Subject: Re: CoWork buggy!?
Posted by Werner on Mon, 24 Mar 2008 10:54:11 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Mon, 24 March 2008 10:48As far as I understand, CoWork detects number of processors present and tries to execute threads on different cores. Is that so?

I had quite a nice time analyzing "CoWork" (and multithreading under Ultimate++ in general) and sometimes I was painfully wrong , but eventually - withs Mirek's help, of course - I was able to document the whole stuff . So just look at this doc cutting (also published in the hope that Mirek will correct possibly still lurking errors ):

Quote:Cf. ...upp/uppsrc/Core/CoWork.cpp

Pool();

Starts number_of_cpu(_core)s_+_2 Threads which will wait for jobs and do them as soon as they are available.

That is: all these Threads are delivered to the OS to be called by the OS when appropriate.

That means: as soon as Pool is constructed, the machine-dependant maximum number of Threads is active (periodically called by the OS) and waiting for jobs to do.

What, in the end, is periodically called by the OS, are number_of_cpu(_core)s_+_2 ThreadRun-methods (identifiable by the index passed while running "Pool()").

```
CoWork::Pool::Pool()
{
 for(int i = 0; i < CPU_Cores() + 2; i++)   // cpu( core)s + 2
  threads.Add().Run(callback1(&ThreadRun, i)); // add Thread to Pool & start Thread
}
```

Werner