



TÌM HIỂU THƯ VIỆN BOOST

Nguyễn Trần Hậu	1612180
Nguyễn Duy Thanh	1612628
Tăng Nguyễn Hoàng Phi	1612493



Danh sách

1. Giới thiệu
2. Kiến trúc
3. Cài đặt
4. Boost.Asio
5. Boost.Thread
6. Boost.Flyweight



1. Giới thiệu



Boost là gì ?

- Thư viện C++
- Free Software
- Chuẩn bị cho chuẩn C++ tiếp theo

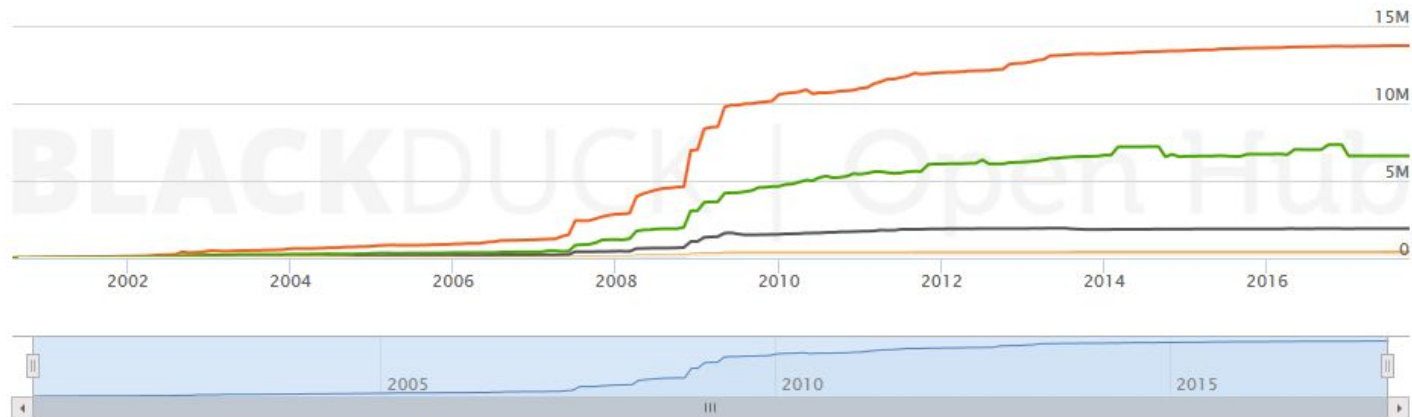


Tại sao phải sử dụng Boost

- Tăng tốc quá trình lập trình
- Code ít lỗi
- Không “phát minh lại bánh xe”



2. Kiến trúc



Language Breakdown

Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
C++	13,725,927	2,917,768	17.5%	2,796,161	19,439,856	<div><div></div></div> 63.4%
HTML	6,605,361	152,492	2.3%	433,586	7,191,439	<div><div></div></div> 23.5%
XML	1,912,950	34,934	1.8%	76,741	2,024,625	<div><div></div></div> 6.6%
C	363,062	99,688	21.5%	70,120	532,870	<div><div></div></div> 1.7%
Autoconf	248,725	319	0.1%	32,029	281,073	<div><div></div></div> 0.9%
CSS	200,738	29,603	12.9%	48,903	279,244	<div><div></div></div> 0.9%
Python	152,073	93,607	38.1%	55,581	301,261	<div><div></div></div> 1.0%
XSL Transformation	130,229	13,313	9.3%	19,937	163,479	<div><div></div></div> 0.5%
JavaScript	115,934	2,336	2.0%	7,006	125,276	<div><div></div></div> 0.4%



>100

Thư viện



80%

Header-only



Tour khám phá Boost

- RAIL và quản lý bộ nhớ
- Xử lý chuỗi
- Container
- Cấu trúc dữ liệu
- Thuật toán
- Giao tiếp
- Lập trình song song
- Functional Programming
- Template Programming
- Language Extensions
- Xử lý lỗi
- Xử lý số học
- Mẫu hình thiết kế
- ...

