] Change Makefile #3  commit a4f518c78d8826f07b11d51fbf0c59dc293ec0d6  Merge: 674c79c ed6156f  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Tue May 23 06:04:06 2017 -0700  Merge branch 'master' of https://github.com/juice500ml/compiler-sogang  commit 5d2a206962726061ea0802e16f981ea8d754f3a6  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Tue May 23 21:17:31 2017 -0700  [Project 3] Explicitly divide by structure name (Issue #1)  commit 2a440ad4d09f728f13876ac5a88cc469b73eca19  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Wed May 24 00:22:32 2017 -0700  [Project 3] Add analyze files and modify main.c  commit 08e9ac250e4d39d167782de936001ac81f246497  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Wed May 24 01:02:58 2017 -0700  [Project 3] Change options in main.c and Add analyze files  commit 922157d94dcb452c86c52fe08a06980b4385492e  Merge: 2a440ad 08e9ac2  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Wed May 24 01:05:14 2017 -0700  [Project 3] Merge branch 'master' of https://github.com/juice500ml/compiler-sogang  commit e47608c177486bac92d4ce4d000949f0414dd8e2  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Wed May 24 01:06:54 2017 -0700  [Project 3] Partly accept @taeguk 's opinion on short names (#1)    Accepted:  compoundStmt --> cmpdStmt  compareExpr --> cmpExpr    Declined:  Decl --> D (Too many words start with D)  func --> fn ("func" is more widely used than "fn")  Prm --> Param (Parameter does not start with "Pr")  commit 03f6a1a16ef415b7ee9f6517c9d79432fb826d89  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Wed May 24 06:41:51 2017 -0700  [Project 3] Make printSymTab function #2  commit 167439b716c6d751e6197267d5a4fa9ad48ab13d  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Sat May 27 08:24:59 2017 -0700  [Project 3] Add .gitignore.  commit e11199ba596ef97c73ff77ad11822ff0152b2d82  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Sat May 27 08:25:45 2017 -0700  Fix bugs in printSymtab. And modify st\_insert for storing tree\_node.  commit 04809d96d96066653fc25e1c8b6ffeda9ca0650d  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Sat May 27 08:27:08 2017 -0700  [Project 3] Implement buildSymtab.  commit f97f18bc25de5bcb681a45301ed02b85a7703b9c  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Sat May 27 08:27:37 2017 -0700  [Project 3] Modify makefile for project 3.  commit 57f28d1acdd338f739c326404d78890cea879dbd  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Sat May 27 08:38:57 2017 -0700  [Project 1,2] Add testcases for project 1 and 2.  commit 18bc7fb76ecf451535590db741110bc6813218ba  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Sat May 27 08:44:28 2017 -0700  [Project 3] Update .gitignore.  commit 7785e91972f69ddc6210f10f542ca6fd3e6d6b6e  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Tue May 30 00:06:14 2017 -0700  [Project 3] Solves issue #4  commit 7a310575e46f6fd3971ef6694ae82695e7e11819  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Tue May 30 01:06:39 2017 -0700  [Project 3] Fix a bug that symbol's node type is always VariableK.  commit 65b420c8abc73ca7a95236a790d7a7e219462786  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Tue May 30 01:51:49 2017 -0700  [Project 3] Added break for TokenTypeK  commit 08f402b3ec2210968a480052d7ac03eb66d7b838  Merge: 65b420c 7a31057  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Tue May 30 01:52:07 2017 -0700  [Project 3] Merge branch 'master' of https://github.com/juice500ml/compiler-sogang  commit d83a7582a1280e75c51caa045e2433e88d757103  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Tue May 30 01:54:16 2017 -0700  [Project 3] Add additional comments  commit a006d68116351a887768f5976a668dda68408081  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Tue May 30 04:35:10 2017 -0700  [Project 3] Correct the printSymtab format(#9) and Add defaults at cases  commit 356a13089baeacee603c4ffe14a8668df1b7b5a2  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Tue May 30 06:09:04 2017 -0700  [Project 2] Fix the bug about ArrayK  commit 3e648e9fd2538ce3c27154af29f1f49971425d44  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Tue May 30 06:49:40 2017 -0700  [Project 3] Fix the bug about VariableK  When nodeKind is VariableK in insertNode, we should add  a line number to a line number list of table  commit 29e39a90110e7865a66d97a64146a904cba5fe6e  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Tue May 30 10:07:13 2017 -0700  [Project 3] #5 Exception handling about undeclaration and redeclaration  commit ca38032de6ca73956658bbfb8be60d59fb83b803  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Tue May 30 21:15:12 2017 -0700  [Project 3] Add varidic arguemnts for typeError function  commit f3fa9752c18d2725f4cfbae6566d9faaa18efccd  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Tue May 30 21:16:25 2017 -0700  [Project 3] Remove unused variable  commit f0a1ee97668506e0718a1697a7bbb6426db8633d  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Tue May 30 23:47:20 2017 -0700  [Project 3] #5 Add testcases for checking exception handling  commit 38fb3d5245d1eae5962bdf9dd1c7e29084b37092  Merge: f0a1ee9 f3fa975  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Wed May 31 00:01:59 2017 -0700  Merge branch 'master' of https://github.com/juice500ml/compiler-sogang  commit 9abacdf276b06e63c2250cb51e1511acaec004c4  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Wed May 31 01:31:04 2017 -0700  [Project 3] Prototypes for type checking (#5)  commit 2c2817524b909d62c4aa5c1e96c15c8bd6ceb663  Merge: 9abacdf 38fb3d5  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Wed May 31 01:31:44 2017 -0700  [Project 3] Merge branch 'master' of https://github.com/juice500ml/compiler-sogang  commit 231a7a05201baab7db51a7d70e3037711fb54661  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Wed May 31 09:02:53 2017 -0700  [Project 3] Fix a bug about redeclaration.  commit cafcd19e8d734027e37c0acd3e4338f76abf5794  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Wed May 31 09:10:12 2017 -0700  [Project 3] Add multiple scope testcase  commit acd805281ee9f2232563cd9ae47de21194bb5fc2  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Wed May 31 09:15:08 2017 -0700  [Project 3] Fix a bug about st\_insert and printSymtab.  commit d9be7b09aa9f8a2b3c50b08ea363903def518d0b  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Wed May 31 10:01:09 2017 -0700  [Project 3] Draft type checking.  