(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112749020 A
(43)申请公布日 2021.05.04

(21)申请号 202011638727.4

(22)申请日 2020.12.31

(71)申请人 普华基础软件股份有限公司
地址 200030 上海市徐汇区虹漕路448号1幢12楼

(72)发明人 骆政强 冯建 邢章威

(74)专利代理机构 上海申新律师事务所 31272
代理人 竺路玲

(51)Int.Cl.
G06F 9/52(2006.01)

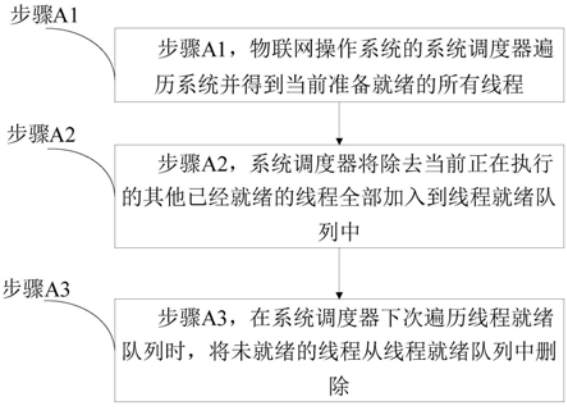
权利要求书2页 说明书6页 附图2页

(54)发明名称

一种物联网操作系统的微内核优化方法

(57)摘要

本发明涉及操作系统技术领域，公开了一种物联网操作系统的微内核优化方法。于物联网中设置有多个物联网设备和服务器设备，所述物联网操作系统被应用于各个所述物联网设备以及所述服务器设备中；微内核优化方法包括一系统接口的合并方法，一线程就绪队列的延迟更新过程，指定线程调用的过程，一线程锁的释放过程，一权能字委托的过程以及一撤销权能字委托的过程。本发明的技术方案有益效果在于：提供一种物联网操作系统的微内核优化方法，能够在实现线程管理、地址空间管理、线程间通信、中断处理等功能的同时，兼顾底层所有功能且灵活性的微内核架构设计，避免不同模块之间的循环依赖性。



CN 112749020 A

1.一种物联网操作系统的微内核优化方法,其特征在于,于物联网中设置有多个物联网设备和服务器设备,所述物联网操作系统被应用于各个所述物联网设备以及所述服务器设备中;

所述微内核优化方法包括一系统接口的合并方法,具体包括:

于所述物联网操作系统中,将负责发送信息的第一系统接口和负责接收信息的第二系统接口均关联到同一个线程中,使得所述物联网设备和所述服务器设备进行一次通信的过程中,每端的所述物联网操作系统分别仅执行两次系统调用。

2.根据权利要求1中所述的一种物联网操作系统的微内核优化方法,其特征在于,还包括一线程就绪队列的延迟更新过程,具体包括:

步骤A1,所述物联网操作系统的系统调度器遍历系统并得到当前准备就绪的所有线程;

步骤A2,所述系统调度器将除去当前正在执行的其他已经就绪的线程全部加入到所述线程就绪队列中;

步骤A3,在所述系统调度器下次遍历所述线程就绪队列时,将未就绪的线程从所述线程就绪队列中删除。

3.根据权利要求1中所述的一种物联网操作系统的微内核优化方法,其特征在于,还包括指定线程调用的过程,具体包括:

在采用一第一类线程执行数据发送的任务完毕后,所述物联网操作系统的系统调度器直接指定下一个要被调度的线程为一第二类线程,所述第二类线程对应于所述第一类线程,并用于执行数据接收的任务。

4.根据权利要求1中所述的一种物联网操作系统的微内核优化方法,其特征在于,还包括一线程锁的释放过程,具体包括:

步骤B1,在具有线程锁的一第一线程运行的过程中,所述物联网操作系统接收到一第二线程的线程锁获取请求;

步骤B2,所述物联网操作系统判断所述第二线程是否符合一预设的线程锁抢占标准:

当所述第二线程符合所述线程锁抢占标准时,所述物联网操作系统将所述线程锁由所述第一线程转移至所述第二线程,随后返回所述步骤B1;

