

ELF Tool Chain Project

<http://sourceforge.net/apps/trac/elftoolchain/>

Developers

- Joseph Koshy <jkoshy@FreeBSD.org>
- Kai Wang <kaiw@FreeBSD.org>
- Hyogeol Lee <hyogeollee@gmail.com>
- S.Sam Arun Raj

Goal

To create BSD licensed implementations of essential compilation tools for the ELF object file format.

libelf

- Implements the SVR4 ELF(3) API
- Basis for many other tools

- Targeting a summer 0.5 milestone with significant improvements

Tuesday, May 11, 2010

libelf: we've improved libelf a lot over the last two years.

I think these improvements and bug fixes will be merged back to FreeBSD after our 0.5 milestone (sometime this summer). This is important since libelf is the base for our other tools and libraries.

libdwarf

- Library for parsing DWARF debug information
- API compatible with the SGI libdwarf
- Also has support for producing dwarf information which can be used by as(1)
- Based on John Birrell's work

Tuesday, May 11, 2010

libdwarf: Based on jb@'s initial work, we've made a complete libdwarf implementation that is API compatible with the SGI libdwarf. The new libdwarf includes support for frame and line number and lots of other things missing in jb@'s original version. It also has support for producing dwarf information which can be used by as(1), for example. Of course the new libdwarf can also be merged back to FreeBSD after some more polishing and testing.

Easy tools done

- nm(1)
- size(1)
- string(1)
- c++filt(1)
- addr2line(1)
- readelf(1)
- elfdump(1)
- objcopy(1)
- strip(1)
- etc

Tuesday, May 11, 2010

The minor tools. We've basically finished all the easier tools/counterparts present in GNU binutils like nm(1), size(1), string(1), c++filt(1), addr2line(1), readelf(1), elfdump(1), objcopy(1), strip(1) etc. We also made a library (libelftc) which provides API for demangling C++ symbols among other things. I believe some of the tools are ready to hit the tree...

Plans for major tools

- Implement `as(1)` and `ld(1)`
- Based on `libmc`
- Intend to support LLVM bit object files using `libLTO`

- Planned after `libdwarf` completion

Tuesday, May 11, 2010

The major tools. We have started making plans for `libmc` (a machine code parsing library, serves as base for `as(1)`) and `ld(1)`. I believe we will focus on these two after we're done with `libdwarf`. About debugger, I found there are several other BSDL debugger projects going on, I think we can collaborate somehow. (`libelf` and `libdwarf` will be useful for a debugger)

LLVM support. At the moment I have plans to teach `ar(1)` and `nm(1)` about LLVM bit object files using the LLVM `libLTO` library. If all goes well, our `ar(1)` and `nm(1)` will be able to handle the symbols found in a LLVM bit object. Later when we start with `ld(1)`, we will also teach `ld(1)` to work with LLVM link time optimization.