commit 0cc159118d40a40e1d67e4bde0ea31b7f125225f  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Wed May 31 11:18:01 2017 -0700  [Project 3] #14 Make 'make debug' and 'make warn'  make debug: Add -g option at gcc  make warn: Add lots of options at gcc for checking hard convention  make or make all: normally compile codes  commit 1ea02d7e4b8bc5f0e1aa456dcc456d19d0262d3e  Merge: 0cc1591 d9be7b0  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Wed May 31 11:20:12 2017 -0700  Merge branch 'master' of https://github.com/juice500ml/compiler-sogang  commit 63a292b2f04dc1ad04a045e03065d7dca255bd7e  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Wed May 31 18:59:35 2017 -0700  [Project 3] Fix typos  commit 6ed96fb5d2fac0ddd2954812c349a183458c0716  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Wed May 31 19:28:40 2017 -0700  [Project 3] Divide registerSymbol into registerSymbol and referSymbol  commit b8e5e4b0e889ccd96f2191bab35e01238d13c149  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Wed May 31 23:04:51 2017 -0700  [Project 3] Add debug macros and apply macros at defaults  By adding an option -D DEBUG to gcc, LOG\_DEBUG output logs  commit 765acec39dac83d83df2f455917000323e023d5f  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Thu Jun 1 02:48:32 2017 -0700  [Project 3] Add testcases that have various case  commit 1383ce61608d17d08b3792d7ed1d2395c4bdd64d  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Thu Jun 1 03:33:37 2017 -0700  [Project 3] Defines SymbolInfo value not to reuse symbol table  commit fc4db6fa929fd1f4bcef48cf2ceb656820c4a209  Merge: 1383ce6 765acec  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Thu Jun 1 03:34:02 2017 -0700  [Project 3] Merge branch 'master' of https://github.com/juice500ml/compiler-sogang  commit 6f039c0f34ac645005d5ba234434550ca0ab1534  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Thu Jun 1 05:06:08 2017 -0700  [Project 3] Make tokenToNodeType and setSymbolInfo functions  NodeType tokenToNodeType(TokenType): convert tokentype to Nodetype.  SymbolInfo\* setSymbolInfo(TreeNode \*): set a symbolinfo variable  and return one.  commit e3b87ca4f140a6cb4d083aa325aacf33e1b14a68  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Thu Jun 1 06:42:46 2017 -0700  [Project 3] Rename testfiles in project 3  commit 1aad6a1701ebbb54e471073f0582171581e2a4ab  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Thu Jun 1 07:00:52 2017 -0700  [Project 3] Draft for issue #5  commit 3bf13a2e5ea6a060e4ae14295535d4f33d6f5085  Merge: 1aad6a1 e3b87ca  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Thu Jun 1 07:01:45 2017 -0700  Merge branch 'master' of https://github.com/juice500ml/compiler-sogang  commit e9a29fe87b1e25514731890443efbb487e33604e  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Thu Jun 1 07:06:24 2017 -0700  Caught null pointer exception bug in setSymbolInfo  commit 024fe4ee920172e96ce3aa25ccbb25d8aa2b5b40  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Thu Jun 1 07:19:30 2017 -0700  Add error strings for printError  commit eee06d01f73028a97816d190c2d68649600e4b17  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Thu Jun 1 07:35:27 2017 -0700  Remove underbar in variable at cminus testcase05  commit 212edef474e1aaf3de04590e31ba391c091400f9  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Thu Jun 1 07:43:56 2017 -0700  Add new line at the end of DONT\_OCCUR\_PRINT  commit 1cfeb9281fa7ed0679aa91ae8de521017b093d8f  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Thu Jun 1 07:46:44 2017 -0700  Caught bug at compoundStatement recursion for typeCheck  commit 2cb951935b4db618d9e19edd54006a4a47c1486d  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Thu Jun 1 08:01:49 2017 -0700  Remove DONT\_OCCUR\_PRINT on symbolInfo==NULL case  commit 40c2ae71d943779621a211881255ecf31e31aa63  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Thu Jun 1 08:06:23 2017 -0700  Fix bug for return statement type checking  commit 9be5e9fcc37067a76e683053659f2942f30580b9  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Thu Jun 1 08:22:57 2017 -0700  Caught bug at cmpdStmt type checking  commit 92f2a0b42203e841c7d825a59a0f203c9d19d6ee  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Thu Jun 1 09:31:35 2017 -0700  Fix the bug about ReturnStatementK  commit 4dafa9bf350cdff0a5083dc3b574c553995f352e  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Thu Jun 1 19:53:57 2017 -0700  Fix a bug about st\_pop\_scope.  commit 4f0c89d86879cb5601fd0177f294dcea03fe992e  Merge: 9be5e9f 4dafa9b  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Thu Jun 1 20:02:31 2017 -0700  Merge branch 'master' of https://github.com/juice500ml/compiler-sogang  commit ae8dcd485dcf452b34946e8958cdf071c0de19f0  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Thu Jun 1 20:13:44 2017 -0700  Fix symbolInfo default value bug  commit e7990fa86606f0bfa156526f6295271075785864  Author: Kwanghee Choi <juice500ml@gmail.com>  Date: Thu Jun 1 20:21:27 2017 -0700  Print error for array errors, Moved info null check after type checking at CallK  commit 58d8e1edb7f4b222d70ff3dac756875579079e90  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Thu Jun 1 20:42:15 2017 -0700  [Project 3] Fix bugs for testcase 5.  commit 50b613271cb8efe43f7a5ec737a80907d41e46f7  Merge: 58d8e1e e7990fa  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Thu Jun 1 20:42:57 2017 -0700  [Project 3] Resolve confict.  commit 1e3881abc79e424bec4071b5329da7663c26389b  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Thu Jun 1 21:22:30 2017 -0700  [Project 3] Fix a bug.  commit 3fbd0965d4164fc83c2987aab070c1d51b0d2ceb  Author: juice500ml <juice500ml@gmail.com>  Date: Fri Jun 2 14:07:33 2017 +0900  Fix for testcase06  commit 26541533c924d57fe2059f5379a9073468301fb7  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Thu Jun 1 22:14:19 2017 -0700  Remove comments in testcases  commit b1af4831c4183f74786cdb1c5cb0a70d66fb79dc  Author: juice500ml <juice500ml@gmail.com>  Date: Fri Jun 2 14:25:18 2017 +0900  Check main function grammar  commit ffdda70dd10984fe4ea0f375b3f31dc99dd7222b  Merge: b1af483 2654153  Author: juice500ml <juice500ml@gmail.com>  Date: Fri Jun 2 14:26:10 2017 +0900  Merge branch 'master' of https://github.com/juice500ml/compiler-sogang  commit 29954c33ed3be9568988b099f48ae9b090143ceb  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Thu Jun 1 22:28:36 2017 -0700  Fix the lexical bug of testcases  commit 9f04cfaeee81e596a6e1b88139734c6048d44c9f  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Thu Jun 1 22:59:22 2017 -0700  Change len to params but this has a bug  commit 337c2895b8a7238251fcf252bc00fbcb02828ede  Author: Taeseung Lee <taeseunglee159@gmail.com>  Date: Thu Jun 1 23:37:41 2017 -0700  Add a testcases 04.array\_param.c  commit 46f95ae03cbe424777be1213afd34e2317103186  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Fri Jun 2 00:45:59 2017 -0700  [Project 3] Fix a bug in testcase 2.  commit 1e72694133f756835aa00a1d00cc064d6f5e9ea2  Merge: 46f95ae 337c289  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Fri Jun 2 00:46:22 2017 -0700  Merge branch 'master' of https://github.com/juice500ml/compiler-sogang  commit b4e005f0f5ef75af1c39192ecd440f4caa55f868  Author: Taeguk Kwon <xornrbboy@gmail.com>  Date: Fri Jun 2 01:30:05 2017 -0700  [Project 3] Fix a bug in boss testcase. |

