DIVA - Debug Information Visual Analyzer

Phillip Power  <phillip.power@sony.com>
EuroLLVM ’17 (March 2017)
Introducing the problem

1. /* example_15.cpp */
2. class bar {
3. public:
4.   void foo();
5. }
6. void bar::foo()
7. {
8. }
9. bar b;
10. 
11. bar b;
Introducing DIVA

DIVA output is:
  a) A high level view of ELF debug information
  b) Same output regardless of the DWARF layout or spec version

Use cases:
  1. Visualize / inspection of the debug information
  2. Compare debug information from two compilations
  3. Regression testing
  4. Creation of debug information reproducible test case
Visualize the debug information

1. /* example_15.cpp */
2. class bar {
3. public:
4.   void foo();
5. }
6. void bar::foo()
7. {
8.   {InputFile} "example_15.o"
9.   {CompileUnit} "example_15.cpp"
10.   {Source} 'example_15.cpp'
11.   {Class} "bar"
      {Function} "foo" -> "void"
          - Is declaration
          {Parameter} "" -> "* bar"
12.     {Function} "foo" -> "void"
13.       - Declaration @ example_15.cpp,4
14.         {Parameter} "this" -> "* bar"
15.     {CodeLine}
16.     {CodeLine}
17.   {Variable} "b" -> "bar"
Compare two compilations

Use standard diff tools to compare debug info in two ELF files
— This is not possible using DWARF directly because it can be structured in many different ways

```
{InputFile} "example_15.o"
  {CompileUnit} "example_15.cpp"
  {Source} 'example_15.cpp'
    2     {Class} "bar"
    4     {Function} "foo" -> "void"
       - {Is declaration}
         {Parameter} **"bar"**
        7     {Function} "foo" -> "void"
           - Declaration @ example_15.cpp,4
          {Parameter} "this" -> "bar"
    8     {CodeLine}
    9     {CodeLine}
   11     {Variable} "b" -> "bar"

{InputFile} "example_15.o"
  {CompileUnit} "example_15.cpp"
  {Source} 'example_15.cpp'
    2     {Class} "bar"
    4     {Function} "foo" -> "void"
       - {Is declaration}
         {Parameter} **"void"**
        7     {Function} "foo" -> "void"
           - Declaration @ example_15.cpp,4
          {Parameter} "this" -> "void"
    8     {CodeLine}
    9     {CodeLine}
   11     {Variable} "b" -> "bar"
```
Comparing generated debug info with the expected

1 Expected debug information
2 Non-optimized debug info
3 Optimized build debug info
Roadmap

Sony are using DIVA:
- to create a debug information regression suite
- to investigate what optimizations do to the debug information
- to aid DWARF 5 implementation in LLVM

DIVA is available in GitHub
- I propose that it is used in the LLVM project for testing DWARF
- I would like to hear your views and requests for features

https://github.com/SNSSystems/DIVA
Thank you to the author of DIVA:
   — Carlos Enciso

and some contributors:
   — Phil Camp
   — Sean Eveson
   — Philip Douglas
   — Tom Weaver

https://github.com/SNSystems/DIVA