

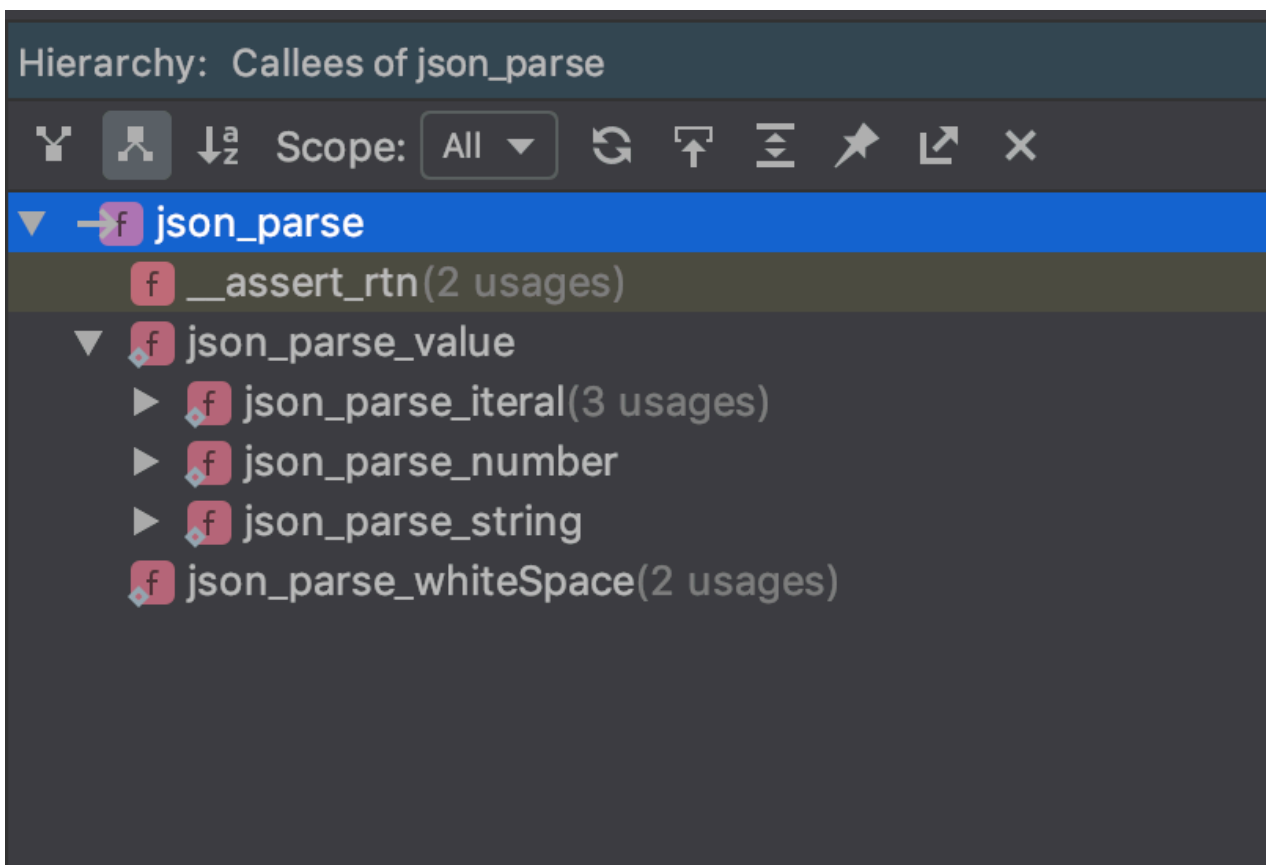
JSON解析器说明

已完成部分

JSON的6种基本数据类型，已经完成了null、boolean、number、string部分。以下是几个重要的函数结构图：

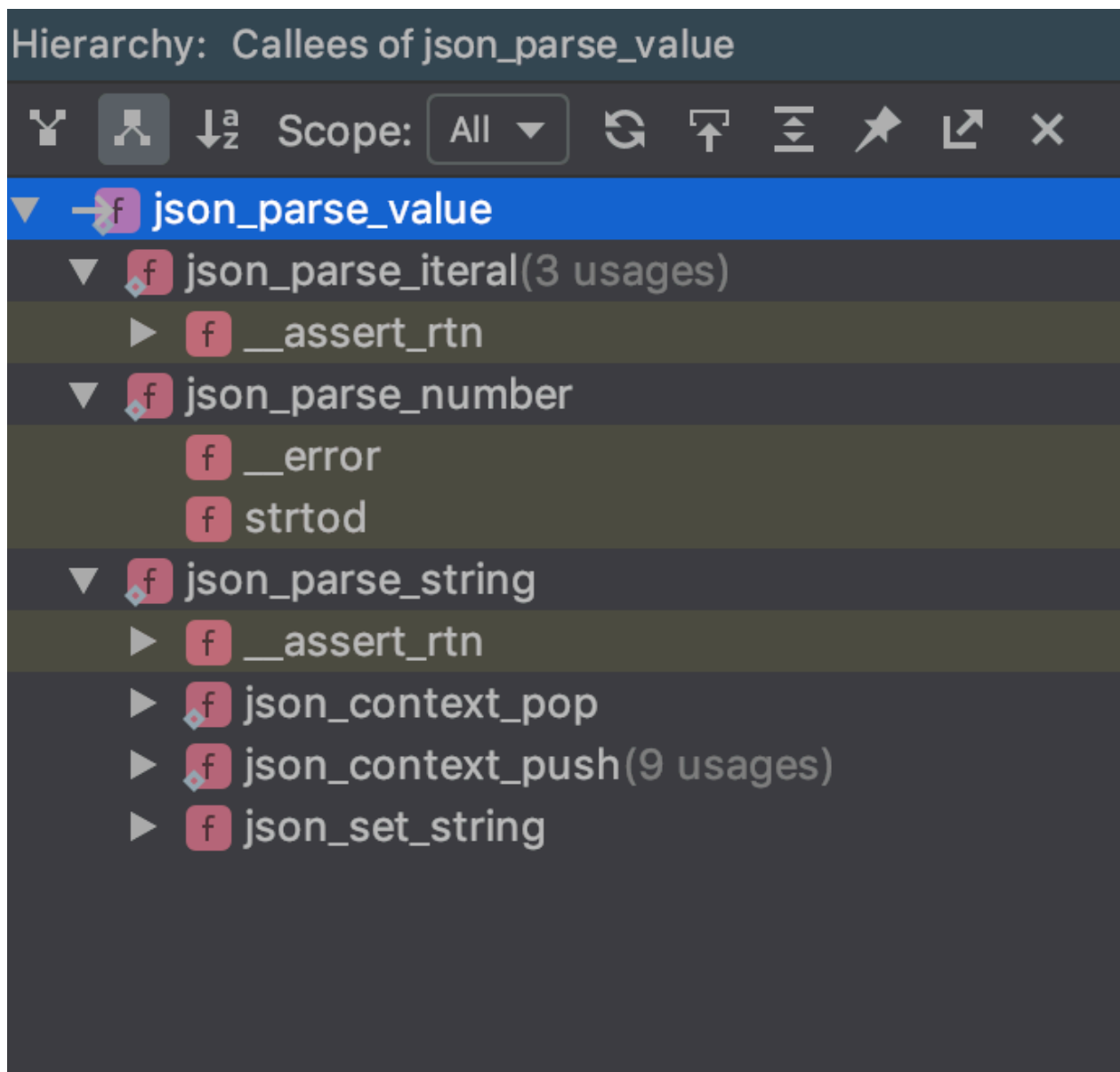
*json_parse()*函数

该函数是主要被调用的函数，`json_parse_iteral()`用来解析boolean和null类型，`json_parse_number()`解析number，`json_parse_string()`主要解析string。



*json_parse_value()*函数

该函数通过switch跳转选择不同的解析函数。遇到'n'、't'、'f'字符则跳转至json_parse_iteral()函数进行解析，遇到\"则跳转至json_parse_string()函数进行解析，default情况则跳转至json_parse_number()进行解析。



