(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2019 年 1 月 3 日 (03.01.2019)

WIPO | PCT

(10) 国际公布号
WO 2019/001332 A1

(51) 国际专利分类号:
G06F 9/48 (2006.01)

(21) 国际申请号: PCT/CN2018/092145

(22) 国际申请日: 2018 年 6 月 21 日 (21.06.2018)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201710501881.9 2017年6月27日 (27.06.2017) CN

(71) 申请人: 阿里巴巴集团控股有限公司 (ALIBABA GROUP HOLDING LIMITED) [—/CN]; 开曼群岛大开曼资本大厦一座四层 847 号邮箱, Grand Cayman (KY)。

(72) 发明人: 陈凌君 (CHEN, Lingjun); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。 王斌 (WANG, Bin); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。 朱亮亮 (ZHU, Liangliang); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。 曾旭 (ZENG, Xu); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。 刘自龙 (LIU, Zilong); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。 蔡俊杰 (CAI, Junjie); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。

(74) 代理人: 北京三友知识产权代理有限公司 (BEIJING SANYOU INTELLECTUAL PROPERTY AGENCY LTD.); 中国北京市金

(54) **Title:** TASK PRIORITY PROCESSING METHOD AND PROCESSING DEVICE

(54) **发明名称**: 任务的优先级处理方法和处理装置

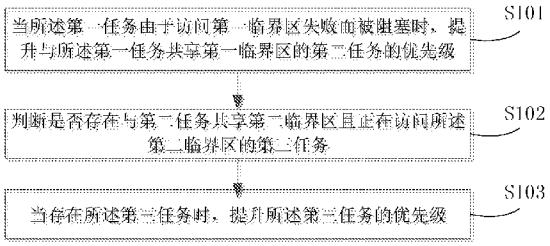


图 1

S101 WHEN A FIRST TASK IS BLOCKED DUE TO THE FAILURE TO ACCESS A FIRST CRITICAL AREA, THEN PROMOTE THE PRIORITY OF A SECOND TASK WHICH SHARES THE FIRST CRITICAL AREA WITH THE FIRST TASK

S102 DETERMINE WHETHER A THIRD TASK EXISTS THAT SHARES A SECOND CRITICAL AREA WITH THE SECOND TASK AND WHICH IS CURRENTLY ACCESSING THE SECOND CRITICAL AREA

S103 IF THE THIRD TASK DOES EXIST, THEN PROMOTE THE PRIORITY OF THE THIRD TASK

(57) **Abstract:** Disclosed in the present application are a task priority processing method and a processing device. The tasks comprise a first task, a second task and a third task, wherein the first task has a priority higher than that of the second task and the third task; The method comprises: when a first task is blocked due to a failure to access a first critical area, then promote the priority of a second task which shares the first critical area with the first task and which is currently accessing the first critical area; determine whether a third task exists that shares a second critical area with the second task and which is currently accessing the second critical area; if the third task does exist, then promote the priority of the third task. The solution provided in the embodiments of the present application prevents a third task which has a lower priority from delaying the implementation of a second task, thus preventing a priority reversion due to a delay in implementing a first task which has a higher priority.

WO 2019/001332 A1

[见续页]

融街35号国际企业大厦A座16层, Beijing 100033 (CN)。

(81) 指定国(除另有指明，要求每一种可提供的国家保护)：AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国(除另有指明，要求每一种可提供的地区保护)：ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

(57) 摘要：本申请公开了一种任务的优先级处理方法和处理装置。所述任务包括第一任务、第二任务和第三任务，所述的第一任务的优先级高于所述第二任务和所述第三任务；所述方法包括：当所述第一任务由于访问第一临界区失败而被阻塞时，提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级；判断是否存在与第二任务共享第二临界区且正在访问所述第二临界区的第三任务；当存在所述第三任务时，提升所述第三任务的优先级。本申请实施例提供的方案避免了优先级低的第三任务拖延第二任务的执行，从而避免优先级高的第一任务延迟执行造成的优先级反转。

任务的优先级处理方法和处理装置

本申请要求 2017 年 06 月 27 日递交的申请号为 201710501881.9、发明名称为“任务的优先级处理方法和处理装置”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及计算机信息处理领域，特别是涉及一种任务的优先级处理方法和处理装置。

背景技术

在多任务并发的环境里，从概念上看，有多个任务在同时执行，具体到单个 CPU 级别，实际上任何时刻只能有一个任务处于执行状态；因此操作系统需要决定哪个任务执行，哪些任务等待，也就是任务的调度。

优先级反转是当一个高优先级任务通过信号量机制访问共享资源时，该信号量已被一低优先级任务占有，因此造成高优先级任务被许多具有较低优先级任务阻塞，实时性难以得到保证。

在多任务系统中，通常可能出现如下优先级反转的情形：

假设有三个优先级不同的任务，分别为 task1、task2、和 task3；task1 的优先级最高，task2 次之，task3 最低。其中 task1 和 task2 有共享的临界区，因此存在互斥锁，即 task1 和 task2 在同一时间只有一者能访问临界区。在任务执行的过程中，如果 task2 已进入临界区，由于互斥锁的存在，task1 在被唤醒且进入临界区之前，就会被阻塞。

由于 task1 只有在 task2 从临界区退出后才能被调度，而 task2 什么时候从临界区退出是未知的，因此 task1 被阻塞的时间也是未知的。这样，低优先级的 task3 先于高优先级的 task1 被调度，造成了优先级反转。

现有技术为了解决这一问题，提出了优先级继承的方法。在上述过程中，task1 在被阻塞的时候会提升 task2 的优先级与 task1 相同，此时其他任务就无法打断 task2，从而保证 task2 执行之后顺利执行高优先级的 task1，解决了三个任务单一互斥锁的情况下优先级反转的问题。

但是，如果此时 task2 与 task3 共享临界区，task3 获得互斥锁的情况下，task2 也被

阻塞，只有 task3 能够继续执行，在执行 task3 的过程中，优先级高于 task3 的任务均可以打断 task3 执行，导致优先级高的 task1 执行推后。

发明内容

鉴于上述问题，本申请一实施例提出一种任务的优先级处理方法和处理装置，以解决现有技术存在的问题。

为了解决上述问题，本申请一实施例公开一种任务的优先级处理方法，所述任务包括第一任务、第二任务和第三任务，所述的第一任务的优先级高于所述第二任务和所述第三任务；所述方法包括：当所述第一任务由于访问第一临界区失败而被阻塞时，提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级；判断是否存在与第二任务共享第二临界区且正在访问所述第二临界区的第三任务；当存在所述第三任务时，提升所述第三任务的优先级。

为了解决上述问题，本申请一实施例还公开一种任务的优先级处理装置，所述任务包括第一任务、第二任务和第三任务，所述的第一任务的优先级高于所述第二任务和所述第三任务；所述装置包括：第一优先级变更模块，用于当所述第一任务由于访问第一临界区失败而被阻塞时，提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级；第一判断模块，用于判断是否存在与第二任务共享第二临界区且正在访问所述第二临界区的第三任务；第二优先级变更模块，用于当存在所述第三任务时，提升所述第三任务的优先级。