**2. 개발 방법**

개발 환경은 cspro9와 cspro10을 사용하여, Ubuntu 16.04.2 LTS 기준으로 작업한다. vi editor을 이용하여 편집을 하며, Indentation은 GNU Standard를 따른다. 주 개발 언어는 C이나, 이번 Phase의 경우 Syntactic Specification 소스 역시 작성한다. 컴파일은 gcc 5.4.0 을 통해 하며, Makefile을 이용하여 빌드 및 Testing의 자동화를 한다. git을 이용하여 Version Control을 하며, Github에서 remote branch를 관리하여, 매 Commit마다 각 case들에 대해 검증한다.

1. **연구 결과**

**1. 합성 내용**



전체 소프트웨어 분석도

**2. 분석 내용**

이번 프로젝트에서는 크게 두 부분으로 나누어 개발했다. 먼저, Symbol Table을 작성하고 Scoping error를 Check하는 단계, 그리고 생성된 Symbol Table을 이용해 Type Check와 semantic error Check하는 단계로 나누어 개발했다. 또한, Type Check의 Symantic analyzer를 구현할 때, “name equivalence”를 사용했다.

개념적으로 symtab.c와 analyze.c를 어떻게 구현했는지 살펴본다면 다음과 같다.

Symbol table을 작성하는 buildSymtab()을 채우는 것이 이번 프로젝트의 세부 목표이다. 먼저, buildSymtab은 insertNode(), printSymTab(), typeCheck(), mainCheck()의 과정을 거친다. insertNode(), typeCheck(), mainCheck()는 analyze.c에, printSymTab은 symtab.c에 존재한다.

* insertNode()의 역할은 syntaxTree을 재귀적으로 탐색하면서, Symbol Table을 작성한다. 이때, 중복 선언을 한 variable/parameter/function(이하 V/P/F)는 registerSymbol()에서 에러를 찾으며, 에러임을 확인 한 경우에는 printError()함수를 통해 에러가 있음을 출력한다. 또한, 선언하지 않은 V/P/F에 대해서는 referSymbol()에서 printError()함수를 통해 선언하지 않은 V/P/F를 사용했음을 알려준다. 또한, 이 insertNode()에서 Function이 선언되었거나, Compound statement를 만났을 때 st\_push\_scope()를 통해 scope를 증가시켜주고, Compound statement에서 부른 함수들이 모두 끝났을 때 st\_pop\_scope()를 호출해 scope를 감소시켜준다. 단, 여기서 Function 자체에도 Compound Statement도 존재하기 때문에, flag를 이용해서 문제를 해결한다. Compound Statement가 function에 의한 것인지 아닌지를 flag에 담아, function에 의해 만들어진 Compound Statement는 st\_push\_scope()를 실행하지 않는다.
* printSymtab()의 역할을 symbol table을 출력해주는 역할이다. Hash table에서 현재 scope에 해당하는 TreeNode\*에 있는 정보를 보면서 내용을 출력해준다.
* typeCheck()의 역할은 생성된 syntaxTree를 재귀적으로 탐색하며 Type Check한다. 변수, 배열 선언과 변수, 배열 paramter에 대해서, 이들의 type이 모두 int이어야한다. 그래서 모두 int인지 아닌지 확인하고, 아닌 경우에는 printError()를 통해 에러 메세지를 출력해준다.
* mainCheck()는 메인 함수의 제한사항들을 확인해주기 위해 존재하는 함수이다. main함수가 가장 마지막에 선언 되었는지, Return Type은 void인지, parameter가 존재하는지를 확인한다. 그리고 이 제한 사항들에 어긋났을 경우에 에러메세지를 출력한다.