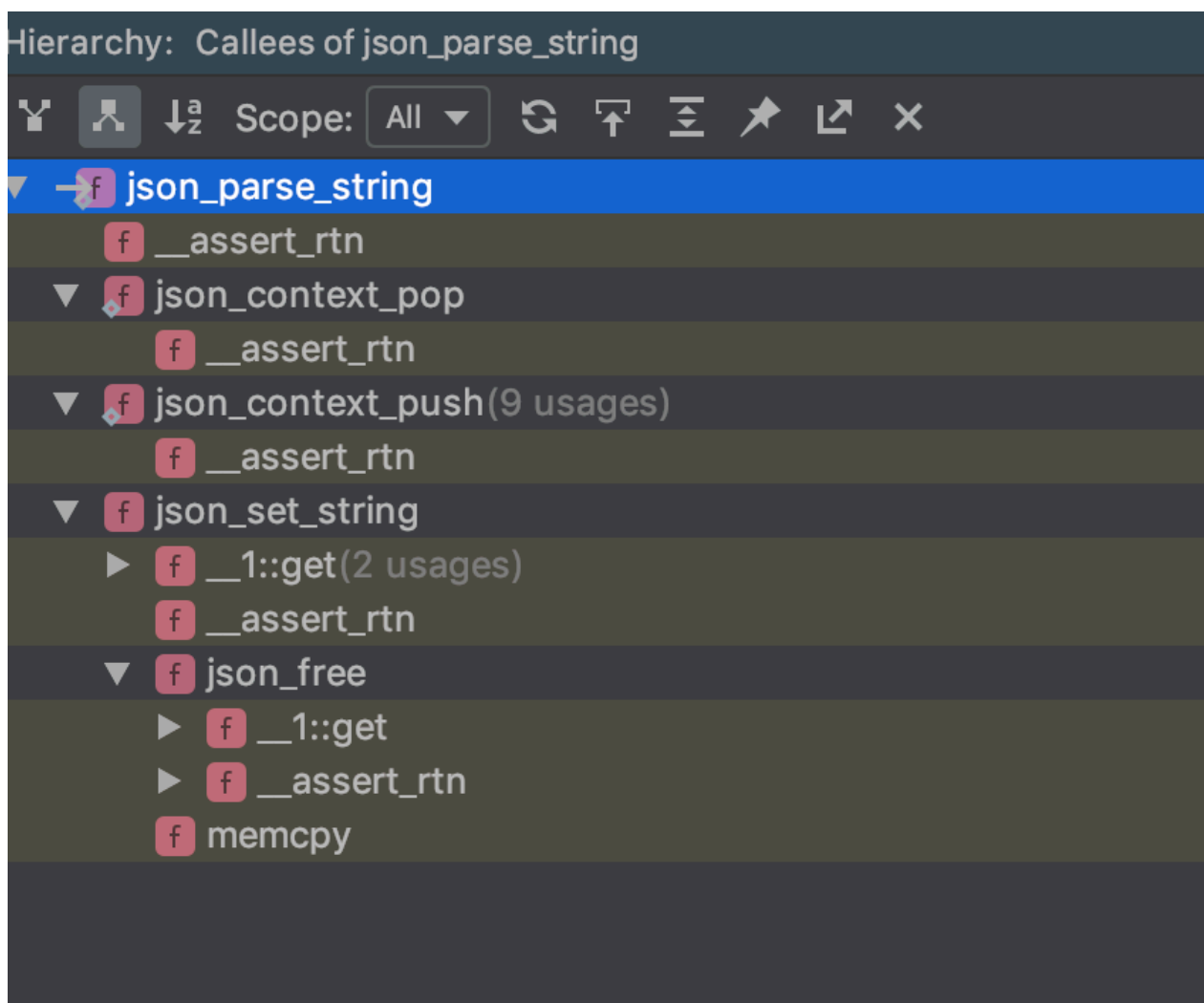
*json_parse_string()*函数

遇到最大的困难就是json_parse_string()函数了。我解析字符串的思路就是由于字符串不是固定的长度，所以要动态分配内存，同时，我需要把解析的结果先储存在一个临时的缓冲区，最后再用json_set_string()把缓冲区的结果设进值之中。在完成解析一个字符串之前，这个缓冲区的大小是不能预知的。因此，我们

可以采用动态数组（dynamic array）这种数据结构，即数组空间不足时，能自动扩展。C++ 标准库的 `std::vector` 就是一种动态数组。

如果每次解析字符串时，都重新建一个动态数组，那么是比较耗时的。我们可以重用这个动态数组，每次解析 JSON 时就只需要创建一个。而且我们将会发现，无论是解析字符串、数组或对象，我们也只需要以先进后出的方式访问这个动态数组。换句话说，我们需要一个动态的堆栈数据结构。

但是我发现我总是会写出 `segmentation fault` 的代码，这是由于代码访问了禁止访问的区域导致的，或者有可能提前 `delete` 了某些指针。排查这类问题花了我大约两周的时间，为了避免这样的问题，准备在以后进行重构，使用 C++11 中推荐使用的智能指针 `smart pointer` 进行值的传递，不再使用裸指针。



单元测试代码

以下单元测试已经全部通过。

```
static void test_parse() {
    test_parse_null();
    test_parse_true();
    test_parse_false();
    test_access_boolean();
    test_parse_invalid_value();
    test_parse_expect_value();
    test_parse_root_not_singular();
    test_parse_number_too_big();
    test_parse_number();
    test_access_number();
    test_parse_string();
    test_access_string();
    test_parse_invalid_string_char();
    test_parse_invalid_string_escape();
    test_parse_missing_quotation_mark();
    /* ... */
}
```

```
rafa_json x
/Users/fszhuangb/Documents/rafa_json/cmake-build-debug/rafa_json
162/162 (100) passed

Process finished with exit code 0
```

未完成部分

1. 当前我已经能解析一般的 JSON 字符串，但仅仅没有处理 `\uXXXX` 这种转义序列，下一步要完成 Unicode 的解析，才能完全符合标准。
2. 继续完成对于 `array` 和 `object` 的解析。
3. 完成生成器的编写。

疑问

1. 一个JSON parser除了以上提到的这些功能和部分之外，还需要其他的什么功能呢？
2. 我的代码编写基本是面向过程的，是否需要面向对象的重构呢？
3. 不知道老师有没有时间，可以当面看一下我的代码，希望老师能不吝赐教，谢谢！