

C 0x：崭新的C，还是另一个Java？PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/167/2021_2022_C__0x_EF_B_C_9A_E5_B4_c104_167022.htm 从全球最活跃的C社区boost传来消息，新的C标准：C 0x，最早将在2007年10月发布.....

最近，从全球最活跃的C社区boost传来消息，新的C标准：C 0x，最早将在2007年10月发布。很早就有人猜测，C 0x中的x到底是8还是9，不过现在看来很可能是7，但是也不尽然，因为上一个C标准本来计划是1997年发布，但是因为STL的引入而推迟到1998年。这次又遇到了相同的情况，大家都不怀疑TR1和TR2(Technical Report)将成为新标准的一部分，但是随着越来越多的boost库被移到TR2中，TR2的进一步完善很可能拖新标准的后腿。不过，C 0x是C的一次重大升级，恐怕编译器厂商最快也只能在2009年推出符合新标准的编译器。新的C标准将在几个方面对C标准进行扩充：1. 增强STL库，标准库TR1和TR2中的新组建，比如“正则表达式”。2. 并发内存模型和并发库(面向多核处理器)。3. 垃圾收集器。4. 可简化的泛型程序的开发。从上一个标准C 98发布到现在，将近有10年的时间了，在这十年的时间里编程语言发生了翻天覆地的变化，首先是JAVA异军突起，然后是微软借助.NET平台的优势推出了C#，C已经不再是编程语言中的明星了，它被挤下了神坛，慢慢沦为边缘编程语言。这中间的主要原因并不是C的语法落伍了，而是使用C开发软件效率低下并且容易出错。效率低下的原因是因为C缺少库的支持，C只提供了编程语言的基本功能，缺少强有力的库支持，比如字符串处理就一直是C的诟病。容易出错的原因是C从C继承了动态

内存分配和指针，这一点被认为是软件发生错误的根源，虽然C引入STL来解决这个问题，比如用vector代替内置数组可以防止内存泄漏，用迭代器代替直接下标操作可以防止地址越界。但是STL毕竟太简单了，比如字符串处理就只有一个简单的string，比起Perl差远了。C唯一的优势就是速度，但是随着处理器和内存的快速发展也慢慢被遗忘了，所以就沦落到了现在的地步。C要想摆脱这种局面，就必须解决这两个问题，而在新标准中我们已经看到了答案。TR1和TR2已经成为新的标准库的一部分，它们不仅提供了象“正则表达式”和“哈希表”这样程序员盼望已久的功能，还有针对多核处理器的并发内存模型和并发库，对网络也有socket库等等。针对内存出错问题，新标准通过强化STL代替内存访问，使用垃圾收集器防止内存泄漏。在C社区上关于新标准讨论最多的就是到底应不应该修改C语言，争论的核心之一就是垃圾收集器。老鸟们总是认为“不要为了新手而将C降格，适合新手的语言已经有很多了！”，很多C程序员认为新标准应该将精力集中于扩充标准库，而不是解决内存泄漏问题，因为那是程序员自己的事。在本人看来这种思想将会害死C，因为现实中总是新手比老鸟多，软件公司的老板招一个程序员是希望他(她，它)能够立即开始编写没有错误的代码，而不是培养他们知道他们成为专家后再开始干活，所以，如果C不能成为一种“菜鸟友好”的语言，那么C的堕落将不可避免。设想某个程序员分配了一块内存而忘记释放它，但是垃圾收集器捕获了这个异常并默默地替他释放了这块内存，从而使内存泄漏这种悲惨的事情得以避免，而程序运行起来状态良好，那还有什么理由不在C中添加垃圾收集功能呢？C0x能够

重振C 的雄风吗?有了垃圾收集器的C ，并且所有内存和指针操作都被“友好地”建议用STL代替，那还是C 吗?还是另一个JAVA?答案当然是“这仍旧是C ，请放心使用”，不过C 0x之后还会有什么新的内容，会不会有一套GUI库?如果是那样的话就真的成JAVA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