

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 11/36 (2006.01)

G06F 9/46 (2006.01)



# [12] 发明专利说明书

专利号 ZL 200610099269.5

[45] 授权公告日 2009 年 4 月 15 日

[11] 授权公告号 CN 100478911C

[22] 申请日 2006.7.25

[21] 申请号 200610099269.5

[73] 专利权人 中兴通讯股份有限公司

地址 518057 广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦法律部

[72] 发明人 毛坚桓

[56] 参考文献

JP11 - 249909A 1999.9.17

CN1464397A 2003.12.31

CN1752936A 2006.3.29

审查员 李 俊

[74] 专利代理机构 北京安信方达知识产权代理有限公司

代理人 龙 洪 霍育栋

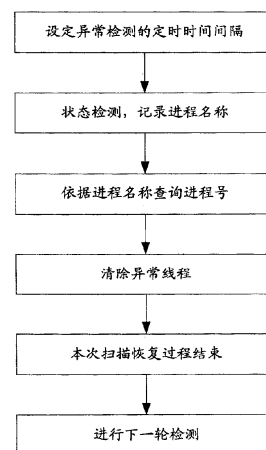
权利要求书 2 页 说明书 4 页 附图 1 页

[54] 发明名称

管理线程异常的防护方法

[57] 摘要

本发明公开了一种管理线程异常的防护方法，包括如下步骤：获取系统进程的状态信息，记录管理线程异常的进程名称；根据获取的该进程名称，查找该进程名称对应的进程号；清除该进程号对应的线程；创建该进程名称的进程。本发明完善了 POSIX 多线程进程的管理，使原来无人监管的管理线程得到了监控，避免了由于管理线程异常而出现的所在进程工作异常，从而影响用户使用的问题。



1、一种管理线程异常的防护方法，包括如下步骤：

- (1) 通过状态检测，获取系统进程的状态信息，记录管理线程异常的进程名称；
- (2) 根据获取的该进程名称，查找该进程名称对应的进程号；
- (3) 清除该进程号对应的线程；
- (4) 创建该进程名称的进程。

2、根据权利要求1所述的管理线程异常的防护方法，其特征在于，步骤(1)进一步包括：设定异常检测的定时时间间隔。

3、根据权利要求2所述的管理线程异常的防护方法，其特征在于，步骤(1)具体为：定时扫描proc目录下所有进程目录，利用linux系统的proc文件系统对应目录下的进程信息进行查询，记录所有管理线程状态为管理线程异常的进程名称。

4、根据权利要求3所述的管理线程异常的防护方法，其特征在于，步骤(1)中，proc目录下，linux系统的proc文件系统下的stat文件存放着进程的状态信息，所述管理线程异常的状态信息存放在所述stat文件中。

5、根据权利要求1所述的管理线程异常的防护方法，其特征在于，步骤(2)中，当管理线程异常的进程名称的数量为一个以上时，依次查找每个进程名称的所有进程号；步骤(3)中，依次清除每个进程号对应的线程；步骤(4)中，依次创建进程名称对应的进程。

6、根据权利要求2所述的管理线程异常的防护方法，其特征在于，进一步包括步骤(5)：等待定时时间间隔后，进行下一轮的检测。

7、根据权利要求6所述的管理线程异常的防护方法，其特征在于，步骤(1)中，在系统进程的当前状态信息中没有检测到管理线程异常的状态，

则执行步骤（5）。

8、根据权利要求3所述的管理线程异常的防护方法，其特征在于，步骤（1）中，所述proc文件系统的所有信息都会由内核定时更新。

9、根据权利要求2所述的管理线程异常的防护方法，其特征在于，步骤（1）中，定时时间间隔的范围是0~50秒。

## 管理线程异常的防护方法

### 技术领域

本发明涉及一种线程维护方法，具体说涉及一种在 POSIX（Portable Operating System Interface，可移植操作系统接口）线程库环境下、多线程进程的管理线程异常的防护方法。

