

GROMACS - Task #2478

avoid use of getenv in static initialization

04/13/2018 11:57 AM - Mark Abraham

Status:	New
Priority:	Low
Assignee:	
Category:	core library
Target version:	
Difficulty:	uncategorized

Description

We sometimes write static initialization code like

```
const bool c_useCudaLaunchKernel = (GMX_CUDA_VERSION >= 7000) && (getenv("GMX_DISABLE_CUDALAUNCH")  
== nullptr);
```

so that we can have configure-time defaults with run-time overrides. However there's a risk that something else being constructed at init time might depend on the order of such initialization and strange non-reproducible behaviour result.

```
/* Implements the "construct on first use" idiom to avoid any static  
* initialization order fiasco.  
*  
* Note that thread-safety of the initialization is guaranteed by the  
* C++11 language standard.  
*  
* The pointer itself (not the memory it points to) has no destructor,  
* so there is no deinitialization issue. See  
* https://isocpp.org/wiki/faq/ctors for discussion of alternatives  
* and trade-offs. */  
const PmeTestEnvironment *getPmeTestEnv()  
{  
    static PmeTestEnvironment *pmeTestEnvironment = nullptr;  
    if (pmeTestEnvironment == nullptr)  
    {  
        // Ownership of the TestEnvironment is taken by GoogleTest, so nothing can leak  
        pmeTestEnvironment = static_cast<PmeTestEnvironment *> (::testing::AddGlobalTestEnvironment  
(new PmeTestEnvironment));  
    }  
    return pmeTestEnvironment;  
}
```

We should probably rework any such use of getenv(), and maybe have the checker prevent us adding new ones.