

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 11/36 (2006.01)



[12] 发明专利说明书

专利号 ZL 02121418.2

[45] 授权公告日 2006年3月29日

[11] 授权公告号 CN 1248114C

[22] 申请日 2002.6.20 [21] 申请号 02121418.2
[71] 专利权人 华为技术有限公司
地址 517057 广东省深圳市科技园科发路华为
用户服务中心大厦知识产权部
[72] 发明人 叶顺福
审查员 白雪涛

[74] 专利代理机构 北京集佳知识产权代理有限公司
代理人 逯长明

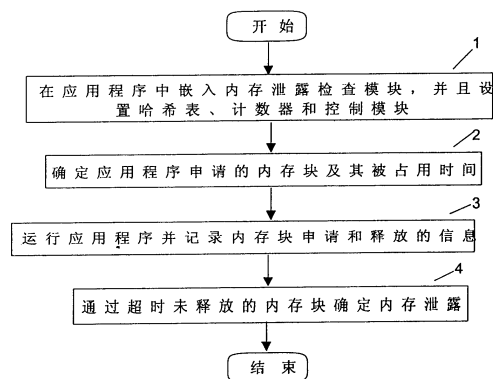
权利要求书 2 页 说明书 6 页 附图 2 页

[54] 发明名称

一种软件内存泄露的检查方法

[57] 摘要

本发明公开了一种软件内存泄露的检查方法，该方法首先确定应用程序申请的内存块，以及所申请的不同内存块被占用的时间，定义内存块记录表，用于记录应用程序申请内存块和释放内存块的信息，设置内存块计数器，用于对应用程序申请和释放的内存块进行管理；在应用程序运行时，记录其申请内存块和释放内存块的信息，最后确定超过占用时间尚未释放的内存块，通过上述内存块确定应用程序的内存泄露点；采用上述方案进行应用程序内存泄露的检查对应用程序具有较小地影响，且易于实现和成本低廉，有利于在应用程序运行过程中对可能产生的内存泄露进行长期检测，有助于发现应用程序在一些实际环境下才出现的或者较隐蔽的内存泄漏错误。



1、一种软件内存泄露的检查方法，包括：

a. 确定应用程序申请的内存块，以及所申请的不同内存块被占用的时间；定义内存块记录表，用于记录应用程序申请内存块和释放内存块的信息；设置内存块计数器，用于对应用程序申请和释放的内存块进行管理；

b. 在应用程序运行时，记录其申请内存块和释放内存块的信息；

c. 确定超过占用时间尚未释放的内存块，通过上述内存块确定应用程序的内存泄露点。

2、根据权利要求1所述的软件内存泄露的检查方法，其特征在于：所述内存块记录表为哈希(HASH)表。

3、根据权利要求1或2所述的软件内存泄露的检查方法，其特征在于：所述方法通过在应用程序中嵌入内存泄露检查模块实现。

4、根据权利要求3所述的软件内存泄露的检查方法，其特征在于：设置控制模块，以从应用程序外部控制内存泄露检查模块进行内存泄露的检查。

5、根据权利要求4所述的软件内存泄露的检查方法，其特征在于：控制模块控制内存泄露检查模块按照下述步骤实现状态的迁移以进行内存泄露的检查：

(51) 首先使内存泄露检查模块处于空闲状态，当需要进行内存泄露分析时，向内存泄露检查模块发送操作记录信号，使其进入操作记录状态记录应用程序的内存申请和释放操作；

(52) 记录一段时间后，向内存泄露检查模块发送停止插入记录信号，使其进入等待释放状态，只记录内存释放操作而不记录内存申请操作，等待哈希表中记录的尚未释放的内存块释放；