### 背景技术

Linux 系统向管理员提供了非常好的方法，使他们可以在系统运行时更改内核，而不需要重新引导内核系统。这是通过 /proc 虚拟文件系统实现的。/proc 文件虚拟系统是一种内核和内核模块用来向进程（process）发送信息的机制（所以叫做 /proc）。这个伪文件系统可以和内核内部数据结构进行交互，可以用于获取运行中的进程的信息。在 /proc 中有一些编号的子目录，每个编号的目录对应一个进程标识 PID。这样，每一个运行中的进程 /proc 中都有一个用它的 PID 命名的目录。这些子目录中包含可以提供有关进程的状态和环境的重要细节信息的文件。

POSIX 线程库是 uclibc 库中提供的标准线程库，用来提供在各个满足 POSIX 标准的系统中可方便移植的通用线程库接口。该线程库主要提供了线程创建，销毁等线程管理的功能，使用该库接口函数编写的多线程程序，可以在任何其他 POSIX 操作系统上编译执行。

在该库的实现中，为了实现信号处理、线程创建和进程管理中其他部分的正确语义，引入了管理线程的概念。管理线程对用户透明，从层次上看，它位于主线程和用户创建的子线程之间。从用户角度来看，线程的层次结构是主线程——子线程的二层模式，但从系统角度来看，实际上已转化为主线程——管理线程——子线程的三层模式。管理线程不执行用户代码，只是集中处理来自其他线程（包括主线程）的创建线程、请求退出等请求，担负着整个进程管理的重大职责。

由于实现方面的种种原因，使管理线程的权力过高，同时又不受任何其他进程监管，这便导致了目前 POSIX 线程库的如下缺陷：一旦管理线程异常（ZOMBIE），所在进程的所有工作可能也会中止，但是整个系统却浑然不知。如果在产品中出现这种情况，必将会影响到用户的正常使用。

## 发明内容

本发明所解决的技术问题是提供一种管理线程异常的防护方法，能够在 POSIX 线程库环境下避免由于管理线程变成 ZOMBIE 状态而引起进程异常。

本发明的技术方案如下：

管理线程异常的防护方法包括如下步骤：

- （1）通过状态检测，获取系统进程的状态信息，记录管理线程异常的进程名称；
- （2）根据获取的该进程名称，查找该进程名称对应的进程号；
- （3）清除该进程号对应的线程；
- （4）创建该进程名称的进程。

进一步，步骤（1）进一步包括：设定异常检测的定时时间间隔。

进一步，步骤（1）具体为：定时扫描 proc 目录下所有进程目录，利用 linux 系统的 proc 文件系统对应目录下的进程信息进行查询，记录所有管理线程状态为管理线程异常的进程名称。

进一步，步骤（1）中，proc 目录下，linux 系统的 proc 文件系统下的 stat 文件存放着进程的状态信息，所述管理线程异常的状态信息存放在所述 stat 文件中。

进一步，步骤（2）中，当管理线程异常的进程名称的数量为一个以上时，依次查找每个进程名称的所有进程号；步骤（3）中，依次清除每个进程号对应的线程；步骤（4）中，依次创建进程名称对应的进程。

进一步包括步骤（5）：等待定时时间间隔后，进行下一轮的检测。

进一步，步骤（1）中，在系统进程的当前状态信息中没有检测到管理

线程异常的状态，则执行步骤（5）。

进一步，步骤（1）中，所述 `proc` 文件系统的所有信息都会由内核定时更新。

进一步，步骤（1）中，定时时间间隔的范围是 0~50 秒。

本发明所提供的管理线程异常的防护方法可以保证：

1、 在用户指定的定时间隔内，可以成功检测出所有管理线程变成 ZOMBIE 状态的进程。

2、 对于检测出上述异常的多线程进程，可以保证及时将这些进程从异常恢复过来。

本发明完善了 POSIX 多线程进程的管理，使原来无人监管的管理线程得到了监控，避免了由于管理线程异常而出现的所在进程工作异常，从而影响用户使用的问题。

## 附图说明

图 1 是本发明优选实施例的流程图。

## 具体实施方式

下面参照附图对本发明的优选实施例作详细描述。

本发明方法的技术方案从整体上来说，包括管理线程异常状态检测和异常后的恢复两方面。

首先，管理线程异常（ZOMBIE）状态的检测。

参照图 1 所示，状态的检测可以定时进行，定时间隔可以根据需要进行设定。查询进程状态时，可凭借 linux 系统自带的 `/proc` 文件系统，其下的数字目录即为该系统上所有的运行进程，每个数字目录下还包括一些文件，表示了这个进程的所有相关信息，其中 `stat` 文件存放着进程的状态信息。由于 `/proc` 文件系统的所有信息都会由内核定时更新，故通过对 `/proc` 中信息的查询，可以即时获得系统中进程的当前状态信息，只需要把状态为 ZOMBIE 的进程名称记录下来，就完成了状态检测的过程。