当所述第二线程不符合所述线程锁抢占标准时,所述物联网操作系统将所述第二线程移至一帮助线程堆栈的顶部,并保持所述第二线程始终处于就绪状态;

步骤B3,所述物联网操作系统将处于所述帮助线程堆栈的顶部的线程的线程资源转移至正在运行的所述第一线程,直至所述第一线程执行完毕并释放所述线程锁后,将所述线程锁转移至处于所述帮助线程堆栈的顶部的线程。

5.根据权利要求4所述的一种物联网操作系统的微内核优化方法,其特征在于,所述步骤B2中,预设的所述线程锁抢占标准为:所述第二线程的优先级大于或等于所述第一线程的优先级。

6.根据权利要求4所述的一种物联网操作系统的微内核优化方法,其特征在于,所述步骤B3中,所述线程资源包括线程的调度上下文、时间片以及优先级。

7.根据权利要求1中所述的一种物联网操作系统的微内核优化方法,其特征在于,于所述物联网操作系统中预先设定包括用于指示不同权限的权能字的权能字列表,所述物联网

操作系统预先为不同的线程分配所述权能字列表中的不同的所述权能字,所述权能字被分别保存于一存储空间内;

于所述物联网操作系统的内存空间内设置一对象地址空间,并于所述对象地址空间内设置多个间接指向对象,每个所述间接指向对象分别指向不同的所述权能字;

则所述微内核优化方法中还包括一权能字委托的过程,具体包括:

将所述间接指向对象设置为指向一个所述权能字,并将所述间接指向对象赋予一个所述线程,从而将被指向的所述权能字委托给所述线程。

8.根据权利要求7所述的一种物联网操作系统的微内核优化方法,其特征在于,还包括一撤销权能字委托的过程,具体包括:

将已被赋予所述线程的所述间接指向对象设定为指向一个空的对象,从而撤销委托给所述线程的所述权能字。

一种物联网操作系统的微内核优化方法

技术领域

[0001] 本发明涉及操作系统技术领域,公开了一种物联网操作系统的微内核优化方法。

背景技术

[0002] 现有的物联网操作系统的微内核已经模块化,模块之间也使用了明确定义的接口。其源代码已经分为子系统、模块和子模块。不同的子系统几乎是微内核的完全独立部分,例如用于简单内存管理的库,简化的C库或主内核映像本身。每个子系统由一个或多个封装逻辑单元的模块组成。例如,线程模块包含处理UHomeOS线程所需的数据结构和方法。这些模块的接口定义良好,隐藏了实现细节。同样,模块可以聚合一个或多个子模块,这些子模块用于进一步将模块细分为更小的逻辑块。

[0003] 传统内核同步机制,例如信号量、自选锁机制,存在线程死锁和优先级反转等问题,尤其对于时间限制要求的消费类与工业类物联网系统实时特性,导致影响更为严重,因为这些内核锁同步机制,在多处理器中会造成线程对临界区竞争热点,造成系统性能与实时性的大幅下降。影响服务响应时间因素包括中断延迟、中断处理延迟、调度器延迟、任务调度延迟,线程间各种同步、互斥与通信措施的延迟,优先级反转现象带来不可预测的延迟。

发明内容

[0004] 针对现有技术中存在的上述问题,现提供一种物联网操作系统的微内核优化方法,于物联网中设置有多个物联网设备和服务器设备,所述物联网操作系统被应用于各个所述物联网设备以及所述服务器设备中;

[0005] 所述微内核优化方法包括一系统接口的合并方法,具体包括:

[0006] 于所述物联网操作系统中,将负责发送信息的第一系统接口和负责接收信息的第二系统接口均关联到同一个线程中,使得所述物联网设备和所述服务器设备进行一次通信的过程中,每端的所述物联网操作系统分别仅执行两次系统调用。

[0007] 优选的,还包括一线程就绪队列的延迟更新过程,具体包括:

[0008] 步骤A1,所述物联网操作系统的系统调度器遍历系统并得到当前准备就绪的所有线程;

