Improve compile-time speed

(C) Richêl Bilderbeek 

March 23, 2015

Chapter 1

Introduction

1.1 Question

- Given a project:
  - with X header (.h) files
  - with Y implementation (.cpp) files
  - in which every .cpp #includes each .h
  - in which every .h #includes each other .h
- How often is each .h file compiled?
- How often is each .cpp file compiled?

1.2 Answer

- Every .h file is compiled Y (the number of implementation files) times
- Every .cpp file is compiled once
- Conclusion: keep every .h file as simple to compile as possible

1.3 Question

```
//derived.h (MUST BE CORRECTED!)
#include <cmath>
#include <iostream>
#include "base.h"
#include "a.h"
#include "b.h"
#include "c.h"
struct Derived: public Base {
  A  m_a;
  B* m_b;
  C& m_c;
  double Sqrt() const;
};
std::ostream& operator<<(
  std::ostream& os, const Derived& myclass);
```

1.4 Answer

- Add #include guards
- Always write internal #include guards[1]
- (never write external #include guards[2])

[1] Herb Sutter and Andrei Alexandrescu. C++ coding standards: 101 rules, guidelines, and best practices. Chapter 24

[2] Herb Sutter and Andrei Alexandrescu. C++ coding standards: 101 rules, guidelines, and best practices. Chapter 24

1.5 Answer

- Remove #include <cmath>: no mathematical function is called
- Never #include unnecessary headers[3]

[3] Herb Sutter. Exceptional C++. Item 26

1.6 Answer

- Do not #include <iostream>, #include <iosfwd> instead
- Prefer to #include <iosfwd> when a forward declaration of a stream will suffice[4]

[4] Herb Sutter. Exceptional C++. Item 26

1.7 Answer

- Add forward declarations for 'B * m_b' and 'C& m_c'
- Remove their header file #includes
- Never #include a header file when a forward declaration will suffice[5]

[5] Herb Sutter. Exceptional C++. Item 26

1.8 Answer

```
//derived.h
#ifndef DERIVED_H
#define DERIVED_H
#include <iosfwd>
#include "base.h"
#include "a.h"
struct B;
struct C;
struct Derived: public Base {
  A m_a; B* m_b; C& m_c;
};
std::ostream& operator<<(
  std::ostream& os, const Derived& myclass);
#endif //~DERIVED_H
```

Chapter 2

Conclusion

2.1 Advice

- Keep a header file as simple to compile as possible
- Always write internal #include guards[1]
- Never #include unnecessary headers[2]
- Prefer to #include <iosfwd> when a forward declaration of a stream will suffice[3]
- Never #include a header file when a forward declaration will suffice[4]

[1] Herb Sutter and Andrei Alexandrescu. C++ coding standards: 101 rules, guidelines, and best practices. Chapter 24
[2] Herb Sutter. Exceptional C++. Item 26
[3] Herb Sutter. Exceptional C++. Item 26
[4] Herb Sutter. Exceptional C++. Item 26

- For widely used classes, prefer to use the compiler-firewall idiom (Pimpl idiom)[5][6]
- Never inherit when composition is sufficient[7]

[5] Herb Sutter, Andrei Alexandrescu. C++ coding standards: 101 rules, guidelines, and best practices. Item 43

[6] ?Meyers, More Effective C++

[7] ?Meyers, More Effective C++

Chapter 3

EOF