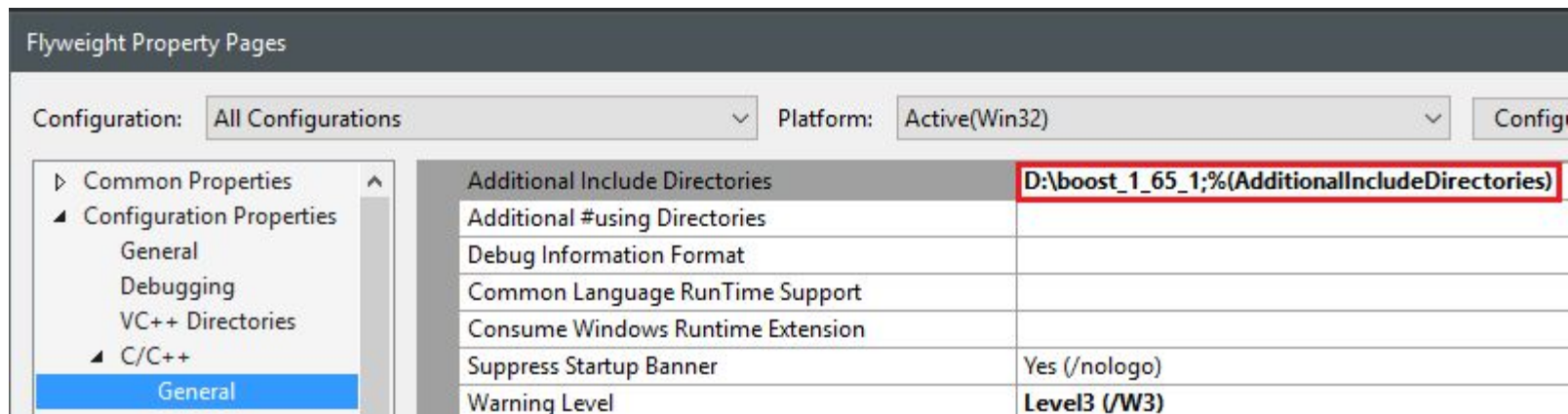


3. Cài đặt

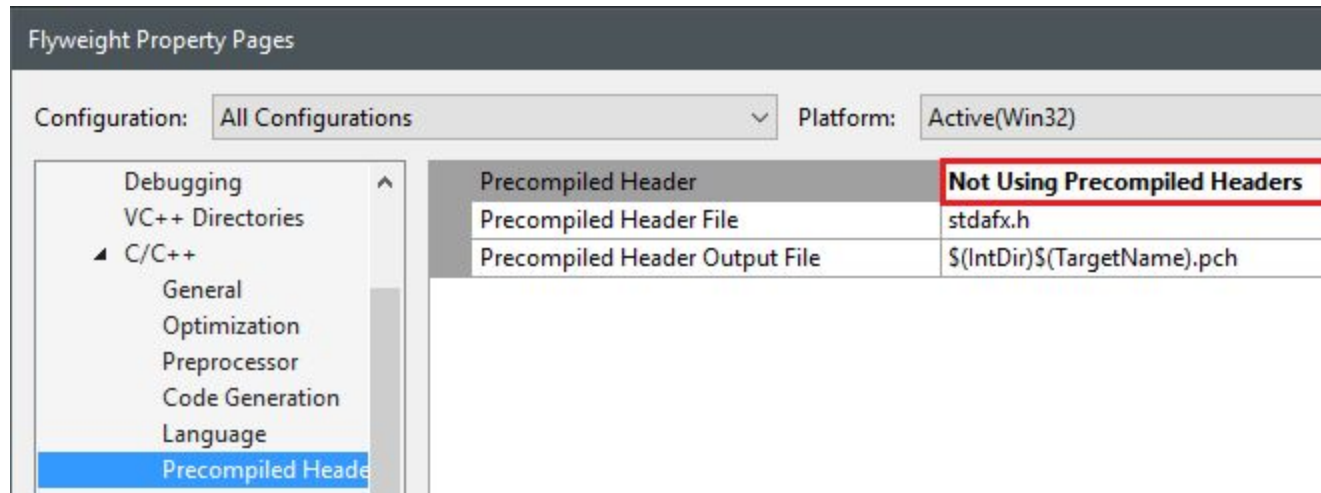


Nothing to build

- Đa số thư viện chỉ cần *#include*
- Chỉ một số thư viện mới cần build trước khi *#include*



