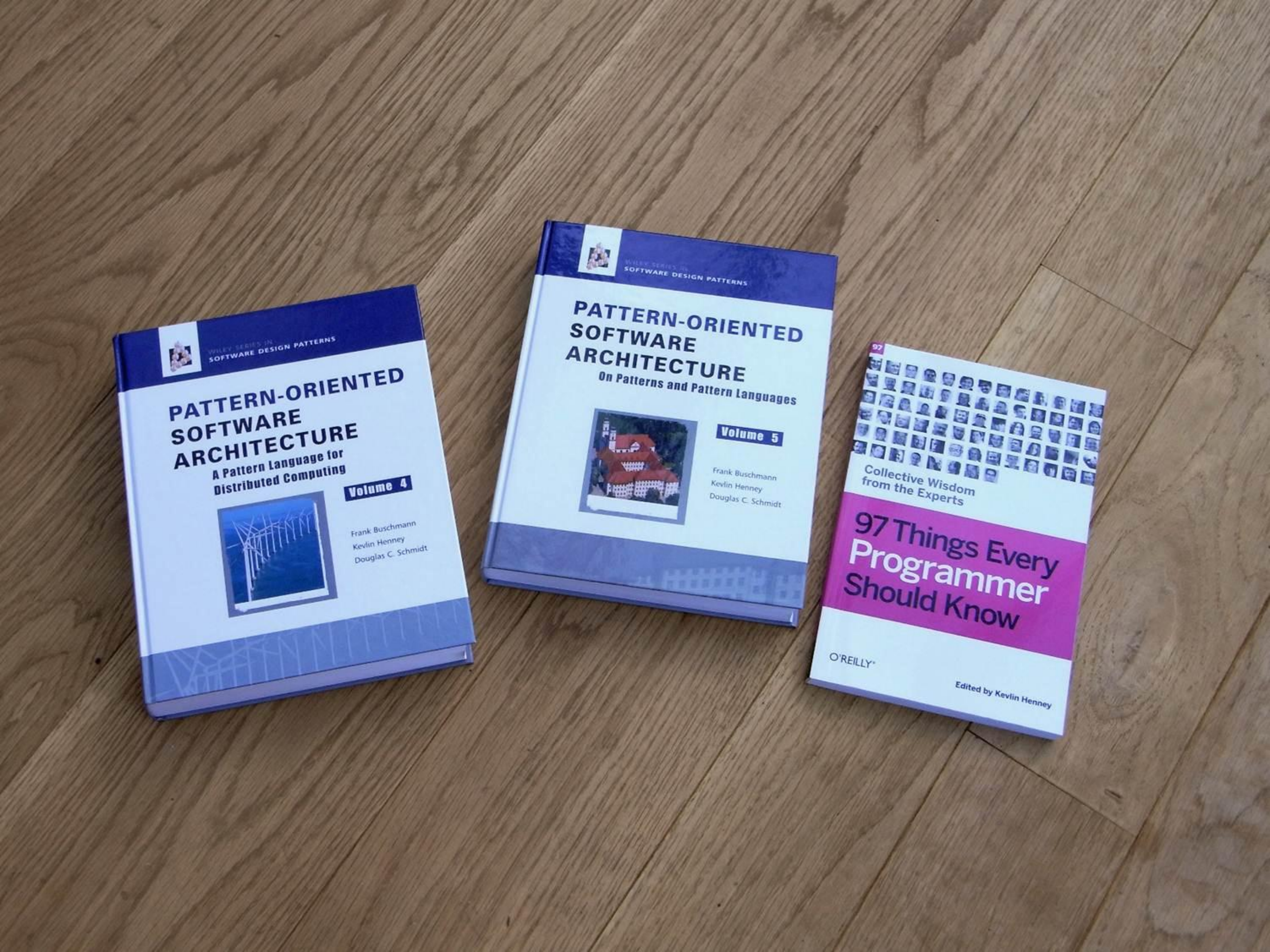
Cool Code

@KevlinHenney

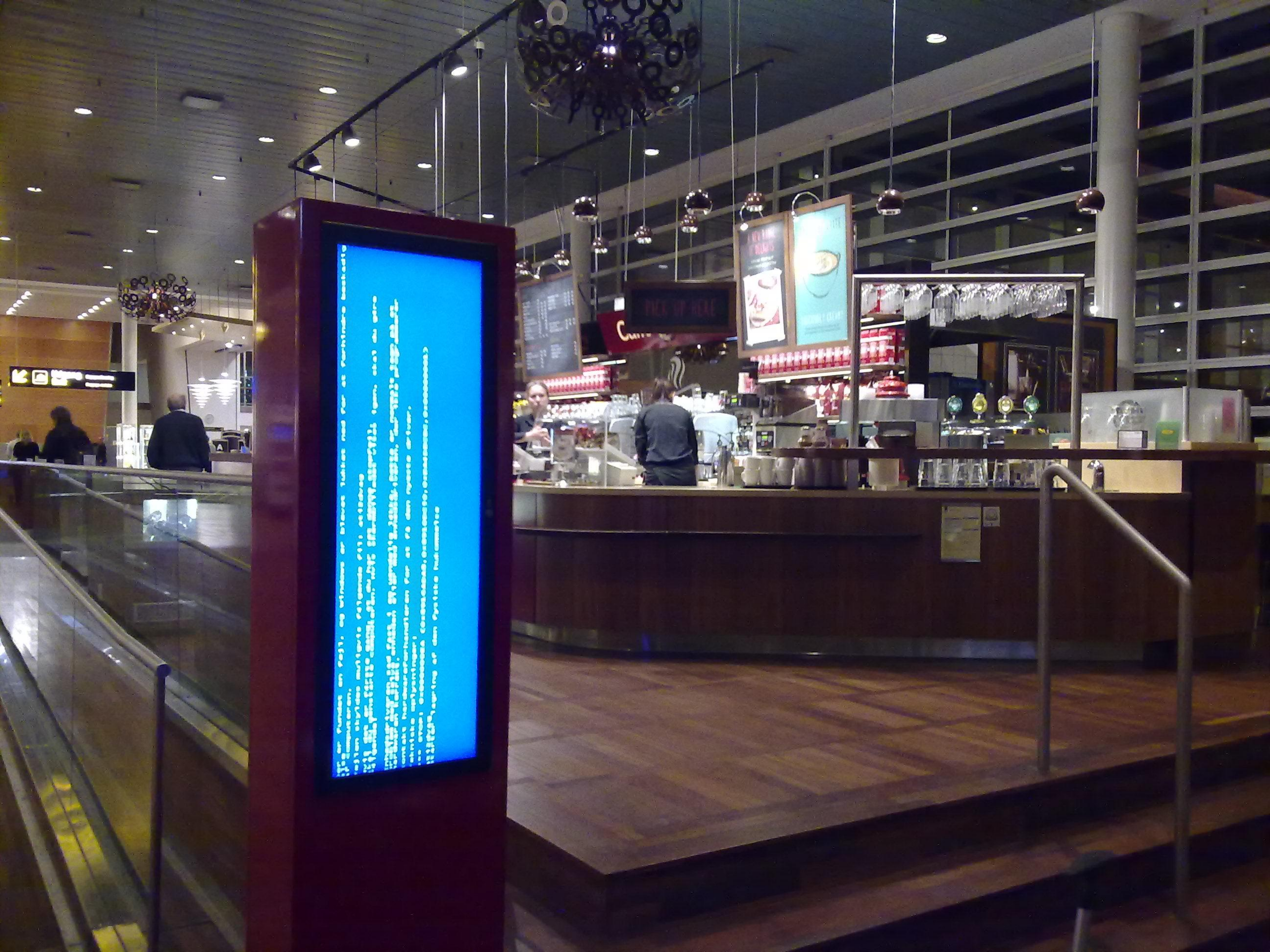

WILEY SERIES IN SOFTWARE DESIGN PATTERNS
PATTERN-ORIENTED SOFTWARE ARCHITECTURE
A Pattern Language for Distributed Computing
Volume 4
Frank Buschmann
Kevlin Henney
Douglas C. Schmidt
WILEY SERIES IN SOFTWARE DESIGN PATTERNS
PATTERN-ORIENTED SOFTWARE ARCHITECTURE
On Patterns and Pattern Languages
Volume 5
Frank Buschmann
Kevlin Henney
Douglas C. Schmidt
97
Collective Wisdom from the Experts
97 Things Every Programmer Should Know
O'REILLY
Edited by Kevlin Henney

Art. Craft. Engineering. Science. These are the swirling muses of design patterns. Art and science are stories; craft and engineering are actions.

Craft is midway between art and science; art and craft stand over against engineering and science. Art is the unique example, the first thing, the story as artifact condensing out of talent and desire. Craft is reliable production of quality. A craftsman might be disappointed but rarely fails. A work of craft is the product of a person and materials. Engineering is reliable and efficient production of things for the use and convenience of people. Science is a process of making a story that can be used for engineering.

Wayne Cool

foreword to
Pattern-Oriented Software Architecture, Volume 5: On Patterns and Pattern Languages


PICK UP HERE

97

Collective Wisdom
from the Experts

プログラマが
知るべき97のこと

97 Things Every Programmer Should Know

O'REILLY®
オライリー・ジャパン

Kevlin Henney 編
和田 卓人 監修
夏目 大 訳

Read Code

Karianne Berg

WE PROGRAMMERS ARE WEIRD CREATURES. We love writing code. But when it comes to reading it, we usually shy away. After all, writing code is so much more fun, and reading code is hard—sometimes almost impossible. Reading other people's code is particularly hard. Not necessarily because other people's code is bad, but because they probably think and solve problems in a different way than you. But did you ever consider that reading someone else's code could improve your own?

```
/*
 * Copyright (c) 1995, 2008, Oracle and/or its affiliates. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 *   - Redistributions of source code must retain the above copyright
 *     notice, this list of conditions and the following disclaimer.
 *
 *   - Redistributions in binary form must reproduce the above copyright
 *     notice, this list of conditions and the following disclaimer in the
 *     documentation and/or other materials provided with the distribution.
 *
 *   - Neither the name of Oracle or the names of its
 *     contributors may be used to endorse or promote products derived
 *     from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
 * IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT OWNER OR
 * CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
 * PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
 * LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
 * NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.
    }
}
```

cool, *adjective*

- fashionably attractive or impressive
- excellent
- used to express acceptance or agreement
- used as an intensive
- used when a conversation goes silent
- marked by deliberate effrontery or lack of due respect or discretion
- restrained or relaxed in style

code, *noun*

- a system of words, figures or symbols used to represent others
- a set of instructions for a computer
- a computer program, or a portion thereof
- a set of conventions or principles governing behaviour or activity in a particular domain
- a system or collection of rules or regulations on any subject
- a collection of writings

