ORACLE®

ORACLE®

G++ diagnostics: present and (near) future

Paolo Carlini

ORACLE®
CONSULTING

Renewed interest...

- Between 4.6 and 4.7 renewed interest in some recent and not-so-recent diagnostic issues, eg:
  - c++/48934 (“no rejection reason given for SFINAE”)
    - Patch contributed by Nathan Froyd
  - **-Wdelete-non-virtual-dtor**
    - Patch contributed by Jonathan Wakely
  - many smaller issues fixed (ICEs on invalid, error recovery...)

- We should thank Clang++ for a lot of this ;)

Renewed interest (2)

- In 4.8 finally we also have “caret diagnostics”
  - Patch contributed by Manuel Lopez Ibanez
  - Quite similar to EDG
  - Location information still needs work (lately making progress)
  - Can be disabled
    - What about expressions in that case?!
  - No ranges (about this more from Dodji)
  - Should we somehow keep the source code around instead of reloading it in case of error??
    - I **do** see the delay!
    - See the wiki for details

-Wunused-local-typedefs

- Resolving libstdc++/33084 boiled down to fixing a library (ie, `<valarrray>`) function with this body:

```
typedef _BinClos<_Name, _Constant, _ValArray,
                 _Tp, _Tp>   _Closure;

typedef typename __fun<_Name, _Tp>::result_type _Rt;

return _Expr<_Closure, _Tp>(_Closure(__t, __v));
```

- Note the pointless typedef...

-Wunused-local-typedefs (2)

- … indeed we had a trivial typo:

```
_Rt → _Tp
```

- In PR33255 I wondered if we could do something about this!
- In 4.7, the new **-Wunused-local-typedefs** warning (implemented by Dodji Seketeli) detects such sort of very suspect “unused” typedef.
- In 4.8 is enabled by default as part of -Wunused
  - had to make sure we don't give spurious warnings in some special cases involving system headers

-Wzero-as-null-pointer-constant

- In C++11 there is a proper type for null pointer constants, std::nullptr_t, with value nullptr (*), eg:

```
int* p = nullptr;
```

preferably replaces:

```
int* p = 0;
```

- Likewise in conditionals, everywhere.

(*) http://en.wikipedia.org/wiki/C++11#Null_pointer_constant

-Wzero-as-null-pointer-constant (2)

- The new **-Wzero-as-null-pointer-constant**, available in c++98 mode too, detects such uses of the legacy “0” literal to mean null pointer and helps moving code to C++11.
- First blush, it seems a very trivial thing – I did the work mostly to address a PR and, while doing that, learning more about the C++ front-end – but apparently quite a few users are finding it useful...
- … because we got **many** PRs when the features was still buggy, and one for 4.7.0 too!
  - The latter fixed for 4.7.1, was about “0” in default arguments.

Conclusions (for Paolo's talk)

Please add to:

http://gcc.gnu.org/wiki/Better_Diagnostics

Thanks!

Random bibliography

- Some recent C++11 books:

  http://www.manning.com/williams/
  http://www.cppstdlib.com/