C Expression Utility for IOCSH

Till Straumann, SSRL/SLAC
<strauman@slac.stanford.edu>

Overview

- Basic idea
- What is Cexp?
- What is Cexp not?
- Features
- Architecture
- Porting work, future
- Demo

Basic Idea

- Available RTEMS shell required ’registering’ functions a la iocsh
- vxWorks shell is nice but has no powerful expressions or data types (try doing a 16bit memory access)
- Needed a framework under RTEMS which can do both:
  - access arbitrary symbols with no registration from the command line
  - evaluate expressions

What is Cexp?

- C expression interpreter with access to ELF symbol table
- command line tool
- trivial (not parameterized) scripts
- simple
- portable
- reasonably small footprint (PPC binary without libreadline: 100k)
- currently tested on
  - RTEMS-ppc
  - linux-i386-gnu
  - linux-ppc-gnu
  - solaris-sparc-cc
  - solaris-sparc-gnu
  - alpha-tru64unix-cc
  - alpha-tru64unix-gnu

What Is Cexp Not?

- A shell (rather part of a shell) - no job control, no command interpreter, no control structs
- A monitor - no memory inspection, thread control/info
- A dynamic loader

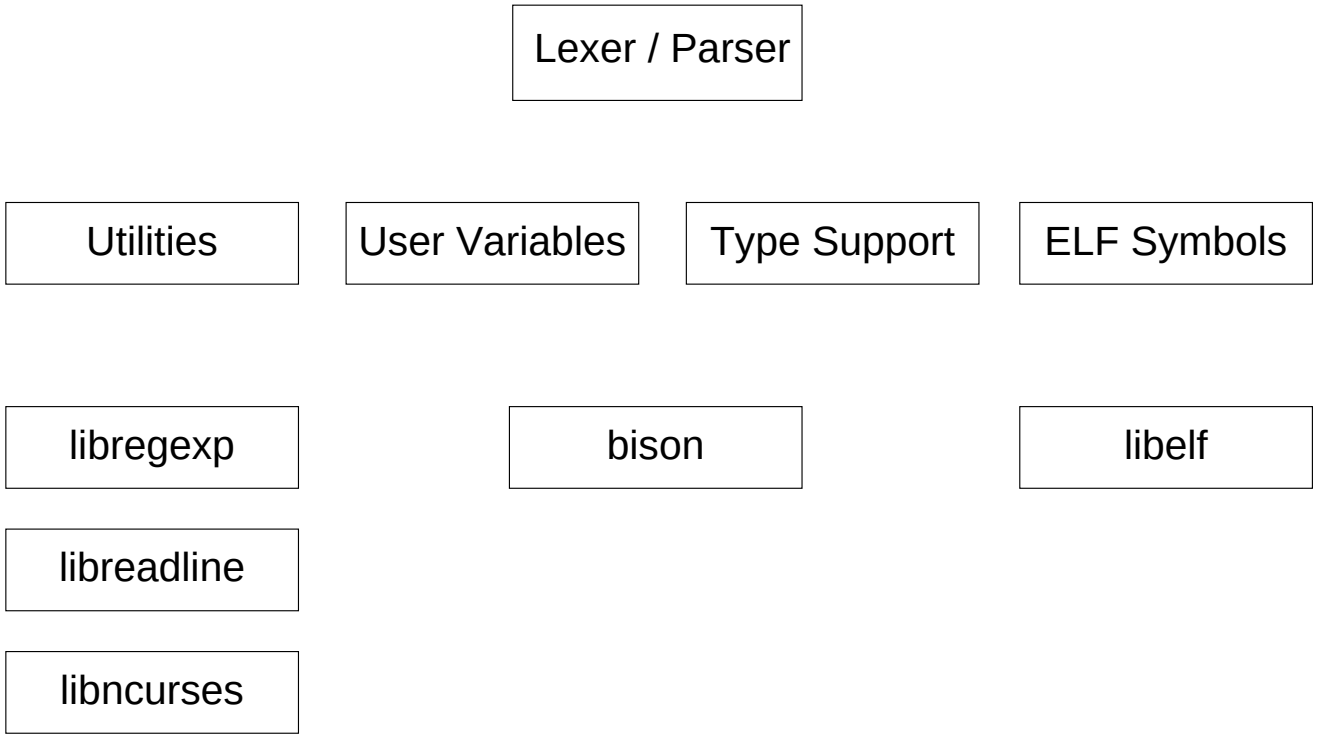
Features

- C-style expression interpreter (no ’?/:’, ’[ ]’, ’->’, ’.’)
- Access to ELF symbol table
- Convenience routines for symbol lookup (regexp support)
- Basic data type support (unsigned char, short, long, double, pointers)
- User definable variables

Examples

- dbl()
- lkup("^[eE]pics")
- sqrt = (double(*)())sqrt@@GLIBC_2.0
- printf("Square root of 2: %g\n", sqrt(2.0))
- *(char*)&somevar &= 0xfe<<1
- somedevProbe(0xdeadbeef) &&
  dbLoadRecords("someRecs.db")

Cexp Architecture



Future

- (optional) iocsh integration
- make symbol lookup more portable (more general than ELF)
- use (enhanced) OSI EPICS symbol table API
  - possible implementation: BFD library as a default
  - override for specific systems, as needed
- BFD could also help supporting disassembling and/or dynamic loading

Demo