bip - business in progress

Heiner Paulsen, Dennis Junker

17. Apr 2018

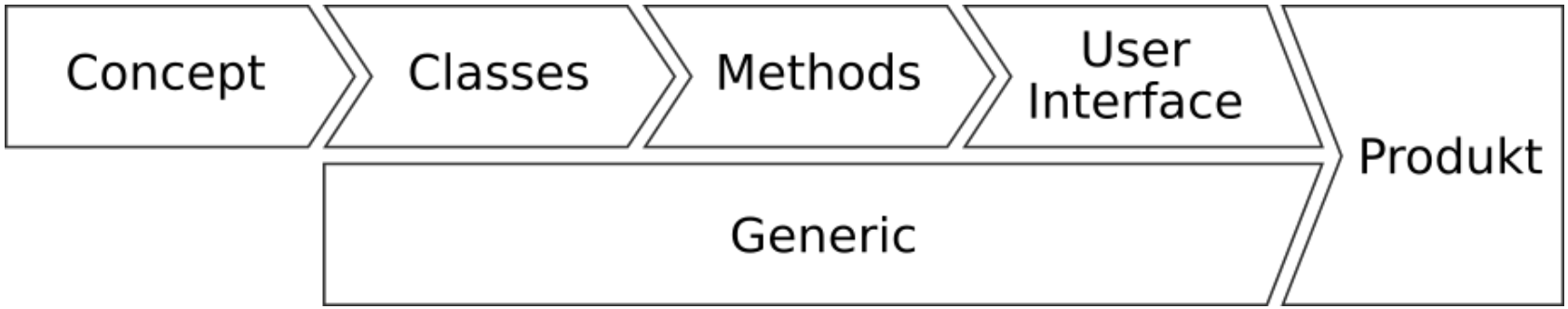
bip

Enterprise Resource Planning (ERP) for

- Finance & Accounting
- Human resources
- Order Processing
- Supply chain management
- Project management
- Customer relationship management
- ...

