# cgi programs

rise from ash then burn again

upon each request

# Continuity

A continuation based web application library for perl.

Brock Wilcox (awwaiid@thelackthereof.org)

Pittsburgh Perl Workshop 2007

http://continuity.tlt42.org/

# Talk Overview

- Traditional Web Apps
- Continuation Based Web Apps
- Seaside
- Continuity

# Memento

MEMENTO
SOME MEMORIES ARE BEST FORGOTTEN
GUY PEARCE CARRIE-ANNE MOSS JOE PANTOLIANO
MEMENTO
otnemem.com

# Memento

- Do an action
- What did I just do? Ask the requestor
- What did I just do? Look for a sticky note (cookie)

# CGI Applications

- Restart on each request
- Start with a switch statement
- GOTO based on input
- Control flow logic spread out
- Switch statement is state machine

# Run Lola Run

# Run Lola Run

- Capture a moment in time
- Put it in a variable
- Execute it later
- Execute it again?
- Welcome to Continuations

# Continuation Conshminuation

- Program state in a variable
  - Not just data! Control flow too!
- One-shot vs Multi-shot
- Coroutines
- Cooperative Multithreading

# Continuations in Web Apps

- Don't restart app
- Continue on source next line
- Keep program state (local vars)
- Centralize control flow logic

# Other Efforts

- ViaWeb (Lisp, Paul Grahm 1999)
- Seaside (Smalltalk)
- UnCommon Web, SISCweb (Scheme)
- RIFE (Java)
- ... AJAX Turns Things Inside Out Too ...

# Seaside

# Overview of Seaside

- Continuations
- No templates
- Components
- Generated HTML with callbacks

# Seaside Demo

- Counter example
- Halos
- In-page code browser
- Inline link-press callbacks

This is

SWEET

I want this in

# Perl

# Continuations in Perl5

- Nothing built-in
- High-level trickery (Contize)
- Low-level trickery (Coro)
- Other trickery?
- use v6;

use Coro;

# Coro

- Written by Marc Lehmann
- Coroutines for Perl
- Saves
  - callchain
  - lexical variables
  - @_  $_  $@  $/
  - C stack
- Effectively one-shot continuations
- Plus other goodies!

# Continuity

# Continuity

- Keep a list of continuations
- Accept incoming HTTP requests
- Map to a continuation
- Repeat

# Continuity Design

- Library Not Framework
- Unobtrusive
- TMTOWTDI
- Foundation for higher abstraction
- Modular
  - Plugable Adaptors: HttpDaemon, FastCGI, ...
  - Session Mapper

# What is it Good For

- High interaction
- Shared state - games
- Command line conversions
- Clean, organized code

# Continuation Mapping

- Incoming request to continuation
  - Cookie
  - IP
  - Path
  - Query
- Path + Cookie is fun
- Query is my favorite

# Counter Example

- Demonstrates:
  - Server setup
  - Cookie sessions
  - Query sessions
  - Parameters
  - Input loop
  - Subroutines if we make a prompt

# Chat Example

- Demonstrates
  - All things as Counter Example
  - AJAX and COMET
  - Multi-Continuation Sessions

# Issues

- Back Button
- Bookmarks
- Security
- Scalability

# Back Buttons and Bookmarks

- URLs should be pretty
- Many techniques to choose from
- Continuity doesn't decide for you
- Seaside mostly gives up, uses Request ID

# Detect the Back Button

- Send a unique Request ID
- Duplicates imply Back, Reload, or Fork
- Set flag, raise exception, ignore

# Bookmarks

- Bring back the CASE statement
- Goto the right spot :)
- Continue from there
- URLs for major parts of application

# Security

- Easy to centralize Input and Output
- Guessing a ContinuationID is bad

# Scalability

- Difficult to scale?
- Serve dynamic data (not static stuff)
- Session Affinity
- Don't fix it if it ain't broke

# Advanced Techniques

- Subref callbacks
- Multiple continuations per session
- Event.pm timers and watchers

# Future Work

- Get people using it
- Feedback!

# Future Work

- High-level layer(s) on top
- Continuity + Moose + DBIx::Class
- Steal from Seaside
- Steal from Magritte
- Steal from everyone

# Future Work

- Easy DBI interface
  - Coro::Util::eval_fork
- Other languages, keep similar minimal API

Get it on CPAN.

# The End