由上述可知，本申请实施例至少包括以下优点：

本申请提出一种任务的优先级处理方法，应用于多任务操作系统。当操作系统中的第一任务和第二任务共享第一临界区，而第二任务与第三任务共享第二临界区时，为了避免现有的优先级继承的方法在多任务系统中仍可能造成的优先级反转的任务，本申请实施例提供的方案不仅在第一任务无法访问第一临界区的情况下提升此时正在访问临界区的第二任务，同时提升与第二任务共享第二临界区的第三任务，避免优先级低的第三任务拖延第二任务的执行，从而避免优先级高的第一任务延迟执行造成的优先级反转。

附图说明

为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作一简单地介绍，显而易见地，下面描述中的附图是本申

请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

图 1 是本申请第一实施例的任务的优先级处理方法的流程图。

图 2 是本申请第二实施例的任务的优先级处理方法的流程图。

图 3 是本申请第三实施例的任务的优先级处理装置的方框图。

图 4 是本申请第四实施例的任务的优先级处理装置的方框图。

具体实施方式

下面将结合本申请实施例中的附图，对本申请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申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员所获得的所有其他实施例，都属于本申请保护的范围。

本申请的核心思想之一在于，提出一种任务的优先级处理方法，所述任务包括第一任务、第二任务和第三任务，所述的第一任务的优先级高于所述第二任务和所述第三任务；所述方法包括：当所述第一任务由于访问第一临界区失败而被阻塞时，提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级；判断是否存在与第二任务共享第二临界区且正在访问所述第二临界区的第三任务；当存在所述第三任务时，提升所述第三任务的优先级。本申请实施例提出的方案在现有的优先级继承的基础上，不仅提高了与第一任务共享临界区的第二任务的优先级，而且提高了可能拖延第二任务执行的其他任务（例如与第二任务共享临界区的其他任务）的优先级，避免优先级低于第一任务的其他任务抢占执行造成的优先级反转，提高了优先级高的第一任务的响应时间。

以下以两段多操作系统中执行的程序为例，说明现有的优先级继承问题存在的实质缺陷和本申请提供的任务的优先级处理方法。

假设有 task1， task2，task3，优先级分别为 5， 20， 25， 数字越小代表任务优先级越高。下面的例子中涉及到的互斥锁有 mutex_1 以及 mutex_2。互斥锁是一种信号量，常用来防止两个任务在同一时刻访问相同的共享资源。下述示例中互斥锁 mutex_1 保护 task1 和 task2 的共享资源，mutex_2 保护 task2 和 task3 的共享资源。每一个共享资源分别对应一个互斥锁标记，这一标记用来保证任意时刻只有一个任务能够访问该共享资源。

程序 1：

```
void task1()
{
mutex_1_get();                          (1)
        /*do something*/                (2)
        mutex_1_free();                 (3)
}
void task2()
{
        mutex_1_get();                  (4)
        mutex_2_get();                  (5)
        /*do something*/                (6)
        mutex_2_free();                 (7)
        mutex_1_free();                 (8)
}
void task3()
{
        mutex_2_get();          (9)
        /*do something*/                (10)
       mutex_2_free();                  (11)
}
```

按照现有的优先级继承提供的方案，假设开始的时候 task1 以及 task2 处于等待某事件的阻塞状态（例如主动睡眠、等待信号量输入等），上述程序执行的步骤如下：

执行步骤 1’． task3 运行，在(9)处获得互斥锁 mutex_2；

执行步骤 2’． task3 运行到(10)处时，task2 被中断唤醒；

执行步骤 3’．由于 task2 优先级高于 task3，运行 task2，task3 被挂起；

执行步骤 4’． task2 在(4)处获得互斥锁 mutex_1；

执行步骤 5’．当 task2 运行到(5)处时，因为互斥锁 mutex_2 已被 task3 获得，因而 task2 得不到互斥锁 mutex_2，会被阻塞；

执行步骤 6’．根据优先级继承的方法，task3 的优先级被 task2 从 25 拉升到 20；

执行步骤 7’. task2 被阻塞当中，假设 task1 被中断唤醒然，task1 开始运行；

执行步骤 8’. 当 task1 运行到(1)处时，由于此时 task2 在步骤（4）时已获得了互斥锁 mutex_1，此时 task1 得不到互斥锁 mutex_1，被阻塞；

执行步骤 9’. 根据优先级继承的方法，task2 的优先级被 task1 从 20 拉升到 5；

执行步骤 10’. task2 继续返回 task3 运行(10)处。

从上述可知，此时 task1 的优先级为 5，task2 的优先级为 5， task3 的优先级为 20。在步骤 5 处时 task2 被阻塞，在步骤 8 处 task1 被阻塞，只有 task3 能够继续执行。task3 在(10)处继续运行的时候会被其他任意优先级为 5-19 的任务可以随便抢占，如果有一个 task4 优先级为 15，可以随时打断 task3 的运行，到时再返回 task3 运行到(11)处的时候，则高优先级的 task1 和 task2 的等的时间会非常长，本质上还是造成了优先级反转的问题。

为了解决这一问题，本申请提出一种任务的优先级处理方法，结合上述程序，对该方法进行说明。在本申请中，假设开始的时候 task1 以及 task2 处于等待某事件，处于阻塞状态，上述程序执行的步骤如下：

执行步骤 1. task3 运行，在（9）处获得互斥锁 mutex_2；

执行步骤 2. task3 运行到(10)处的时候，此时 task2 被中断唤醒；

执行步骤 3. 由于 task2 优先级高于 task3，运行 task2，task3 被挂起；

执行步骤 4. task2 在(4)处获得互斥锁 mutex_1；

执行步骤 5. 当 task2 运行到(5)处时，因为互斥锁 mutex_2 已被 task3 获得，因而 task2 得不到互斥锁 mutex_2，会被阻塞；

执行步骤 6. 根据优先级继承的方法，task3 的优先级被 task2 从 25 拉升到 20；

执行步骤 7. task2 被阻塞当中，假设 task1 被中断唤醒然，task1 开始运行；

执行步骤 8. 当 task1 运行到(1)处时，由于此时 task2 在步骤（4）时已获得了互斥锁 mutex_1，因此 task1 得不到互斥锁 mutex_1，被阻塞；

执行步骤 9. 根据优先级继承的方法，task1 将 task2 的优先级从 20 拉升到 5；

执行步骤 10. task1 同时将 task3 的优先级从 20 拉升到 5；

执行步骤 11. task2 继续返回 task3 运行(10)处。

从上述可知，此时 task1 的优先级为 5，task2 和 task3 的优先级也被提升至 5。task3 在(10)处继续运行的时候不会被其他任意优先级为 5-19 的任务抢占执行，task3 结束之后

释放互斥锁 mutex_2，进一步执行 task2，并在 task2 执行完毕之后释放互斥锁 mutex_1，执行 task1，保证了优先级高的 task1 顺利执行，不会被优先级低的任务抢占执行。