bip
business in progress

project development

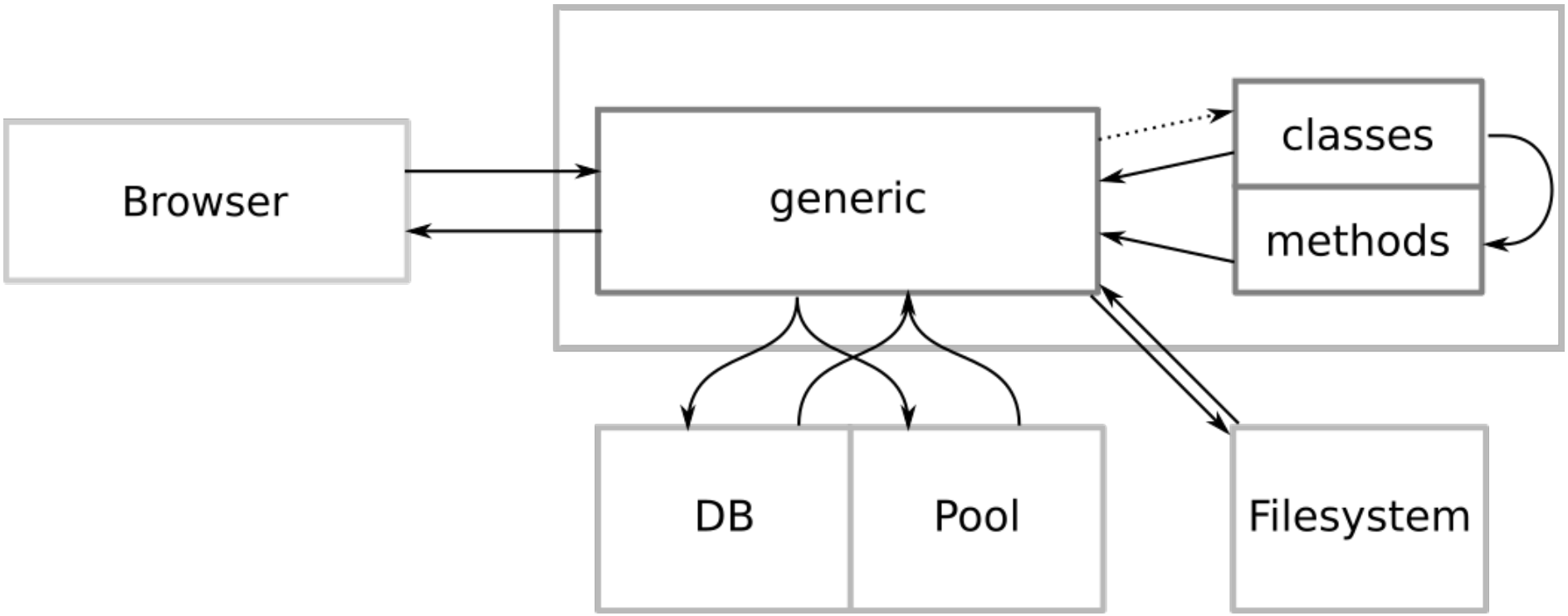


generic

- DB-access
- HTTP-Server
- E-Mailing
- CSV/Excel import/export
- XBRL
- odt/pdf printfiles
- ...