**3. 제작 내용**

이번 프로젝트에서 크게 잡은 주제들을 중심으로 제작 내용을 소개하도록 하겠다.

* 기존의 프로젝트에서 수정한 내용

코드 베이스는 최광희의 코드로 진행하였다. 해당 코드에서는 각 NodeKind마다 해당 node를 생성해주는 함수가 쪼개져있었으나, Struct의 경우 Union-struct으로 섞여있던 상태였다. 해당 코드를, C++에서의 상속과 인터페이스 개념을 차용하여, attr라는 union을 재설계하여, 해당 attr에서 각 NodeKind의 특성에 맞는 값들을 접근할 수 있도록 설계하였다. 예를 들면, TreeNode \*t에 대해서, t->attr.iterStmt.loop\_stmt와 같이, 해당 struct의 값을 접근할 때 명시적으로 할 수 있도록 하였다. 해당 struct 구조가 바뀌었으므로, 그에 맞게 기존에 각 node를 allocate해주는 util.c의 각 함수들도 수정했다. global.h에 해당 sturct 구조를 고침에 더불어, SymbolInfo라는 pointer을 추가했는데, 이는 밑에서 설명하도록 하겠다.

* Symbol Table API

기존 TINY Compiler은 scope의 개념이 없고, 모두 전역이기 때문에 Symbol table에 scope 정보를 추가할 필요가 없었으나, C minus의 경우 존재하기 때문에 해당 api를 설계하였다. 먼저, st\_push\_scope와 st\_pop\_scope를 설계했는데, 이는 symtab.c에 존재하는 static int cur\_scope\_level 변수를 통하여 scope를 통제하는 api이다. 해당 api st\_push\_scope를 할 경우 valid한 hash array base를 array count로 업데이트해준다. 또한, st\_pop\_scope를 할 경우 기존의 가장 top에 있는 bucket을 삭제해준다.

또한, st\_insert와 st\_lookup의 경우 기존 TINY 코드 베이스를 크게 수정했는데, 이전에 언급한 cur\_scope\_level-dependent하도록 설게하였다. 또한, SymbolInfo 역시 해당 table record에 들어가게 되는데, 이에 대해서 바로 다음에 설명하도록 하겠다. st\_lookup의 경우, variable 이름을 통하여 접근을 할 수 있는데, is\_cur\_scope 포인터를 통하여 현재 스코프에 존재하는 variable인지 return할 수 있도록 했다.

printSymtab의 경우, 위 함수들을 구현하면서 hash table에 대한 설계가 끝났기 때문에 다소 쉽게 해결할 수 있었는데, 주어진 명세서의 포맷에 맞추어 각 node의 nodekind마다의 정보를 갖고 와서 출력을 해준다.

* analyze.c에서의 Symbol access API

위의 symbol table을 access할 때, 추가적인 code dependency를 막기 위하여 단일 api로 접근을 허용했는데, 그 두 함수가 registerSymbol과 referSymbol이다. 두 함수는 각각 symbol을 hash table에 새로 register할 때와, 기존에 hash table에 있는 symbol에 추가적으로 reference해주기 위한 함수로, 여기에서도 Redeclaration과 Undeclared symbol에 대한 예외 처리를 해 주었다.

* Semantic 정보를 저장한 tree의 생성

Tree의 경우 insertNode 함수를 재귀적으로 부르면서 생성해주었다. 이 함수에서는, symbol의 scope-dependent한 코드가 등장하는 Variable, Array, Function declaration과 Variable, Array parameter을 제외하고는 모두 해당 타입을 저장하고 재귀적으로 하위 노드들을 부르는 형태로 설계되어 있다. 해당 scope-dependent한 node들의 경우 위에서 소개한 symbol access api와 symbol table api의 st\_push\_scope와 st\_pop\_scope를 통해 접근하여, scope에 dependent한 해당 symbol의 정보를 찾아 생성하여 hash table에 생성해주었다. symbol의 정보는 setSymbolInfo라는 함수를 통해 추상화시켰는데, 해당 함수에서는 위 다섯가지 node kind에 대해 여러 예외 케이스를 검사하고, 실제 해당 symbol에 추가적으로 필요한 정보를 node로부터 extract하여 저장한다. 예를 들면, function의 경우 parameter의 갯수와 타입, return 타입이 있어야 하는데, 이러한 정보를 위에서 잠시 언급한 SymbolInfo라는 struct 안에 hiding시켜 node에 다시 저장한다.

* Type checking

각 타입마다 check해주어야 하는 범위가 다른데, 이를 해결하기 위하여 typeCheck 함수를 위에서 생성한 tree에서 재귀적으로 방문하여 해결했다. 그러므로, 먼저 leaf node부터 시작하여 차근차근 type을 determine해서 점차적으로 위로 올라오는 bottom-up 방식으로 진행했다. 이렇게 개발을 한 결과, 정확히 어느 line에서 에러가 나는지 명확히 보여줄 수 있었다. 자명한 예시들을 제외하고 다루기 까다로운 NodeKind들에 대해 설명하면, Declaration 계열의 경우 cminus 문법에 맞는지, 예를 들면 변수 생성시 void 불가능 등에 대해 체크를 했다. 그런데 여기서 Function의 declaration의 경우 좀 까다로웠는데, return type을 체킹해주기 위해서이다. 이를 위하여 재귀 호출을 하기 직전에 자식 node에 retType 변수를 통해 return해야 하는 type을 injection해주어, 하위 노드인 compound statement에서 재귀적으로 다시 불릴 수 있게 하였고, return statement에서 위로부터 inject받은 type과 현재 return statement가 갖고 있는 expression의 type을 비교하여 에러 체킹을 하였다. Call의 경우, parameter checking이 필수적인데, 이를 구현하기 위하여 이전에 semantic 정보를 저장한 symbolInfo를 이용하여 해당 node에 저장되어있는 정보를 통해 checking을 하였다.