程序 2：

```
void task1()
{
   mutex_1_get()        (1)
   mutex_1_free()       (2)
}

void task2()
{
   mutex_1_get()       (3)

   mutex_1_free()     (4)
 }

void task3()
{
   mutex_1_get()        (5)
 /* do something */     (6)
   mutex_1_free()       (7)
}
```

假设利用现有技术提供的方案，假设优先级 task1 为 5 ， task2 为 20 ，task3 为 25，假设开始的时候 task1 以及 task2 处于等待某事件的阻塞状态（例如主动睡眠、等待信号量输入等），上述程序执行的步骤如下：

执行步骤 1’. task3 获得互斥锁 mutex_1；

执行步骤 2’. task3 运行到(6)的时候，task2 被唤醒，运行 task2， task3 被挂起；

执行步骤 3’. task2 运行到(3)处时，无法获得互斥锁 mutex_1，task2 被阻塞，此时将 task3 的优先级 25 提升到 task2 优先级 20，task2 睡眠；

执行步骤 4’. task3 继续运行(6)处时，假设 task1 被唤醒运行；

执行步骤 5’. 由于 task1 无法获得互斥锁 mutex_1，task1 被阻塞，此时将 task3 的

优先级 20 提升到与 task1 的优先级 5 相同；

执行步骤 6’. task1 睡眠，回到 task3 运行；

执行步骤 7’. task3 在 (7)处解锁，唤醒 task2 继续运行，此时 task3 的优先级被还原到初始值 25。

由上述可知，由于 task3 此时的优先级相对较低，在步骤 6 中 task3 运行的过程中，优先级高于 task3 的任务，即使优先级低于 task1 和 task2，均可以抢占任务，如果有一个 task4 优先级高于 task3，其随时打断 task3 的运行，到时再返回 task3 运行到(7)处的时候，task1 和 task2 的等的时间会非常长，同样造成了优先级反转的问题。

为了解决这一问题，本申请提出一种任务的优先级处理方法，结合上述程序，对该方法进行说明。在本申请中，假设开始的时候 task1 以及 task2 处于等待某事件，处于阻塞状态，上述程序执行的步骤如下：

执行步骤 1. task3 获得互斥锁 mutex_1；

执行步骤 2. task3 运行到(6)的时候，task2 被唤醒，资源被 task2 抢占；

执行步骤 3. task2 运行到(3)处时，无法获得互斥锁 mutex_1，task2 被阻塞，此时将 task3 提升到 task2 优先级，然后 task2 睡眠；

执行步骤 4. task3 继续运行(6)处时，task1 被唤醒运行；

执行步骤 5. 由于 task1 无法获得互斥锁 mutex_1，task1 被阻塞，此时 task1 将 task2 的优先级提高到与 task1 相同；

执行步骤 6.task1 将 task3 的优先级同时拉升到与 task1 的优先级相同；

执行步骤 7. task1 睡眠，回到 task3 运行；

执行步骤 8. task3 在 (7)处解锁，然后唤醒 task2 继续运行。

由上述可知，在步骤 8 的 task3 运行的过程中，由于 task3 的优先级已被提升至与 task1 相同，因此在 task3 运行（6）的过程中，优先级低于 5 的其他任务不会打断 task3 运行，从而在 task3 结束之后释放互斥锁 mutex_1，进一步执行 task2 和 task1，保证了优先级高的 task1 顺利执行，不会被优先级低的任务抢占执行。

以下通过具体实施例说明本申请提出的任务的优先级处理方法和处理装置。

第一实施例

本申请第一实施例提出一种任务的优先级处理方法。图 1 所示为本申请第一实施例的任务的优先级处理方法的步骤流程图。

如图 1 所示，本申请实施例提出一种任务的优先级处理方法，所述任务包括第一任务、第二任务和第三任务，所述的第一任务的优先级高于所述第二任务和所述第三任务；该方法包括如下步骤：

S101，当所述第一任务由于访问第一临界区失败而被阻塞时，提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级；

在这一步骤中，参考上述程序 1 的执行步骤 9 和程序 2 的执行步骤 5，根据优先级继承方法，在满足如下情况时第二任务的优先级可以被提升至与第一任务相同：

1，第二任务的优先级低于第一任务；

2，此时第一任务和第二任务的临界区由第二任务访问，即第二任务获得了保护临界区（共享资源）的互斥锁导致当前第一任务无法获得。

在这一步骤中，当第一任务由于访问第一临界区失败而被阻塞时，第一任务将当前访问第一临界区的第二任务的优先级提升。

在一优选实施例中，上述第二任务的优先级优选可以被提升至与第一任务的优先级相同。在其他实施例中，上述第二任务的优先级也可以提升至仅次于第一任务，或者提升至高于其他任务的优先级，本申请并不以此为限。

在实际操作中，可以在任务创建的接口设置任务和对应的优先级。进一步地，程序中例如可以包括如下语句，提升第二任务的优先级至第一任务的优先级：

```
if (g_active_task->prio < mutex_task->prio) {
        task_pri_change(mutex_task, g_active_task->prio);
        TRACE_TASK_PRI_INV(g_active_task, mutex_task);
    }
pend_to_blk_obj((blk_obj_t *)mutex, g_active_task, ticks);
```

从 if (g_active_task->prio < mutex_task->prio) {

task_pri_change(mutex_task, g_active_task->prio)；可知：如果当前的任务的优先级高于获得锁的任务的优先级，就会调用 task_pri_change(mutex_task, g_active_task->prio)这个程序，改变这个任务的优先级到当前任务的优先级。其中 task_pri_change 这个函数会提

升对方的优先级。

S102，判断是否存在与第二任务共享第二临界区且正在访问所述第二临界区的第三任务；

S103，当存在所述第三任务时，提升所述第三任务的优先级。

在上述两个步骤中，参考上述程序 1 的执行步骤 10 和程序 2 的执行步骤 6，本申请实施例提出的改进相比于现有技术而言，不仅提升 task2 的优先级，同时提升与 task2 相关的 task3 的优先级，以避免 task3 拖延 task2 的执行时间，导致 task1 无法及时执行。

在本发明实施例中，task_pri_change 里可以包括下面的代码，用于步骤 S102 中发现 task3 并提升 task3 的优先级。

```
case K_PEND:
                    case K_PEND_SUSPENDED:
                        old_pri      = task->prio;
                        task->prio = new_pri;
                        pend_list_reorder(task);

                        if (task->blk_obj->obj_type == YUNOS_MUTEX_OBJ_TYPE)
{
                            mutex_tmp    = (kmutex_t *)(task->blk_obj);
                            mutex_task = mutex_tmp->mutex_task;
                            if (mutex_task->prio > task->prio) {
                                task = mutex_task;
```

在上述代码中，如果 task1 发现 task2 是在阻塞状态下，先把 task2 的优先级改成和 task1 一致，往下判断如果 task2 阻塞在 task3 的互斥锁上（即当前 task3 正在访问第二临界区，导致 task2 无法访问）的时候，会通过阻塞的互斥锁 mutex_2 得到 task3，如下：

```
mutex_tmp    = (kmutex_t *)(task->blk_obj);
                            mutex_task = mutex_tmp->mutex_task;
```