bip
business in progress

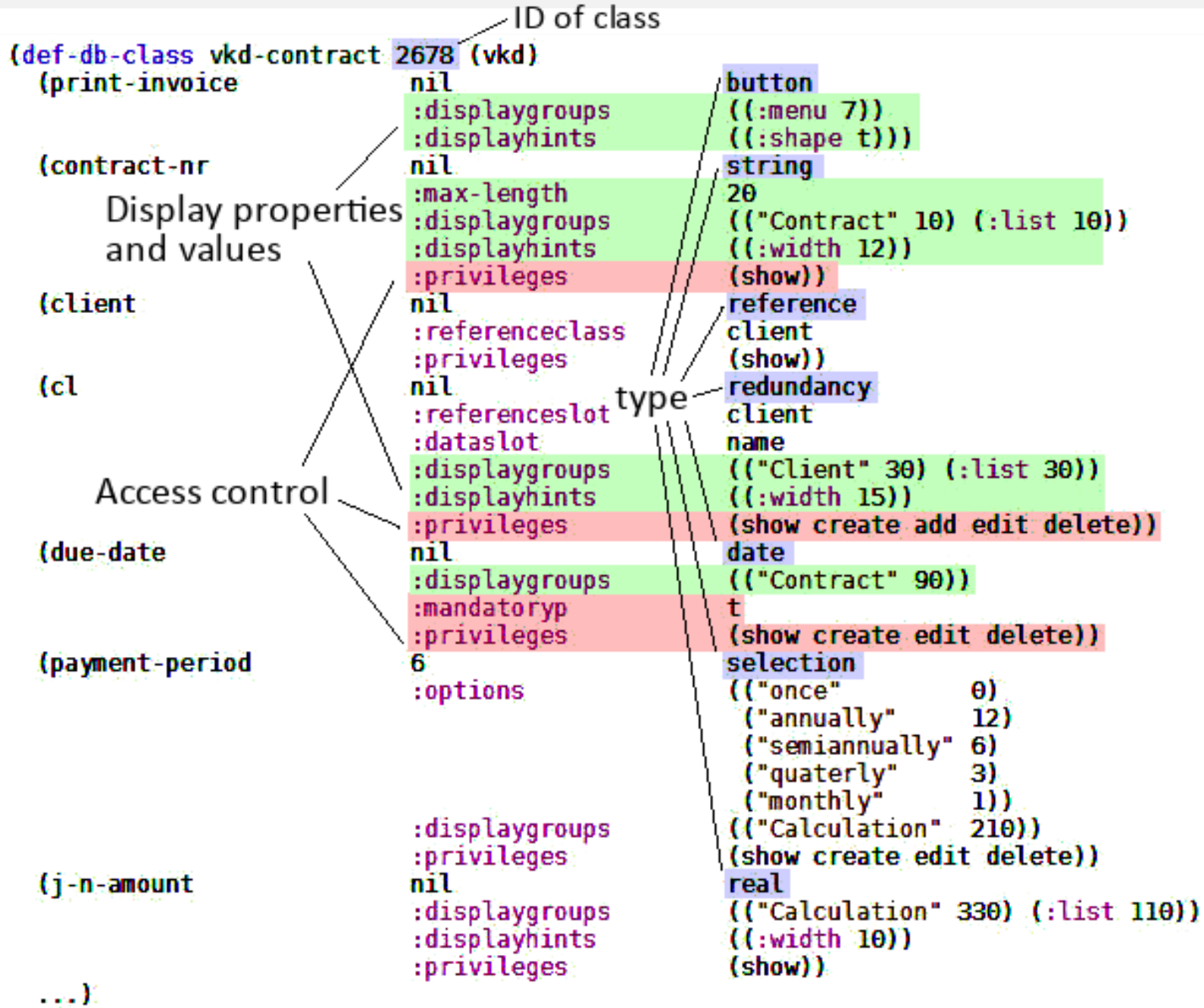
overview



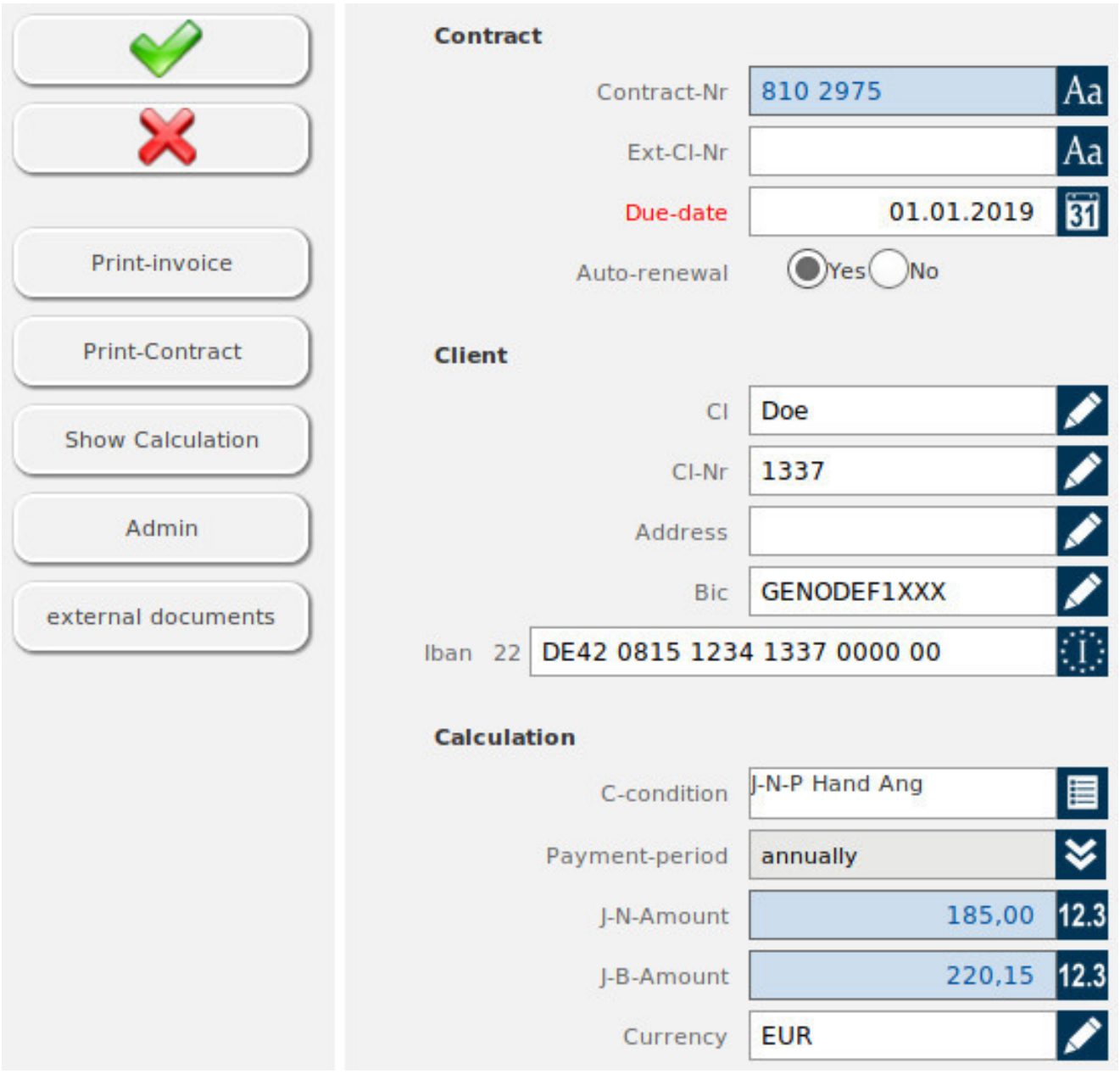
classes

```
(def-db-class vkd-contract 2678 (vkd)
  (print-invoice            nil                 button
                            :displaygroups      ((:menu 7))
                            :displayhints       ((:shape t)))
  (contract-nr              nil                 string
                            :max-length         20
                            :displaygroups      (("Contract" 10) (:list 10))
                            :displayhints       ((:width 12))
                            :privileges         (show))
  (client                   nil                 reference
                            :referenceclass     client
                            :privileges         (show))
  (cl                       nil                 redundancy
                            :referenceslot      client
                            :dataslot           name
                            :displaygroups      (("Client" 30) (:list 30))
                            :displayhints       ((:width 15))
                            :privileges         (show create add edit delete))
  (due-date                 nil                 date
                            :displaygroups      (("Contract" 90))
                            :mandatoryp         t
                            :privileges         (show create edit delete))
  (payment-period           6                   selection
                            :options            (("once"          0)
                                                 ("annually"      12)
                                                 ("semiannually"  6)
                                                 ("quaterly"      3)
                                                 ("monthly"       1))
                            :displaygroups      (("Calculation"  210))
                            :privileges         (show create edit delete))
  (j-n-amount               nil                 real
                            :displaygroups      (("Calculation" 330) (:list 110))
                            :displayhints       ((:width 10))
                            :privileges         (show))
  ...)
```

