07.09.2011

Boost
Compressed Pair

Raluca Pahontu

dkfz.
DEUTSCHES
KREBSFORSCHUNGSZENTRUM
IN DER HELMHOLTZ-GEMEINSCHAFT

Compressed Pair

struct S {};
sizeof(S) ?
(msvc: 1 byte)

“Why is the size of an empty class not zero?”

Requirement from the c++ standard.
From Bjarne Stroustrup:
“To ensure that the addresses of two different objects will be different
For the same reason, "new" always returns pointers to distinct objects”

Compressed Pair

- std::pair vs boost::compressed_pair

```
1  #include <boost/compressed_pair.hpp>
2  #include <utility>
3
4  int main(int argc, char* argv[])
5  {
6    struct S {};
7
8    std::pair< char , S > p1;
9    boost::compressed_pair< char, S > p2;
10
11   printf("sizeof(p1)=%d sizeof(p2)=%d\n",sizeof(p1),sizeof(p2));
12
13   return 0;
14 }
```

- sizeof(p1) = 2
- sizeof(p2) = 1

Compressed Pair

- If the template arguments are empty classes, then the "**empty base-class optimisation**" is applied to compress the size of the pair

- **Empty base-class optimisation** technique:

```
5  struct S ();
6
7  struct T1 { S s; char c; };
8  struct T2 : public S { char c; };
9
10 // sizeof( T1 ) == 2
11 // sizeof( T2 ) == 1
```

Compressed Pair

From Google c++ coding guidelines:

Following boost libraries are permitted:

- Call Traits from boost/call_traits.hpp
- **Compressed Pair from boost/compressed_pair.hpp <- HERE**
- Pointer Container from boost/ptr_container
- Array from boost/array.hpp
- The Boost Graph Library (BGL) from boost/graph
- Property Map from boost/property_map