(53) 当需要确定是否存在内存泄露时，向内存泄露检查模块发送停止信号，使其进入停止状态以分析确定是否存在内存泄露；

(54) 当需要重新开始内存泄露检查时，向内存泄露检查模块发送复位信号，然后转步骤(51)进行新一轮的内存泄露检查。

一种软件内存泄露的检查方法

技术领域

本发明涉及软件性能测试方法，具体地说涉及到软件内存泄露的检查方法。

背景技术

在软件（或者说应用程序）的诸多缺陷中，内存泄漏是一项比较难发现的错误。导致软件内存泄露的原因可能多种多样的，例如可能是由软件设计缺陷造成的，也可能是由操作系统造成的。无论如何，如果软件存在内存泄漏的缺陷，将会导致系统可用内存减少，运行效率下降；对那些需长期不间断运行的软件，即使一次只泄漏很少的内存，经过长期积累也会导致系统内存耗尽直至瘫痪。因此，软件的内存泄漏检查是测试中一个重要的检查项。目前的软件内存泄露检查一般采用一些专用内存检查工具进行，尽管这些工具可以检测出大部分软件的内存错误操作。但使用这些工具也有几个不足之处：一是这些工具大多都是大型商业软件，成本较高；二是这些工具是通过在可执行文件中插入监视代码来实现内存的操作监视，使得软件的程序代码或容量要增大很多，运行起来也要慢很多，只能在软件处于测试阶段时使用，即只能进行静态检查，软件正式在线运行时无法进行检测，而一些比较隐蔽的内存泄漏，只有在正式运行环境下才能出现，要捕捉这些内存泄漏，上述专用内存检查工具提供的方法已无能为力。

发明内容

本发明的目的在于提供一种成本较低、能够在软件在线运行时进行检查的软件内存泄露的检查方法。

为达到上述目的，本发明提供的软件内存泄露的检查方法，包括：

- a、确定应用程序申请的内存块，以及所申请的不同内存块被占用的时间；定义内存块记录表，以记录应用程序申请内存块和释放内存块的信息；设置内存块计数器，用于对应用程序申请和释放的内存块进行管理；
- b. 在应用程序运行时，记录其申请内存块和释放内存块的信息；
- c. 确定超过占用时间尚未释放的内存块，通过上述内存块确定应用程序的内存泄露点。

上述方法通过在应用程序中嵌入内存泄露检查模块实现。

由于本发明在应用程序运行时，记录其申请内存块和释放内存块的信息，通过确定超过占用时间尚未释放的内存块来确定软件的内存泄露点，上述方法的实现可以通过在应用程序中嵌入专用的内存泄露模块完成，这样，采用本发明的方法不但适合对应用程序的内存泄露进行在线检查，还由于在应用程序运行时仅记录其申请内存块和释放内存块的信息，对应用程序具有较小地影响，且易于实现和成本低廉；因此，采用本发明有利于在应用程序运行过程中对可能产生的内存泄露进行长期检测，有助于发现一些实际环境下才出现的或者较隐蔽的内存泄漏错误；同时，通过专用的控制模块对内存泄露检查模块进行控制，使得内存泄露检查的各个步骤时

间长短（主要是等待释放的状态或步骤的时长）得以控制，从而可以适应占用时长不同的内存块。另外，使用本方法进行的内存泄漏检查可以随时启动和停止，从而使检查对应用程序的影响可以得到进一步的控制，例如可以避免程序的事务繁忙期，在程序空闲时启动检查，或者缩短检查的时长等等。

附图说明

图1是本发明所述方法的实施例流程图；

图2是图1所述实施例采用的内存泄漏检查模块和应用程序其它模块的关系图；

图3是内存泄漏检查模块内部逻辑结构图；

图4是内存泄漏检查模块在进行内存泄露检查时的状态迁移图。

具体实施方式

下面结合附图对本发明作进一步详细的描述。

图1是本发明所述方法的实施例流程图。图1所述方法的实质，是通过在应用程序中嵌入一个内存泄漏检查模块，使应用程序中所有的内存申请和释放操作都通过该模块进行，再通过该模块记录应用程序申请和释放的内存块的地址、时间等信息，来实现对内存使用状况的监视，从而完成应用程序内存泄露的检查。

按照图1实施本发明，首先经步骤1设置一个内存泄漏检查模块，并将其嵌入到应用程序中，并且定义一个内存块记录表来记录应用程序其他模块所申请的内存块和释放的内存块的信息，本例中的内存块记录表为哈希表。同时还设置一个内存块计数器，用于对哈希表中的内存块进行管理；

