

<<垃圾收集>>

图书基本信息

书名：<<垃圾收集>>

13位ISBN编号：9787115120700

10位ISBN编号：7115120706

出版时间：1900-01-01

出版时间：人民邮电出版社

作者：(美国)琼斯等著、谢之易译

译者：琼斯

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<垃圾收集>>

### 内容概要

书围绕着动态内存自动回收的话题，介绍了垃圾收集机制，详细分析了各种算法和相关技术。  
本书共12章。

第1章首先介绍计算机存储器管理的演化和自动内存回收的需求，并引入了本书所使用的术语和记法。

第2章介绍了3种“经典”的垃圾收集技术：引用计数（reference counting）、标记 - 清扫（mark-sweep）和节点复制（copying）。

随后的4章更详细地讨论了上述这些垃圾收集方式和标记 - 缩并（mark-compact）收集。

第7章和第8章分别介绍了在现代垃圾收集实现中具有重要地位的分代式（generational）垃圾收集和渐进式（incremental）垃圾收集。

第9章和第10章扩展了垃圾收集的领域，讨论了如何让垃圾收集能够在无法得到来自语言编译器的支持的环境（分别是C和C++）中运行。

第11章讨论了一个相对较新的研究领域 -- 垃圾收集和硬件数据cache的相互作用。

第12章简要地考察了用于分布式系统的垃圾收集。

本书适合对动态内存管理感兴趣的读者阅读，可供专业的研究人员参考。

## &lt;&lt;垃圾收集&gt;&gt;

## 书籍目录

第1章 简介 11.1 内存分配的历史 21.1.1 静态分配 21.1.2 栈分配 31.1.3 堆分配 31.2 状态、存  
和指针可达性 41.3 显式堆分配 51.3.1 一个简单的例子 51.3.2 垃圾 51.3.3 悬挂引用 61.3.4 共  
享 61.3.5 失败 61.4 为什么需要垃圾收集 71.4.1 语言的需求 71.4.2 问题的需求 71.4.3 软件工程  
题 81.4.4 没有银弹 91.5 垃圾收集的开销有多大 111.6 垃圾收集算法比较 111.7 记法 151.7.1  
堆 151.7.2 指针和子女 151.7.3 伪代码 161.8 引文注记 16第2章 经典算法 182.1 引用计数算法  
算法 182.1.2 一个例子 202.1.3 引用计数算法的优势和弱点 212.1.4 环形数据结构 222.2 标记 - 清  
扫算法 232.2.1 算法 232.2.2 标记 - 清扫算法的优势和弱点 252.3 节点复制算法 262.3.1 算法 272.  
一个例子 282.3.3 节点复制算法的优势和弱点 302.4 比较标记 - 清扫技术和节点复制技术 312.5 需要  
考虑的问题 322.6 引文注记 36第3章 引用计数 383.1 非递归的释放 383.1.1 算法 383.1.2 延迟释  
优点和代价 393.2 延迟引用计数 393.2.1 Deutsch-Bobrow算法 403.2.2 一个例子 413.2.3 ZCT溢  
出 433.2.4 延迟引用计数的效率 433.3 计数域大小受限的引用计数 443.3.1 “粘住的”计数值 443.3.  
追踪式收集恢复计数值 443.3.3 仅有一位的计数值 453.3.4 恢复独享信息 463.3.5 “Ought to be two  
”缓冲区 473.4 硬件引用计数 483.5 环形引用计数 493.5.1 函数式程序设计语言 493.5.2 Bobrow的  
术 503.5.3 弱指针算法 513.5.4 部分标记 - 清扫算法 543.6 需要考虑的问题 603.7 引文注记 63第4  
标记 - 清扫垃圾收集 664.1 与引用计数技术的比较 664.2 使用标记栈 674.2.1 显式地使用栈来实现递  
归 674.2.2 最小化栈的深度 684.2.3 栈溢出 704.3 指针反转 724.3.1 Deutsch-Schorr-Waite算法 734.  
可变大小节点的指针反转 754.3.3 指针反转的开销 754.4 位图标记 764.5 延迟清扫 784.5.1 Hughes  
延迟清扫算法 784.5.2 Boehm-Demers-Weiser清扫器 794.5.3 Zorn的延迟清扫器 814.6 需要考虑的问  