* Main function check

Cminus의 경우, procedure의 entry point가 main 함수이기 때문에 strict한 문법을 적용시킨다. 이를 체크하기 위하여 type check를 끝낸 뒤에 별도의 procedure을 분리시켰다. 실제 구현은 다소 trivial한 편이나, 내용을 살펴보자면 가장 마지막에 선언되었는지 확인하기 위해 declaration list를 traverse했고, 해당 type specification을 확인하여 void인지 점검했고, parameter이 없는지 확인했다.

* Build symbol table

이로써 buildSymtab의 4개의 component를 설명하였다. 순서대로, insertNode를 통해 tree를 생성한 뒤에, 해당 tree를 생성하면서 각 scope에 대해 printSymTab함수를 부른 뒤에 마지막으로 global scope의 symbol table을 출력하고, 해당 tree에 대해 symbol table의 정보를 이용하여 typeCheck와 mainCheck을 통해 Semantic analysis를 진행하였다.

**4. 시험 내용**

각 오류 체크 사항에 대해서 간단한 테스트 케이스를 만들어 시험 해봤다.

1. 선언되지 않은 변수나 함수는 사용 불가.

|  |
| --- |
| void main (void) {  a = 1;  func(a); } |
| Filename: 01.undecl.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : main  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  Expression Statement  > Expression :  Assignment Expression  > Variable associated to assignment :  Variable Id : a  > Value assigned :  Constant : 1  Expression Statement  > Expression :  Function Call  Variable Id : func  > Function arguments :  Variable Id : a  Building Symbol Table...  Declaration error at line 3: Undeclared symbol "a"  Declaration error at line 4: Undeclared symbol "func"  Declaration error at line 4: Undeclared symbol "a"  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  main 0 0 Func No - void 1  Type error at line 4: 'func' is never declared. |

* 변수나 함수, 함수 파라미터 선언 시 중복된 이름 검사.

|  |
| --- |
| void b(void) {} void func(void) {} void func(int a) {} void main(void) {  int a;  int a;  int b; } |
| Filename: 02.redecl.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : b  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Function Declaration  Token : void  Variable Id : func  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Function Declaration  Token : void  Variable Id : func  > Parameters :  Parameter (Variable)  Token : int  Variable Id : a  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Function Declaration  Token : void  Variable Id : main  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  Variable Declaration  Token : int  Variable Id : a  Variable Declaration  Token : int  Variable Id : a  Variable Declaration  Token : int  Variable Id : b  > Local Statements :  (null)  Building Symbol Table...  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Declaration error at line 3: Redeclaration of symbol "func"  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Par No - int 3  Declaration error at line 7: Redeclaration of symbol "a"  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Var No - int 6  b 1 0 Var No - int 8  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  main 0 0 Func No - void 4  b 0 0 Func No - void 1  func 0 0 Func No - void 2 |

* 변수나 Parameter 선언 시 void type으로 선언할 수 없음.

|  |
| --- |
| void main(void par) {  void var; } |
| Filename: 03.void\_decl.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : main  > Parameters :  Parameter (Variable)  Token : void  Variable Id : par  > Function Block :  Compound Statement  > Local Declarations :  Variable Declaration  Token : void  Variable Id : var  > Local Statements :  (null)  Building Symbol Table...  Type error at line 1: Variable parameter 'par' cannot be void type.  Type error at line 3: Variable 'var' cannot be void type.  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  main 0 0 Func No - void 1  Main error at line 4: Main function cannot have parameter. |

* 변수나 Parameter 선언 시 Array 변수인지 확인해야 함.

|  |
| --- |
| void arrParam(int a, int arr[], int arrTwo[]) { }   void main(void) {  int a; } |
| Filename: 04.array\_param.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : arrParam  > Parameters :  Parameter (Variable)  Token : int  Variable Id : a  Parameter (Array)  Token : int  Variable Id : arr  Parameter (Array)  Token : int  Variable Id : arrTwo  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Function Declaration  Token : void  Variable Id : main  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  Variable Declaration  Token : int  Variable Id : a  > Local Statements :  (null)  Building Symbol Table...  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  arrTwo 1 0 Par Array 0 array 1  a 1 0 Par No - int 1  arr 1 0 Par Array 0 array 1  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Var No - int 8  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  main 0 0 Func No - void 6  arrParam 0 0 Func No - void 1 |

* 변수에 값을 assign 하는 경우 type에 대한 검사.

|  |
| --- |
| void voidFunc(void) {  int a;  a = 1; }  int intFunc(void) {  int a;  return a = 1; }  void main(void) {  int g;  g = intFunc(); /\* pass \*/  g = voidFunc(); /\* fail \*/ } |
| Filename: 05.asssign\_type\_check.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : voidFunc  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  Variable Declaration  Token : int  Variable Id : a  > Local Statements :  Expression Statement  > Expression :  Assignment Expression  > Variable associated to assignment :  Variable Id : a  > Value assigned :  Constant : 1  Function Declaration  Token : int  Variable Id : intFunc  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  Variable Declaration  Token : int  Variable Id : a  > Local Statements :  Return Statement  > Returning expression :  Assignment Expression  > Variable associated to assignment :  Variable Id : a  > Value assigned :  Constant : 1  Function Declaration  Token : void  Variable Id : main  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  Variable Declaration  Token : int  Variable Id : g  > Local Statements :  Expression Statement  > Expression :  Assignment Expression  > Variable associated to assignment :  Variable Id : g  > Value assigned :  Function Call  Variable Id : intFunc  > Function arguments :  (null)  Expression Statement  > Expression :  Assignment Expression  > Variable associated to assignment :  Variable Id : g  > Value assigned :  Function Call  Variable Id : voidFunc  > Function arguments :  (null)  Building Symbol Table...  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Var No - int 3 4  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Var No - int 9 10  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  g 1 0 Var No - int 15 16 17  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  main 0 0 Func No - void 13  intFunc 0 0 Func No - int 7  voidFunc 0 0 Func No - void 1  Type error at line 17: Assignment and assignee type mismatch |