Oxford English Dictionary · Concise Oxford English Dictionary · Merriam-Webster's Collegiate Dictionary · Urban Dictionary

```
STR
16571  LD C A      79
(187)  LD L A      105
( 64)  LD H N      38 67
16575  LD A (HL)   126
(191)  LD B N      6  1
( 64)  AND N       230 127
       CP N        254 0
       JP Z DIS    40 20
       INC B       4
       CP N        254 110
       JPZDIS      40 15
       CP N        254 39
       JP C DIS    56 11
       LD A (HL)   126           '0' DIFF
       INC B       4
       LD L N      46 55         '1' EMPTY
       ADD (HL)    134
       BIT 7,A     203 127       '2' WALL
       JP Z DIS    40  2
       LD B N      6   0         '3' SAME
       LD A B      120 ·
       LD L C      105
       RET         201
TABLES
16607  1 11 -1 -11 -10 -12 12 10
16615  13 -13 21 -21 23 -23 -9 9
16623  11 10 12
16626  54 55 39 51 53
PIECE
16621  XOR A       175
(247)  LD(NN) A    50 70 64
( 64)  LD A (HL)   126
       AND N       230 127
       CP P        254 53
       JP Z DIS    40 79
       LD C N      14 1
       LD B N      6  8
       LD HL NN    33 231 64
       CP N        254 51
       JP Z DIS    40 22
       LD L N      46 223
       CP K        254 40
       JP Z DIS    40 16
       LD C B      72
       CP Q        254 54
       JP Z DIS    40 11
       LD B N      6  4
       CP R        254 55
       JP Z DIS    40  5
       LD L N      46 227
       CP B        254 39
       RET NZ      192

SHIFT
16882  LD HL NN    33 99 64
(242)  LD DE NN    17 70 64
( 65)  LD BC NN    1  28 0
       JP C DIS    56 1
       EX DE HL    235
       LDIR        237 176
       RET         201
PSC
17162  AND N       230 127
(10)   LD HL NN    33 242 64
(67)   LD B  N     6  5
       CP (HL)     190
       RET Z       200
       INC HL      35
       DJNZ DIS    16 251
       LD A B      120
       RET         201
MPSCAN
17046  XOR A       175
(150)  LD (NN) A   50 65 64
( 66)  LD B N      6  86
       LD HL NN    33 62 67
       INC HL      35
       PUSH HL     229
       PUSH BC     197
       LD E L      93
       CALL STR2   205 191 64
       CP N        254 3
       JP NZ DIS   32 41
       LD L E      107
       LD (NN) HL  34 7 64
       CALL MOVE   205 247 64
       CALL TL     205 130 66
       JP Z DIS    40 29
       LD E A      95
       LD D N      22 67
       CALL PMOVE  205 255 66
       EXX         217
       AND A       167
```

```
       CALL SHIFT  205 242 65
       CALL CHK    205 1 66
       EXX         217
       LD(HL) B'   112
       LD A C      121
       LD(DE) A    18
       JP C DIS    56 3
       CALL SCORE  205 153 65
       SCF         55
       CALL SHIFT  205 242 65
       JP DIS      24 222
       POP BC      193
       POP HL      225
       DJNZ DIS    16 200
       LD A (NN)   58 65 64
       CP N        254 0
       JP Z DIS    40 254 •
       LD HL NN    33 69 64
       LD A (HL)   126
       DEC HL      43
       DEC HL      43
       LD E (HL)   94
       LD D N      22 67
       LD (DE) A   18
       DEC HL      43
       LD L (HL)   110
       LD H D      98
17131  BIT 0.L     203 69
(235)  LD(HL) N    54 0
( 66)  JP Z DIS    40 2
       LD(HL) N    54 128
       CALL CHGMV  205 247 66
       RET         201
INC
17176  LD A L      125
(24)   EXX         217
(67)   LD(NN) A    50 128 64
       CALL SQ.AT  205 16 66
       EXX         217
       LD A C      121
       RET         201
DRIVER
16959  LD B N      6  5
(63)   LD A N      62 8
(66)   LD HL NN    33 159 67
       INC HL      35
       LD(HL) A    119
       DJNZ DIS    16 252
       CALL KT     205 160 64
       CP N        254 3
       JP NZ DIS   32 238
       LD (NN) HL  34 7 64
       LD E L      93
       CALL MOVE   205 247 64
       LD HL NN    33 161 67
       CALL KT     205 160 64
       CP N        245 2
       EX DE HL    235
       JP NC DIS   48 220
       CALL TL     205 130 66
       JP Z DIS    40 215
       CP C        185
       JP NZ DIS   32 248
       CALL PMOVE  205 255 66
       EXX         217
       CALL CHK    205 1 66
       EXX         217
       JP C DIS    56 8
       CALL CHGSQ  205 235 66
       CALL MPSCAN 205 150 66
       JP DIS      24 194
       LD (HL) B   112
       LD A C      121
       LD (DE) A   18
       JP DIS      24 249
TEST LIST
17026  LD HL NN    33 70 64
(130)  DEC (HL)    53
( 66)  LD A (HL)   126
       INC A       60
       RET Z       200
       ADD L       133
       LD L A      111
       LD A (HL)   126
       RET         201
```

```
PAWN
16721  LD A (HL)   126
(81)   AND N       230 128
(65)   LD HL NN    33 228 64
       JP NZ DIS   32 2
       LD L  N     46 241
       LD D  N     22 3
       LD A  E     123
       ADD (HL)    134
       PUSH HL     229
       PUSH AF     245
       CP N        254 63
       JP C DIS    56 32
       CP N        254 148
       JP NC DIS   48 28
       CALL STR    205 187 64
       CP N        254 0
       JP Z DIS    40 20
       CP N        254 1
       JP NZ DIS   32 17
       LD A D      122
       CP N        254 1
       JP NZ DIS   32 12
       CALL ALIST  205 141 66
       LD A E      123
       CP N        254 82
       JP C DIS    56 19
       CP N        254 126
       JP NC DIS   48 15
       POP AF      241
       POP HL      225
   DEC ~~INC~~ HL      43
       DEC D       21
       JP NZ DIS   32 210
       RET         201
       LD A D      122
       CP N        254 1
       CALL NZ ALIST 196 141 66
       JP DIS      24 241
       POP AF      241
       POP HL      225
       LD E A      95
       JP DIS      24 197
CHK
16897  LD A (NN)   58 55 67
( 1)   ADD N       198 48
(66)   LD HL NN    33 62 67
       LD B A      71
       CPIR        237 177
       DEC HL      43
       LD (NN) HL  34 128 64
SQ.AT  LD B N      6 86
16912  LD HL NN    33 62 67
(16)   INC HL      35
(66)   PUSH HL     229
       PUSH BC     197
       LD E L      93
       CALL STR2   205 191 64
       CP 0        254 0
       JP NZ DIS   32 25
       CALL CHEMV  205 247 66
       LD L E      107
       CALL MOVE   205 247 64
       CALL CHEMV  205 247 66
       CALL TL     205 130 66
       JP Z DIS    40 10
       LD HL(NN)   42 128 64
```

```
       CP L        189
       JP NZ DIS   32 245
       POP BC      193
       POP HL      225
       SCF         55
       RET         201
       POP BC      193
       POP HL      225
       DJNZ DIS    16 216
       AND A       167
       RET         201
SCORE
16793  PUSH HL     229
(153)  PUSH BC     197
( 65)  PUSH DE     213
       PUSH HL     229
       PUSH BC     197
       LD D L      85
       LD HL NN    33 64 64
       CALL NN     205 36 7
       CALL PSC    205 10 67
       LD A B      120
       ADD A H     132
       LD C A      79
       POP AF      241
       CALL PSC    205 10 67
       POP HL      225
       CALL INC    205 24 67
       JP NC DIS   48 1
       ADD A B     128
       LD C A      79
       POP HL      225
       POP DE      209
       LD E (HL)   94
       LD (HL) D   114
       PUSH HL     229
       PUSH DE     213
       CALL INC    205 24 67
       JP NC DIS   48 1
       SUB B       144
       PUSH AF     245
       CALL CHEMV  205 247 66
       CALL CHK    205 1 66