其次，异常（ZOMBIE）后的恢复。

异常后的恢复包括清除原来进程和创建新进程两步。

在管理线程异常（ZOMBIE）后，由于管理线程层次和作用的特殊性，采用通常的清除主线程的方法是清除不了整个进程的所有线程的。要把进程清除干净，需要根据获得的进程名称，在/proc 文件系统中依次查找每个进程名称的所有线程的进程号，逐个清除进程号对应的线程，从而达到把整个进程清除干净的目的。接着，再次创建相应进程，从而将异常的进程恢复。

以下结合一个例子，对本发明涉及的管理线程 ZOMBIE 状态防护的整个流程进行详细的阐述：

假设系统运行了一段时间后，系统中有两个多线程进程的管理线程状态先后异常（ZOMBIE）。设定异常检测的定时间间隔为 30 秒，则在第一个进程管理线程异常后 30 秒内执行的第一次进程状态扫描过程中，将检测到有两个管理线程处于 ZOMBIE 状态，并记录这两个进程名称。假设这两个进程名称为“proc1”、“proc2”。

再次对/proc 文件系统中的进程的当前状态进行查询，首先查询名称为“proc1”的所有进程号，将会找到和异常的管理线程属于同一进程的所有线程，将找到的所有线程清除后，则“proc1”进程清除干净。然后重新创建“proc1”的进程，使其恢复正常执行。

接着，再处理“proc2”的进程。查询名称为“proc2”的所有进程号，找到和异常的管理线程属于同一进程的所有进程号，将找到的所有线程清除后，则“proc2”进程清除干净。然后重新创建“proc2”的进程，使其恢复正常执行。

等到两个进程处理完毕，本次扫描检测和恢复过程结束，将等待 30 秒进行下一轮的监测。

假设系统运行了一段时间后，系统中发现多个管理线程状态异常（ZOMBIE），则对多个进程名称对应的进程依次进行记录、清除和恢复，从而将异常的进程恢复。

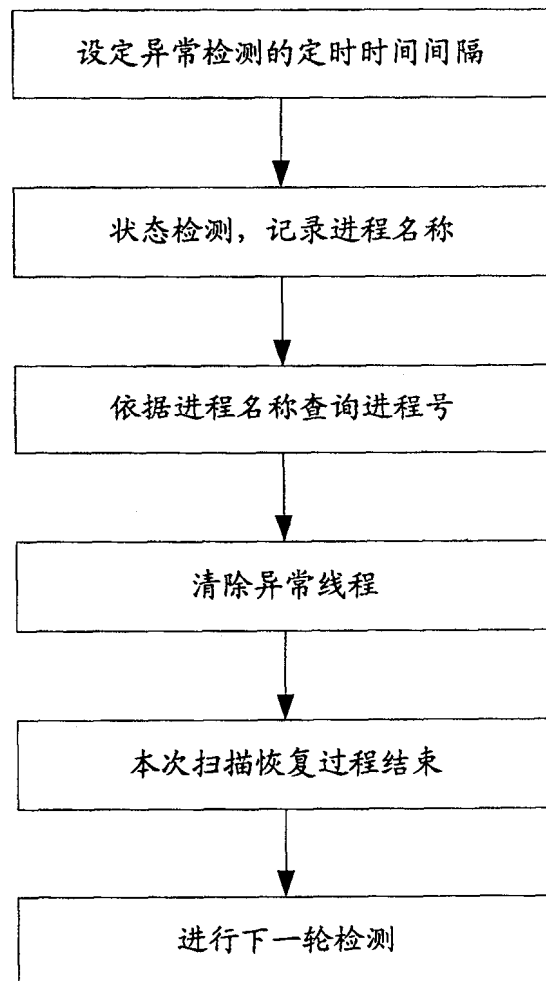


图 1