之后，如果 task3 的优先级比 task1 低，如步骤 S103，则提升 task3 的优先级：

```
if (mutex_task->prio > task->prio) {
                                task = mutex_task;
```

具体地，在本实施例中，影响 task2 执行的情况包括：task3 与 task2 共享第二临界区

且 task3 正在访问第二临界区（即 task3 获得了互斥锁）导致 task2 无法获得互斥锁时，只有 task3 执行完毕释放互斥锁，task2 才能执行，继而 task1 才能执行。因此，为了保证在 task3 执行过程中不被其他低与 task1 的优先级的任务打断，在上述步骤 S103 中将 task3 的优先级提高至与 task1 相同。

在一优选实施例中，上述第三任务的优先级优选可以被提升至与第一任务的优先级相同。在其他实施例中，上述第三任务的优先级也可以提升至仅次于第一任务，或者提升至高于其他任务的优先级，本申请并不以此为限。

由上述可知，本申请第一实施例提出的任务的优先级处理方法至少具有如下技术效果：

本发明提出一种任务的优先级处理方法，应用于多任务操作系统。当操作系统中的第一任务和第二任务共享第一临界区，而第二任务与第三任务共享第二临界区时，为了避免现有的优先级继承的方法在多任务系统中仍可能造成的优先级反转的任务，本发明实施例提供的方案不仅在第一任务无法访问第一临界区的情况下提升此时正在访问临界区的第二任务的优先级，同时提升与第二任务共享第二临界区的第三任务的优先级，避免优先级低的第三任务拖延第二任务的执行，从而避免优先级高的第一任务延迟执行造成的优先级反转。相比于现有技术，本发明实施例改进了现有的优先级继承的方法，解决了优先级反转的问题，进而解决了高优先级的任务由于等待低优先级任务的互斥锁的时间太长，而导致系统实时性跟不上的问题。本申请提出的方法提高了系统的实用性，在各种领域，特别是工业，语音，航天，军工，等领域具有重要的应用价值。

第二实施例

本申请第二实施例提出一种任务的优先级处理方法。图 2 所示为本申请第二实施例的任务的优先级处理方法的步骤流程图。如图 2 所示，本申请实施例提出一种任务的优先级处理方法，所述任务包括第一任务、第二任务和第三任务，所述的第一任务的优先级高于所述第二任务和所述第三任务；该方法包括如下步骤：

S201，当所述第一任务由于访问第一临界区失败而被阻塞时，提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级；

S202，判断是否存在与第二任务共享第二临界区且正在访问所述第二临界区的第三任务；

S203，当存在所述第三任务时，提升所述第三任务的优先级。

上述步骤 S201 至步骤 S203 与上一实施例的步骤 S101 至步骤 S103 相同或相似，在此不再赘述。本实施例重点说明与上一实施例的不同之处。

在这一实施例中，与步骤 S101 相似地，步骤 S201 中，可以将第二任务的优先级提升至与第一任务的优先级相同，或者例如可以仅次于第一任务的优先级等，在此不再赘述。

在步骤 S203 中，与步骤 S203 相似地，可以将第三任务的优先级提升至与第一任务的优先级相同，或者例如可以仅次于第一任务的优先级等，在此不再赘述。

在本实施例中，第一临界区与第二临界区可以是同一临界区，也可以是不同临界区。

同一临界区的情况可以参考上述程序 2，其中 task1、task2 和 task3 共享临界区，在进入临界区之前需要获得互斥锁 mutex_1。当其中一者获得互斥锁 mutex_1 并访问临界区时，其余的任务无法访问该临界区。

不同临界区的情况可以参考上述程序 1，其中 task1、task2 共享第一临界区，该临界区由互斥锁 mutex_1 保护；task2 和 task3 共享第二临界区，该临界区由互斥锁 mutex_2 保护。针对每个临界区，当其中一者获得互斥锁 mutex_1 并访问临界区时，另一者无法访问该临界区。

在一实施例中，上述任务的优先级处理方法针对的任务可以不限为三个。例如可以包括第四任务、第五任务……第 N 任务等。

在提升第三任务的优先级之后，所述方法还可以包括：

S204，判断是否存在与所述第三任务共享第三临界区且正在访问所述第三临界区的第四任务；

S205，当存在时，判断所述第四任务的优先级是否低于所述第一任务的优先级；

S206，当判断出所述第四任务的优先级低于所述第一任务时，提升所述第四任务的优先级。

在这一步骤中，按照与上述相同的方法判断出是否存在与第三任务共享第三临界区的第四任务，当存在时，且第四任务的优先级低于第一任务的优先级时，可以提升第四任务的优先级，例如提升至与第一任务相同。

可选地，在步骤 S206，即当判断出所述第四任务的优先级低于所述第一任务时，提升所述第四任务的优先级之后，该方法还可以包括：

S207，判断是否存在与被提高优先级的第 N 任务共享临界区且正在访问该临界区的第 N+1 任务；

S208，当存在时，判断所述第 N+1 任务的优先级是否低于所述第一任务的优先级；

S209，当判断出所述第 N+1 任务的优先级低于所述第一任务时，提升所述第 N+1 任务的优先级提高至所述第一任务的优先级，其中为 N 大于等于 4 的正整数。

在上述步骤中，假设 N 为 4，在第四任务被提升优先级之后，判断是否存在与第四任务共享临界区的第五任务。当存在时，且第五任务的优先级低于第一任务的优先级时，可以提升第五任务的优先级，例如提升至与第一任务相同。由此，可以保证多于三个任务执行时，每一个与第三任务相关的任务的优先级均被提升，避免了低优先级任务拖延高优先级任务的执行。上述的“与第三任务相关的任务”是指需要先于第三任务执行、否则第三任务无法执行的任务。

综上所述，本实施例提出的任务的优先级处理方法至少具有如下优点：

本申请提出一种任务的优先级处理方法，应用于多任务操作系统。当操作系统中的第一任务和第二任务共享第一临界区，而第二任务与第三任务共享第二临界区时，为了避免现有的优先级继承的方法在多任务系统中仍可能造成的优先级反转的任务，本申请实施例提供的方案不仅在第一任务无法访问第一临界区的情况下提升此时正在访问临界区的第二任务，同时提升与第二任务共享第二临界区的第三任务，避免优先级低的第三任务拖延第二任务的执行，从而避免优先级高的第一任务延迟执行造成的优先级反转。

除此之外，本实施例提出的任务的优先级处理方法至少还包括如下优点：

本申请优选实施例提出一种任务的优先级处理方法，应用于多任务操作系统。在操作系统中的任务高于三个的情况下，本申请实施例提供的方案不仅在第一任务无法访问第一临界区的情况下提升此时正在访问临界区的第二任务，同时提升与第二任务共享第二临界区的第三任务，以及与第三任务共享临界区的第四、第五……第 N+1 任务的优先级，避免优先级低的任务直接或间接地拖延第二任务的执行，从而避免优先级高的第一任务延迟执行造成的优先级反转。相比于现有技术，本申请实施例改进了现有的优先级继承的方法，解决了优先级反转的问题，进而解决了高优先级的任务由于等待时间过长导致延迟响应的问题。