       POP BC      193
       JP NC DIS   48 2
       INC B       4
       INC B       4
       POP DE      209
       POP HL      225
       LD (HL) E   115
       POP HL      225
       CALL CHG    205 250 66
       CALL INC    205 24 67
       JP NC DIS   48 1
       DEC B       5
       CALL CHG    205 250 66
       CALL CHEMV  205 247 66
       LD A B      120
       LD HL NN    33 60 64
       LD (HL) A   119
       EX DE HL    235
       LD HL NN    33 65 64
       CP (HL)     190
       RET C       216
       LD BC NN    1  5  0
       JP DIS      24 11
```

```
# HERE IS THE PHILOSOPHY OF GUILDENSTERN:       ON EVERY APPEARANCE OR DISAPPEARANCE OF THE MANUAL THROTTLE
# DISCRETE TO SELECT P67 OR P66 RESPECTIVELY:   ON EVERY APPEARANCE OF THE ATTITUDE-HOLD DISCRETE TO SELECT P66
# UNLESS THE CURRENT PROGRAM IS P67 IN WHICH CASE THERE IS NO CHANGE

GUILDEN         EXTEND                          # IS UN-AUTO-THROTTLE DISCRETE PRESENT?
# STERN                                         # RSB 2009: Not originally a comment.
                READ CHAN30
                MASK    BIT5
                CCS     A
                TCF     STARTP67                # YES
P67NOW?         TC      CHECKMM                 # NO:  ARE WE IN P67 NOW?
                DEC     67
                TCF     STABL?                  # NO
STARTP66        TC      FASTCHNG                # YES
                TC      NEWMODEX
DEC66           DEC     66
                EXTEND
                DCA     HDOTDISP                # SET DESIRED ALTITUDE RATE = CURRENT
                DXCH    VDGVERT                 #       ALTITUDE RATE.
STRTP66A        TC      INTPRET
                SLOAD   PUSH
                        PBIASZ
                SLOAD   PUSH
                        PBIASY
                SLOAD   VDEF
                        PBIASX
                VXSC    SET
                        BIASFACT
                        RODFLAG
                STOVL   VBIAS
                        TEMX
                VCOMP
                STOVL   OLDPIPAX
                        ZEROVECS
                STODL   DELVROD
                        RODSCALE
                STODL   RODSCAL1
                        PIPTIME
                STORE   LASTTPIP
                EXIT
                CAF     ZERO
                TS      FCOLD
                TS      FWEIGHT
                TS      FWEIGHT +1
VRTSTART        TS      WCHVERT
# Page 801
                CAF     TWO                     # WCHPHASE = 2 ---> VERTICAL: P65,P66,P67
                TS      WCHPHOLD
                TS      WCHPHASE
                TC      BANKCALL                # TEMPORARY, I HOPE HOPE HOPE
                CADR    STOPRATE                # TEMPORARY, I HOPE HOPE HOPE
                TC      DOWNFLAG                # PERMIT X-AXIS OVERRIDE
                ADRES   XOVINFLG
                TC      DOWNFLAG
                ADRES   REDFLAG
                TCF     VERTGUID

STARTP67        TC      NEWMODEX                # NO HARM IN "STARTING" P67 OVER AND OVER
                DEC     67                      # SO NO NEED FOR A FASTCHNG AND NO NEED
                CAF     ZERO                    # TO SEE IF ALREADY IN P67.
                TS      RODCOUNT
                CAF     TEN
                TCF     VRTSTART

STABL?          CAF     BIT13                   # IS UN-ATTITUDE-HOLD DISCRETE PRESENT?
                EXTEND
                RAND    CHAN31
                CCS     A
                TCF     GUILDRET                # YES ALL'S WELL

P66NOW?         CS      MODREG
                AD      DEC66
                EXTEND
                BZF     RESTART?

                CA      RODCOUNT                # NO. HAS THE ROD SWITCH BEEN "CLICKED"?
                EXTEND
                BZF     GUILDRET                # NO. CONTINUE WITH AUTOMATIC LANDING
                TCF     STARTP66                # YES. SWITCH INTO THE ROD MODE.

RESTART?        CA      FLAGWRD1                # HAS THERE BEEN A RESTART?
                MASK    RODFLBIT
                EXTEND
                BZF     STRTP66A                # YES.  REINITIALIZE BUT LEAVE VDGVERT AS
                                                #       IS.

                TCF     VERTGUID                # NO: CONTINUE WITH R.O.D.
```

```
/* grep: search for regexp in file */
int grep(char *regexp, FILE *f, char *name)
{
        int n, nmatch;
        char buf[BUFSIZ];

        nmatch = 0;
        while (fgets(buf, sizeof buf, f) != NULL) {
                n = strlen(buf);
                if (n > 0 && buf[n-1] == '\n')
                        buf[n-1] = '\0';
                if (match(regexp, buf)) {
                        nmatch++;
                        if (name != NULL)
                                printf("%s:", name);
                        printf("%s\n", buf);
                }
        }
        return nmatch;
}

/* matchhere: search for regexp at beginning of text */
int matchhere(char *regexp, char *text)
{
        if (regexp[0] == '\0')
                return 1;
        if (regexp[1] == '*')
                return matchstar(regexp[0], regexp+2, text);
        if (regexp[0] == '$' && regexp[1] == '\0')
                return *text == '\0';
        if (*text!='\0' && (regexp[0]=='.' || regexp[0]==*text))
                return matchhere(regexp+1, text+1);
        return 0;
}

/* match: search for regexp anywhere in text */
int match(char *regexp, char *text)
{
        if (regexp[0] == '^')
                return matchhere(regexp+1, text);
        do {    /* must look even if string is empty */
                if (matchhere(regexp, text))
                        return 1;
        } while (*text++ != '\0');
        return 0;
}

/* matchstar: search for c*regexp at beginning of text */
int matchstar(int c, char *regexp, char *text)
{
        do {    /* a * matches zero or more instances */
                if (matchhere(regexp, text))
                        return 1;
        } while (*text != '\0' && (*text++ == c || c == '.'));
        return 0;
}
```

```java
/**
 * Runs the bare test sequence.
 * @exception Throwable if any exception is thrown
 */
public void runBare() throws Throwable {
    setUp();
    try {
        runTest();
    }
    finally {
        tearDown();
    }
}
/**
 * Override to run the test and assert its state.
 * @exception Throwable if any exception is thrown
 */
protected void runTest() throws Throwable {
    Method runMethod= null;
    try {
        // use getMethod to get all public inherited
        // methods. getDeclaredMethods returns all
        // methods of this class but excludes the
        // inherited ones.
        runMethod= getClass().getMethod(fName, null);
    } catch (NoSuchMethodException e) {
        fail("Method \""+fName+"\" not found");
    }
    if (!Modifier.isPublic(runMethod.getModifiers())) {
        fail("Method \""+fName+"\" should be public");
    }

    try {
        runMethod.invoke(this, new Class[0]);
    }
    catch (InvocationTargetException e) {
        e.fillInStackTrace();
        throw e.getTargetException();
    }
    catch (IllegalAccessException e) {
        e.fillInStackTrace();
        throw e;
    }
}
```