设置内存泄露检查模块的控制模块。在哈希表中记录的内存块的信息包括内存块的起始地址、内存块大小、起始时间等，还可以记录一些调试信息，如执行申请操作的代码所在的文件、行号等，以内存块的起始地址作为哈希表的关键字，由于同一进程申请的两个内存块的起始地址不可能相同，因此可以保证哈希表中的元素关键字不重复。当程序中的其它模块调用内存泄露检查模块申请内存时，首先调用操作系统提供的接口申请内存，若申请内存成功，则在哈希表中添加一条元素，记录该内存块的信息，并增加计数器的值。若应用程序中的其它模块调用内存泄露检查模块释放内存，则以内存块起始地址为关键字到哈希表中查找相应的记录，找到后删除该记录并减少相应的计数器的值。

步骤1设置的内存泄露检查模块内部逻辑结构参考图3。图3所示，在应用程序成功申请内存块后，通过内存泄露检查模块的申请内存接口在哈希表中插入一条内存块信息记录，同时更新计数器的值；在应用程序成功释放内存块后，通过内存泄露检查模块的释放内存接口在哈希表中删除一条相应的内存块信息记录，同时更新计数器的值。

在步骤1中设置的内存泄露检查模块和应用程序其它模块以及控制模块的关系参考图2。图2所示，应用程序的其它模块在进行内存块的申请和释放操作时，都要经过内存泄露检查模块管理，即在调用操作系统进行内存块的申请和释放时要由内存泄露检查模块管理申请和释放的内存块；内存泄露检查模块由控制模块控制，并将检查结果送给控制模块分析处理。

然后在步骤2根据应用程序的具体情况确定应用程序需要申请的内存块，以及所申请的不同内存块被占用的时间，或者可能被占用的时间。接

着运行应用程序，并在步骤3用哈希表记录应用程序申请内存块和释放内存块的信息；在需要进行内存泄露分析时，由控制模块在步骤4通过内存泄露检查模块确定的超过占用时间尚未释放的内存块，确定应用程序的内存泄露点。

为便于控制模块控制内存泄露检查模块进行应用程序内存泄露的检查，本例中在内存泄露检查模块另设一个状态字段来记录模块当前的状态，并据之进行相应的动作。在控制模块对内存泄漏模块进行控制进行内存泄漏检查时，通过向嵌入到应用程序中的内存泄露检查模块发送信号或消息，促使内存泄漏检测模块进行适合的状态迁移，执行内存泄漏检查操作。内存泄露检查模块的各个状态以及状态间的迁移关系如下：

1) 空闲状态：在此状态下，内存泄漏检查模块并不对内存申请和释放操作进行记录，这些操作都被透明地传递给操作系统，模块中的哈希表和计数器也处于清空状态。

2) 操作记录状态：在“空闲状态”下收到控制模块发来的“启动记录”信号时，进入此状态。在此状态下，内存泄漏检查模块按照前面的方法记录程序中的内存申请和释放操作。

3) 等待释放状态：在“操作记录状态”下收到到控制模块发来的“停止插入”信号时进入此状态。在该状态下，内存泄漏检查模块只对释放内存的操作进行记录，而不记录申请内存的操作。之所以要有这个阶段，是因为程序申请的内存块总是使用一段时间后才释放，为了完整的记录所申请的内存块的释放情况，防止出现误报，对释放操作的记录必须长于对申请操作的记录。需要说明的是，这个阶段的时间长度可以根据需要

进行调整，原则是要有足够的时间让应用程序释放在“操作记录状态”中所申请的内存。经过这个阶段还未释放的内存就认为是可能的内存泄漏。例如，如果知道应用程序中申请的某部分内存资源最多使用一个小时后就会释放，首先让内存泄漏检查模块在“操作记录状态”记录一些内存申请，然后进入“等待释放状态”，等待一个小时，若还有这部分的内存资源没有释放，则可以认为是内存泄漏。

4) 停止状态：在“等待释放状态”下收到控制模块发来的“停止”信号时进入此状态。在该状态下，内存泄漏检查也不记录内存申请和释放操作，但哈希表和计数器中还保留着状态2和状态3时记录的数据，可以根据需要对这些数据进行查询或让内存泄漏检查模块将数据输出到文件由控制模块进行分析处理。