classes


ID of class
Display properties and values
Access control
type
(def-db-class vkd-contract 2678 (vkd)
(print-invoice
nil
button
:displaygroups
((:menu 7))
:displayhints
((:shape t)))
(contract-nr
nil
string
:max-length
20
:displaygroups
(("Contract" 10) (:list 10))
:displayhints
((:width 12))
:privileges
(show))
(client
nil
reference
:referenceclass
client
:privileges
(show))
(cl
nil
redundancy
:referenceslot
client
:dataslot
name
:displaygroups
(("Client" 30) (:list 30))
:displayhints
((:width 15))
:privileges
(show create add edit delete))
(due-date
nil
date
:displaygroups
(("Contract" 90))
:mandatoryp
t
:privileges
(show create edit delete))
(payment-period
6
selection
:options
(("once" 0)
("annually" 12)
("semiannually" 6)
("quaterly" 3)
("monthly" 1))
:displaygroups
(("Calculation" 210))
:privileges
(show create edit delete))
(j-n-amount
nil
real
:displaygroups
(("Calculation" 330) (:list 110))
:displayhints
((:width 10))
:privileges
(show))
...)

classes


Print-invoice
Print-Contract
Show Calculation
Admin
external documents
Contract
Contract-Nr
810 2975
Ext-Cl-Nr
Due-date
01.01.2019
Auto-renewal
Yes
No
Client
Cl
Doe
Cl-Nr
1337
Address
Bic
GENODEF1XXX
Iban 22
DE42 0815 1234 1337 0000 00
Calculation
C-condition
J-N-P Hand Ang
Payment-period
annually
J-N-Amount
185,00
J-B-Amount
220,15
Currency
EUR

why Common LISP?