[0009] 步骤A2,所述系统调度器将除去当前正在执行的其他已经就绪的线程全部加入到所述线程就绪队列中;

[0010] 步骤A3,在所述系统调度器下次遍历所述线程就绪队列时,将未就绪的线程从所述线程就绪队列中删除。

[0011] 优选的,还包括指定线程调用的过程,具体包括:

[0012] 在采用一第一类线程执行数据发送的任务完毕后,所述物联网操作系统的系统调度器直接指定下一个要被调度的线程为一第二类线程,所述第二类线程对应于所述第一类线程,并用于执行数据接收的任务。

[0013] 优选的，还包括一线程锁的释放过程，具体包括：

[0014] 步骤B1，在具有线程锁的一第一线程运行的过程中，所述物联网操作系统接收到一第二线程的线程锁获取请求；

[0015] 步骤B2，所述物联网操作系统判断所述第二线程是否符合一预设的线程锁抢占标准：

[0016] 当所述第二线程符合所述线程锁抢占标准时，所述物联网操作系统将所述线程锁由所述第一线程转移至所述第二线程，随后返回所述步骤B1；

[0017] 当所述第二线程不符合所述线程锁抢占标准时，所述物联网操作系统将所述第二线程移至一帮助线程堆栈的顶部，并保持所述第二线程始终处于就绪状态；

[0018] 步骤B3，所述物联网操作系统将处于所述帮助线程堆栈的顶部的线程的线程资源转移至正在运行的所述第一线程，直至所述第一线程执行完毕并释放所述线程锁后，将所述线程锁转移至处于所述帮助线程堆栈的顶部的线程。

[0019] 优选的，所述步骤B2中，预设的所述线程锁抢占标准为：所述第二线程的优先级大于或等于所述第一线程的优先级。

[0020] 优选的，所述步骤B3中，所述线程资源包括线程的调度上下文、时间片以及优先级。

[0021] 优选的，于所述物联网操作系统中预先设定包括用于指示不同权限的权能字的权能字列表，所述物联网操作系统预先为不同的线程分配所述权能字列表中的不同的所述权能字，所述权能字被分别保存于一存储空间内；

[0022] 于所述物联网操作系统的内存空间内设置一对象地址空间，并于所述对象地址空间内设置多个间接指向对象，每个所述间接指向对象分别指向不同的所述权能字；

[0023] 则所述微内核优化方法中还包括一权能字委托的过程，具体包括：

[0024] 将所述间接指向对象设置为指向一个所述权能字，并将所述间接指向对象赋予一个所述线程，从而将被指向的所述权能字委托给所述线程。

[0025] 优选的，还包括一撤销权能字委托的过程，具体包括：

[0026] 将已被赋予所述线程的所述间接指向对象设定为指向一个空的对象，从而撤销委托给所述线程的所述权能字。

[0027] 本发明的技术方案有益效果在于：提供一种物联网操作系统的微内核优化方法，能够在实现线程管理、地址空间管理、线程间通信、中断处理等功能的同时，兼顾底层所有功能且灵活性的微内核架构设计，避免不同模块之间的循环依赖性。

附图说明

[0028] 参考所附附图，以更加充分的描述本发明的实施例。然而，所附附图仅用于说明和阐述，并不构成对本发明范围的限制。

[0029] 图1为现有技术中，微内核通信的结构示意图；

[0030] 图2为本发明的优选实施方式中，延迟更新过程的流程示意图；

[0031] 图3为本发明的优选实施方式中，线程锁的释放过程的流程示意图。

具体实施方式

[0032] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有作出创造性劳动的前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0033] 需要说明的是，在不冲突的情况下，本发明中的实施例及实施例中的特征。

[0034] 现有技术中，针对微内核IPC(Inter Process Communication，进程间通信)可如图1所示，包括网络子系统01、驱动服务02、微内核03以及硬件04，当网络子系统01与驱动服务02进行IPC通信时，4次内核陷入与退出和2次上下文切换，而内核只需要调用内核函数，由此可见，进程间通信影响到微内核IPC性能的高效与否。