第三实施例

本申请第三实施例提出一种任务的优先级处理装置，图 3 是本申请第三实施例的任务的优先级处理装置的方框图。如图 3 所示，所述任务包括第一任务、第二任务和第三任务，所述的第一任务的优先级高于所述第二任务和所述第三任务；所述装置包括：

第一优先级变更模块 301，用于当所述第一任务由于访问第一临界区失败而被阻塞时，提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级；

参考上述程序 1 的执行步骤 9 和程序 2 的执行步骤 5，根据优先级继承方法，在满足如下情况时第二任务的优先级可以被提升至与第一任务相同：

1，第二任务的优先级低于第一任务；

2，此时第一任务和第二任务的临界区由第二任务访问，即第二任务获得了保护临界区（共享资源）的互斥锁导致当前第一任务无法获得。

在这一模块中，当第一任务由于访问第一临界区失败而被阻塞时，第一任务将当前访问第一临界区的第二任务的优先级提升。

在一优选实施例中，上述第二任务的优先级优选可以被提升至与第一任务的优先级相同。在其他实施例中，上述第二任务的优先级也可以提升至仅次于第一任务，或者提升至高于其他任务的优先级，本申请并不以此为限。

第一判断模块 302，用于判断是否存在与第二任务共享第二临界区且正在访问所述第二临界区的第三任务；

第二优先级变更模块 303，用于当存在所述第三任务时，提升所述第三任务的优先级。

在上述两个模块中，参考上述程序 1 的执行步骤 10 和程序 2 的执行步骤 6，本申请实施例提出的改进相比于现有技术而言，不仅提升 task2 的优先级，同时提升与 task2 相关的 task3 的优先级，以避免 task3 拖延 task2 的执行时间，导致 task1 无法及时执行。

在本发明实施例中，task_pri_change 里可以包括下面的代码，用于步骤 S102 中发现 task3 并提升 task3 的优先级。

在一优选实施例中，上述第三任务的优先级优选可以被提升至与第一任务的优先级相同。在其他实施例中，上述第三任务的优先级也可以提升至仅次于第一任务，或者提升至高于其他任务的优先级，本申请并不以此为限。

本实施例提出的任务的优先级处理装置至少具有如下优点：

本发明提出一种任务的优先级处理装置，应用于多任务操作系统。当操作系统中的第一任务和第二任务共享第一临界区，而第二任务与第三任务共享第二临界区时，为了避免现有的优先级继承的方法在多任务系统中仍可能造成的优先级反转的任务，本发明实施例提供的方案不仅在第一任务无法访问第一临界区的情况下提升此时正在访问临界区的第二任务的优先级，同时提升与第二任务共享第二临界区的第三任务的优先级，避免优先级低的第三任务拖延第二任务的执行，从而避免优先级高的第一任务延迟执行造成的优先级反转。

相比于现有技术，本发明实施例改进了现有的优先级继承的手段，解决了优先级反转的问题，进而解决了高优先级的任务由于等待低优先级任务的互斥锁的时间太长，而导致系统实时性跟不上的问题。本申请提出优先级处理装置提高了系统的实用性，在各种领域，特别是工业，语音，航天，军工，等领域具有重要的应用价值。

第四实施例

本申请第四实施例提出一种任务的优先级处理装置，图 4 是本申请第四实施例的任务的优先级处理装置的方框图。所述任务包括第一任务、第二任务和第三任务，所述的第一任务的优先级高于所述第二任务和所述第三任务；如图 4 所示，该任务的优先级处理装置包括：

第一优先级变更模块 401，用于当所述第一任务由于访问第一临界区失败而被阻塞时，提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级；

第二判断模块 402，用于判断是否存在与第二任务共享第二临界区且正在访问所述第二临界区的第三任务；

第二优先级变更模块 403，用于当存在所述第三任务时，提升所述第三任务的优先级。

上述模块 401 至 403 与上一实施例的模块 301 至 303 相同或相似，在此不再赘述，相关内容可以参考模块 301 至 303 的描述。

可选地，所述第一临界区与所述第二临界区为同一临界区或不同临界区。

可选地，所述提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级中，所述第二任务的优先级被提升至与所述第一任务的优先级相同；和/或

所述提升所述第三任务的优先级中，所述第三任务的优先级被提升至与所述第一任务的优先级相同。

可选地，所述装置还包括：

第一判断模块 404，用于判断是否存在与所述第三任务共享第三临界区且正在访问所述第三临界区的第四任务；

第一优先级判断模块 405，用于当存在第四任务时，判断所述第四任务的优先级是否低于所述第一任务的优先级；

第三优先级变更模块 406，用于当判断出所述第四任务的优先级低于所述第一任务时，提升所述第四任务的优先级。

可选地，所述装置还包括：

第三判断模块 407，判断是否存在与被提高优先级的第 N 任务共享临界区且正在访问该临界区的第 N+1 任务；

第二优先级判断模块 408，用于当存在第 N+1 任务时，判断所述第 N+1 任务的优先级是否低于所述第一任务的优先级；

第四优先级变更模块 409，用于当判断出所述第 N+1 任务的优先级低于所述第一任务时，提升所述第 N+1 任务的优先级。

综上所述，本实施例提出的任务的优先级处理装置至少具有如下优点：

本申请提出一种任务的优先级处理装置，应用于多任务操作系统。当操作系统中的第一任务和第二任务共享第一临界区，而第二任务与第三任务共享第二临界区时，为了避免现有的优先级继承的手段在多任务系统中仍可能造成的优先级反转的任务，本申请实施例提供的方案不仅在第一任务无法访问第一临界区的情况下提升此时正在访问临界区的第二任务，同时提升与第二任务共享第二临界区的第三任务，避免优先级低的第三任务拖延第二任务的执行，从而避免优先级高的第一任务延迟执行造成的优先级反转。

除此之外，本实施例提出的任务的优先级处理装置至少还包括如下优点：

本申请优选实施例提出一种任务的优先级处理装置，应用于多任务操作系统。在操作系统中的任务高于三个的情况下，本申请实施例提供的方案不仅在第一任务无法访问第一临界区的情况下提升此时正在访问临界区的第二任务，同时提升与第二任务共享第二临界区的第三任务，以及与第三任务共享临界区的第四、第五……第 N+1 任务的优先级，避免优先级低的任务直接或间接地拖延第二任务的执行，从而避免优先级高的第一

任务延迟执行造成的优先级反转。相比于现有技术，本申请实施例改进了现有的优先级继承的方案，解决了优先级反转的问题，进而解决了高优先级的任务由于等待时间过长导致延迟响应的问题。

