

C 0x中的垃圾收集 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/268/2021_2022_C__0x_E4_B8_AD_E7_9A_c97_268069.htm 最近两年我对 gc 很感兴趣 :D 已经在项目中用了两年。项目从 C 转到 C ， gc 模块的实现发生了变化，但是本质却没有变。我对 C 加入 gc 是非常欢迎的，这点在以前写的另一篇 blog 中已经表明过态度。记得两年前，当有机会当面问 Bjarne Stroustrup 关于 C 发展的问题时，我毫不犹豫的讲出自己对 gc 的迫切期待，并希望能够以最小代价的把 gc 加入 C。因为已经实现过一些 C 的 gc 模块，我有一些语言上的需求。当时描述了自己的想法，可惜英文实在是太差了，完全说不清楚 :(因为没听明白我的意思，Bjarne Stroustrup 他老人家似乎也很无奈，最后只是建议中国的程序员应该参与于到语言的标准化事务当中去，一直以来，C 标准委员会中似乎没有来至中国大陆的程序员。既然 C 是你的工具，你就应该努力把自己对工具的改进需求说出来。其实我的需求很简单，就是 C 中应该加入一些对数据结构中数据类型的有限描述。其实只做内存管理的话，类型信息只需要区分数据还是指针就够了。具体数据类型可以忽略。我的想法就是像虚表一样，给每个类多加一张表，描述这个对象中指针的位置(记录一个偏移量)。由于 C 的对象布局比较复杂，这个工作如果不在编译器里做，会相当麻烦。有这样的信息，gc 就可以容易的遍历内存了。看那篇文章的介绍，C 的 gc 似乎用 gc_strict gc_relax 这样的关键字来描述一整块内存区内有没有指针，而没有更细致的精确到每个数据上。这跟已有的 C 的 gc 库实现类似。我猜测这些是为了兼容 POD 类型设

计的，对于 C 自己的类，应该可以更好的解决。毕竟编译器知道全部的类型信息。除此之外，遍历堆栈依旧是个问题，但已经好多了。遍历可以用各种语法糖来实现，反正 C 有了 template 后，什么诡异的写法都弄的出来。最后说两句 gc 的效率问题。gc 没有人肉内存管理效率高是一种普遍的误解。如果不是靠臆测，而是自己实现一个 gc 模块，然后做代码剖析的话，很容易相信 gc 可以带来更高的性能。关于 gc 和人肉内存管理之间的性能话题，以前写过太多，吵的太多，嚼着都没味道了。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com