- Macros
  - class description
  - method definition
  - print settings
- error handling
  - writing log files
  - message to browser
  - roll-back on error
- process clean up
  - restart while debugging
  - accomplished shutdown
- recompiling
- ...

bip
business in progress

bip - business in progress

Heiner Paulsen, Dennis Junker

17. Apr 2018

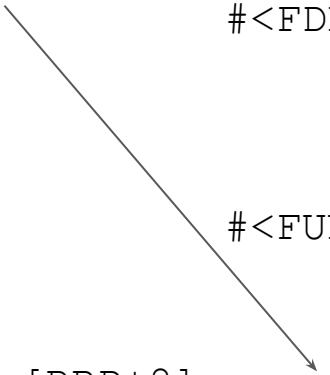
Self-modifying code (for fun and profit)

Simple steps

- Ensure that DISASSEMBLE + reassemble is perfectly faithful to the original machine code
- Manipulate the asm (fun) prior to re-assembly
- Reap benefits (profit)

Example 1: static linking / unlinking

```
* (defun f (x) (g x))
* (disassemble 'f)
...
; B0:       FF7508          PUSH QWORD PTR [RBP+8]
; B3:       E98002F5FD      JMP #x204BA038          #<FDEFN G>

* (disassemble-memory #x204BA038 5)
; Size: 5 bytes. Origin: #x204BA038
; 8:       E973FA0A02      JMP #x22569AB0          #<FUNCTION G>

After static linking:
; B0:       FF7508          PUSH QWORD PTR [RBP+8]
; B3:       E9F8FCFFFF      JMP #x22569AB0          #<FUNCTION G>
```



Did it work?

```
Before: perf stat baseline/sbcl --load make-host-1.sh

27,054,177,733 branches                   #  766.426 M/sec

  35.313061039 seconds time elapsed

After: perf stat experiment/sbcl --load make-host-1.sh

26,135,644,543 branches                   #  809.109 M/sec

  32.360553825 seconds time elapsed
```

Example 2: Precise heap use profiler

```
(defun f (n)
  (list (make-string n :element-type 'base-char)))

; 6B1:   4D8B5D18          MOV R11, [R13+24]
; 6B5:   4D85DB            TEST R11, R11
; 6B8:   EB16              JMP L0
; 6BA:   660F1F440000      NOP
; 6C0:   FF14253800B021    CALL QWORD PTR [#x21B00038]     ;
ENABLE-SIZED-ALLOC-COUNTER
; 6C7:   90                NOP
; 6C8:   4885C0            TEST RAX, RAX
; 6CB:   0F1F440000        NOP
; 6D0: L0
```

Run it

```
* (sb-aprof::aprof-run (lambda () (f 15)))
       %           Bytes         Count    Function
 100.0              48                    F
      66.7             32            1        SIMPLE-BASE-STRING
      33.3             16            1        LIST


* (disassemble ’f)
; 6C0:       F049FF8318000000 LOCK INC QWORD PTR [R11+24]
; 6C8:       F04901832000000 LOCK ADD [R11+32], RAX
```

Example 3: ELFinating an executable

1. Disassemble everything
2. Emit as textual assembly into ‘tempfile.s’
3. “cc -o exename tempfile.s $(SBCL_OBJS)”
4. “perf record exename && perf report --stdio”

```
 3.35%  exename (flet with-recursive-system-lock-thunk in
gethash3)
 3.24%  libc-2.24.so  __memset_erms
 1.68%  exename       sb-c::lz-compress
 1.26%  exename       cl:equal
```

Learning Kanji with Emacs

Wojciech Gac
wojciech.s.gac@gmail.com

April 17, 2018

Motivation

Motivation

- Japanese writing system

Motivation

- Japanese writing system
  - Hiragana

Motivation

- Japanese writing system
  - Hiragana - 46 characters (plus modifiers)

Motivation

- Japanese writing system
  - Hiragana - 46 characters (plus modifiers)
  - Katakana

Motivation

- Japanese writing system
  - Hiragana - 46 characters (plus modifiers)
  - Katakana - 46 characters (plus modifiers)

Motivation

- Japanese writing system
  - Hiragana - 46 characters (plus modifiers)
  - Katakana - 46 characters (plus modifiers)
  - Kanji

Motivation

- Japanese writing system
  - Hiragana - 46 characters (plus modifiers)
  - Katakana - 46 characters (plus modifiers)
  - Kanji - 2,000 - 3,000 in common use,