[0035] 并且，进程间通信IPC通常分为同步与异步两种方式，同步IPC是发送者阻塞等待接受方接受信息，一般发送信息不进行缓冲，发送信息由发送者直接传递给接收者。异步IPC是发送者不必等待接受者接受信息，发送者继续执行。一般带有发送缓冲区，如果接收者到来，把缓冲区信息传送给接收者。同步IPC优势简单而快效率高，不太容易遭受拒绝式攻击。

[0036] 本发明提供一种物联网操作系统的微内核优化方法，于物联网中设置有多个物联网设备和服务器设备，物联网操作系统被应用于各个物联网设备以及服务器设备中；

[0037] 微内核优化方法包括一系统接口的合并方法，具体包括：

[0038] 于物联网操作系统中，将负责发送信息的第一系统接口和负责接收信息的第二系统接口均关联到同一个线程中，使得物联网设备和服务器设备进行一次通信的过程中，每端的物联网操作系统分别仅执行两次系统调用。

[0039] 具体地，本发明采用了多服务器的系统架构设计模型，即同步的客户端/服务器系统模型，可调用第一系统接口call和第二系统接口serve，通过第一系统接口call和第二系统接口serve，将发送的消息和接受的消息绑定到同一个线程对象，客户端第一系统接口call执行发送与接受操作，服务器系统调用第二系统接口serve执行应答与接受操作，在IPC实现机制上，不仅支持独立发送与接受操作处理，同时实现应答与接受合并处理操作。由此可见，应用于本发明提供的合并方法，能够减少IPC系统调用次数，即每个RPC仅使用2个系统调用而不是每个4个系统调用的实现，从而解决现有技术中IPC调用过程非常耗时的问题。

[0040] 本发明优选的实施方式中，还包括一线程就绪队列的延迟更新过程，如图2所示，具体包括：

[0041] 步骤A1，物联网操作系统的系统调度器遍历系统并得到当前准备就绪的所有线程；

[0042] 步骤A2，系统调度器将除去当前正在执行的其他已经就绪的线程全部加入到线程就绪队列中；

[0043] 步骤A3，在系统调度器下次遍历线程就绪队列时，将未就绪的线程从线程就绪队列中删除。

[0044] 具体地，在客户端/服务器模型中，客户端调用请求第一系统接口call发送到服务器，然后阻塞并等待答复。服务器通常调用第二系统接口serve阻塞，直到有请求进入，然后

处理该请求，发回答复并再次阻塞。对于调度器的就绪队列来说，线程就绪就要排队就绪队列中，这样才能保持就绪队列一致性，那么客户端/服务器模型需要对该就绪队列进行多次更新，通常实现该操作的现有技术包括：(1)客户端发送请求后，客户端进入阻塞并等待，不再准备就绪，必须从就绪队列出队；(2)服务器收到请求后，服务器进入就绪阶段，必须进入就绪队列；(3)服务器发回答复后，服务器再次阻塞，必须从就绪队列出队；(4)客户端收到答复后，客户端将再次准备就绪，必须进入就绪队列。

[0045] 然而上述技术均会导致就绪队列的频繁操作，进而导致微内核IPC性能严重下降，本发明，于UHomeOS微内核中，通过线程就绪队列的延迟更新过程，控制除了当前正在执行的线程外的所有就绪线程排入就绪队列，未就绪的线程在阻塞时不会从就绪队列中删除，而是在下次调度程序遍历就绪队列时将其从就绪队列中删除，能够减少就绪队列的操作过程，提高IPC性能。

[0046] 本发明优选的实施方式中，还包括指定线程调用的过程，具体包括：

[0047] 在采用一第一类线程执行数据发送的任务完毕后，物联网操作系统的系统调度器直接指定下一个要被调度的线程为一第二类线程，第二类线程对应于第一类线程，并用于执行数据接收的任务。