Cài đặt thư viện boost trên Visual Studio 2013



Cài đặt thư viện boost trên Visual Studio 2013



```
#include <boost/algorithm/string.hpp>
#include <iostream>
#include <string>

using namespace boost::algorithm;

int main()
{
    std::string s = "Boost C++ Libraries";
    std::cout << to_upper_copy(s) << '\n';
    return 0;
}
```



4. Boost.Asio



```
#include <iostream>
#include <boost/asio.hpp>

int main()
{
    boost::asio::io_service io;
    boost::asio::deadline_timer t(io,
        boost::posix_time::seconds(5));
    t.wait();
    std::cout << "Good night, world!" << std::endl;
    return 0;
}
```



Boost.Asio ●

● Xử lý I/O

- Sockets
- Files
- Serial Ports
- Timers

● Cung cấp mô hình làm việc đồng bộ và không đồng bộ



Boost.Asio sử dụng cho?



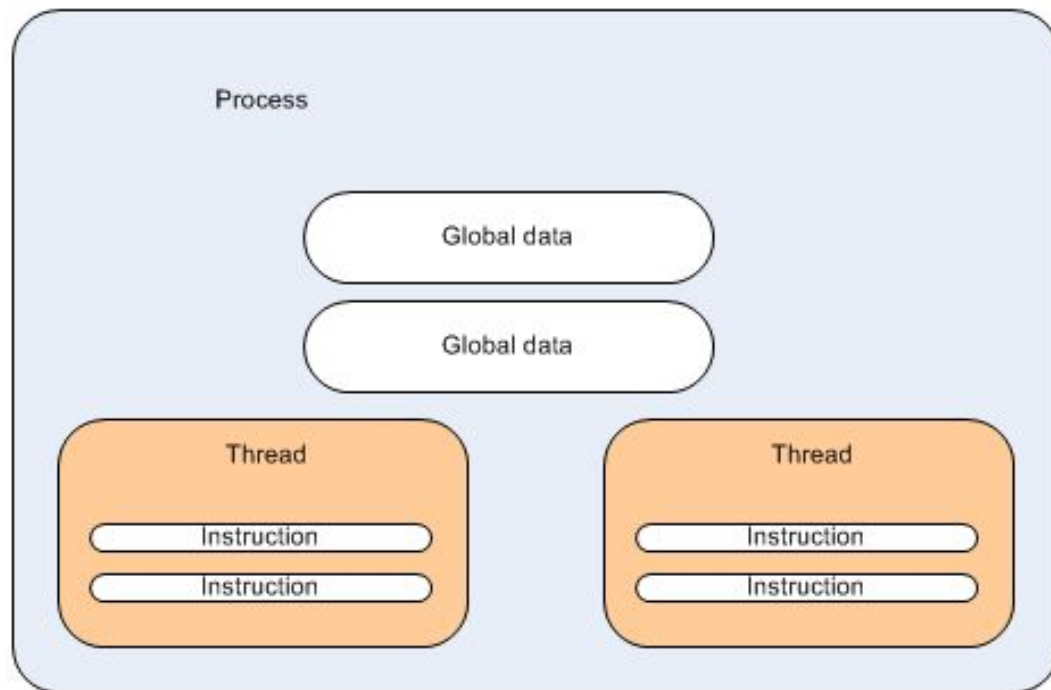


5. Boost.Thread



Boost.Thread

Free Threading





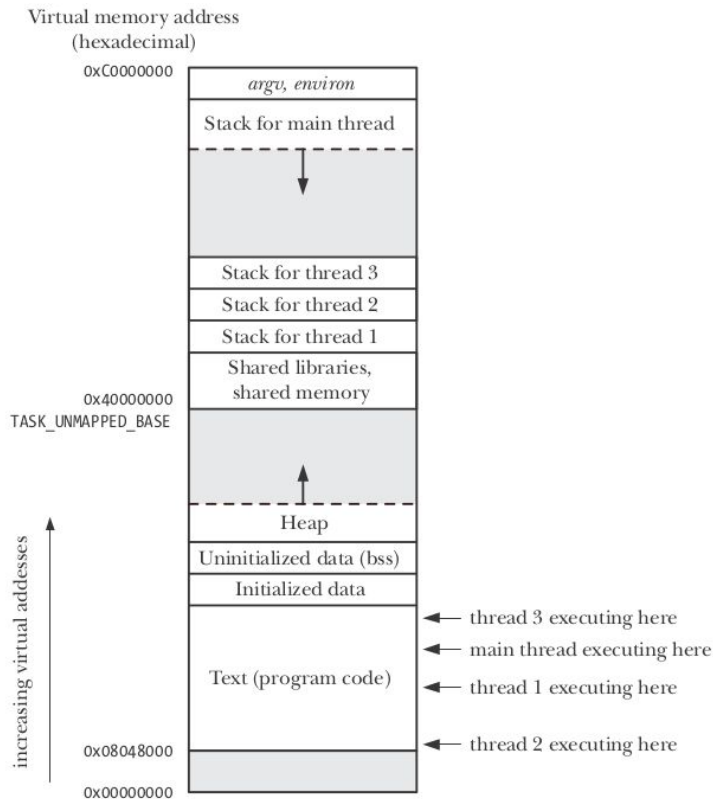
Tại sao lại sử dụng Boost.Thread

- ❖ Nền tảng độc lập
