Don’t forget: std::pair does lexicographical ordering, so you don’t have to

devblogs.microsoft.com/oldnewthing/20181226-00

December 26, 2018

Raymond Chen

A feature perhaps not as widely known as I thought is that the `std::pair` type performs lexicographical ordering, so you don’t have to.

```
// Suppose we record versions as std::pair<int, int>
// where the first is the major version
// and the second is the minor version.

std::map<ComponentId, std::pair<int, int>> requiredVersions;

bool IsSupported(ComponentId id, std::pair<int, int> actualVersion)
{
 auto item = requiredVersions.find(id);
 if (item == requiredVersions.end()) {
  return true;
 }

 auto& requiredVersion = item->second;

 if (actualVersion.first > requiredVersion.first ||
     (actualVersion.first == requiredVersion.first &&
      actualVersion.second >= requiredVersion.second)) {
  return true;
 }

 return false;
}
```

First, we try to find the component in our list of required versions. If it’s not found, then the component has no version requirements, and we say, “Sure, it’s supported!” (This is just an example. Maybe you want to say that if it’s not on the list, then it’s not supported at all.)

Otherwise, we check the actual version number against the required version. If the major version is greater, or if the major version is equal but the minor version is greater or equal, then we decide that we have met the minimum requirements.

Writing the comparison of major and minor versions is easy to get wrong,

So don't write the code that's easy to get wrong. Let the standard library do it.

```
bool IsSupported(ComponentId id, std::pair<int, int> actualVersion)
{
  auto item = requiredVersions.find(id);
 if (item == requiredVersions.end()) {
  return true;
 }

 auto& requiredVersion = item->second;

 return actualVersion >= requiredVersion;
}
```

Bonus chatter: I saw this mistake in some code that used the `std::pair` as the key in a map.

```
std::map<std::pair<int, int>, CoolThing> sortedThings;
```

The idea is that the cool things would be sorted by a sort key that behaved like major/minor. The code compared the keys manually, presumably because the author didn't think that `std::pair` supported the relational operators.

But of course `std::pair` supports the relational operators bcause that's one of the prerequisites for being the key of a `std::map`. (Okay, technically, `std::map` requires only `operator<`, but once you have `operator<`, you can synthesize the rest.)

Raymond Chen

Follow