



中山大學

SUN YAT - SEN UNIVERSITY

编译原理 Project 2

(2022 - 2023 学年春季学期)

课程名称	编译原理
学院	计算机学院
授课教师	张献伟
学生姓名	唐喆
学号	20337111
专业	计算机科学与技术
Email	tangzh_33@163.com
完成时间	2023 年 04 月 10 日

编译原理 Project 2 Parser

20337111 唐喆

2023 年 04 月 10 日

目录

1	概述与感想	3
1.1	概述	3
1.2	感想	3
2	实验结果	4

1 概述与感想

1.1 概述

本次实验需在SYsU-lang实现的代码框架下，基于 Bison 提供的接口完善编译器的语法分析器（Sysu-Parser）。

相较于实验一，实验二的学习曲线更陡峭，需要学习的知识点更多。直观来说，代码量都呈几何倍数增加。在历时 10 天、踩了无数坑、重构了无数次后，本次实验我的提交代码累计 1700 行 +，数十次的 Commit 着实记录了曲折的 Debug 过程：

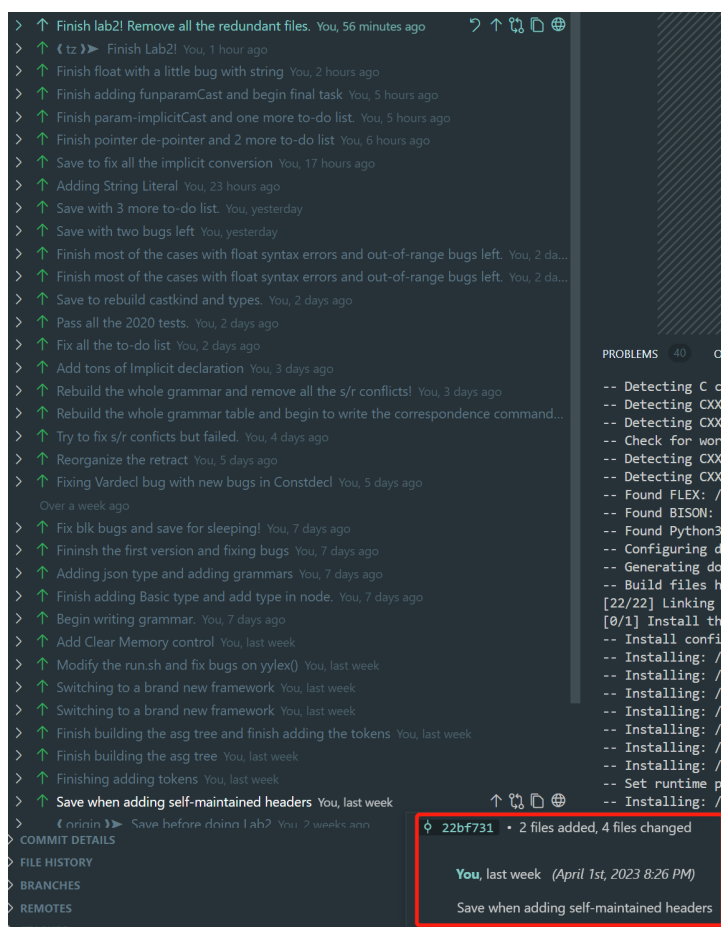


图 1: 实验二 Commit 记录

1.2 感想

在踩了这么多坑后，细节都记录在代码中了，但完成项目知道思想比细节更值得记录。以下是本次实验我的一些感想：

- **及时重构！** 本次实验有 455 个测例（不含 ThirdParty 测例）；任何打补丁的尝试最后很容易让代码无从下手且漏洞百出；所以及时总结通用操作（如二元操作的类型转换，

强制类型转换等) 并重构, 可以大大减少代码量, 提高代码的可读性和可维护性。

- **及时开始!** 不仅是因为本次实验代码量很大, 同时我们很容易在及其广阔的要求和极高的实现自由度之间矛盾。所以在大致学习相关知识后应尽快开始码代码, 在实践中完善代码设计 (反正也要重构 QAQ), 而不是等到想的尽善尽美再开始。
- **善用调试工具:** 相较于 Python 和 Java 等自带虚拟机的语言, C++ 比较痛苦的点在于发生段错误时没有调用栈, 从而很难定位问题。我使用的方法是 **Valgrind + GDB** 联合调试, 方便打印调用栈。

