Subject: Re: Set thread priority for linux
Posted by tojocky on Sun, 03 Jun 2012 13:17:10 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 03 June 2012 14:15I would like to add this patch, but (Ubuntu 10.04):

/home/cxl/upp.src/uppsrc/Core/Mt.cpp: In member function ‘void Upp::Thread::Priority(int)’:
/home/cxl/upp.src/uppsrc/Core/Mt.cpp:255:12: error: ‘SCHED_IDLE’ was not declared in this scope

...I guess more work is needed to resolve all compatiblity issues.

It has to work with Debian since 5, FreeBSD, MacOSX.

Mirek

You are right,
Sorry, I don't use old versions.

Please try this one:

```
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
 }
 int percen_min, percen_max;
 if(percent <= 25){
```

```
 #if defined(SCHED_IDLE)
  policy = SCHED_IDLE;
  percen_min = 0;
  percen_max = 25;
 #elif defined(SCHED_BATCH)
  policy = SCHED_BATCH;
  percen_min = 0;
  percen_max = 75;
 #else
  policy = SCHED_OTHER;
  percen_min = 0;
  percen_max = 125;
 #endif
}else if(percent <= 75){
 #if defined(SCHED_IDLE)
  policy = SCHED_BATCH;
  percen_min = 25;
  percen_max = 75;
 #elif defined(SCHED_BATCH)
  policy = SCHED_BATCH;
  percen_min = 0;
  percen_max = 75;
 #else
  policy = SCHED_OTHER;
  percen_min = 0;
  percen_max = 125;
 #endif
}else if(percent <= 125){
 policy = SCHED_OTHER;
 #if defined(SCHED_IDLE)
  percen_min = 75;
  percen_max = 125;
 #elif defined(SCHED_BATCH)
  percen_min = 25;
  percen_max = 125;
 #else
  percen_min = 0;
  percen_max = 125;
 #endif
}else if(percent <= 175){// should be a root
 policy = SCHED_FIFO;
 percen_min = 125;
 percen_max = 175;
}else{
 policy = SCHED_RR;
}
 param.sched_priority = (sched_get_priority_max(policy) -
sched_get_priority_min(policy))*(minmax(percent, percen_min,
```

```
percen_max)-percen_min)/(percen_max - percen_min);

 if ((res = pthread_setschedparam(handle, policy, &param)) != 0){
  // don't have privileges? I'm trying maxim possible! I do not use EPERM because not all os
support this one
  policy = SCHED_OTHER;
  percen_max = 125;
  percen_min = minmax(percen_min, 0, percen_max);
  param.sched_priority = (sched_get_priority_max(policy) -
sched_get_priority_min(policy))*(minmax(percent, percen_min,
percen_max)-percen_min)/(percen_max - percen_min);
  if ((res = pthread_setschedparam(handle, policy, &param)) != 0){
   return;
   //TODO: should returns the error
  }
 }
#endif
}
```

I have checked this functionality for MacOS, iOS, Linux and freebsd if it exists. My code I compared with QT.

Thank you Mirek!

Ion.