[0048] 具体地，对于客户端/服务器模型中，在IPC通信第一种情况中，双方就绪情况下，不同于现有技术，控制调用系统调度器去选择下一个调度线程，而是将控制调度器直接切换到通信对方上下文，为了加快通信对方执行速度，并向其捐赠当前剩余时间片，尽快完成其接受任务。具体地实现过程可包括指定线程调用的过程，就是发送数据的线程发送数据后可直接指定下一个调度的线程为接受线程，这时接受数据的线程可以直接被调度，以便接受数据。通过此处直接进行线程切换，有针对性进行线程的调度切换，能够显著地改善通信双方等待时间，提高通信对方的IPC性能。

[0049] 本发明优选的实施方式中，还包括一线程锁的释放过程，如图3所示，具体包括：

[0050] 步骤B1，在具有线程锁的一第一线程运行的过程中，物联网操作系统接收到一第二线程的线程锁获取请求；

[0051] 步骤B2，物联网操作系统判断第二线程是否符合一预设的线程锁抢占标准：

[0052] 当第二线程符合线程锁抢占标准时，物联网操作系统将线程锁由第一线程转移至第二线程，随后返回步骤B1；

[0053] 当第二线程不符合线程锁抢占标准时，物联网操作系统将第二线程移至一帮助线程堆栈的顶部，并保持第二线程始终处于就绪状态；

[0054] 步骤B3，物联网操作系统将处于帮助线程堆栈的顶部的线程的线程资源转移至正在运行的第一线程，直至第一线程执行完毕并释放线程锁后，将线程锁转移至处于帮助线程堆栈的顶部的线程。

[0055] 本发明优选的实施方式中，步骤B2中，预设的线程锁抢占标准为：第二线程的优先级大于或等于第一线程的优先级。

[0056] 本发明优选的实施方式中，步骤B3中，线程资源包括线程的调度上下文、时间片以及优先级。

[0057] 具体地，应用于上述所述的释放过程，本发明可实现免等待锁。免等待锁是非阻塞的同步机制，在访问临界区之前如果线程检测的冲突，不是阻塞等待临界区内的线程操作

完成，而是继续运行帮助临界区内线程完成操作，然后在进行临界区操作，保证所有线程都能在有限的时间完成临界区的操作，也使系统实时性得到明显提高。但是由于免等待锁算法复杂性较高，目前很少操作系统实现这种同步机制锁。在本发明中，结合UHomeOS线程调度机制特性，通过步骤B1至B3，通过时间片捐赠、优先级继承等方式进行线程资源的转移，能够大大简化免等待锁的实现流程，实现无优先级反转问题的免等待锁的机制。

[0058] 一具体实施例中，如果第二线程A要获取当前由第一线程B持有的锁，第一线程B已准备就绪运行，此时必须满足第二线程A的优先级等于或大于第一线程B的优先级，否则它将不能抢占第一线程B线程运行。第二线程A无法立即获取锁，所以将其置于该锁的帮助线程堆栈的顶部，并帮助第一线程B释放锁。通过将第二线程A其调度上下文以及时间和优先级捐赠给第一线程B。帮助线程堆栈中的线程保持就绪状态，并在每次调度程序重新激活第二线程A时，继续捐赠给锁所有者第一线程B，直到第一线程B释放锁，并将其锁传递给帮助线程堆栈顶部的第二线程A。

[0059] 由上述可见，不仅能够实现免等待锁机制，并且，在实际运行过程中，减少了临界区同步性能开销，更好的解决了物联网系统同步锁的实时性问题。

[0060] 本发明优选的实施方式中，于物联网操作系统中预先设定包括用于指示不同权限的权能字的权能字列表，物联网操作系统预先为不同的线程分配权能字列表中的不同的权能字，权能字被分别保存于一存储空间内；

[0061] 于物联网操作系统的内存空间内设置一对象地址空间，并于对象地址空间内设置多个间接指向对象，每个间接指向对象分别指向不同的权能字；

[0062] 则微内核优化方法中还包括一权能字委托的过程，具体包括：

[0063] 将间接指向对象设置为指向一个权能字，并将间接指向对象赋予一个线程，从而将被指向的权能字委托给线程。

[0064] 本发明优选的实施方式中，还包括一撤销权能字委托的过程，具体包括：

[0065] 将已被赋予线程的间接指向对象设定为指向一个空的对象，从而撤销委托给线程的权能字。

[0066] 具体地，权能字是授予拥有者访问计算机系统中的实体或对象的权限的令牌，票证或密钥，通常包括标识符和访问权限，前者是名称，例如线程，任务，线程等，后者是读取，写入，执行，访问等权限，权能字表示对标识符对象的操作权限，权能字委托是主体将其权能字委托给其他主体，撤消是主体撤消它委派给其他主体的权能字。