最后是记录一些本次实验的工作流:

- **构造语法:** 在根据 Wiki 上的 ENBF 语法实现 Bison 适用的语法后, 我们可以借助 Bison 本身提供的 Debug 工具首先检验语法是否有 *shift/reduce* 或 *reduce/reduce* 冲突, 确认语法无问题再进行下一步。值得注意的是, 我们可以适当修改、扩展提供 ENBF 文法, 从而更加优雅的实现接下来的步骤。
- **设计合适的数据结构:** 通过合适的数据结构我们可以很好的结合 Bison 维护的栈, 从而构建出易维护的抽象语法树。要注意树的内存管理方法, 防止内存泄露。
- **根据文法设计 Driver 行为:** 这里我们可以参照 Clang 的输出, 理解抽象语法树的构建过程, 进而设计 Driver 的行为。值得注意的是, 在这一阶段我们可能会发现 Wiki 上提供的文法不够全面, 我们可以根据实际情况进行修改, 但一定要返回第一步验证冲突。

2 实验结果

本次实现通过了 Parser-1 的所有测例。以下是 455 个测例 (不含 Thirdparty 测例) 的输出结果:

```

[440/455] tester/h_functional/104_long_array.sysu.c
[441/455] tester/h_functional/105_long_array2.sysu.c
[442/455] tester/h_functional/106_long_code.sysu.c
[443/455] tester/h_functional/107_long_code2.sysu.c
TimeoutExpired. Skip.
[444/455] tester/h_functional/108_many_params.sysu.c
[445/455] tester/h_functional/109_many_params2.sysu.c
[446/455] tester/h_functional/110_many_params3.sysu.c
[447/455] tester/h_functional/111_many_globals.sysu.c
[448/455] tester/h_functional/112_many_locals.sysu.c
[449/455] tester/h_functional/113_many_locals2.sysu.c
[450/455] tester/h_functional/114_register_allloc.sysu.c
[451/455] tester/h_functional/115_nested_calls.sysu.c
[452/455] tester/h_functional/116_nested_calls2.sysu.c
[453/455] tester/h_functional/117_nested_loops.sysu.c
[454/455] tester/mizuno_ai/mizuno_ai.sysu.c
tanggang@tang2he:/mnt/c/Users/TangZhe/Desktop/compiler_dockerfiles/Sysu-lang$ history | tail -n 5
150 export PATH=$HOME/sysu/bin:$PATH
151 echo "20337111 TangZhe"
152 export PATH=$HOME/sysu/bin:$PATH CPATH=$HOME/sysu/include:$CPATH LIBRARY_PATH=$HOME/sysu/lib:$LIBRARY_
PATH LD_LIBRARY_PATH=$HOME/sysu/lib:$LD_LIBRARY_PATH && sysu-compiler --unittest-parser-1(i) "**/*.sysu.c"
153 export PATH=$HOME/sysu/bin:$PATH CPATH=$HOME/sysu/include:$CPATH LIBRARY_PATH=$HOME/sysu/lib:$LIBRARY_
PATH LD_LIBRARY_PATH=$HOME/sysu/lib:$LD_LIBRARY_PATH && sysu-compiler --unittest-parser-1 "**/*.sysu.c"
154 history | tail -n 5
  
