CHICKEN

History of an Open-Source Programming Language Project

CHICKEN

A Scheme-to-C Compiler and Runtime system

Started 1998, Open source since 2000

Mature, efficient, portable
(Linux, *BSD, Windows, Android, iOS, MacOS, Solaris, AIX, Haiku, ...)

Excellent FFI, good support for cross-development

5 mailing lists, ~500 subscribers, 7 core developers

IRC #chicken on freenode

Scheme

- Derived from Lisp, created ~1970
- Very simple, very elegant, very general
- Strong academic background
- Code == Data, syntactical abstractions
- Continuations, generalized control-flow, TCO

Cheney-on-the-MTA

- Originally invented by Henry Baker
- (Nearly) direct translation of CPS-converted Scheme to C
- Functions never return, stack used as a 1st generation heap, simple Cheney-style copying collector as 2nd
- Once stack reaches a certain size, longjmp back to base
- Very efficient continuations, full TCO, safe-for-space

"Eggs"

~800 extensions + libraries

Individual eggs can be downloaded and installed by a single command

minimal library manager ("chicken-install")

download from centralized mirrors, but published in a decentralized manner ("THE SYSTEM")

Continuously tested ("Salmonella")
Instances run for several OSes

Initial state / How it began

Found CoMTA paper, read "Compiling with Continuations", got hooked, started coding

covers all crucial concepts in a natural manner

Sources released, after long hesitation

no makefile, no VCS, no clue, but immense amount of work

Considered to be "bug free" ... haha

First steps as OSS

- Strangely enough, people reported bugs and offered suggestions
- Some people love playing with new tools
- And some people are just plain helpful, investing time and effort
- NLP-guy relentlessly threw 50kLOC Scheme codebase at it
- Bugs, bugs, bugs

A user base builds up

- More bugs
- Moved to Savannah, using CVS, created mailing list
- Slowly gains traction, popularity, including Usenet flamewars
- Furious development, trying to please everybody
- Core system grows and becomes more complex
- Egg-system starts to develop to move stuff out of core

Development model shows first signs of strain

- Too many disruptive changes, people start using this for real stuff
- Features accumulate, some necessary, some not
- Complexity piles up, no real processes exist
- Move between different VCSes (CVS, SVN, Darcs, SVN again, later git)
- Move between different build systems (make, autoconf, automake, CMake, make)

Development model shows first signs of strain

- Varying support for different platforms (Windows, Mac)
- Changing dependencies
- Users come and go, some immensely helpful ones, some totally crazy
- First face-to-face meetings
- Something has to be done

Politics

- Standards
- Scheme community is not easy
- People have different ideas + tastes
- Delegation, trust, patience, humbleness (hard)

Trying to get stability into the process

- Rigourous, automated testing
- "Change Requests"
- Patch-review with signoff by core-developer
- Proper release-cycle
- "Stability" branch

Target area

- System-programming / infrastructure
- Embedded systems
- Commercial systems
- Web programming
- Games

Lessons

Make it easy to use with existing projects

Avoid dependencies at all costs

All your assumptions about performance are wrong

Keep things simple

Listen to your users, trust that others may know better

… and still try to keep the system coherent

Mistakes

- Too much fear to fail
- Too many disruptive changes
- “Fixing” things without thinking of the implications
- Featuritis
- Trying to stay in control

Future

- CHICKEN 5 currently being prepared
- Better standards support (R7RS)
- Full support for static linking, easier deployment
- Internal structure cleaned up and modularized
- Cleaning up old cruft, fixing bugs long postponed

Thanks!

felix.winkelmann@bevuta.com