[0067] 针对委托过程，本发明通过预先设定权能字，将权能字委托操作授予客户与服务器，客户可以使用初始权能字与服务器通信，服务器在其对象地址空间查询权能字，保证了通信安全可靠。

[0068] 针对撤销过程，现有技术实现权能字权限撤销通常是十分困难的，而本发明，在对象地址空间中，通过间接指向对象，而不是存放对象本身，只须设置间接对象执行一个空的对象，非常有效地解决权能字失效问题，

[0069] 由上述可见，应用于本发明的技术方案，从安全性的角度来看，初始权能字，即对象的访问权限，完全定义了本发明应用程序的执行环境所需权限。并且，通过设置任务的权能字可以定义强制性安全策略，从而解决了代码的沙箱问题，当对应于本发明的程序实际运行时，设置任务只包含它所需要的权能字，就能实现物联网操作系统的操作安全。

[0070] 进一步地，需要说明的是，本发明可用利用PendSV中断来完成任务切换：在每个中断处理函数中都内嵌了触发PendSV中断的代码，其中也包括systick(ktimer)中断。在pendsv_handler中断处理函数中，分别获取schedule_in_irq()函数调用前后的时间差即可计算得知任务切换的时间。

[0071] 更进一步地，应用于本发明，可以实现快速任务切换，在空载时，最小切换时间为6us，最大切换时间为14us，平均切换时间为6.67us，而在满载时，最小切换时间为9us，最大切换时间为16us，平均切换时间为9.35us。

[0072] 更进一步地，应用于本发明，可以实现中断的快速响应，在空载时，最小响应时间为6us，最大响应时间为10us，平均响应时间为7.70us，而在满载时，最小响应时间为8us，最大响应时间为14us，平均切换时间为10.40us。

[0073] 以上仅为本发明较佳的实施例，并非因此限制本发明的实施方式及保护范围，对于本领域技术人员而言，应当能够意识到凡运用本发明说明书及图示内容所作出的等同替换和显而易见的变化所得到的方案，均应当包含在本发明的保护范围内。