* Array 변수의 경우 Array index가 int가 맞는지 확인.

|  |
| --- |
| void helloworld(int a) {  a; }  void main(void) {  int a;  int arr[2];  a = 1;   arr[a] = 3;  arr[helloworld(a)] = 3; } |
| Filename: 06.array\_idx\_void.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : helloworld  > Parameters :  Parameter (Variable)  Token : int  Variable Id : a  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  Expression Statement  > Expression :  Variable Id : a  Function Declaration  Token : void  Variable Id : main  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  Variable Declaration  Token : int  Variable Id : a  Array Declaration  Token : int  Variable Id : arr  Constant : 2  > Local Statements :  Expression Statement  > Expression :  Assignment Expression  > Variable associated to assignment :  Variable Id : a  > Value assigned :  Constant : 1  Expression Statement  > Expression :  Assignment Expression  > Variable associated to assignment :  Array  Variable Id : arr  > Expression inside subscript [\*]  Variable Id : a  > Value assigned :  Constant : 3  Expression Statement  > Expression :  Assignment Expression  > Variable associated to assignment :  Array  Variable Id : arr  > Expression inside subscript [\*]  Function Call  Variable Id : helloworld  > Function arguments :  Variable Id : a  > Value assigned :  Constant : 3  Building Symbol Table...  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Par No - int 1 3  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Var No - int 8 10 12 13  arr 1 0 Var Array 2 array 9 12 13  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  helloworld 0 0 Func No - void 1  main 0 0 Func No - void 6  Type error at line 13: Array subscript must be type 'int'. |

* array가 아닌 변수를 array처럼 사용하려 하는 지 검사.

|  |
| --- |
| void main(void) {  int a;  a[2]; } |
| Filename: 07.simulate\_array.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : main  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  Variable Declaration  Token : int  Variable Id : a  > Local Statements :  Expression Statement  > Expression :  Array  Variable Id : a  > Expression inside subscript [\*]  Constant : 2  Building Symbol Table...  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Var No - int 3 4  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  main 0 0 Func No - void 1  Type error at line 4: Variable 'a' is not subscriptable. |

* 함수 호출 시 Parameter의 개수와 type이 일치하는지 검사.

|  |
| --- |
| void func(int param, int paramTwoo, int paramThreee[]) { }  void main(void) {  int arr[30];  int arrTwo[20];  int arrThree;   func(1, 2, arr);  func(3, 4, arrTwo);  func(5, 6, arrThree);   func();  func(1,2); } |
| Filename: 08.check\_param.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : func  > Parameters :  Parameter (Variable)  Token : int  Variable Id : param  Parameter (Variable)  Token : int  Variable Id : paramTwoo  Parameter (Array)  Token : int  Variable Id : paramThreee  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Function Declaration  Token : void  Variable Id : main  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  Array Declaration  Token : int  Variable Id : arr  Constant : 30  Array Declaration  Token : int  Variable Id : arrTwo  Constant : 20  Variable Declaration  Token : int  Variable Id : arrThree  > Local Statements :  Expression Statement  > Expression :  Function Call  Variable Id : func  > Function arguments :  Constant : 1  Constant : 2  Variable Id : arr  Expression Statement  > Expression :  Function Call  Variable Id : func  > Function arguments :  Constant : 3  Constant : 4  Variable Id : arrTwo  Expression Statement  > Expression :  Function Call  Variable Id : func  > Function arguments :  Constant : 5  Constant : 6  Variable Id : arrThree  Expression Statement  > Expression :  Function Call  Variable Id : func  > Function arguments :  (null)  Expression Statement  > Expression :  Function Call  Variable Id : func  > Function arguments :  Constant : 1  Constant : 2  Building Symbol Table...  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  param 1 0 Par No - int 1  paramThreee 1 0 Par Array 0 array 1  paramTwoo 1 0 Par No - int 1  Declaration error at line 12: Undeclared symbol "func"  Declaration error at line 13: Undeclared symbol "func"  Declaration error at line 15: Undeclared symbol "func"  Declaration error at line 16: Undeclared symbol "func"  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  arrTwo 1 0 Var Array 20 array 8 12  arrThree 1 0 Var No - int 9 13  arr 1 0 Var Array 30 array 7 11  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  main 0 0 Func No - void 5  func 0 0 Func No - void 1  Type error at line 11: Type mismatch of parameter at 3 while calling 'func'.  Type error at line 12: 'func' is never declared.  Type error at line 13: 'func' is never declared.  Type error at line 15: 'func' is never declared.  Type error at line 16: 'func' is never declared. |

* 함수 호출 시 호출된 것이 함수가 맞는지 (즉 변수 등이 아니었는지) 확인.

|  |
| --- |
| void funcOne(int a) {} int funcTwo(void) {} int funcThree(int aa[]) {}   void main (void) {  int a;  int arr[23];   helloworld(); } |
| Filename: 09.check\_really\_function.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : funcOne  > Parameters :  Parameter (Variable)  Token : int  Variable Id : a  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Function Declaration  Token : int  Variable Id : funcTwo  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Function Declaration  Token : int  Variable Id : funcThree  > Parameters :  Parameter (Array)  Token : int  Variable Id : aa  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Function Declaration  Token : void  Variable Id : main  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  Variable Declaration  Token : int  Variable Id : a  Array Declaration  Token : int  Variable Id : arr  Constant : 23  > Local Statements :  Expression Statement  > Expression :  Function Call  Variable Id : helloworld  > Function arguments :  (null)  Building Symbol Table...  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Par No - int 1  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  aa 1 0 Par Array 0 array 3  Declaration error at line 11: Undeclared symbol "helloworld"  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Var No - int 8  arr 1 0 Var Array 23 array 9  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  funcOne 0 0 Func No - void 1  main 0 0 Func No - void 6  funcThree 0 0 Func No - int 3  funcTwo 0 0 Func No - int 2  Type error at line 11: 'helloworld' is never declared. |