Would you do anything differently in the development of AWK looking back?

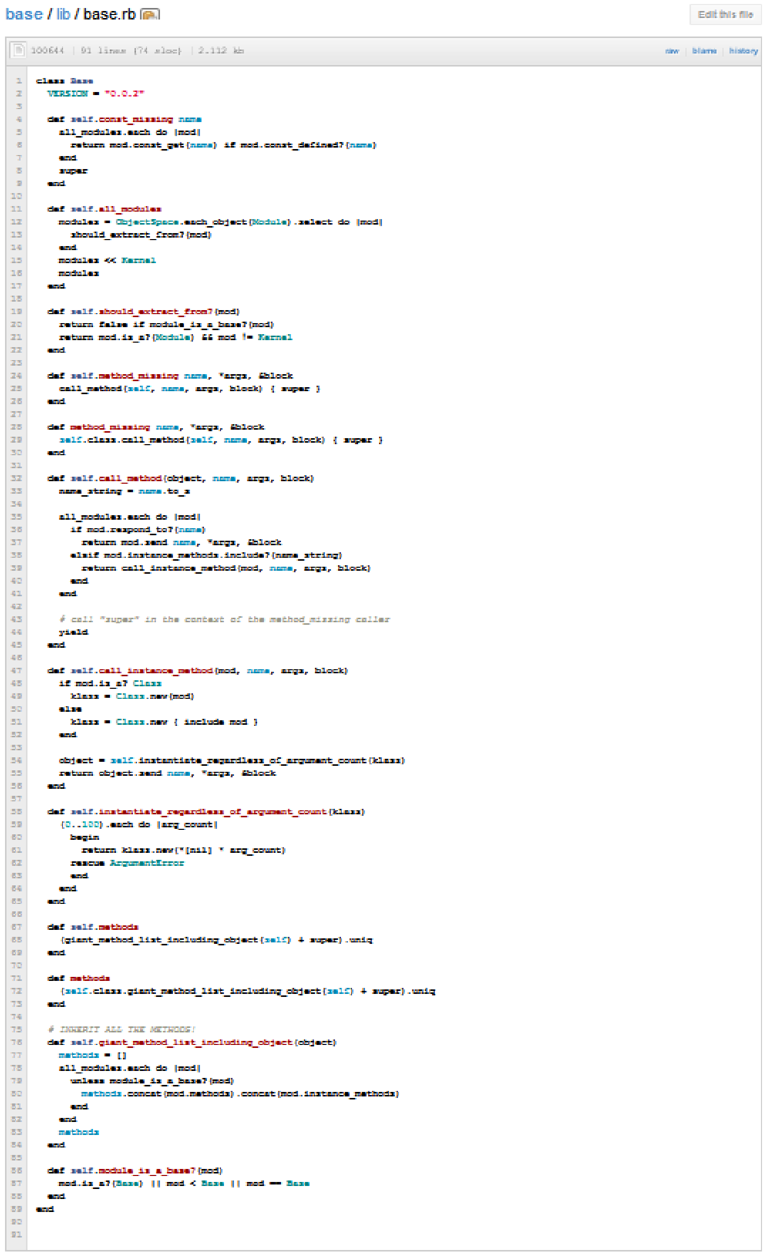
One of the things that I would have done differently is instituting rigorous testing as we started to develop the language. We initially created AWK as a 'throw-away' language, so we didn't do rigorous quality control as part of our initial implementation.

I mentioned to you earlier that there was a person who wrote a CAD system in AWK. The reason he initially came to see me was to report a bug in the AWK compiler. He was very testy with me saying I had wasted three weeks of his life, as he had been looking for a bug in his own code only to discover that it was a bug in the AWK compiler! I huddled with Brian Kernighan after this, and we agreed we really need to do something differently in terms of quality control. So we instituted a rigorous regression test for all of the features of AWK. Any of the three of us who put in a new feature into the language from then on, first had to write a test for the new feature.

```
#!/usr/bin/perl
# --------------------------------------------------------  PerlInterpreter
# PerlInterpreter must be the first line of the file.
#
# Copyright (c) 1995, Cunningham & Cunningham, Inc.
#
# This program has been generated by the HyperPerl
# generator.  The source hypertext can be found
# at http://c2.com/cgi/wikibase.  This program belongs
# to Cunningham & Cunningham, Inc., is to be used
# only by agreement with the owner, and then only
# with the understanding that the owner cannot be
# responsible for any behaviour of the program or
# any damages that it may cause.
# --------------------------------------------------------  InitialComments
```

```
# InitialComments
print "Content-type: text/html\n\n";
$DBM = "/usr/ward/$ScriptName";
dbmopen(%db, $DBM , 0666) || &AbortScript("can't open $DBM");
$CookedInput{browse} && &HandleBrowse;
$CookedInput{edit}   && &HandleEdit;
$CookedInput{copy}   && &HandleEdit;
$CookedInput{links}  && &HandleLinks;
$CookedInput{search} && &HandleSearch;
dbmclose (%db);
if ($ENV{REQUEST_METHOD} eq POST) {
$CookedInput{post}   && &HandlePost;
}
# &DumpBinding(*CookedInput);
# &DumpBinding(*old);
# &DumpBinding(*ENV);
# --------------------------------------------------------  WikiInHyperPerl
```

```
// Erwin Unruh, untitled program,
// ANSI X3J16-94-0075/ISO WG21-462, 1994.

template <int i>
struct D
{
    D(void *);
    operator int();
};

template <int p, int i>
struct is_prime
{
    enum { prim = (p%i) && is_prime<(i>2?p:0), i>::prim };
};

template <int i>
struct Prime_print
{
    Prime_print<i-1>    a;
    enum { prim = is_prime<i,i-1>::prim };
    void f() { D<i> d = prim; }
};

struct is_prime<0,0> { enum { prim = 1 }; };
struct is_prime<0,1> { enum { prim = 1 }; };
struct Prime_print<2>
{
    enum { prim = 1 };
    void f() { D<2> d = prim; }
};

void foo()
{
    Prime_print<10> a;
}

// output:
// unruh.cpp 30: conversion from enum to D<2> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<3> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<5> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<7> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<11> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<13> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<17> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<19> requested in Prime_print
```



