Boost.ライブラリー周の旅

ver.1.50.0

高橋晶 (Akira Takahashi)
id:faith_and_brave
@cpp_akira

Boost.勉強会#10 2012/07/28(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 1.48.0までのライブラリを紹介しました。
今回は1.49.0と1.50.0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. Heap
2. Algorithm
3. Functional/OverloadedFunction
4. LocalFunction
5. Utility/IdentityType

Heap 1/2

優先順位付きキューのデータ構造

```
boost::heap::fibonacci_heap<int> que; // フィボナッチヒープ

que.push(3);
que.push(1);
que.push(4);

while (!que.empty()) {
    std::cout << que.top() << std::endl;
    que.pop();
}
```

```
4
3
1
```

Heap 2/2

Boost.Heapの特徴：

- 要素の修正ができる (Mutability)
- イテレータを持っている (Iterators)
- マージできる (Mergable)
- 安定 (Stability)

Boost C++ Libraries

Algorithm 1/3

アルゴリズム集。文字列検索、C++11アルゴリズム、
ユーティリティが含まれる

```
std::string text =
   "the stick, and made believe to worry it: then Alice dodged behind a";
std::string pattern = "behind";

// BM法で文字列検索
decltype(text)::const_iterator it =
   boost::algorithm::boyer_moore_search(text.begin(), text.end(),
                                        pattern.begin(), pattern.end());

if (it != text.end()) std::cout << "found" << std::endl;
else                  std::cout << "not found" << std::endl;
```

```
found
```

Boost C++ Libraries

Algorithm 2/3

アルゴリズム集。文字列検索、C++11アルゴリズム、ユーティリティが含まれる

```
const std::vector<int> v = {2, 4, 6, 8, 10};

// 全ての値が偶数かを調べる
bool result = boost::algorithm::all_of(v, is_even);

std::cout << std::boolalpha << result << std::endl;
```

```
true
```

Boost C++ Libraries

Algorithm 3/3

アルゴリズム集。文字列検索、C++11アルゴリズム、
ユーティリティが含まれる

```
using boost::algorithm::clamp;

// xを0〜10の範囲に丸める : min(max(a, x), b)

int x = 11;
x = clamp(x, 0, 10); // x == 10

int y = -1;
y = clamp(y, 0, 10); // x == 0
```

Boost C++ Libraries — Functional/OverloadedFunction

複数の関数から、オーバーロードする
関数オブジェクトを作る。

```
const std::string& identity_s(const std::string& s) { return s; }
int identity_n(int x)                                { return x; }
double identity_d(double x)                          { return x; }

boost::overloaded_function<
    const std::string& (const std::string&),
    int (int),
    double (double)
> identity(identity_s, identity_n, identity_d);

std::string s = "hello"; s = identity(s);
int n = 2;               n = identity(n);
double d = 3.14;         d = identity(d);
```

Local Function

ローカル関数を定義する

```
int main()
{
    int sum = 0;

    void BOOST_LOCAL_FUNCTION(bind& sum, int x) {
        sum += x;
    } BOOST_LOCAL_FUNCTION_NAME(add);

    const std::vector<int> v = {1, 2, 3, 4, 5};
    boost::for_each(v, add);

    std::cout << sum << std::endl;
}
```

```
15
```

Utility/IdentityType

関数マクロに渡す引数でカンマを付けれるようにする

```
std::map<int, int> m = {{1, 3}, {2, 4}};

BOOST_FOREACH(
  BOOST_IDENTITY_TYPE((std::map<int, int>))::const_reference x, m) {
    std::cout << x.first << "," << x.second << std::endl;
}
```

```
1,3
2,4
```

Boost C++ Libraries

まとめ(?)

- まとめはとくにありません。
- ここでは差分のみを紹介しましたが、1.50.0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 1.50.0(all)

http://www.slideshare.net/faithandbrave/boost-tour-1480-all