本申请实施例还提出一种电子装置，用于任务的优先级处理，所述任务包括第一任务、第二任务和第三任务，所述的第一任务的优先级高于所述第二任务和所述第三任务；包括：

存储器，用于存储计算机可读程序；

处理器，当所述处理器读取所述存储器中的计算机可读程序时，所述电子装置执行如下操作：

当所述第一任务由于访问第一临界区失败而被阻塞时，提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级；

判断是否存在与第二任务共享第二临界区且正在访问所述第二临界区的第三任务；

当存在所述第三任务时，提升所述第三任务的优先级。

对于装置实施例而言，由于其与方法实施例基本相似，所以描述得比较简单，相关之处参见方法实施例的部分说明即可。

本说明书中的各个实施例均采用递进的方式描述，每个实施例重点说明的都是与其他实施例的不同之处，各个实施例之间相同相似的部分互相参见即可。

本领域内的技术人员应明白，本申请实施例的实施例可提供为方法、装置、或计算机程序产品。因此，本申请实施例可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本申请实施例可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

在一个典型的配置中，所述计算机设备包括一个或多个处理器 (CPU)、输入/输出接口、网络接口和内存。内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器(RAM) 和/或非易失性内存等形式，如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存 (PRAM)、

静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括非持续性的电脑可读媒体(transitory media)，如调制的数据信号和载波。

本申请实施例是参照根据本申请实施例的方法、终端设备(系统)、和计算机程序产品的流程图和／或方框图来描述的。应理解可由计算机程序指令实现流程图和／或方框图中的每一流程和／或方框、以及流程图和／或方框图中的流程和／或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理终端设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理终端设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和／或方框图一个方框或多个方框中指定的功能的装置。

这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理终端设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和／或方框图一个方框或多个方框中指定的功能。

这些计算机程序指令也可装载到计算机或其他可编程数据处理终端设备上，使得在计算机或其他可编程终端设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程终端设备上执行的指令提供用于实现在流程图一个流程或多个流程和／或方框图一个方框或多个方框中指定的功能的步骤。

尽管已描述了本申请实施例的优选实施例，但本领域内的技术人员一旦得知了基本创造性概念，则可对这些实施例做出另外的变更和修改。所以，所附权利要求意欲解释为包括优选实施例以及落入本申请实施例范围的所有变更和修改。

最后，还需要说明的是，在本文中，诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来，而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者终端设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者终端设备所固有的要素。在没有更多限制的情况下，由语句“包

括一个……”限定的要素，并不排除在包括所述要素的过程、方法、物品或者终端设备中还存在另外的相同要素。

以上对本申请所提供的一种任务的优先级处理方法和处理装置，进行了详细介绍，本文中应用了具体个例对本申请的原理及实施方式进行了阐述，以上实施例的说明只是用于帮助理解本申请的方法及其核心思想；同时，对于本领域的一般技术人员，依据本申请的思想，在具体实施方式及应用范围上均会有改变之处，综上所述，本说明书内容不应理解为对本申请的限制。

权 利 要 求 书

1. 一种任务的优先级处理方法，所述任务包括第一任务、第二任务和第三任务，所述的第一任务的优先级高于所述第二任务和所述第三任务；其特征在于，所述方法包括：

当所述第一任务由于访问第一临界区失败而被阻塞时，提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级；

判断是否存在与第二任务共享第二临界区且正在访问所述第二临界区的第三任务；

当存在所述第三任务时，提升所述第三任务的优先级。

2. 如权利要求 1 所述的方法，其特征在于，所述提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级包括：将所述第二任务的优先级提升至所述第一任务的优先级。

3. 如权利要求 1 所述的方法，其特征在于，所述提升所述第三任务的优先级包括：将所述第三任务的优先级提升至所述第一任务的优先级。

4. 如权利要求 1 所述的方法，其特征在于，第一临界区与所述第二临界区为同一临界区。

5. 如权利要求 1 所述的方法，其特征在于，所述第一临界区与所述第二临界区为不同临界区。

6. 如权利要求 1 所述的方法，其特征在于，所述当存在所述第三任务时，提升所述第三任务的优先级之后，所述方法还包括：

判断是否存在与所述第三任务共享第三临界区且正在访问所述第三临界区的第四任务；

当存在时，判断所述第四任务的优先级是否低于所述第一任务的优先级；

当判断出所述第四任务的优先级低于所述第一任务时，提升所述第四任务的优先级。

7. 如权利要求 6 所述的方法，其特征在于，所述提升所述第四任务的优先级包括：将所述第四任务的优先级提升至所述第一任务的优先级。

8. 如权利要求 6 所述的方法，其特征在于，所述当判断出所述第四任务的优先级低于所述第一任务时，提升所述第四任务的优先级之后，所述方法还包括：

判断是否存在与被提高优先级的第 N 任务共享临界区且正在访问该临界区的第 N+1 任务；

当存在时，判断所述第 N+1 任务的优先级是否低于所述第一任务的优先级；

当判断出所述第 N+1 任务的优先级低于所述第一任务时，提升所述第 N+1 任务的优先级，其中为 N 大于等于 4 的正整数。

9. 如权利要求 8 所述的方法，其特征在于，所述提升所述第 N+1 任务的优先级包括：将所述第 N+1 任务的优先级提升至所述第一任务的优先级。

10. 一种任务的优先级处理装置，所述任务包括第一任务、第二任务和第三任务，所述的第一任务的优先级高于所述第二任务和所述第三任务；所述装置包括：

第一优先级变更模块，用于当所述第一任务由于访问第一临界区失败而被阻塞时，将与所述第一任务共享第一临界区的第二任务的优先级提高至第一任务的优先级；

第一判断模块，用于判断是否存在与第二任务共享第二临界区且正在访问所述第二临界区的第三任务；

第二优先级变更模块，用于当存在所述第三任务时，提升所述第三任务的优先级。

11. 如权利要求 10 所述的装置，其特征在于，所述提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级包括：将所述第二任务的优先级提升至所述第一任务的优先级。

