
Subject: Set thread priority for linux

Posted by [tojocky](#) on Sat, 02 Jun 2012 16:50:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello,

Having some experience with thread priority on linux I propose to implment Thread::Priority(int);

Ok the implementation for linux is:

```
....
#ifdef PLATFORM_POSIX
int policy, res;
struct sched_param param;

if ((res = pthread_getschedparam(handle, &policy, &param)) != 0){
    return;
    //TODO: should returns the error
}
if(percent <= 25){
    policy = SCHED_IDLE;
    param.sched_priority = (sched_get_priority_max(policy) -
sched_get_priority_min(policy))*percent/25;
}else if(percent <= 75){
    policy = SCHED_BATCH;
    param.sched_priority = (sched_get_priority_max(policy) -
sched_get_priority_min(policy))*(percent-25)/50;
}else if(percent <= 125){
    policy = SCHED_OTHER;
    param.sched_priority = (sched_get_priority_max(policy) -
sched_get_priority_min(policy))*(percent-75)/50;
}else if(percent <= 175){// should be a root
    policy = SCHED_FIFO;
    param.sched_priority = (sched_get_priority_max(policy) -
sched_get_priority_min(policy))*(percent-125)/50;
}else{
    policy = SCHED_RR;
    param.sched_priority = (sched_get_priority_max(policy) -
sched_get_priority_min(policy))*(minmax(percent, 175, 225)-175)/25;
}

if ((res = pthread_setschedparam(handle, policy, &param)) != 0){
    return;
    //TODO: should returns the error
}
#endif
....
```

and the hole method:

```
void Thread::Priority(int percent)
{
    ASSERT(IsOpen());
#ifdef PLATFORM_WIN32
    int prior;
    if(percent <= 25)
        prior = THREAD_PRIORITY_LOWEST;
    else if(percent <= 75)
        prior = THREAD_PRIORITY_BELOW_NORMAL;
    else if(percent <= 125)
        prior = THREAD_PRIORITY_NORMAL;
    else if(percent <= 175)
        prior = THREAD_PRIORITY_ABOVE_NORMAL;
    else
        prior = THREAD_PRIORITY_HIGHEST;
    SetThreadPriority(handle, prior);
#endif
#ifdef PLATFORM_POSIX
    int policy, res;
    struct sched_param param;

    if ((res = pthread_getschedparam(handle, &policy, &param)) != 0){
        return;
        //TODO: should returns the error
    }
    if(percent <= 25){
        policy = SCHED_IDLE;
        param.sched_priority = (sched_get_priority_max(policy) -
        sched_get_priority_min(policy))*percent/25;
    }else if(percent <= 75){
        policy = SCHED_BATCH;
        param.sched_priority = (sched_get_priority_max(policy) -
        sched_get_priority_min(policy))*(percent-25)/50;
    }else if(percent <= 125){
        policy = SCHED_OTHER;
        param.sched_priority = (sched_get_priority_max(policy) -
        sched_get_priority_min(policy))*(percent-75)/50;
    }else if(percent <= 175){// should be a root
        policy = SCHED_FIFO;
        param.sched_priority = (sched_get_priority_max(policy) -
        sched_get_priority_min(policy))*(percent-125)/50;
    }else{
        policy = SCHED_RR;
        param.sched_priority = (sched_get_priority_max(policy) -
        sched_get_priority_min(policy))*(minmax(percent, 175, 225)-175)/25;
```

```
}  
  
if ((res = pthread_setschedparam(handle, policy, &param)) != 0){  
    return;  
    //TODO: should returns the error  
}  
#endif  
}
```

Any comments are welcome!

Best regards,
Ion.