题 824.7 引文注记 84第5章 标记 - 缩并垃圾收集 865.1 碎片现象 865.2 缩并的方式 875.3 “双指  
”算法 895.3.1 算法 905.3.2 对“双指针”算法的分析 915.3.3 可变大小的单元 915.4 Lisp 2算法 9  
基于表的方法 935.5.1 算法 935.5.2 间断表 945.5.3 更新指针 955.6 穿线方法 955.6.1 穿线指  
针 955.6.2 Jonkers的缩并算法 965.6.3 前向指针 975.6.4 后向指针 985.7 需要考虑的问题 995.8 引  
注记 101第6章 节点复制垃圾收集 1036.1 Cheney的节点复制收集器 1046.1.1 三色抽象 1046.1.2 算  
法 1056.1.3 一个例子 1066.2 廉价地分配 1096.3 多区域收集 1106.3.1 静态区域 1116.3.2 大型对  
域 1116.3.3 渐进的递增缩并垃圾收集 1116.4 垃圾收集器的效率 1136.5 局部性问题 1146.6 重组策  
略 1156.6.1 深度优先节点复制与广度优先节点复制 1166.6.2 不需要栈的递归式节点复制收集 1176.6.3  
近似于深度优先的节点复制 1196.6.4 层次分解 1206.6.5 哈希表 1216.7 需要考虑的问题 1226.8 引  
文注记 124第7章 分代式垃圾收集 1267.1 分代假设 1267.2 分代式垃圾收集 1297.2.1 一个简单例  
子 1297.2.2 中断时间 1317.2.3 次级收集的根集合 1327.2.4 性能 1337.3 提升策略 1347.3.1 多个分  
代 1347.3.2 提升的阈值 1357.3.3 Standard ML of New Jersey收集器 1367.3.4 自适应提升 1377.4 分代  
织和年龄记录 1407.4.1 每个分代一个半区 1407.4.2 创建空间 1407.4.3 记录年龄 1417.4.4 大型对象  
域 1447.5 分代间指针 1457.5.1 写拦截器 1457.5.2 入口表 1467.5.3 记忆集 1477.5.4 顺序保存缓冲  
区 1487.5.5 硬件支持的页面标记 1497.5.6 虚存系统支持的页面标记 1507.5.7 卡片标记 1517.5.8 记  
集还是卡片 1537.6 非节点复制的分代式垃圾收集 1547.7 调度垃圾收集 1557.7.1 关键对象 1567.7.2  
成熟对象空间 1577.8 需要考虑的问题 1597.9 引文注记 160第8章 渐进式和并发垃圾收集 1628.1 同  
步 1638.2 拦截器方案 1658.3 标记 - 清扫收集器 1678.3.1 写拦截器 1678.3.2 新单元 1718.3.3 初  
和终止 1738.3.4 虚存技术 1768.4 并发引用计数 1778.5 Baker的算法 1808.5.1 算法 1818.5.2 Bake  
的延迟的界限 1828.5.3 Baker的算法的局限 1828.5.4 Baker算法的变种 1838.5.5 动态重组 1848.6  
Appel-Ellis-Li收集器 1868.6.1 各种改进 1888.6.2 大型对象 1888.6.3 分代 1898.6.4 性能 1898.7  
复制收集器 1908.7.1 Nettle的应变复制收集器 1908.7.2 Huelsbergen和Larus的收集器 1918.7.3  
Doligez-Leroy-Gonthier收集器 1928.8 Baker的工作环收集器 1948.9 对实时垃圾收集的硬件支  
持 1968.10 需要考虑的问题 1978.11 引文注记 199第9章 C语言的垃圾收集 2029.1 根不确定收集的  
个分类 2039.2 保守式垃圾收集 2059.2.1 分配 2059.2.2 寻找根和指针 2069.2.3 内部指针 2099.2.4  
守式垃圾收集的问题 2099.2.5 识别错误 2119.2.6 效率 2129.2.7 渐进式、分代式垃圾收集 2149.3 准  
制式收集 2159.3.1 堆的布局 2159.3.2 分配 2169.3.3 垃圾收集 2169.3.4 分代式垃圾收集 2189.3.5