12. 如权利要求 10 所述的装置，其特征在于，所述提升所述第三任务的优先级包括：将所述第三任务的优先级提升至所述第一任务的优先级。

13. 如权利要求 10 所述的装置，其特征在于，第一临界区与所述第二临界区为同一临界区。

14. 如权利要求 10 所述的装置，其特征在于，所述第一临界区与所述第二临界区为不同临界区。

15. 如权利要求 10 所述的装置，其特征在于，所述装置还包括：

第一判断模块，用于判断是否存在与所述第三任务共享第三临界区且正在访问所述第三临界区的第四任务；

第一优先级判断模块，用于当存在第四任务时，判断所述第四任务的优先级是否低于所述第一任务的优先级；

第三优先级变更模块，用于当判断出所述第四任务的优先级低于所述第一任务时，提升所述第四任务的优先级。

16. 如权利要求 15 所述的装置，其特征在于，所述提升所述第四任务的优先级包括：将所述第四任务的优先级提升至所述第一任务的优先级。

17. 如权利要求 15 所述的装置，其特征在于，所述装置还包括：

第二判断模块，判断是否存在与被提高优先级的第 N 任务共享临界区且正在访问该临界区的第 N+1 任务；

第二优先级判断模块，用于当存在第 N+1 任务时，判断所述第 N+1 任务的优先级是否低于所述第一任务的优先级；

第四优先级变更模块，用于当判断出所述第 N+1 任务的优先级低于所述第一任务时，提升所述第 N+1 任务的优先级，其中为 N 大于等于 4 的正整数。

18. 如权利要求 17 所述的装置，其特征在于，所述提升所述第 N+1 任务的优先级包括：将所述第 N+1 任务的优先级提升至所述第一任务的优先级。

19. 一种电子装置，用于任务的优先级处理，所述任务包括第一任务、第二任务和第三任务，所述的第一任务的优先级高于所述第二任务和所述第三任务；包括：

存储器，用于存储计算机可读程序；

处理器，当所述处理器读取所述存储器中的计算机可读程序时，所述电子装置执行如下操作：

当所述第一任务由于访问第一临界区失败而被阻塞时，提升与所述第一任务共享第一临界区且正在访问所述第一临界区的第二任务的优先级；

判断是否存在与第二任务共享第二临界区且正在访问所述第二临界区的第三任务；

当存在所述第三任务时，提升所述第三任务的优先级。

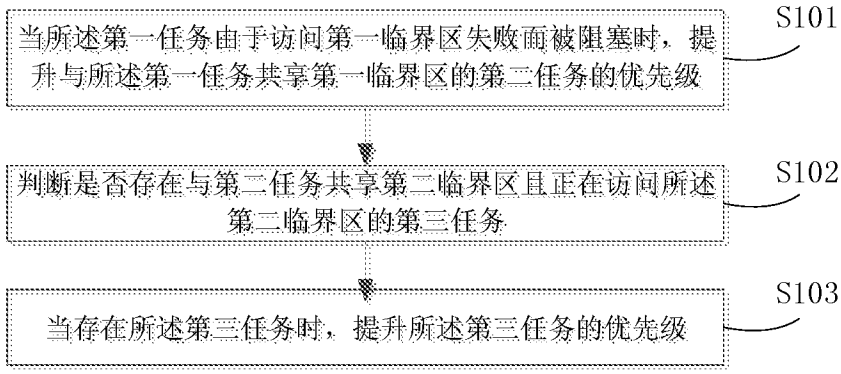

当所述第一任务由于访问第一临界区失败而被阻塞时，提升与所述第一任务共享第一临界区的第二任务的优先级
S101
判断是否存在与第二任务共享第二临界区且正在访问所述第二临界区的第三任务
S102
当存在所述第三任务时，提升所述第三任务的优先级
S103

图 1

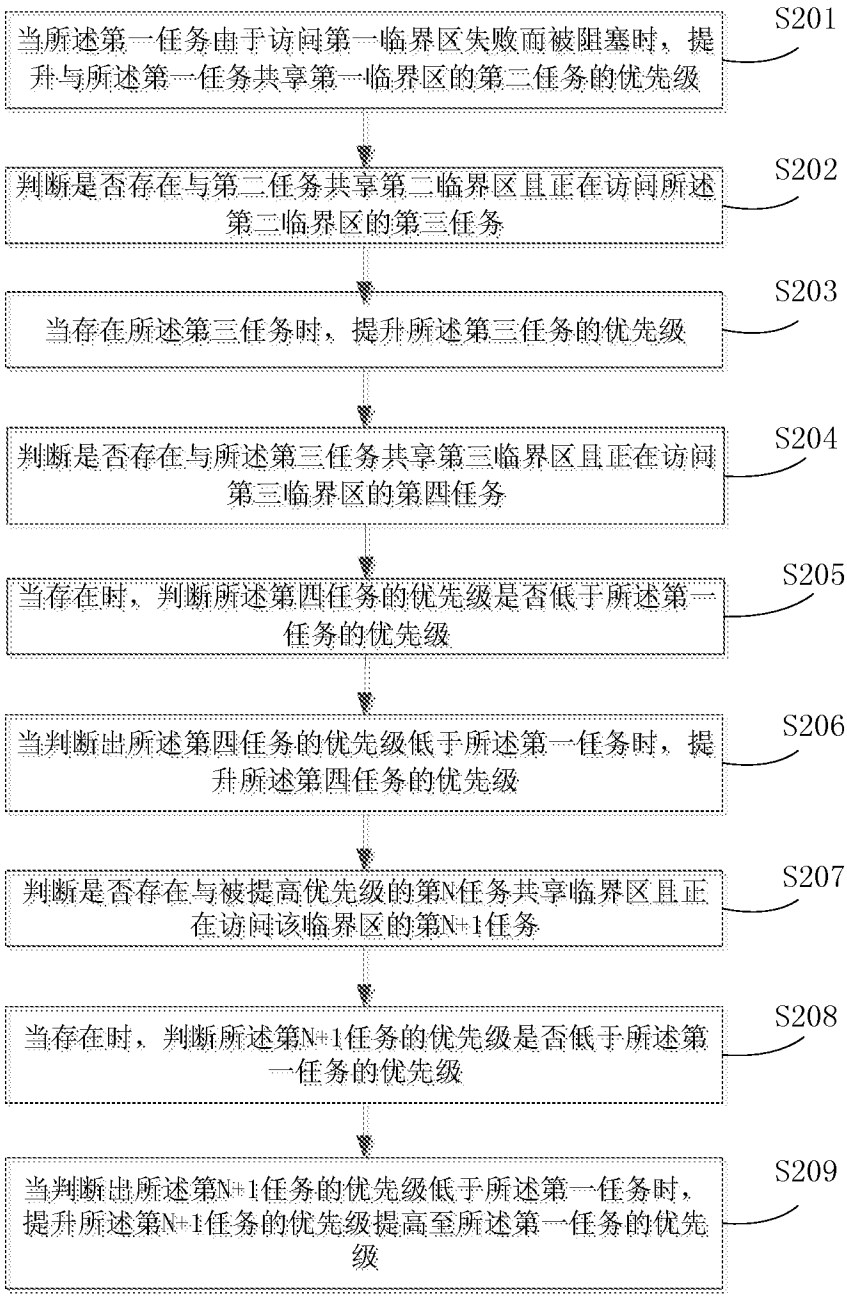

当所述第一任务由于访问第一临界区失败而被阻塞时，提升与所述第一任务共享第一临界区的第三任务的优先级
S201
判断是否存在与第二任务共享第二临界区且正在访问所述第三临界区的第三任务
S202
当存在所述第三任务时，提升所述第三任务的优先级
S203
判断是否存在与所述第三任务共享第三临界区且正在访问第三临界区的第四任务
S204
当存在时，判断所述第四任务的优先级是否低于所述第一任务的优先级
S205
当判断出所述第四任务的优先级低于所述第一任务时，提升所述第四任务的优先级
S206
判断是否存在与被提高优先级的第N任务共享临界区且正在访问该临界区的第N+1任务
S207
当存在时，判断所述第N+1任务的优先级是否低于所述第一任务的优先级
S208
当判断出所述第N+1任务的优先级低于所述第一任务时，提升所述第N+1任务的优先级提高至所述第一任务的优先级
S209

