# X0-Compiler Test Document

Liu, Altair @ ilovehanhan1120@hotmail.com

November 29, 2018

## 1 Introduction

### 1.1 Purpose

The purpose of conducting as technical proposal to describe the global designing of this project, containing basic functionality of the system, run-time designing and error detecting methods. This document is aimed to provide a schema of designing and implement all functionality, which will be the critical document during the process of developing. This document will be read by developers and testers.

### 1.2 Background

This project is to develop a **X0 Language Compiler**, which is a C-like language. This project is mainly for research and study purpose.

| Item | Detail |
|---|---|
| Project Name | X0-Compiler(mini-C) |
| Developing Platform | Ubuntu 18.04 64-bit |
| Developing Tools | **Flex** and **Bison** |
| Open Source or not | Yes |

All source files can be found at: http://github.com/SubjectNoi/X0-Compiler, star and follow it, please^_^.

### 1.3 Remarks

Usage:

```
Ubuntu>$ git clone http://github.com/SubjectNoi/X0-Compiler
Ubuntu>$ cd X0-Compiler
Ubuntu>$ make
Ubuntu>$ ./X0 [Your source file]
```

20 test sample in every sub-modules are provided, named as **TestXX_[Test Content]**.

## 2 Testing Environment

### 2.1 Hardware

| Item | Detail |
|---|---|
| CPU | Intel Xeon E5-2699v3@2.30GHz(18C36T) |
| Main Board | ASUS ROG Rampage V Extreme |
| RAM | Corsair DDR4 2133@15-15-36-50 64GB |
| GPU | Nvidia Geforce RTX 2080Ti 11GB × 2 |
| Hard Disk | Intel 750 NVMe SSD 1.2TB × 2 |

### 2.2 Software

| Item | Detail |
|---|---|
| OS | Ubuntu 18.04 64-bit LTS |
| Lexical Analyzer | Flex |
| Grammar Analyzer | Bison |
| Version Control | git |
| IDE | Visual Studio Code |

### 2.3 Personnel

Developer, students, researcher using this software can be the tester, all bug will be issued by git so that developer will know them in time. All bug will be fixed ASAP. All interested users are welcome to test this software.

## 3 Testing Strategies

### 3.1 Functionality test

| Item | Detail |
|---|---|
| Test Scale | Basic functionality of the software |
| Test Aim | Ensure the correct behaviors of the software |
| Test Technology | Black-Box Testing, Marginal Testing |
| Tool | Manual Testing |
| Starting Point | Completing of corresponding modeles |
| Ending Point | Pass the standard of the developer |
| Testing Priority | 0 |

### 3.2 Performance test

| Item | Detail |
|---|---|
| Test Scale | Execution speed and efficiency |
| Test Aim | Ensure the fastest speed and highest efficiency |
| Test Technology | Black-Box Testing, Marginal Testing |
| Tool | Manual Testing |
| Starting Point | Completing of corresponding modeles |
| Ending Point | Pass the standard of the developer |
| Testing Priority | 2 |

### 3.3 Compatibility test

| Item | Detail |
|---|---|
| Test Scale | Run in different OS, version, etc. |
| Test Aim | Ensure the highest compatibility |
| Test Technology | Black-Box Testing, Marginal Testing |
| Tool | Manual Testing |
| Starting Point | Completing of corresponding modeles |
| Ending Point | Pass the standard of the developer |
| Testing Priority | 2 |

### 3.4 Security test

| Item | Detail |
|---|---|
| Test Scale | Influence to the system |
| Test Aim | Ensure the software will not lead to the crash of the system |
| Test Technology | Black-Box Testing, Marginal Testing |
| Tool | Manual Testing, Karma |
| Starting Point | Completing of corresponding modeles |
| Ending Point | Pass the standard of the developer |
| Testing Priority | 1 |

### 3.5 Conducting test

| Test Type | Unit Test | Integrated Test | System Test | Acceptation Test |
|---|---|---|---|---|
| Functionality Test | ○ | × | × | ○ |
| Performance Test | ○ | × | × | ○ |
| Compatibility Test | ○ | ○ | × | ○ |
| Security Test | ○ | × | × | ○ |

## 4 Standard of Software Testing

Following is the rank of bug in a software:

- A: Error that influence the system seriously.
- B: Normal error in functionality, influence the system a little bit.
- C: Warning, little error or unfriendly UI.
- D: Error which will not influence the normal use, like display error.
- E: Further recommendation.

There doesn't exist any error rank A. It is confirmed that all error, warning, etc. are solved based on discussion and evaluation. All test samples are executed again and there is one functionality bug that will only lead to incorrect result rather than system crash, which will be fixed in later release.

# 5 Test Details

All test sample will be appended in appendix or in the release folder.