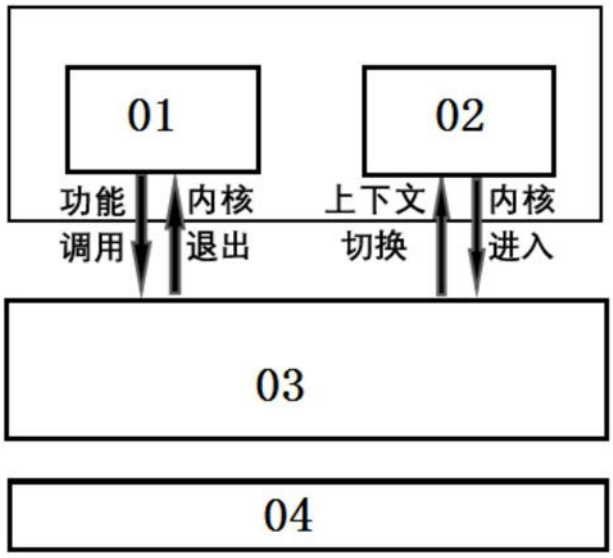

01
02
功能
调用
内核
退出
上下文
切换
内核
进入
03
04

图1

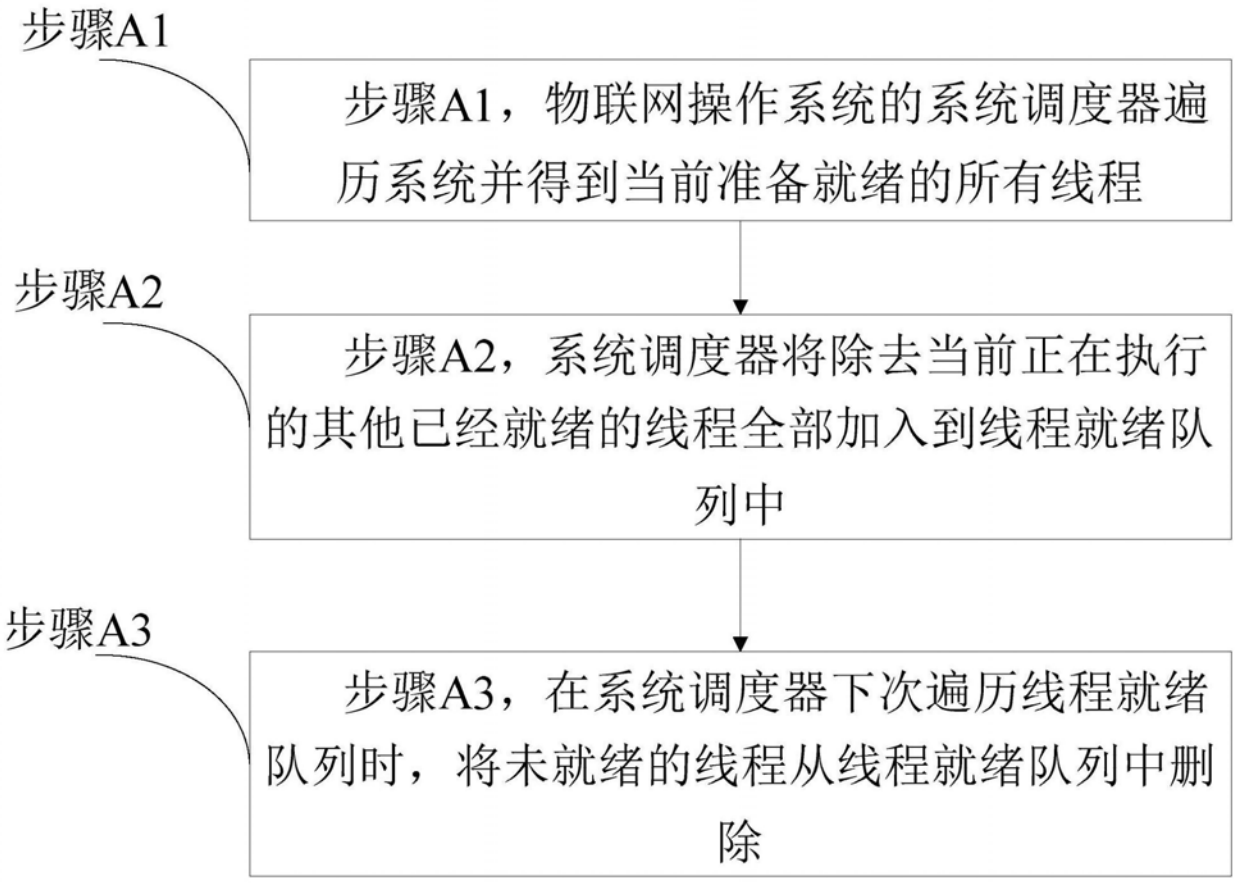

步骤A1
步骤A1，物联网操作系统的系统调度器遍历系统并得到当前准备就绪的所有线程
步骤A2
步骤A2，系统调度器将除去当前正在执行的其他已经就绪的线程全部加入到线程就绪队列中
步骤A3
步骤A3，在系统调度器下次遍历线程就绪队列时，将未就绪的线程从线程就绪队列中删除