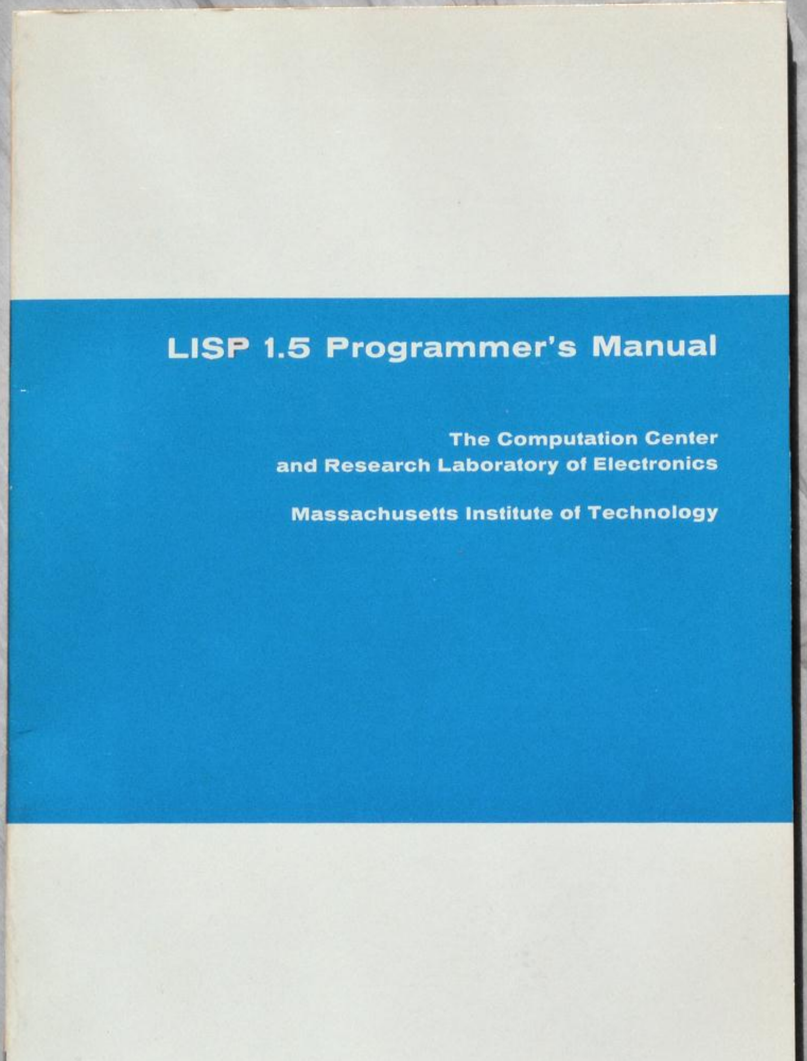
base / lib / base.rb

Edit this file

100644 | 91 lines (74 sloc) | 2.112 kb

raw | blame | history

```ruby
class Base
  VERSION = "0.0.2"

  def self.const_missing name
    all_modules.each do |mod|
      return mod.const_get(name) if mod.const_defined?(name)
    end
    super
  end

  def self.all_modules
    modules = ObjectSpace.each_object(Module).select do |mod|
      should_extract_from?(mod)
    end
    modules << Kernel
    modules
  end

  def self.should_extract_from?(mod)
    return false if module_is_a_base?(mod)
    return mod.is_a?(Module) && mod != Kernel
  end

  def self.method_missing name, *args, &block
    call_method(self, name, args, block) { super }
  end

  def method_missing name, *args, &block
    self.class.call_method(self, name, args, block) { super }
  end

  def self.call_method(object, name, args, block)
    name_string = name.to_s

    all_modules.each do |mod|
      if mod.respond_to?(name)
        return mod.send name, *args, &block
      elsif mod.instance_methods.include?(name_string)
        return call_instance_method(mod, name, args, block)
      end
    end

    # call "super" in the context of the method_missing caller
    yield
  end

  def self.call_instance_method(mod, name, args, block)
    if mod.is_a? Class
      klass = Class.new(mod)
    else
      klass = Class.new { include mod }
    end

    object = self.instantiate_regardless_of_argument_count(klass)
    return object.send name, *args, &block
  end

  def self.instantiate_regardless_of_argument_count(klass)
    (0..100).each do |arg_count|
      begin
        return klass.new(*[nil] * arg_count)
      rescue ArgumentError
      end
    end
  end

  def self.methods
    (giant_method_list_including_object(self) + super).uniq
  end

  def methods
    (self.class.giant_method_list_including_object(self) + super).uniq
  end

  # INHERIT ALL THE METHODS!
  def self.giant_method_list_including_object(object)
    methods = []
    all_modules.each do |mod|
      unless module_is_a_base?(mod)
        methods.concat(mod.methods).concat(mod.instance_methods)
      end
    end
    methods
  end

  def self.module_is_a_base?(mod)
    mod.is_a?(Base) || mod < Base || mod == Base
  end
end


