xh

Spencer Tipping

May 11, 2014

Contents

Part I

Bootstrap implementation

Chapter 1

Self-replication

Note: This implementation requires Perl 5.14 or later, but the self-compiled xh image will run on anything back to 5.10. For this and other reasons, mostly performance-related, you should always use the xh-compiled image rather than bootstrapping in production.

Listing 1.1 boot/xh-header

```
#!/usr/bin/env perl
BEGIN {eval(our $xh_bootstrap = q{
# xh: the X shell | https://github.com/spencertipping/xh
# Copyright (C) 2014, Spencer Tipping
# Licensed under the terms of the MIT source code license

# For the benefit of HTML viewers (long story):
# <body style='display:none'>
# <script src='http://spencertipping.com/xh/page.js'></script>
use 5.010;
package xh;
our %modules;
our @module_ordering;
our %eval_numbers = (1 => '$xh_bootstrap');

sub with_eval_rewriting(&) {
  my @result = eval {$_[0]->(@_[1..$#_])};
  $@ =~ s/\(eval (\d+)\)/$eval_numbers{$1}/eg if $@;
  die $@ if $@;
  @result;
}

sub named_eval {
  my ($name, $code) = @_;
  $eval_numbers{$1 + 1} = $name if eval('__FILE__') =~ /\(eval (\d+)\)/;
  with_eval_rewriting {eval $code};
}

our %compilers = (pl => sub {
  my $package = $_[0] =~ s/\./::/gr;
  named_eval $_[0], "{package ::$package;\n$_[1]\n}";
  die "error compiling module $_[0]: $@" if $@;
});

sub defmodule {
  my ($name, $code, @args) = @_;
  chomp($modules{$name} = $code);
  push @module_ordering, $name;
  my ($base, $extension) = split /\.(\w+$)/, $name;
  die "undefined module extension ’$extension’ for $name"
    unless exists $compilers{$extension};
  $compilers{$extension}->($base, $code, @args);
}

chomp($modules{bootstrap} = $::xh_bootstrap);
undef $::xh_bootstrap;
```

At this point we need a way to reproduce the image. Since the bootstrap code is already stored, we can just wrap it and each defined module into an appropriate `BEGIN` block.

Listing 1.2 boot/xh-header (continued)

```
sub image {
  my @pieces = "#!/usr/bin/env perl";
  push @pieces, "BEGIN {eval(our \$xh_bootstrap = <<’_’)}",
                $modules{bootstrap},
                ’_’;
  push @pieces, "BEGIN {xh::defmodule(’$_’, <<’_’)}",
                $modules{$_},
                ’_’ for @module_ordering;
  push @pieces, "xh::main::main;\n__DATA__";
  join "\n", @pieces;
}
})}
```