图 2

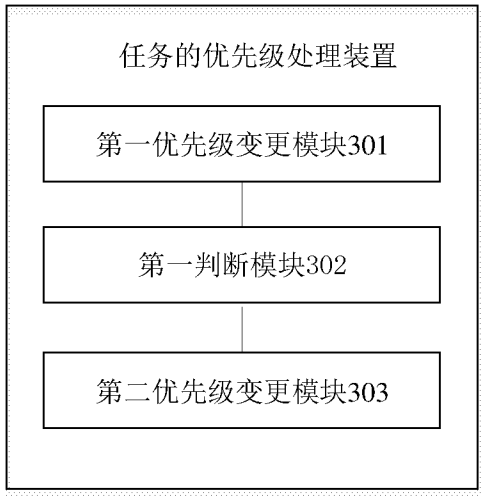

任务的优先级处理装置
第一优先级变更模块301
第一判断模块302
第二优先级变更模块303

图 3

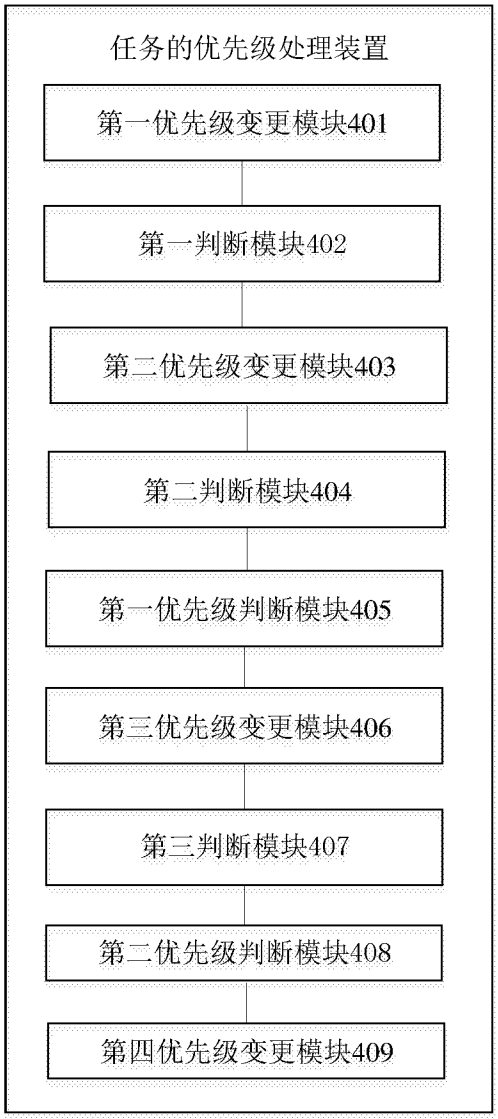

任务的优先级处理装置
第一优先级变更模块401
第一判断模块402
第二优先级变更模块403
第二判断模块404
第一优先级判断模块405
第三优先级变更模块406
第三判断模块407
第二优先级判断模块408
第四优先级变更模块409

图 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2018/092145

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/48(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNKI, CNPAT, WPI, EPODOC: 临界, 资源, 临界区, 阻塞, 优先级, 反转, 翻转, 互斥锁, 互斥, 占用, 调整, 提升, 提高, 改变, 任务, PRI, priority, task, block, adjust, change, upgrade, increase, mutex, reverse, rollback

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 103699437 A (HUAWEI TECHNOLOGIES CO., LTD.) 02 April 2014 (2014-04-02) description, paragraphs [0002]-[0041]	1-19
Y	CN 103870330 A (DATANG MOBILE COMMUNICATIONS EQUIPMENT CO., LTD.) 18 June 2014 (2014-06-18) description, paragraphs [0061]-[0085]	1-19
Y	CN 101216785 A (SAMSUNG ELECTRONICS CO., LTD.) 09 July 2008 (2008-07-09) description, page 1, paragraph 4 to page 4, paragraph 7	1-19
Y	CN 104636206 A (BEIJING INNOFIDEI TECHNOLOGY CO., LTD.) 20 May 2015 (2015-05-20) description, paragraphs [0146]-[0184]	1-19
A	US 2014317380 A1 (DENSO CORPORATION) 23 October 2014 (2014-10-23) entire document	1-19

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
15 August 2018	**04 September 2018**
Name and mailing address of the ISA/CN **State Intellectual Property Office of the P. R. China** **No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing** **100088** **China** Facsimile No. **(86-10)62019451**	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2018/092145

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 103699437 A	02 April 2014	None	
CN 103870330 A	18 June 2014	None	
CN 101216785 A	09 July 2008	EP 1942413 A2	09 July 2008
		KR 20080064608 A	09 July 2008
		US 2008168454 A1	10 July 2008
CN 104636206 A	20 May 2015	None	
US 2014317380 A1	23 October 2014	JP 2014211743 A	13 November 2014

国际检索报告

国际申请号
PCT/CN2018/092145

A. 主题的分类

G06F 9/48(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用）)

CNKI, CNPAT, WPI, EPODOC: 临界，资源，临界区，阻塞，优先级，反转，翻转，互斥锁，互斥，占用，调整，提升，提高，改变，任务，PRI, priority, task, block, adjust, change, upgrade, increase, mutex, reverse, rollback

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
Y	CN 103699437 A (华为技术有限公司) 2014年 4月 2日 (2014-04-02) 说明书第[0002]-[0041]段	1-19
Y	CN 103870330 A (大唐移动通信设备有限公司) 2014年 6月 18日 (2014-06-18) 说明书第[0061]-[0085]段	1-19
Y	CN 101216785 A (三星电子株式会社) 2008年 7月 9日 (2008-07-09) 说明书第1页第4段-第4页第7段	1-19
Y	CN 104636206 A (北京创毅视讯科技有限公司) 2015年 5月 20日 (2015-05-20) 说明书第[0146]-[0184]段	1-19
A	US 2014317380 A1 (DENSO CORPORATION) 2014年 10月 23日 (2014-10-23) 全文	1-19

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

"A" 认为不特别相关的表示了现有技术一般状态的文件

"E" 在国际申请日的当天或之后公布的在先申请或专利

"L" 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

"O" 涉及口头公开、使用、展览或其他方式公开的文件

"P" 公布日先于国际申请日但迟于所要求的优先权日的文件

"T" 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

"X" 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

"Y" 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

"&" 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2018年 8月 15日	2018年 9月 4日

ISA/CN的名称和邮寄地址	受权官员
中华人民共和国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088	李晓利
传真号 (86-10)62019451	电话号码 86-(10)-53961772

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2018/092145

检索报告引用的专利文件			公布日（年/月/日）	同族专利			公布日（年/月/日）
CN	103699437	A	2014年 4月 2日	无			
CN	103870330	A	2014年 6月 18日	无			
CN	101216785	A	2008年 7月 9日	EP	1942413	A2	2008年 7月 9日
				KR	20080064608	A	2008年 7月 9日
				US	2008168454	A1	2008年 7月 10日
CN	104636206	A	2015年 5月 20日	无			
US	2014317380	A1	2014年 10月 23日	JP	2014211743	A	2014年 11月 13日