Motivation

- Japanese writing system
  - Hiragana - 46 characters (plus modifiers)
  - Katakana - 46 characters (plus modifiers)
  - Kanji - 2,000 - 3,000 in common use, **13,108** covered by industrial standards

Two-fold path of suffering

Two-fold path of suffering

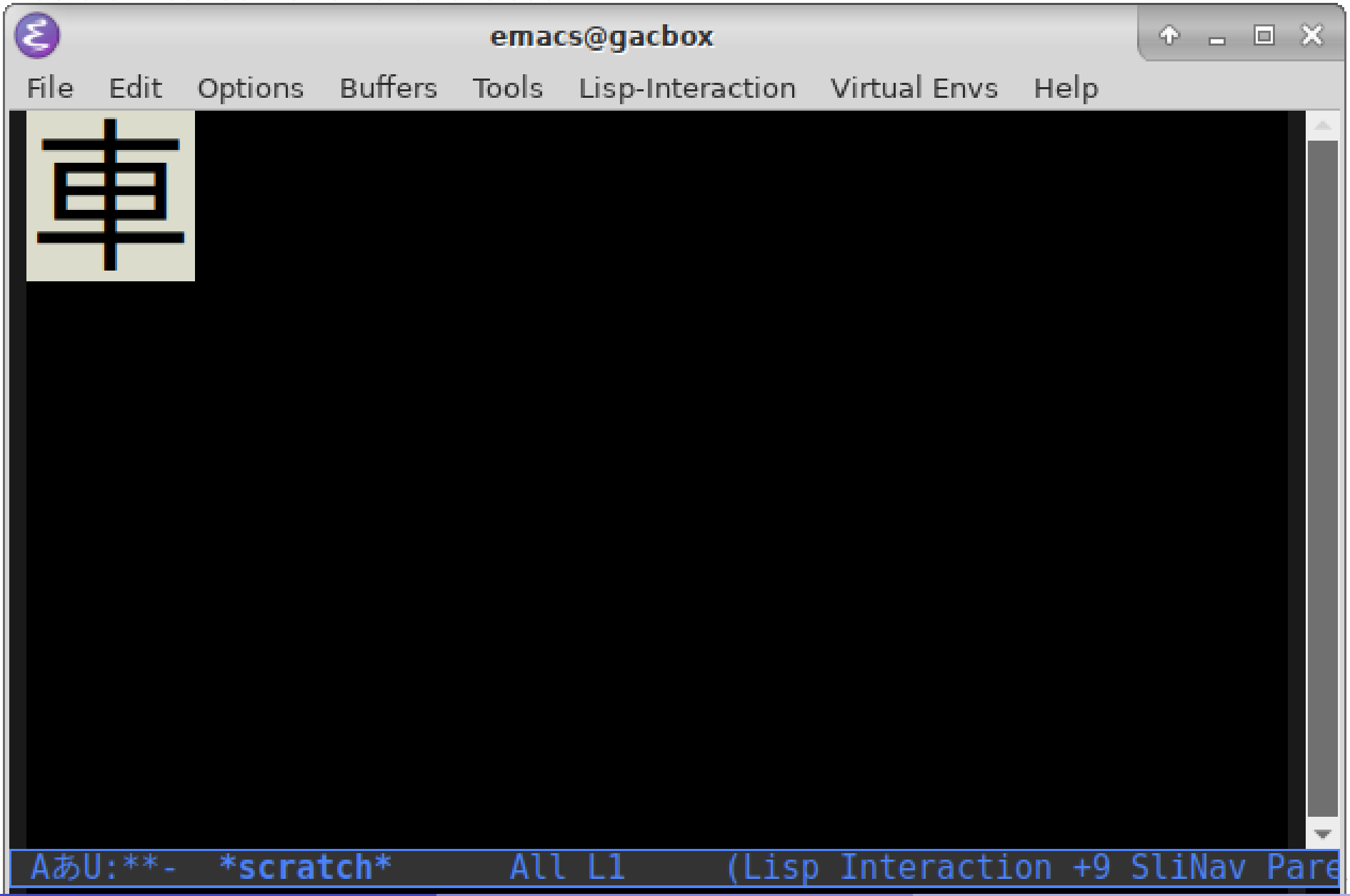
- Learning the stroke order of characters