```



LISP 1.5 Programmer's Manual

The Computation Center
and Research Laboratory of Electronics

Massachusetts Institute of Technology

```
(define (eval exp env)
  (cond ((self-evaluating? exp) exp)
        ((variable? exp) (lookup-variable-value exp env))
        ((quoted? exp) (text-of-quotation exp))
        ((assignment? exp) (eval-assignment exp env))
        ((definition? exp) (eval-definition exp env))
        ((if? exp) (eval-if exp env))
        ((lambda? exp)
         (make-procedure (lambda-parameters exp)
                         (lambda-body exp)
                        env))
        ((begin? exp)
         (eval-sequence (begin-actions exp) env))
        ((cond? exp) (eval (cond->if exp) env))
        ((application? exp)
         (apply (eval (operator exp) env)
                (list-of-values (operands exp) env)))
        (else
         (error "Unknown expression type - EVAL" exp))))
```

```
def eval(x, env=global_env):
    "Evaluate an expression in an environment."
    if isa(x, Symbol):              # variable reference
        return env.find(x)[x]
    elif not isa(x, list):          # constant literal
        return x
    elif x[0] == 'quote':           # (quote exp)
        (_, exp) = x
        return exp
    elif x[0] == 'if':              # (if test conseq alt)
        (_, test, conseq, alt) = x
        return eval((conseq if eval(test, env) else alt), env)
    elif x[0] == 'set!':            # (set! var exp)
        (_, var, exp) = x
        env.find(var)[var] = eval(exp, env)
    elif x[0] == 'define':          # (define var exp)
        (_, var, exp) = x
        env[var] = eval(exp, env)
    elif x[0] == 'lambda':          # (lambda (var*) exp)
        (_, vars, exp) = x
        return lambda *args: eval(exp, Env(vars, args, env))
    elif x[0] == 'begin':           # (begin exp*)
        for exp in x[1:]:
            val = eval(exp, env)
        return val
    else:                           # (proc exp*)
        exps = [eval(exp, env) for exp in x]
        proc = exps.pop(0)
        return proc(*exps)

isa = isinstance

Symbol = str
```

```
def to_string(exp):
    "Convert a Python object back into a Lisp-readable string."
    return '('+' '.join(map(to_string, exp))+')' if isa(exp, list) else str(exp)

def repl(prompt='lis.py> '):
    "A prompt-read-eval-print loop."
    while True:
        val = eval(parse(raw_input(prompt)))
        if val is not None: print to_string(val)
```

```python
import re, collections

def words(text): return re.findall('[a-z]+', text.lower())

def train(features):
    model = collections.defaultdict(lambda: 1)
    for f in features:
        model[f] += 1
    return model

NWORDS = train(words(file('big.txt').read()))

alphabet = 'abcdefghijklmnopqrstuvwxyz'

def edits1(word):
   splits     = [(word[:i], word[i:]) for i in range(len(word) + 1)]
   deletes    = [a + b[1:] for a, b in splits if b]
   transposes = [a + b[1] + b[0] + b[2:] for a, b in splits if len(b)>1]
   replaces   = [a + c + b[1:] for a, b in splits for c in alphabet if b]
   inserts    = [a + c + b     for a, b in splits for c in alphabet]
   return set(deletes + transposes + replaces + inserts)

def known_edits2(word):
    return set(e2 for e1 in edits1(word) for e2 in edits1(e1) if e2 in NWORDS)

def known(words): return set(w for w in words if w in NWORDS)

def correct(word):
    candidates = known([word]) or known(edits1(word)) or known_edits2(word) or [word]
    return max(candidates, key=NWORDS.get)
```

```c
                                                                        char
                                                            _3141592654[3141
          ],__3141[3141];_314159[31415],_3141[31415];main(){register char*
      _3_141,*_3_1415, *_3__1415; register int _314,_31415,__31415,*_31,
    _3_14159,__3_1415;*_3141592654=__31415=2,_3141592654[0][_3141592654
   -1]=1[__3141]=5;__3_1415=1;do{_3_14159=_314=0,__31415++;for( _31415
  =0;_31415<(3,14-4)*__31415;_31415++)_31415[_3141]=_314159[_31415]= -
1;_3141[*_314159=_3_14159]=_314;_3_141=_3141592654+__3_1415;_3_1415=
__3_1415    +__3141;for                 (_31415 = 3141-
           __3_1415  ;                  _31415;_31415--
           ,_3_141 ++,                  _3_1415++){_314
           +=_314<<2 ;                  _314<<=1;_314+=
          *_3_1415;_31                   =_314159+_314;
          if(!(*_31+1)                   )* _31 =_314 /
          __31415,_314                   [_3141]=_314 %
          __31415 ;* (                   _3__1415=_3_141
         )+= *_3_1415                     = *_31;while(*
         _3__1415 >=                      31415/3141 ) *
         _3__1415+= -                     10,(*--_3__1415
        )++;_314=_314                     [_3141]; if ( !
        _3_14159 && *                     _3_1415)_3_14159
        =1,__3_1415 =                     3141-_31415;}if(
        _314+(__31415                      >>1)>=__31415 )
        while ( ++ *                       _3_141==3141/314
       )*_3_141--=0                        ;}while(_3_14159
       ) ; { char *                        __3_14= "3.1415";
       write((3,1),                        (--*__3_14,__3_14
       ),(_3_14159                          ++,++_3_14159))+
      3.1415926; }                          for ( _31415 = 1;
     _31415<3141-                           1;_31415++)write(
    31415% 314-(                            3,14),_3141592654[
  _31415    ] +                            "0123456789","314"
  [ 3]+1)-_314;                            puts((*_3141592654=0
,_3141592654))                              ;_314= *"3.141592";}
```

```
#define _ -F<00||--F-OO--;
int F=00,OO=00;main(){F_OO();printf("%1.3f\n",4.*-F/OO/OO);}F_OO()
{
            _-_-_-_
       _-_-_-_-_-_-_-_-_
    _-_-_-_-_-_-_-_-_-_-_-_
  _-_-_-_-_-_-_-_-_-_-_-_-_-_
 _-_-_-_-_-_-_-_-_-_-_-_-_-_-_
 _-_-_-_-_-_-_-_-_-_-_-_-_-_-_
_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
 _-_-_-_-_-_-_-_-_-_-_-_-_-_-_
 _-_-_-_-_-_-_-_-_-_-_-_-_-_-_
  _-_-_-_-_-_-_-_-_-_-_-_-_-_
    _-_-_-_-_-_-_-_-_-_-_-_
        _-_-_-_-_-_-_-_
            _-_-_-_
}
```

```
v=0000;eval$s=%q~d=%!^Lcf<LK8,                  _@7gj*LJ=c5nM)Tp1g0%Xv.,S[<>YoP
4ZojjV)O>qIH1/n[|2yE[>:ieC       "%.#%  :::##"       97N-A&Kj_K_><wS5rtWk@*a+Y5
yH?b[F^e7C/56j|pmRe+:)B     "##%      ::##########"     O98(Zh)'Iof*nm.,$C5Nyt=
PPu01Avw^<IiQ=5$'D-y?    "##:         ###############"    g6`YT+qLw9k^ch|K'),tc
6ygIL8xI#LNz3v}T=4W    "#            #.   .####:#######"    lL27FZ0ij)7TQCI)P7u
}RT5-iJbbG5P-DHB<.   "              ##### # :############"   R,YvZ_rnv6ky-G+4U'
$*are@b4U351Q-ug5   "              #######################"   00x8RR%`Om7VDp4M5
PFixrPvl&<p[]1IJ   "              ############:####  %#####"   EGgDt8Lm#;bc4zS^
y]0`_PstfUxOC(q   "              .#############:##%   .##  ."   /,}.YOIFj(k&q_V
zcaAi?]^lCVYp!;  " %%            .################.     #.   "  ;s="v=%04o;ev"%
(;v=(v-($*+[45,  ":####:          :##############%       :   "  ])[n=0].to_i;)%
360)+"al$s=%q#{  "%######.              #########            "  ;;"%c"%126+$s<<
126}";d.gsub!(/  "##########.           #######%             "  |\s|".*"/,"");;
require"zlib"||  "###########           :######.             "  ;d=d.unpack"C*"
d.map{|c|n=(n||  ":#########:           .######: .           "  )*90+(c-2)%91};
e=["%x"%n].pack   " :#######%           :###### #:          "   &&"H*";e=Zlib::
Inflate.inflate(   "  ######%           .####% ::          "   &&e).unpack("b*"
)[0];22.times{|y|   "  ####%             %###             "   ;w=(Math.sqrt(1-(
(y*2.0-21)/22)**(;   " .###:             .#%             "   ;2))*23).floor;(w*
2-1).times{|x|u=(e+    " %##                           "    )[y*z=360,z]*2;u=u[
90*x/w+v+90,90/w];s[(    " #.                        "    ;y*80)+120-w+x]=(""<<
32<<".:%#")[4*u.count((     " .                   "     ;"0"))/u.size]}};;puts\
s+";_ The Qlobe#{" "*18+ (       "#  :#######"       ;"Copyright(C).Yusuke End\
oh, 2010")}";exit~;_ The Qlobe                  Copyright(C).Yusuke Endoh, 2010
```

```
#!/bin/bash
function f() {
    sleep "$1"
    echo "$1"
}
while [ -n "$1" ]
do
    f "$1" &
    shift
done
wait
```

```
/^1?$|^(11+?)\1+$/
```

```
:;while [ $? -eq 0 ];do nc -vlp 8080 -c'(r=read;e=echo;$r a b
c;z=$r;while [ ${#z} -gt 2 ];do $r z;done;f=`$e $b|sed 's/[^a-
z0-9_.-]//gi'`;h="HTTP/1.0";o="$h 200 OK\r\n";c="Content";if [
-z $f ];then($e $o;ls|(while $r n;do if [ -f "$n" ]; then $e
"<a href=\"/$n\">`ls -gh $n`</a><br>";fi;done););elif [ -f $f
];then $e "$o$c-Type: `file -ib $f`\n$c-Length: `stat -c%s
$f`";$e;cat $f;else $e -e "$h 404 Not Found\n\n404\n";fi)';done
```

"After 20 years, this is still the best exposition of the workings of a 'real' operating system."
Ken Thompson
Lions' Commentary on UNIX® 6th Edition
with Source Code
John Lions
"BOOK OF THE YEAR" Unix Review
1980
Foreword by Dennis Ritchie

Summary--what's most important.

To put my strongest concerns in a nutshell:

1. We should have some ways of coupling programs like garden hose--screw in another segment when it becomes when it becomes necessary to massage data in another way. This is the way of IO also.