## 5.1 Functionality Test

| Test No. | Test Method | Result | Accept or not |
|---|---|---|---|
| Test0100 | Using **make** in terminal | Software constructed correctly | ○ |
| Test0101 | Run Test00_ArrayAccess | Return correct result | ○ |
| Test0102 | Run Test01_IfTest | Return correct result | ○ |
| Test0103 | Run Test02_WhileTest | Return correct result | ○ |
| Test0104 | Run Test03_SimpleWhileBreak | Return correct result | ○ |
| Test0105 | Run Test04_ComplexWhileBreak | Return correct result | ○ |
| Test0106 | Run Test05_WhileArray | Return correct result | ○ |
| Test0107 | Run Test06_ForBreak | Return correct result | ○ |
| Test0108 | Test $a := b := 10$ | Assignment failed | × |
| Test0109 | Run Test08_ForLoopPrimeNumber | Return correct result | ○ |
| Test0110 | Run Test09_TwoWhileLoop | Return correct result | ○ |
| Test0111 | Run Test10_TypeTest | Return correct result | ○ |
| Test0112 | Run Test11_WhileContinue | Return correct result | ○ |
| Test0113 | Run Test12_ForContinue | Return correct result | ○ |
| Test0114 | Run Test13_IfElseInIf | Return correct result | ○ |
| Test0115 | Run Test14_BasicDoWhile | Return correct result | ○ |
| Test0116 | Run Test15_BoolTest | Return correct result | ○ |
| Test0117 | Run Test16_WhileSwitchCase | Return correct result | ○ |
| Test0118 | Run Test17_2DimArrayInOut | Return correct result | ○ |
| Test0119 | Run Test18_For(;;)Test | Return correct result | ○ |
| Test0120 | Run Test19_FastPow | Return correct result | ○ |
| Test0121 | Run Test07_IncPlusMinus | Return correct result | ○ |
| Test0122 | Define multiple variable with one name | Raise error | ○ |
| Test0123 | Accessing variable without declaration | Raise error | ○ |
| Test0124 | Miss one semi-colon | Syntax Error | ○ |
| Test0125 | Miss multiple semi-colon | Raise only one syntax error | × |
| Test0126 | Accessing element beyond array bound | Raise error | ○ |
| Test0127 | Adjust or read constant | Raise error | ○ |
| Test0128 | Mixed using of int and real | Implicit convert occurred | ○ |
| Test0129 | Mixed using of int and bool | Implicit convert forbidden | ○ |
| Test0130 | Mixed using of char and int | Implicit convert occurred | × |
| **Test0130-2** | Mixed using of char and int | Implicit convert forbidden | ○ |
| Test0131 | Initialize array with size >8000 | Raise error | ○ |
| Test0132 | Declare constant without initialization | Raise error | ○ |
| Test0133 | Generate more than 2000 codes | Raise error | ○ |
| Test0134 | PC jump to -1 | System crash | × |
| **Test0134-2** | PC jump to -1 | Raise error | ○ |
| Test0135 | Test **for(;;)** | Wrong behavior | × |
| **Test0135-2** | Test **for(;;)** | Unlimited loop | ○ |
| Test0136 | Test **while(true)** | Unlimited loop | ○ |
| Test0137 | Test **yarimasune;** | Display Yaju Senpai | Soudayo |
| Test0137-2 | Test **114514;** | Raise magic output | Soudayo |
| Test0138 | Declare array with more than 20 dim | Raise error | ○ |
| Test0139 | Use recursive while with more than 20 level | Raise error | ○ |

## 5.2 Performance Test

Since this project use only **Flex** and **Bison**, which means only **IR** will be generated for Virtual Machine, so, the performance is mainly based on the interpreter, that means it is hard and meaningless to test the run time of the **IR** codes generated by the compiler. So, in this test, only samples with 10 or 100 times scale are will be tested.

### 5.3 Compatibility Test

| Test No. | Test Method | Result | Accept or not |
|---|---|---|---|
| Test0300 | Launch with developing platform | Launch correctly | ○ |
| Test0301 | Launch with Ubuntu 18.04 32-bit | Launch correctly | ○ |
| Test0302 | Launch with Ubuntu 16.04 64-bit | Launch correctly | ○ |
| Test0303 | Launch with Ubuntu 16.04 32-bit | Launch correctly | ○ |
| Test0304 | Launch with Windows 10 64-bit | Launch correctly | ○ |
| Test0305 | Launch with Windows 8 64-bit | Launch correctly | ○ |
| Test0306 | Launch with Windows 7-64-bit | Launch correctly | ○ |
| Test0307 | Launch with Windows XP | Launch incorrectly | ○ |
| Test0308 | Launch with old **Lex** | Launch correctly | ○ |
| Test0309 | Launch with old **Yacc** | Launch correctly | ○ |

### 5.4 Security Test

| Test No. | Test Method | Result | Accept or not |
|---|---|---|---|
| Test0400 | Launch in compatible mode | Launch correctly | ○ |
| Test0401 | Accessing array element out of bound | Influence system data | × |
| **Test0401-2** | Accessing array element out of bound | Raise error | ○ |
| Test0402 | Set pc to be -1 | System crash | × |
| **Test0402-2** | Set pc to be -1 | Raise error | ○ |

## 6 Conclusion

Software passed the test and run stably.