- ❖ Sử dụng ít tài nguyên hơn vì dùng chung không gian và bộ nhớ
- ❖ Tạo và xóa nhanh hơn
- ❖ Là một C++ API hiện đại
- ❖ Cú pháp tương thích với `std::thread` C++11
- ❖ Thiết kế các chương trình đa nhiệm.

Vấn đề khi sử dụng Boost.Thread



Quá nhiều
Threads





Vấn đề khi sử dụng Boost.Thread

Cùng truy
cập một
vùng dữ liệu

	THREAD 1	THREAD 2
Original Code	<pre>t = x x = t + 1</pre>	<pre>u = x x = u + 2</pre>
Interleaving #1	<pre>(x is 0) t = x x = t+1 (x is 1)</pre>	<pre>u = x x = u + 2 (x is 2)</pre>
Interleaving #2	<pre>t = x x = t + 1 (x is 1)</pre>	<pre>(x is 0) u = x x = u + 2 (x is 2)</pre>
Interleaving #3	<pre>(x is 0) t = x x = t + 1 (x is 1)</pre>	<pre>u = x x = u + 2 (x is 3)</pre>



6. Boost.Flyweight



Giới thiệu Boost.Flyweight

Là mẫu hình thiết kế áp dụng cho
lượng lớn đối tượng dùng chung dữ liệu



```
struct person
```

```
{
```

```
    int id;
```

```
    string city;
```

```
}
```

```
int main()
```