2. Our loader should be able to do link-loading and controlled establishment.

3. Our library filing scheme should allow for rather general indexing, responsibility, generations, data path switching.

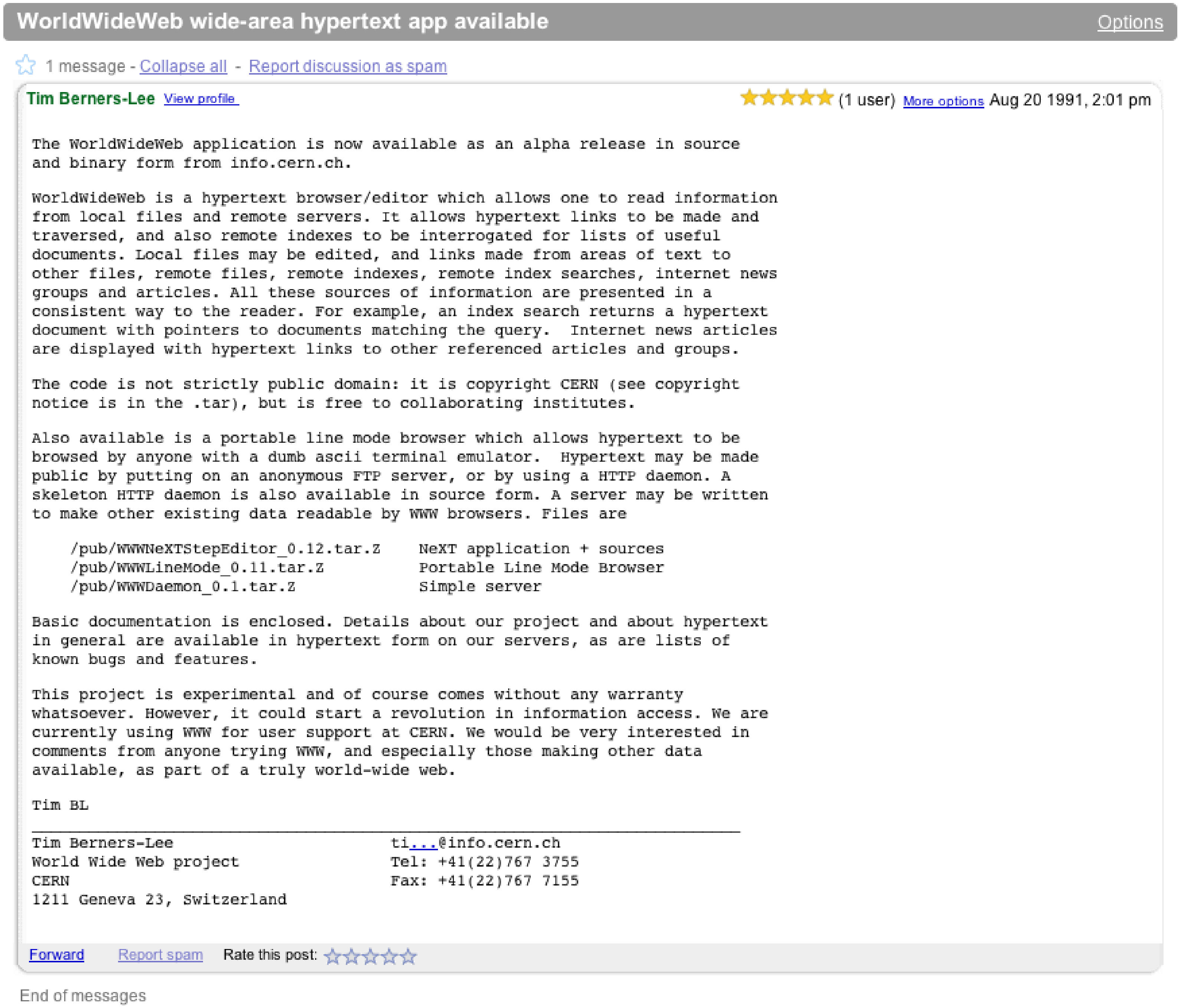
4. It should be possible to get private system components (all routines are sytem components) for buggering around with.

M. D. McIlroy
Oct. 11, 1964

While Thompson and Ritchie were laying out their file system, McIlroy was "sketching out how to do data processing by connecting together cascades of processes and looking for a kind of prefix-notation language for connecting processes together."

Over a period from 1970 to 1972, McIlroy suggested proposal after proposal. He recalls the break-through day: "Then one day, I came up with a syntax for the shell that went along with the piping, and Ken said, I'm gonna do it. He was tired of hearing all this stuff." Thompson didn't do exactly what McIlroy had proposed for the pipe system call, but "invented a slightly better one. That finally got changed once more to what we have today. He put pipes into Unix." Thompson also had to change most of the programs, because up until that time, they couldn't take standard input. There wasn't really a need; they all had file arguments. "GREP had a file argument, CAT had a file argument."

The next morning, "we had this orgy of `one liners.' Everybody had a one liner. Look at this, look at that. ...Everybody started putting forth the UNIX philosophy. Write programs that do one thing and do it well. Write programs to work together. Write programs that handle text streams, because that is a universal interface." Those ideas which add up to the tool approach, were there in some unformed way before pipes, but they really came together afterwards. Pipes became the catalyst for this UNIX philosophy. "The tool thing has turned out to be actually successful. With pipes, many programs could work together, and they could work together at a distance."



WorldWideWeb wide-area hypertext app available Options

1 message - Collapse all - Report discussion as spam

Tim Berners-Lee View profile ★★★★★ (1 user) More options Aug 20 1991, 2:01 pm

The WorldWideWeb application is now available as an alpha release in source and binary form from info.cern.ch.

WorldWideWeb is a hypertext browser/editor which allows one to read information from local files and remote servers. It allows hypertext links to be made and traversed, and also remote indexes to be interrogated for lists of useful documents. Local files may be edited, and links made from areas of text to other files, remote files, remote indexes, remote index searches, internet news groups and articles. All these sources of information are presented in a consistent way to the reader. For example, an index search returns a hypertext document with pointers to documents matching the query. Internet news articles are displayed with hypertext links to other referenced articles and groups.

The code is not strictly public domain: it is copyright CERN (see copyright notice is in the .tar), but is free to collaborating institutes.

Also available is a portable line mode browser which allows hypertext to be browsed by anyone with a dumb ascii terminal emulator. Hypertext may be made public by putting on an anonymous FTP server, or by using a HTTP daemon. A skeleton HTTP daemon is also available in source form. A server may be written to make other existing data readable by WWW browsers. Files are

```
    /pub/WWWNeXTStepEditor_0.12.tar.Z    NeXT application + sources
    /pub/WWWLineMode_0.11.tar.Z          Portable Line Mode Browser
    /pub/WWWDaemon_0.1.tar.Z             Simple server
```

Basic documentation is enclosed. Details about our project and about hypertext in general are available in hypertext form on our servers, as are lists of known bugs and features.

This project is experimental and of course comes without any warranty whatsoever. However, it could start a revolution in information access. We are currently using WWW for user support at CERN. We would be very interested in comments from anyone trying WWW, and especially those making other data available, as part of a truly world-wide web.

Tim BL

```
___________________________________________________________________________
Tim Berners-Lee                      ti...@info.cern.ch
World Wide Web project               Tel: +41(22)767 3755
CERN                                 Fax: +41(22)767 7155
1211 Geneva 23, Switzerland
```

Forward Report spam Rate this post: ☆☆☆☆☆

End of messages

If you don't have time to read, you don't have the time or the tools to write.

Stephen King