```

图 2: 实验二实现完整截图

值得指出的是, 本次实验有几个较为麻烦的测例, 我也截图如下。首先是浮点测例 (相对路径: `tester/function_test2022/95_float.sysu.c`):

```

[309/455] tester/function_test2022/95_float.sysu.c
tester/function_test2022/95_float.sysu.c:18:28: warning: implicit conversion from 'double' to 'int' changes value from 2.9 to 2 [-Wliteral-conversion]
const int MAX = 1e9, TAO = 2.9, THREE = 3.2, FIVE = TAO + THREE;
~~~~~~^
tester/function_test2022/95_float.sysu.c:18:41: warning: implicit conversion from 'double' to 'int' changes value from 3.2 to 3 [-Wliteral-conversion]
const int MAX = 1e9, TAO = 2.9, THREE = 3.2, FIVE = TAO + THREE;
~~~~~~^
tester/function_test2022/95_float.sysu.c:78:7: warning: implicit conversion from 'double' to 'Bool' changes value from 1.5 to true [-Wliteral-conversion]
if (1.5 < 0);
~^
tester/function_test2022/95_float.sysu.c:79:9: warning: implicit conversion from 'double' to 'Bool' changes value from 3.3 to true [-Wliteral-conversion]
if ((113.3) < 0);
~^
tester/function_test2022/95_float.sysu.c:81:12: warning: implicit conversion from 'double' to 'Bool' changes value from 0.3 to true [-Wliteral-conversion]
if (0 || 0.3) < 0;
~^
3 warnings generated.
[318/455] tester/function_test2022/96_matrix_add.sysu.c

```

顺利通过Float测例

图 3: 实验二浮点测例截图

其次是整数上限测例（即整数类型超限，需要隐式转换的测例，相对路径：`tester/h_functional/100_int_literal.sysu.c`）:

```

[435/455] tester/h_functional/099_skin_spaces.sysu.c
[436/455] tester/h_functional/100_int_literal.sysu.c
[437/455] tester/h_functional/101_scope2.sysu.c
[438/455] tester/h_functional/102_short_circuit3.sysu.c
tester/h_functional/102_short_circuit3.sysu.c:40:25: warning: self-comparison always evaluates to false [-Wtautological-compare]
if (i0 == i11 && i3 < i3 || i4 >= i4) { _sysy_putch(74); }
~^
tester/h_functional/102_short_circuit3.sysu.c:40:36: warning: self-comparison always evaluates to true [-Wtautological-compare]
if (i0 == i11 && i3 < i3 || i4 >= i4) { _sysy_putch(74); }
~^
tester/h_functional/102_short_circuit3.sysu.c:41:25: warning: self-comparison always evaluates to false [-Wtautological-compare]
if (i0 == i11 || i3 < i3 && i4 >= i4) _sysy_putch(75);
~^
tester/h_functional/102_short_circuit3.sysu.c:41:36: warning: self-comparison always evaluates to true [-Wtautological-compare]
if (i0 == i11 || i3 < i3 && i4 >= i4) _sysy_putch(75);
~^
4 warnings generated.

```

顺利通过整数溢出测例

图 4: 实验二整数上限测例

以及最后的字符串拼接测例（相对路径：`tester/mizuno_ai/mizuno_ai.sysu.c`）:

```

4 warnings generated.
[439/455] tester/h_functional/103_long_func.sysu.c
[440/455] tester/h_functional/104_long_array.sysu.c
[441/455] tester/h_functional/105_long_array2.sysu.c
[442/455] tester/h_functional/106_long_code.sysu.c
[443/455] tester/h_functional/107_long_code2.sysu.c
TimeoutExpired. Skip.
[444/455] tester/h_functional/108_many_params.sysu.c
[445/455] tester/h_functional/109_many_params2.sysu.c
[446/455] tester/h_functional/110_many_params3.sysu.c
[447/455] tester/h_functional/111_many_globals.sysu.c
[448/455] tester/h_functional/112_many_locals.sysu.c
[449/455] tester/h_functional/113_many_locals2.sysu.c
[450/455] tester/h_functional/114_register_alloc.sysu.c
[451/455] tester/h_functional/115_nested_calls.sysu.c
[452/455] tester/h_functional/116_nested_calls2.sysu.c
[453/455] tester/h_functional/117_nested_loops.sysu.c
[454/455] tester/mizuno_ai/mizuno_ai.sysu.c
+ set +x

```

顺利通过字符串
拼接测例

图 5: 实验二字符串拼接测例

以上即为实验二的感想以及实验结果。