图2

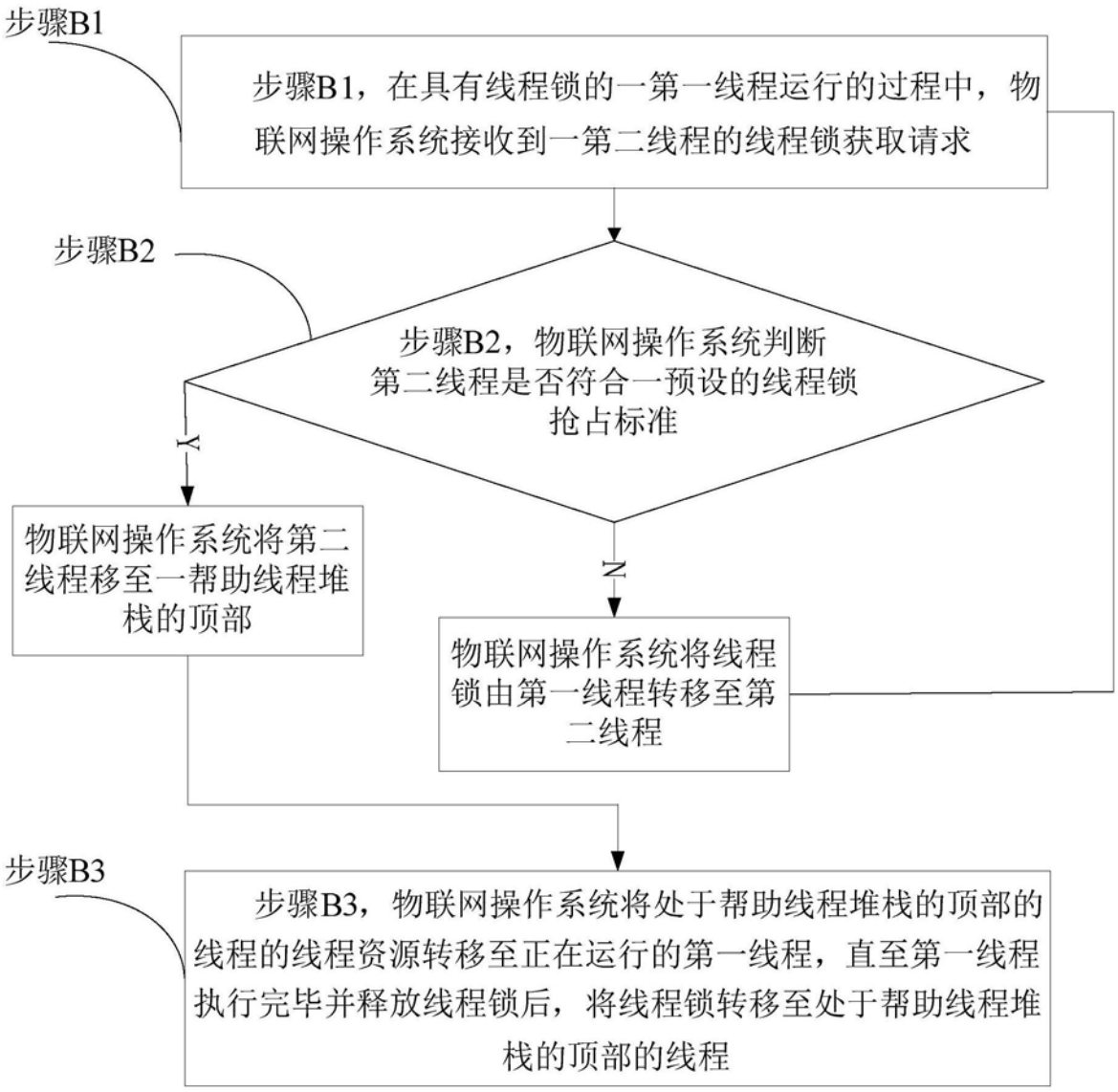

步骤B1
步骤B1，在具有线程锁的一第一线程运行的过程中，物联网操作系统接收到一第二线程的线程锁获取请求
步骤B2
步骤B2，物联网操作系统判断第二线程是否符合一预设的线程锁抢占标准
Y
N
物联网操作系统将第二线程移至一帮助线程堆栈的顶部
物联网操作系统将线程锁由第一线程转移至第二线程
步骤B3
步骤B3，物联网操作系统将处于帮助线程堆栈的顶部的线程的线程资源转移至正在运行的第一线程，直至第一线程执行完毕并释放线程锁后，将线程锁转移至处于帮助线程堆栈的顶部的线程

图3