

进程篇—优先级调度

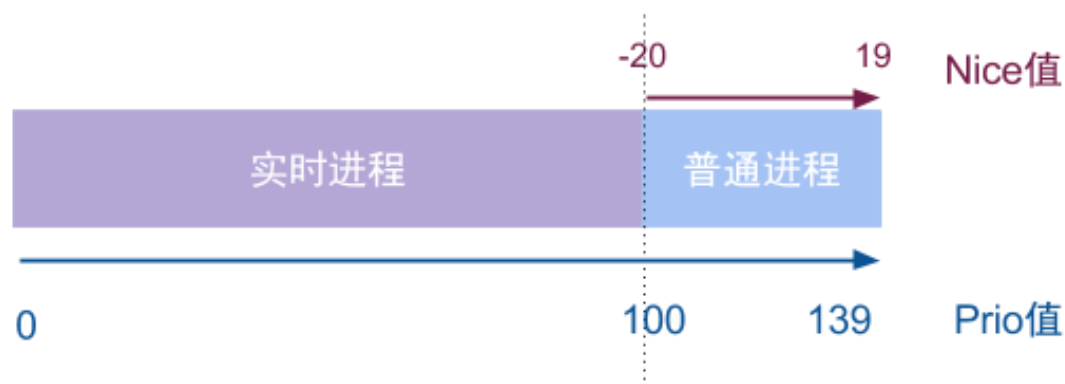
Oct 1, 2015

- 一、 进程优先级
- 二、 Framework调度策略
 - 2.1 进程优先级
 - 2.2 组优先级
 - 2.3 调度器选择
- 三、 Kernel调度策略
 - 3.1 优先级
 - 3.2 调度器

线程与进程的最大区别就是是否共享父进程的地址空间，内核角度来看没有线程与进程之分，都用task_struct结构体来表示，调度器操作的实体便是task_struct。

一、 进程优先级

进程可划分为普通进程和实时进程，那么优先级与nice值的关系图：



优先级值越小表示进程优先级越高，3个进程优先级的概念：

- 静态优先级：不会时间而改变，内核也不会修改，只能通过系统调用改变nice值的方法区修改。优先级映射公式： $\text{static_prio} = \text{MAX_RT_PRIO} + \text{nice} + 20$ ，其中 $\text{MAX_RT_PRIO} = 100$ ，那么取值区间为[100, 139]；对应普通进程；

- 实时优先级：只对实时进程有意义，取值区间为[0, MAX_RT_PRIO - 1]，其中 MAX_RT_PRIO = 100，那么取值区间为[0, 99]；对应实时进程；
- 动态优先级：调度程序通过增加或减少进程静态优先级的值，来达到奖励IO消耗型或惩罚cpu消耗型的进程，调整后的进程称为动态优先级。区间范围[0, MX_PRIO-1]，其中MX_PRIO = 140，那么取值区间为[0,139]；

nice值

nice∈[-20, 19]，可通过adb直接修改某个进程的nice值： `renice prio pid`

二、Framework调度策略

代码路径： framework/base/core/android/os/Process.java

2.1 进程优先级

Android进程优先级，总分10级

优先级调度方法：

setThreadPriority(int tid, int priority)

进程优先级级别：

进程优先级	nice 值	解释
THREAD_PRIORITY_LOWEST	19	最低优先级
THREAD_PRIORITY_BACKGROUND	10	后台
THREAD_PRIORITY_LESS_FAVORABLE	1	比默认略低
THREAD_PRIORITY_DEFAULT	0	默认
THREAD_PRIORITY_MORE_FAVORABLE	-1	比默认略高
THREAD_PRIORITY_FOREGROUND	-2	前台
THREAD_PRIORITY_DISPLAY	-4	显示相关
THREAD_PRIORITY_URGENT_DISPLAY	-8	显示(更为重要)，input事件
THREAD_PRIORITY_AUDIO	-16	音频相关
THREAD_PRIORITY_URGENT_AUDIO	-19	音频(更为重要)

2.2 组优先级

进程/线程组优先级调度方法：

```
setProcessGroup(int pid, int group)
setThreadGroup(int tid, int group)
```

进程组优先级级别：

组优先级	取值	解释
THREAD_GROUP_DEFAULT	-1	仅用于setProcessGroup，将优先级 ≤ 10 的进程提升到-2
THREAD_GROUP_BG_NONINTERACTIVE	0	CPU分时的时长缩短
THREAD_GROUP_FOREGROUND	1	CPU分时的时长正常
THREAD_GROUP_SYSTEM	2	系统线程组
THREAD_GROUP_AUDIO_APP	3	应用程序音频
THREAD_GROUP_AUDIO_SYS	4	系统程序音频

2.3 调度器选择

调度器设置方法：

```
setThreadScheduler(int tid, int policy, int priority)
```

调度器类别

调度器	名称	解释
SCHED_OTHER	默认	标准round-robin分时共享策略
SCHED_BATCH	批处理调度	针对具有batch风格（批处理）进程的调度策略
SCHED_IDLE	空闲调度	针对优先级非常低的适合在后台运行的进程
SCHED_FIFO	先进先出	实时调度策略，android暂未实现
SCHED_RR	循环调度	实时调度策略，android暂未实现

三、Kernel调度策略

设置优先级，Kernel不区别线程和进程，都对应同一个数据结构Task。Linux kernel用nicer值来描述进程的调度优先级，该值越大，表明该进程越友（nice），其被调度运行的几率越低。

3.1 优先级

```
int setpriority(int which, int who, int prio);
```

参数说明：

- which和who参数联合使用：
 - 当which为PRIO_PROGRESS时，who代表一个进程；
 - 当which为PRIO_PGROUUP时，who代表一个进程组；
 - 当which为PRIO_USER时，who代表一个uid。
- prio参数用于设置应用进程的nicer值，可取范围从-20到19。

3.2 调度器

```
int sched_setscheduler(pid_t pid, int policy, const struct sched_param *param);
```

参数说明：

- pid为进程id；
- policy为调度策略；
- param最重要的是该结构体中的sched_priority变量；
 - 针对Android中的三种非实时Scheduler策略，该值必须为NULL。

选择和设置合理的进程优先级和调度器是性能优化的一个方向，后续再以内核调度器的角度来分析调度策略的抉择问题。

喜欢

0条评论

最新 最早 最热

还没有评论，沙发等你来抢

嘿嘿参北斗哇 (<http://www.baidu.com/p/嘿嘿参北斗哇>) 帐号管理



(<http://duoshuo.com/settings/avatar/>)

说点什么吧...

☐ 分享到:

发布

多说 (<http://duoshuo.com>)

✉ gityuan@gmail.com (<mailto:gityuan@gmail.com>) ·  Github

(<https://github.com/yuanhuihui>) · 天道酬勤 · © 2015 Yuanhh · Jekyll

(<https://github.com/jekyll/jekyll>) theme by HyG (<https://github.com/Gaohaoyang>)