* 함수의 return 값과 return type이 일치하는지 검사

|  |
| --- |
| void funcOne(int a) { return a; } int funcTwo(int b) { } int funcThree(int c) { return ; } int funcFour(int c) { return funcOne(c); }  void main(void) {} |
| Filename: 10.unmatched\_return.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : funcOne  > Parameters :  Parameter (Variable)  Token : int  Variable Id : a  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  Return Statement  > Returning expression :  Variable Id : a  Function Declaration  Token : int  Variable Id : funcTwo  > Parameters :  Parameter (Variable)  Token : int  Variable Id : b  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Function Declaration  Token : int  Variable Id : funcThree  > Parameters :  Parameter (Variable)  Token : int  Variable Id : c  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  Return Statement  > Returning expression :  (null)  Function Declaration  Token : int  Variable Id : funcFour  > Parameters :  Parameter (Variable)  Token : int  Variable Id : c  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  Return Statement  > Returning expression :  Function Call  Variable Id : funcOne  > Function arguments :  Variable Id : c  Function Declaration  Token : void  Variable Id : main  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Building Symbol Table...  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Par No - int 1 1  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  b 1 0 Par No - int 2  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  c 1 0 Par No - int 3  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  c 1 0 Par No - int 4 4  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  funcOne 0 0 Func No - void 1  main 0 0 Func No - void 6  funcFour 0 0 Func No - int 4  funcThree 0 0 Func No - int 3  funcTwo 0 0 Func No - int 2  Type error at line 1: Returning expression must match pre-declared function return type.  Type error at line 3: Returning expression must match pre-declared function return type.  Type error at line 4: Returning expression must match pre-declared function return type. |

* 함수의 return type이 void일 경우 return이 있는지 확인. (없어야 함)

|  |
| --- |
| void func(void) {  int a;  return ;  return a; }  void main(void) {  } |
| Filename: 11.void\_return.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : func  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  Variable Declaration  Token : int  Variable Id : a  > Local Statements :  Return Statement  > Returning expression :  (null)  Return Statement  > Returning expression :  Variable Id : a  Function Declaration  Token : void  Variable Id : main  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Building Symbol Table...  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Var No - int 3 5  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  main 0 0 Func No - void 8  func 0 0 Func No - void 1  Type error at line 4: Returning expression must match pre-declared function return type.  Type error at line 5: Returning expression must match pre-declared function return type. |

* main함수는 가장 마지막에 선언되어야 함.

|  |
| --- |
| void func(void) {}  void main(void) {  int a; }  void funcTwo(void) {} int funcThree(void) {} void funcFour(int a, int arr[]) {} |
| Filename: 12.last\_main.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : func  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Function Declaration  Token : void  Variable Id : main  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  Variable Declaration  Token : int  Variable Id : a  > Local Statements :  (null)  Function Declaration  Token : void  Variable Id : funcTwo  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Function Declaration  Token : int  Variable Id : funcThree  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Function Declaration  Token : void  Variable Id : funcFour  > Parameters :  Parameter (Variable)  Token : int  Variable Id : a  Parameter (Array)  Token : int  Variable Id : arr  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Building Symbol Table...  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Var No - int 5  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  a 1 0 Par No - int 12  arr 1 0 Par Array 0 array 12  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  main 0 0 Func No - void 3  funcFour 0 0 Func No - void 12  funcThree 0 0 Func No - int 10  funcTwo 0 0 Func No - void 8  func 0 0 Func No - void 1  Main error at line 13: Main function must be declared at the very last of program. |

* main함수는 반드시 void로 선언되어야 함.

|  |
| --- |
| int main(void) {  return 0; } |
| Filename: 13.int\_main.c |
| Syntax tree:  Function Declaration  Token : int  Variable Id : main  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  Return Statement  > Returning expression :  Constant : 0  Building Symbol Table...  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  main 0 0 Func No - int 1  Type error at line 4: Main function must be type void |

* main함수는 반드시 Parameter가 없어야 함.

|  |
| --- |
| void main(int argv) { } |
| Filename: 14.param\_main.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : main  > Parameters :  Parameter (Variable)  Token : int  Variable Id : argv  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Building Symbol Table...  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  argv 1 0 Par No - int 1  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  main 0 0 Func No - void 1  Main error at line 3: Main function cannot have parameter. |

* loop문의 반복 조건을 표시하는 부분에 오는 statement는 int값을 가져야 한다.

|  |
| --- |
| void func(void) {  }  void main(void) {  int i;   while (i)  {  i = i + 1;  }   while (func())  {  i = i + 1;  } } |
| Filename: 15.loop\_statement\_int.c |
| Syntax tree:  Function Declaration  Token : void  Variable Id : func  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  (null)  Function Declaration  Token : void  Variable Id : main  > Parameters :  (null)  > Function Block :  Compound Statement  > Local Declarations :  Variable Declaration  Token : int  Variable Id : i  > Local Statements :  Iteration Statement  > Expression inside while(\*) :  Variable Id : i  > Statements inside while clause :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  Expression Statement  > Expression :  Assignment Expression  > Variable associated to assignment :  Variable Id : i  > Value assigned :  Additive Expression  Token : +  > Left expression added / subtracted :  Variable Id : i  > Right expression added / subtracted :  Constant : 1  Iteration Statement  > Expression inside while(\*) :  Function Call  Variable Id : func  > Function arguments :  (null)  > Statements inside while clause :  Compound Statement  > Local Declarations :  (null)  > Local Statements :  Expression Statement  > Expression :  Assignment Expression  > Variable associated to assignment :  Variable Id : i  > Value assigned :  Additive Expression  Token : +  > Left expression added / subtracted :  Variable Id : i  > Right expression added / subtracted :  Constant : 1  Building Symbol Table...  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Declaration error at line 12: Undeclared symbol "i"  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Declaration error at line 17: Undeclared symbol "i"  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  i 1 0 Var No - int 8 10  Name Scope Loc V/P/F Array? ArrSize Type Line Numbers  --------------------------------------------------------------------  main 0 0 Func No - void 6  func 0 0 Func No - void 1  Type error at line 18: Expression inside iteration statement should be type 'int'. |