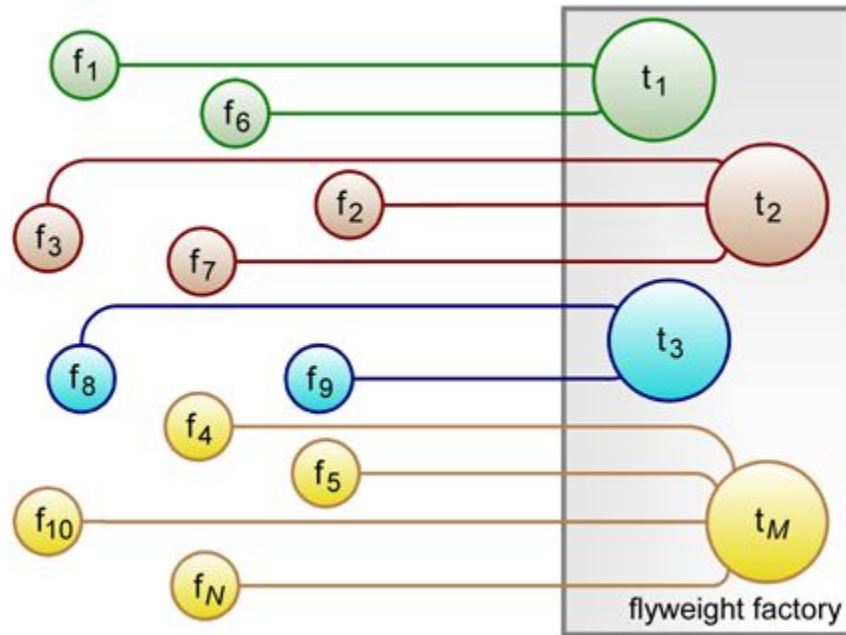
```
{
```

```
    vector<person> persons;
```

```
    for (int i = 0; i < 100000; ++i)
```

```
        persons.push_back({i, "Sai Gon"});
```

```
}
```



N đối tượng, M dữ liệu



Sử dụng

```
#include <boost\flyweight.hpp>

using namespace ::boost;
using namespace ::boost::flyweights;
```



Sử dụng

Không sử dụng flyweight

```
struct user_entry
{
    string first_name;
    string last_name;
    int age;
};
```

Sử dụng flyweight

```
struct user_entry
{
    flyweight<string> first_name;
    flyweight<string> last_name;
    int age;
};
```



Sử dụng

`flyweight<T>`

T là kiểu biến:

- Gán được
- So sánh được
- Sử dụng được với Boost.Hash

Ví dụ: `int`, `float`, `string`, ..



Hiệu suất

Bài test so sánh xử lý với string trong thư viện chuẩn và flyweight<string>

Code http://www.boost.org/doc/libs/1_65_1/libs/flyweight/example/perf.cpp

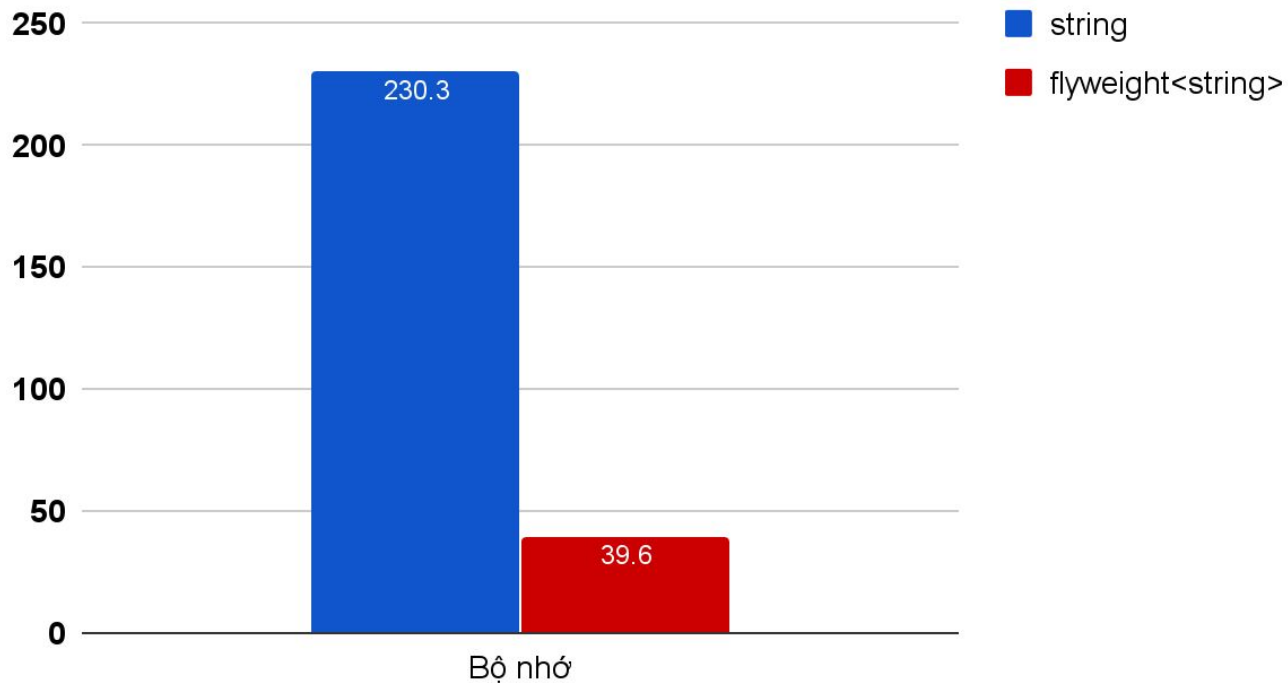
File text <http://www.gutenberg.org/ebooks/2000.txt.utf-8>

