

Google Summer of Code 2014

Manasij Mukherjee

Introduction

- Manasij Mukherjee (manasij7479@gmail.com)
- From Kolkata, India
- 3rd year undergraduate student at St. Xavier's College
- Studying Computer Science

Original Project Goal

- (Cling) Name Autodetection and Library Autoloading
 - with the guidance of Vassil Vassilev.
- Basically, interactive suggestions.
- If you forget to `#include` something, instead of giving up, we want the Interpreter to help you find out what is missing.

Example:


```
[cling]$ Graph g = Graph::ReadGraphFromFile("a.txt");  
Autoloading Diagnostics: Graph can be found in "include/graph.hpp"  
Autoloading Prompt: Include and continue ? (y/n):|
```


- Ultimate goal was to have this extension function without user intervention, when possible.

What Was Accomplished:

- A robust utility for generating Forward Declarations.
- Interactive hints for Incomplete Types:

```
[cling]$ .T test.h fwd.h
[cling]$ #include "fwd.h"
[cling]$ TestClass testObj;
In file included from input_line_3:1:
./fwd.h:5:46: warning: Note: 'TestClass' can be found in ./test.h
class __attribute__((annotate("./test.h"))) TestClass;
input_line_4:2:12: error: variable has incomplete type 'TestClass'
TestClass testObj;
```

 Injects annotated forward declarations.

 Hint

- More practical use cases for the Callback system in cling:
 - This helped in uncovering a few design and implementation oversights in the existing code.

What else could have been done:

- Having the Hint system work for all cases.
Most importantly, templates and functions.
- More Integration with Clang.
- Coding some 'Policy' about how this system would be practically used.
For example, generating forward declarations for a source tree.

Why Forward Declarations ?

- Forward Decl's convey information that 'something' exists and exactly what it is.
 - No implementation details retained.
- We attach annotations on these forward declarations.
 - These annotation strings contain the location of the full definition.
- Reduced memory footprint for the environment including these forward declarations.

Tricking the Compiler

- Some Forward Declarations incompatible with the Original Source files.
- Default Arguments (both template and function) can not be in two places:
 - So we remove one of them (transparently)..Simple!
- Enum forward declarations must fix a type for the underlying data.
 - We just remove this 'fixed' notion just before the original file is included.

Experience Gained

- **Communicating:**

Explain the WHY and not the WHAT.

The WHAT should be evident from the code!

- **Technical Skills:**

I am much more confident with version control.

Large codebases are no longer that intimidating.

Now able to 'see' overall design with minimal effort.

The End

Thank You