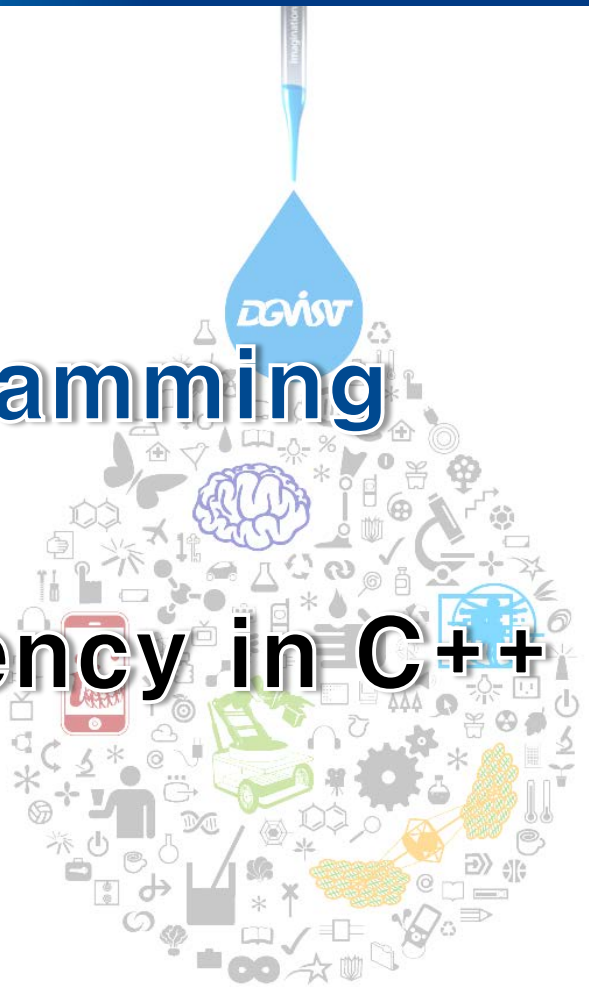
InfoLab



IC619: System Programming

Lecture 1

Hello, world of concurrency in C++

2012.09.03

Min-Soo Kim

DGIST 대구경북과학기술원
Daegu Gyeongbuk Institute of Science & Technology

InfoLab

Lab Quiz: Installation & Compilation

DGIST 대구경북과학기술원
Daegu Gyeongbuk Institute of Science & Technology

Purpose of Lab

- **Actual programming skill is much more important than knowing the grammar and the theory**
  - learning programming languages is very similar with learning human natural languages
- **You can quickly review the contents you've just learned through Lab**
- **Caution**
  - Different from IC512, there is no detailed explanation about how to solve the quiz or how to do C++ programming

Environment

- **Machine**
  - lab computer
  - your computer
- **OS**
  - Microsoft Windows
  - Windows + VMWare/Cygwin/…
  - Linux
  - MacOS
  - …

Environment

- **Compiler**
  - Microsoft Visual C++ + Just Library (on Windows)
  - latest version of g++ (on Linux)
  - …
- **Editor**
  - Visual C++ / Eclipse / … (on Windows)
  - vi / emacs / Eclipse / .. (on Linux)
  - …
- **Debugger**
  - Visual C++ (on Windows)
  - gdb [+ Eclipse] (on Linux)
  - …

Just Library

- **Implementation of C++11 Standard Thread Library**
- **Homepage**
  - http://www.stdthread.co.uk/
    (full documentation available online)
- **Author : Anthony Williams**
  - author or co-author of many of the threading-related proposals for C++11
  - author of C++ Concurrency in Action (our textbook)
  - maintainer of the Boost thread library
- **Installation**
  - see readme.txt (provided in lab)

Features of Just Library

- std::thread class for launching and managing threads
- std::async function for starting asynchronous tasks
- Mutex classes (std::mutex, std::timed_mutex, etc.) for protecting shared data
- Condition variables (std::condition_variable and std::condition_variable_any) for synchronizing operations
- Atomic types (std::atomic<int>, std::atomic<long>, etc.) for low level atomic access
- Futures and promises (std::future, std::promise, etc.) for communicating data between threads
- Thread-local variables using the JSS_THREAD_LOCAL macro

Compatibility of Just Library

- **Compatible with Microsoft Visual Studio 2005, 2008, and 2010 and TDM gcc 4.5.2 for both 32-bit and 64-bit Windows targets**
- **Compatible with g++ 4.3, 4.4, 4.5 and 4.6 for 32-bit and 64-bit Debian, Ubuntu, Fedora and Centos linux (x86/x86_64) targets, making full use of the C++11 support from g++ including rvalue references and variadic templates**
- **Compatible with MacPorts g++ 4.3, 4.4 and 4.5 for 32-bit Intel MacOSX targets, again making full use of the C++11 support from g++ including rvalue references and variadic templates**
- **Special debug mode for identifying the call chain leading to a deadlock (Not available with the TDM gcc compiler for Windows)**

Lab 1

- Make your environment
- Compile the test code in next page

Test Code

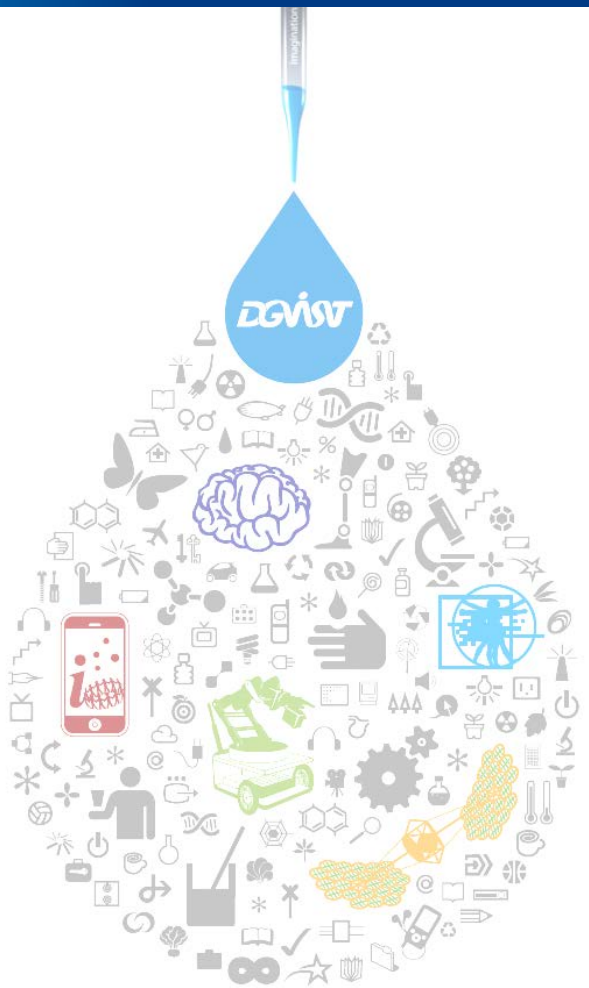
```
#include <iostream>
#include <thread>

void hello()
{
    std::cout<<"Hello Concurrent World\n";
}

int main()
{
    std::thread t(hello);
    t.join();
}
```

InfoLab

Thank you!



DGIST 대구경북과학기술원
Daegu Gyeongbuk Institute of Science & Technology