一次完整的内存泄漏检查中，内存泄漏检查模块要依次经过上面的四种状态，其状态的迁移参考图4。从图4可以看到，通过控制模块的控制，可以让内存泄漏检查模块在任何状态下直接回到空闲状态。各个状态之间的切换由控制模块发送的信号或消息控制，一次检查结束后，就可以根据哈希表中记录的还未释放的内存块的信息来分析应用程序中可能产生泄漏的地方。

在内存泄漏检查模块的具体实现中，可以选择不同的哈希算法，并根据程序中内存申请操作的频繁度来调整哈希表的大小，以适应各种检测环境的需要。同时，控制模块在内存泄露检查时的各个状态的时长可以控制，以满足实际的控制需要。

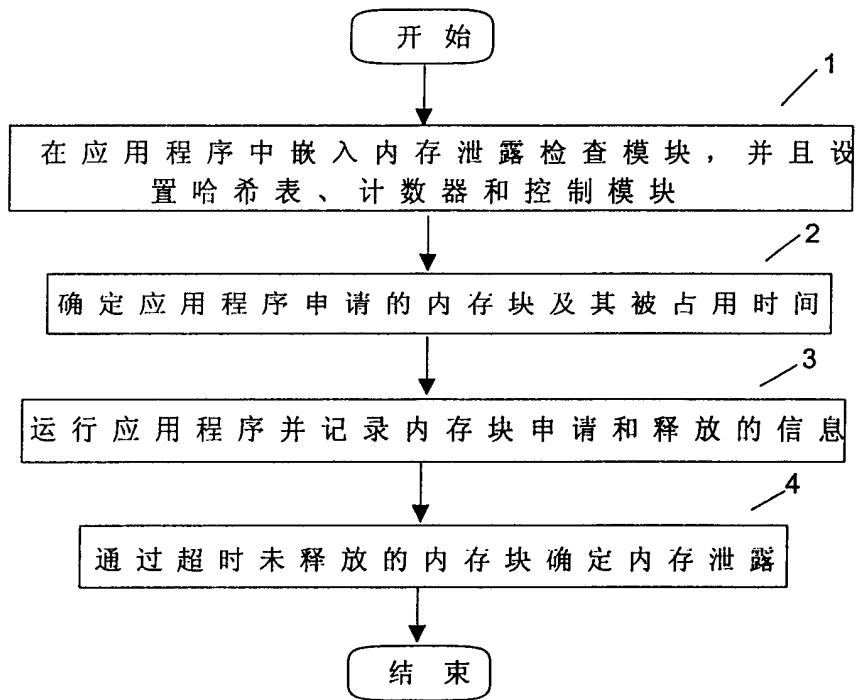


图1

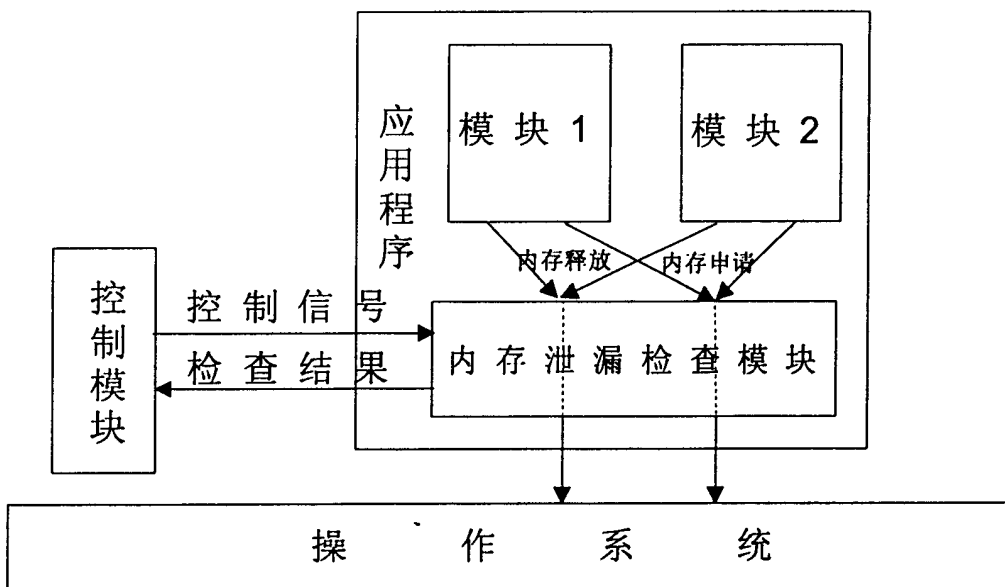


图2

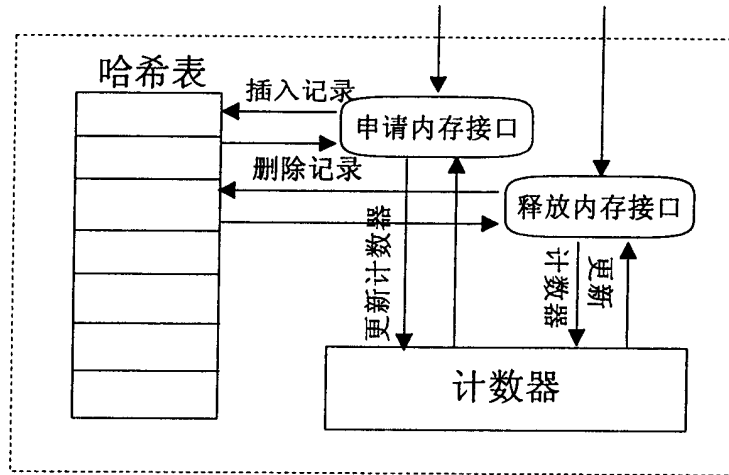


图3

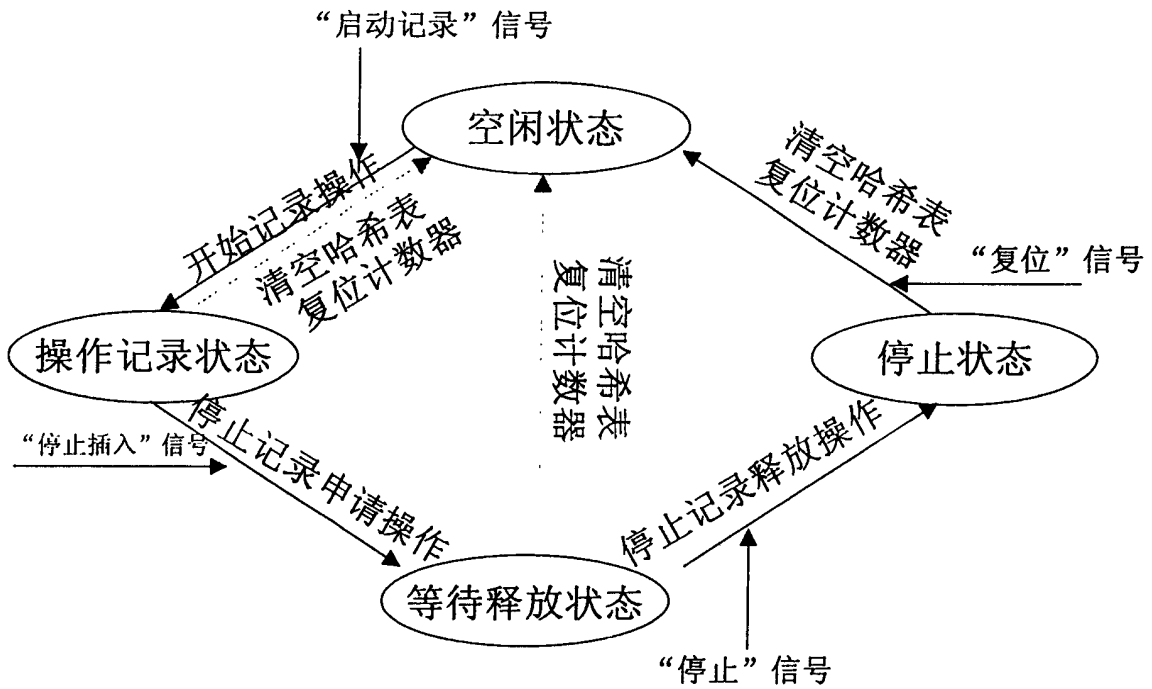


图4