Boost+ライブラリー周の旅

ver.1.53.0

高橋晶 (Akira Takahashi)
id:faith_and_brave
@cpp_akira

Boost.勉強会#11 2013/06/01(土)

Boost C++ Libraries

はじめに

この発表は、Boostのなるべく全てのライブラリを紹介していこうという企画です。
毎回、これまでの差分を紹介し、全体にマージした資料を同時に公開しています。

前回は、Boost 1.50.0までのライブラリを紹介しました。
今回は1.51.0と1.53.0で追加されたライブラリを紹介していきます。

Boost C++ Libraries

Boostとは

- C++標準化委員会の人たちが作ったC++のライブラリ群
- 普段のプログラミング全般で使える基本的なものから、専門特化したものまでいろいろなライブラリがある
- Google、Intel、Adobeも開発に関わっている
- ライセンスはBoost Software License 1.0
  - 無償で商用利用可能
  - 著作権表記の必要なし
  - ソースコードの改変自由

Boost C++ Libraries

本日紹介するライブラリ

1. Atomic
2. Lockfree
3. Coroutine
4. Multiprecision
5. Odeint

Boost C++ Libraries

Atomic

C++11アトミックライブラリのC++03実装。

共有データ

```
int data;
atomic<bool> ready(false); // アトミックなbool型変数
```

スレッド1

```
while (!ready.load(memory_order_acquire)) {} // 書き込まれるまで待機
std::cout << data << std::endl; // 3が出力されることが保証される
```

スレッド2

```
data = 3;
ready.store(true, memory_order_release); // 書き込み
```

Boost C++ Libraries

Lockfree

Boost.Atomicベースのロックフリーコンテナライブラリ。
キュー、スタック、優先順位付きキューの実装がある。

```
lockfree::queue<int> que(128);

void producer() {
    for (int i = 0;; ++i) {
        while (!que.push(i)) {}
    }
}

void consumer() {
    for (;;) {
        int x = 0;
        if (que.pop(x))
            std::cout << x << std::endl;
    }
}
```

Boost C++ Libraries

Coroutine

処理の中断と再開を制御する、コルーチンのライブラリ。

```
typedef coroutine<void()> coroutine;

void f(coroutine::caller_type& coro) {
    for (int i = 0; i < 10; ++i) {
        std::cout << "a ";
        coro(); // 中断
    }
}

coroutine c(f); // 関数f()をコルーチン実行可能にする
for (int i = 0; i < 10; ++i) {
    std::cout << "b ";
    c(); // 再開
}
```

```
a b a b a b a b …
```

Boost C++ Libraries

Multiprecision

多倍長演算ライブラリ。無限長の整数などを扱える。

```
// 100の階乗を計算する
cpp_int x = 1;
for(std::size_t i = 1; i <= 100; ++i)
   x *= i;

std::cout << x << std::endl;
```

```
933262154439441526816992388562667004907159682643816214685929638952175999932299156089414639761565182862536979208272237582511852109168640000000000000000000
```

Odeint

常微分方程式を解くためのライブラリ。カオス理論、振り子の計算など。以下はローレンツ方程式の例。

```
const double sigma = 10.0;
const double R = 28.0;
const double b = 8.0 / 3.0;
typedef boost::array< double , 3 > state_type;

void lorenz( const state_type &x , state_type &dxdt , double t ) {
    dxdt[0] = sigma * ( x[1] - x[0] );
    dxdt[1] = R * x[0] - x[1] - x[0] * x[2];
    dxdt[2] = -b * x[2] + x[0] * x[1];
}
void write_lorenz( const state_type &x , const double t )
{ cout << t << '¥t' << x[0] << '¥t' << x[1] << '¥t' << x[2] << endl; }

state_type x = { 10.0 , 1.0 , 1.0 }; // initial conditions
integrate( lorenz , x , 0.0 , 25.0 , 0.1 , write_lorenz );
```

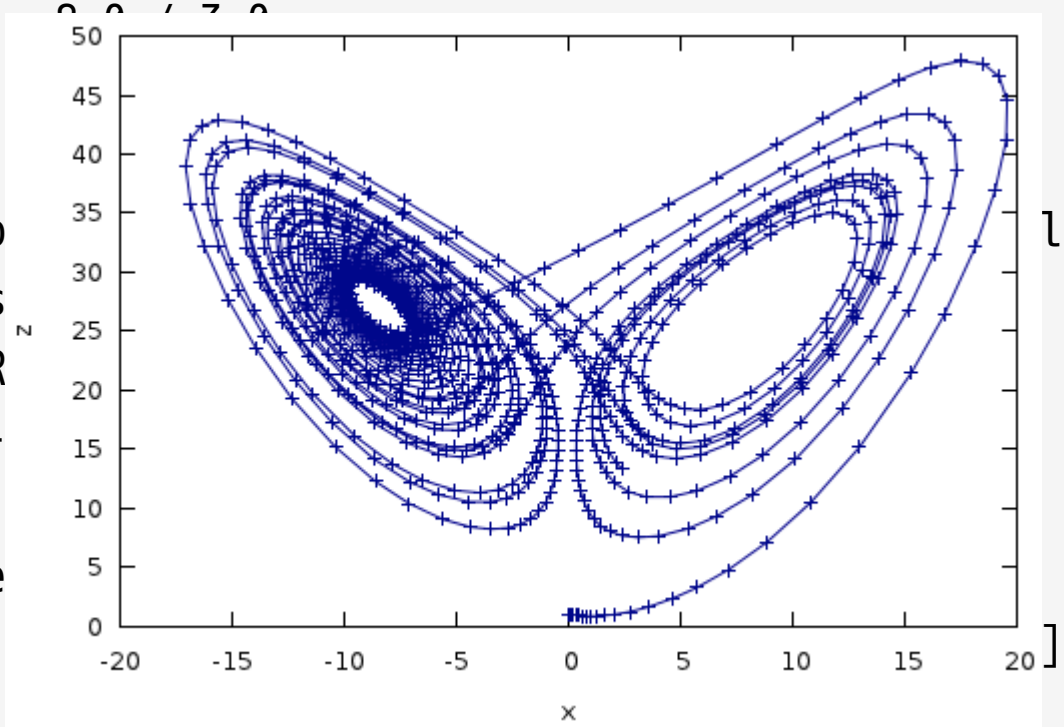
Odeint

常微分方程式を解くためのライブラリ。カオス理論、振り子の計算など。以下はローレンツ方程式の例。

```
const double sigma = 10.0;
const double R = 28.0;
const double b
typedef boost::

void lorenz( co                                    le t ) {
    dxdt[0] = s
    dxdt[1] = R
    dxdt[2] = -
}
void write_lore
{ cout << t <<                                     ] << endl; }

state_type x = { 10.0 , 1.0 , 1.0 }; // initial conditions
integrate( lorenz , x , 0.0 , 25.0 , 0.1 , write_lorenz );
```



Boost C++ Libraries

まとめ(?)

- ここでは差分のみを紹介しましたが、1.53.0までのライブラリをまとめたスライドも別途用意しています。
  全体を知りたい方はそちらを参照してください。

Boostライブラリー周の旅 1.53.0(merge)

http://www.slideshare.net/faithandbrave/boost-tour-1530-merge