Lisp in Summer Projects Submission

Submission Date	2013-10-19 17:11:19
Full Name	Jovan Trujillo
Country	US
Project Name	Vacietis
Type of software	API
General category	development tool
LISP dialect	Commmon Lisp
GitHub URL	https://github.com/navoj/Vacietis
Did you start this project?	No, I'm modifying or extending an existing project.
Which file or directory contains the majority of your work?	Vacietis/libc/include/Sys/time.lisp
Briefly describe your modifications	Vacietis is a C interpreter for Common Lisp. It is still missing a few important C libraries. I worked on getting a basic time.h stub working. It's not complete. I didn't have much time to work on the project :-(
Project Description	I want to describe my project in this form.
Purpose	Vacietis is a C compiler for Common Lisp Systems. You can read standard C code and Vacietis will translate the code into Lisp functions. There is also the possibility of compiling the code into an executable.
Function	You will need clisp and quicklisp installed for this to work. Download Vacietis in your quicklisp/local-projects folder. Run (ql:quickload "vacietis") Load C file with (vacietis:load-c-file "code.c") Run C code with (vacietis:run-c-program *package*) Compile to an executable using the vcc program.
Motivation	I want Lisp to take over the world so I thought it would be useful to have a C interpreter in Lisp to make future projects

	easier. I want to be able to do everything in Lisp on any system so gluing existing source code into Lisp helps make that possible. I imagine it would be useful to speed up development of the Movitz lisp kernel into a full operating system by writing wrappers around Linux drivers for various hardware. Vacietis could also be a very flexible C development environment for dynamically modifying and debugging C code. But my personal reasons are to create a more homogeneous software development environment where I can take anybody's work and easily integrate it into a Lisp platform regardless of what languages and operating systems they are using.
Audience	Software developers who want to do more things in Lisp with less effort.
Methodology	Below is Vladimir Sedach's explanation of how to use Vacietis. From examining the code he ported the zeta-c compiler to common lisp and rewrote the parser to utilize readtables. Vacietis is a C compiler for Common Lisp systems. Vacietis works by loading C code into a Common Lisp runtime as though it were Lisp code, where it can then be compiled or evaled. The loaded C code has the same function calling convention as regular CL code and uses the same numerical representations. C memory is backed by regular Common Lisp arrays. Vacietis comes with a libc implemented in portable Common Lisp. * INSTALLING: ---------- You can obtain Vacietis from github: git clone https://github.com/vsedach/Vacietis.git All Vacietis dependencies are available via Quicklisp (http://www.quicklisp.org/). If you put Vacietis in quicklisp/local-projects/ you can just load it with (ql:quickload "vacietis") * USAGE: ----- C code can be read in the same way as regular Lisp code by using readtables: (let ((*readtable* vacietis:c-readtable) (vacietis:*compiler-state* (vacietis:make-compiler-state))) (read)) The Vacietis reader keeps track of type and preprocessor macro declarations in a compiler state object bound by *compiler-state*. This mechanism is exposed to make it

possible to
create things like C REPLs.

To simplify loading C files, a convenience function is provided that
sets up the readtable, compiler state, and additional debugging
information before calling LOAD:

(vacietis:load-c-file "/foo/bar/file.c")

* COMPILER EXECUTABLE:

The system vacietis.vcc produces a toy C compiler executable that can
take a single-file C program and produce an executable program. Currently it needs CCL, CLISP, or SBCL to work. Sample run:

(ql:quickload "vacietis.vcc") will produce the executable vcc/vcc in
the Vacietis source directory.

$ ./vcc ../test/programs/hanly-83-scanf/main.c

Produces the file a.out in the current directory.

$ ./a.out
Enter 8 numbers separated by blanks or s
>

* TECHNICAL DETAILS:

Vacietis uses the memory model of Common Lisp as is, so sizeof of the
primitive data types (char, int, float etc.) is all 1. This shouldn't
be a problem for most C code, but some C programs claim to be portable
while making assumptions that things can be cast into an array of
chars to be manipulated. These programs won't work under Vacietis.