Visual Studio 2013, Windows 10, Core i5 7200U, 4GB ram

Hiệu suất



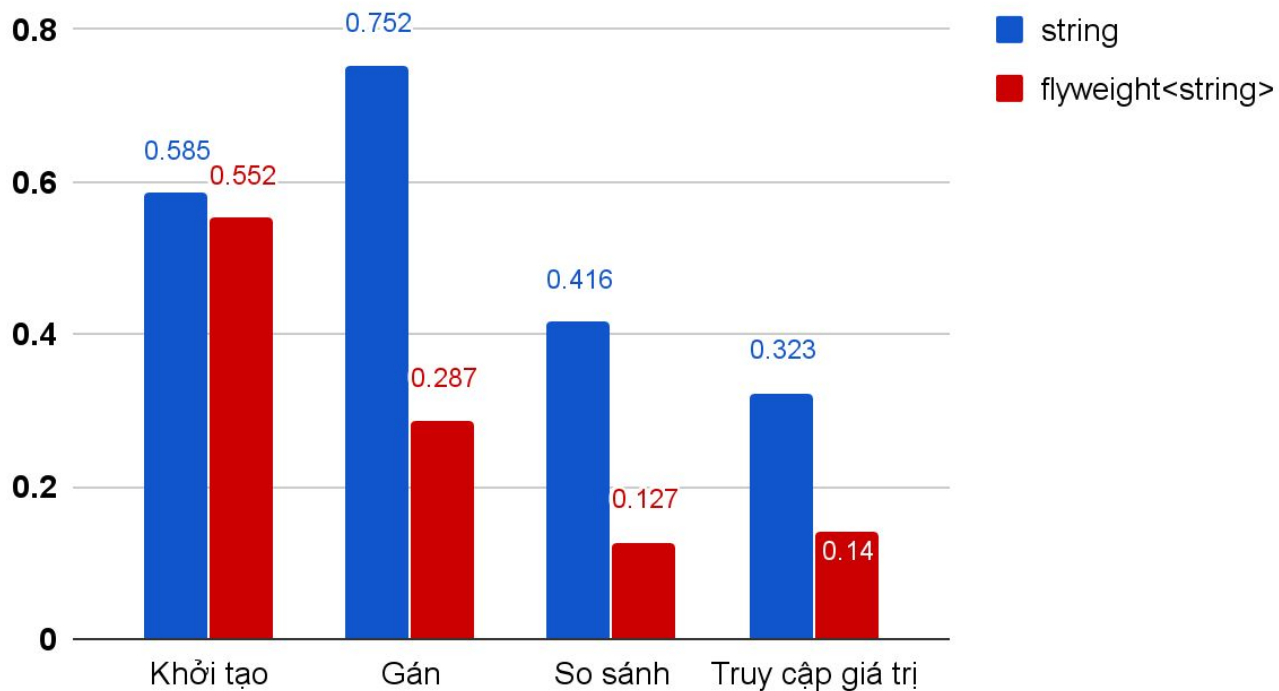
Bộ nhớ tiêu thụ (đơn vị MB)



Hiệu suất



Thời gian thực hiện (đơn vị giây)





Cảm ơn!