

xh

Spencer Tipping

June 12, 2014

Contents

I	xh runtime	2
1	Self-replication	3

Part I

xh runtime

Chapter 1

Self-replication

Listing 1.1 boot/xh-header

```
1  #!/usr/bin/env perl
2  BEGIN {eval(our $xh_bootstrap = q{
3  # xh | https://github.com/spencertipping/xh
4  # Copyright (C) 2014, Spencer Tipping
5  # Licensed under the terms of the MIT source code license
6
7  # For the benefit of HTML viewers (long story):
8  # <body style='display:none'>
9  # <script src='http://spencertipping.com/xh/page.js'></script>
10 use 5.014;
11 package xh;
12 our %modules;
13 our @module_ordering;
14 our %eval_numbers = (1 => '$xh_bootstrap');
15
16 sub with_eval_rewriting(&) {
17     my @result = eval {$_[0]->(@_[1..$#])};
18     $@ =~ s/\(eval (\d+)\)/$eval_numbers{$1}/eg if $@;
19     die $@ if $@;
20     @result;
21 }
22
23 sub named_eval {
24     my ($name, $code) = @_;
25     $eval_numbers{$1 + 1} = $name if eval('__FILE__') =~ /\(eval (\d+)\)/;
26     with_eval_rewriting {eval $code};
27 }
28
29 our %compilers = (pl => sub {
```

```

30 my $package = $_[0] =~ s/\./::/gr;
31 named_eval $_[0], "{package ::$package;\n$_[1]\n}";
32 die "error compiling module $_[0]: $@" if $@;
33 });
34
35 sub defmodule {
36 my ($name, $code, @args) = @_;
37 chomp($modules{$name} = $code);
38 push @module_ordering, $name;
39 my ($base, $extension) = split /\.(\\w+$)/, $name;
40 die "undefined module extension '$extension' for $name"
41 unless exists $compilers{$extension};
42 $compilers{$extension}->($base, $code, @args);
43 }
44
45 chomp($modules{bootstrap} = $::xh_bootstrap);
46 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)

```

1 sub image {
2 my @pieces = "#!/usr/bin/env perl";
3 push @pieces, "BEGIN {eval(our \\\$xh_bootstrap = <<'_')}",
4             $modules{bootstrap},
5             '_';
6 push @pieces, "BEGIN {xh::defmodule('$_', <<'_')}",
7             $modules{$_},
8             '_' for @module_ordering;
9 push @pieces, "xh::main::main;\n__DATA__";
10 join "\n", @pieces;
11 }
12 }}

```