The basic idea for the Vacietis runtime and memory model comes from
Scott L. Burson's Zeta-C compiler for Lisp Machines:
http://www.bitsavers.org/bits/TI/Explorer/zeta-c/

The technique for representing pointers to arbitrary C lvalues as
closures was first demonstrated by Oleg Kiselyov:
http://okmij.org/ftp/Scheme/pointer-as-closure.txt

The idea for a combined single-pass preprocessor/tokenizer/parser
comes from Fabrice Bellard's TCC: http://bellard.org/tcc/

* OBTAINING CODE AND HELP:

The official Vacietis repository is at:
https://github.com/vsedach/vacietis

There is a Vacietis mailing list on the web:
http://groups.google.com/group/vacietis

Bug reports can be sent to the mailing list:
http://groups.google.com/group/vacietis
the github issue tracker:
https://github.com/vsedach/vacietis
or directly to the author:
vsedach@gmail.com

* UNIT TESTS:

(ql:quickload "vacietis.test")
(vacietis.test:run-tests)

The Vacietis test suite includes a variety of code that tests the
compiler and libc.

* TODO:

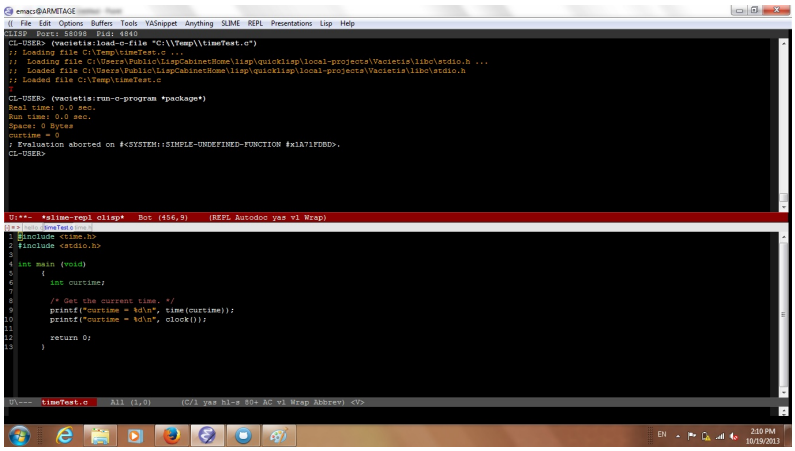
- pointer scaling
- enums: assignment of arbitrary values to enum labels
- struct call by value
- pass arguments to main()
- implement overloading class scope correctly (see H&S p. 147)
- libc stdio: binary streams
- libc stddef: offsetof
- libc signal
- libc stdlib: div/ldiv, srand, exit cleanup, bsearch, qsort
- libc time
- libc setjmp

* THINGS THAT PROBABLY WON'T BE SUPPORTED:

- trying to cast arrays of chars to other types (mmap)
- any kind of GCC extension

* LICENSING INFORMATION

Portions of the Vacietis libc may be derived from Zeta-C (released
into the public domain by its author, Scott L. Burson) and Erik Andersen's LGPL-licensed uClibc

	(http://www.uclibc.org/)
Conclusion	I did not have much time to improve Vacietis this summer. From Vladimir's TODO list I started tackling an implementation of the time.h library. It is not complete and doesn't follow the C standard correctly yet. I would like to complete his TODO list and add the possibility of reading make files into lisp for compiling large C programs. Other limitations that need to be addressed is porting the code to other implementations of common lisp. Right now the software is known to only work with CLisp and nothing else. The vcc compiler Vladimir created isn't working right now either with the latest version of CLisp. This needs to be fixed as well.
Build Instructions	You need CLisp 2.48 and quicklisp installed. Download Vacietis into your quicklisp/local-projects folder and run in clisp: (ql:quickload "vacietis")
Test Instructions	Write a simple helloworld.c program to your favorite path. In clisp, call (vacietis:load-c-file "path-to-helloworld.c") In clisp, now run (vacietis:run-c-program *package*)
Describe any bugs or caveats	time.h doesn't work because I do not fully understand how the library development is structured.
Screen shots	 Untitled.png
Official	I have read rules and have abided by them. I am 18 years of age or older. I am not living in Brazil, Quebec, Saudi Arabia, Cuba, Iran, Myanmar (Burma), North Korea, Sudan, or Syria.