## &lt;&lt;垃圾收集&gt;&gt;

精确识别的数据结构 2189.3.6 准复制式收集的效率 2199.4 优化的编译器是“魔鬼” 2209.5 需要考虑的问题 2239.6 引文注记 224第10章 C++语言的垃圾收集 22610.1 用于面向对象语言的垃圾收集 22710.2 对C++垃圾收集器的需求 22810.3 在编译器中还是在库中 23010.4 保守式垃圾收集 23010.5 准复制式收集器 23110.6 智能指针 23410.6.1 在没有智能指针类层次的情况下进行转换 23410.6.2 多重继承 23510.6.3 不正确的转换 23510.6.4 某些指针无法“智能化” 23610.6.5 用const和volatile修饰的指针 23610.6.6 智能指针的“泄漏” 23610.6.7 智能指针和引用计数 23710.6.8 一个简单的引用计数指针 23810.6.9 用于灵活的垃圾收集的智能指针 23810.6.10 用于追踪式垃圾收集的智能指针 24010.7 为支持垃圾收集而修改C++ 24110.8 Ellis和Detlefs的建议 24210.9 终结机制 24310.10 需要考虑的问题 24510.11 引文注记 246第11章 垃圾收集与cache 24811.1 现代处理器体系结构 24811.2 cache的体系结构 25011.2.1 cache容量 25011.2.2 放置策略 25111.2.3 写策略 25211.2.4 特殊的cache指令 25411.3 存访问的模式 25411.3.1 标记 - 清扫技术, 使用标记位图和延迟清扫 25411.3.2 节点复制垃圾收集 25511.3.3 渐进式垃圾收集 25611.3.4 避免读取 25611.4 改进cache性能的标准方法 25711.4.1 cache的容量 25711.4.2 块大小 26011.4.3 相联度 26011.4.4 特殊指令 26211.4.5 预取 26211.5 失误率和体cache性能 26311.6 专用硬件 26511.7 需要考虑的问题 26511.8 引文注记 266第12章 分布式垃圾收集 26812.1 需求 26912.2 虚拟共享存储器 27012.2.1 共享虚拟存储器模型 27112.2.2 共享数据对象模型 27112.2.3 分布式共享存储器之上的垃圾收集 27212.3 与分布式垃圾收集有关的课题 27212.3.1 分类原则 27212.3.2 同步 27412.3.3 鲁棒性 27412.4 分布式标记 - 清扫 27512.4.1 Hudak和Keller 27512.4.2 Ali的算法 27712.4.3 Hughes的算法 27712.4.4 Liskov-Ladin算法 27812.4.5 Augusteijn的算法 27812.4.6 Vestal的算法 27812.4.7 Schelvis-Bledoeg算法 27812.4.8 Emerald收集器 27912.4.9 IK收集器 28012.5 分布式节点复制 28012.6 分布式引用计数 28112.6.1 Lermen-Maurer协议 28112.6.2 间接引用计数 28112.6.3 Mancini-Shrivastava算法 28212.6.4 SPG协议 28212.6.5 “Garbage collecting the world” 28312.6.6 网络对象 28312.6.7 带权引用计数 28412.6.8 世代引用计数 28412.7 对actor进行垃圾收集 28512.7.1 Halstead算法 28512.7.2 标记算法 28512.7.3 逻辑上集中式的收集器 28612.8 引文注记 参考文献 298术语表 288索引 331算法列表 339

<<垃圾收集>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>