**5. 평가 내용**

개발시에 프로그램에 존재하는 버그를 제거하기 편하게, 여러 가지 장치를 했다. 첫번째로, makefile에 all 이외에 debug, warn 이라는 target을 추가적으로 구현했다. Make debug의 경우에는 -g 옵션(gdb)와 -D DEBUG(Define DEBUG)를 했다. 이를 통해, make debug만 하면 바로 gdb를 사용할 수 있는 실행 파일이 만들어진다. 또한, -D DEBUG를 이용해서, log\_debugger.h에 있는 LOG\_DEBUG macro function이 정보들을 출력해준다. 정보의 종류에는 LOG\_DEBUG가 출력하고자 하는 메세지, 그리고 메세지에 출력 변수로 들어갈 변수들이 있으며, LOG\_DEBUG가 있는 파일, line number, 함수를 출력하여, 어디서 문제가 발생했는지 쉽게 알 수 있게 한다. -D DEBUG 옵션이 없는 경우에는 LOG\_DEBUG는 아무런 출력을 하지 않는다. 이를 통해, 개발을 편하게 하였다. 그리고 절대로 발생하지 말아야하는 경우(예를 들면, malloc의 할당 실패 또는 switch case에서 처리하지 않아야하는 default case, 잘못된 값이 들어옴을 의미합니다.)에 대해서 DONT\_OCCUR\_PRINT macro를 넣어줬다. 이 매크로는 절대 발생하지 않아야하는 부분이 발생했을 때, -D DEBUG옵션과 같이, 파일, 라인, 함수과 같이 에러 메세지를 출력해준다. 이를 통해, 좀 더 빠르게 버그 위치를 확인할 수 있고 고치는 데 유용하였다. Make warn의 경우에 런타임 에러가 발생할 수 있는 가능성을 줄이기 위해, gcc에 warning 옵션을 추가해준다. 여기에 많은 warning 옵션들이 들어가는데 이를 입력해주는 수고를 덜어주며, 리팩토링을 쉽게 해준다.

또한, 개발시에 Git을 사용하여 Commit을 잘게 쪼개 각 변경마다 어떤 사항이 변경되었는가를 Tracking 하도록 하여, 신뢰성 있는 Code 작성 및 Continuous Integration 뿐만 아니라 과제를 제출한 뒤에 지속적으로 타인들과 함께 Open Source Contribution을 할 수 있도록 했다. 또한 문제가 생겼을 때 Revert나 Cherry-pick 등 여러 Version Control 기능을 사용할 수 있어, 개발을 훨씬 안정적으로 할 수 있었다. 해당 코드를 Github에 올림으로써, Local Repository에서만 관리하는 것이 아닌 Remote Repository를 두어, 한 쪽 Code가 날아가더라도 전혀 지장이 없도록 했다.

1. **기타**
2. **연구 조원 기여도**

20141589 최광희 (33%)

20141570 이태승 (33%)

20141500 권태국 (34%)

1. **자체 평가**

우리 팀이 진행한 프로젝트의 경우, 설계 측면의 경우 기존에 TINY Compiler에서 Symbol Table을 keep하는 것과는 다르게, 고민을 통해 table을 오직 Temporary tool로 사용하여, 실제 Code generation에 훨씬 간편한 Tree 구조로 record를 저장할 수 있는 구조를 사용하였다. 개발 측면의 경우, make에 make debug, make warn 등 여러 버전을 준비하여 개발 중 찾아오는 여러 상황들에 대해 효과적으로 대처할 수 있도록 하였다. 또한, Makefile을 hierarchical 구조로 만들어 추후 확장될 코드들에 대해 합리적으로 빌드할 수 있도록 하였다. Test를 할 때도 주어진 예외 뿐만 아니라 여러 예외들에 대해서 모두 잡을 수 있도록 다소 많은 cminus code를 작성했고,

개발 방법론에 있어서는, 기본적으로 주어진 명세서를 기준으로 Kanban 방식을 도입하고, 크지 않은 팀원 수에 알맞게 Scrum과 Pair-programming을 섞어 사용하였다. Github의 Issue system을 통해 Root issue는 Kanban처럼 관리하고, 해당 issue에 소issue들을 생성하거나, Kanban 외의 issue 총 두 가지의 issue 종류로 쪼개, 첫 번째 종류의 issue들은 간단한 issue의 경우 발생한 당일 각자의 시간에 Scrum을 진행하고, 서로의 합의가 필요한 복잡한 issue의 경우 모여서 회의를 1~2시간가량 진행한 뒤에 개발을 진행했다. 복잡한 이슈의 경우, Pair-programming을 하여 만든 Draft를 통해 Pseudo-code를 먼저 생성을 한 뒤에 이후 해당 code에 대한 fix를 하는 방식으로 진행했다.

위의 5. 평가에서 언급된 바와 같이, C- 언어의 Syntax가 명확하지 않은 부분이 있는 경우 기존에 존재하는 textbook과 C Language family가 주로 갖게 되는 규칙들에 어긋나지 않도록 design 결정을 했고, 그 과정에서 여러 예외에 대해 합리적으로 처리할 수 있었다. 또한, 개발시에 Git을 사용하여 신뢰성 있는 Continuous Integration 뿐만 아니라 이후에도 지속적으로 Open Source Contribution을 할 수 있도록 했다. 문제가 생기더라도 여러 Version Control 기능을 통해 안정적인 개발을 할 수 있었다. 해당 코드는 Github를 통해 Remote Repository를 두어 한 쪽 Repo에 문제가 생기더라도 문제 없도록 했다.