Two-fold path of suffering

- Learning the stroke order of characters
- Transcribing Kanji to Hiragana

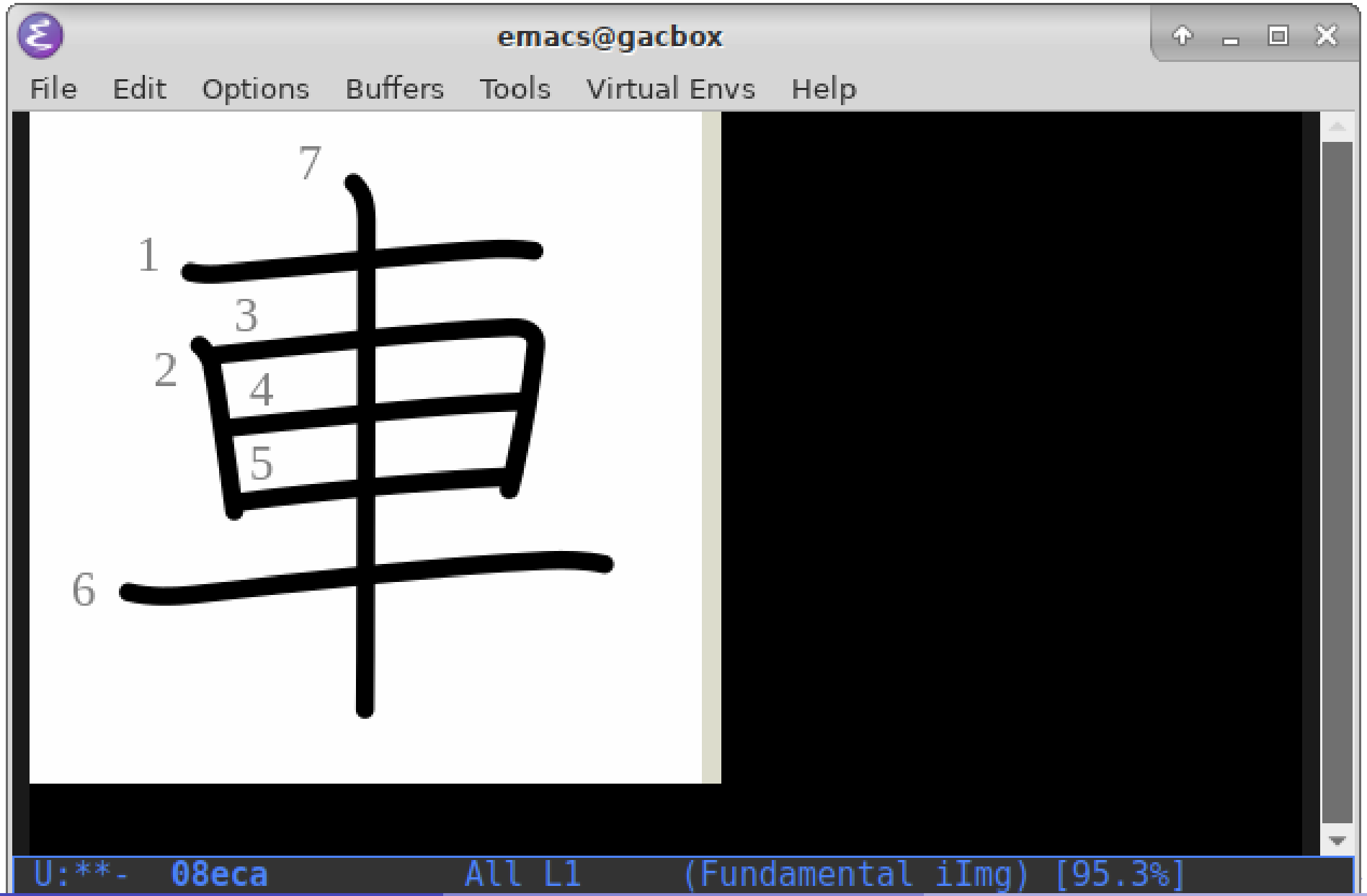
Stroke order

Position cursor over a character



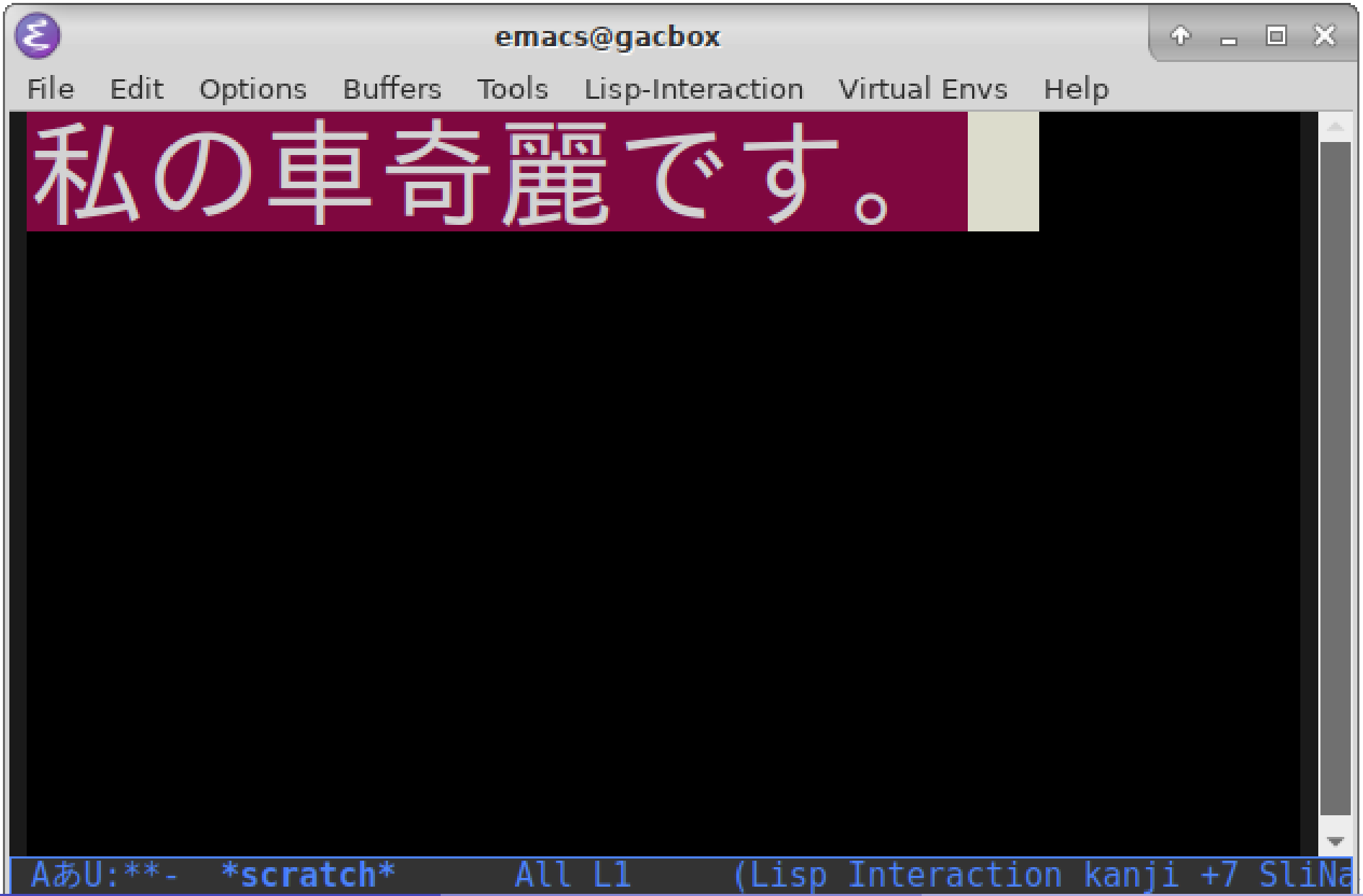
Stroke order

Get a diagram of stroke orders



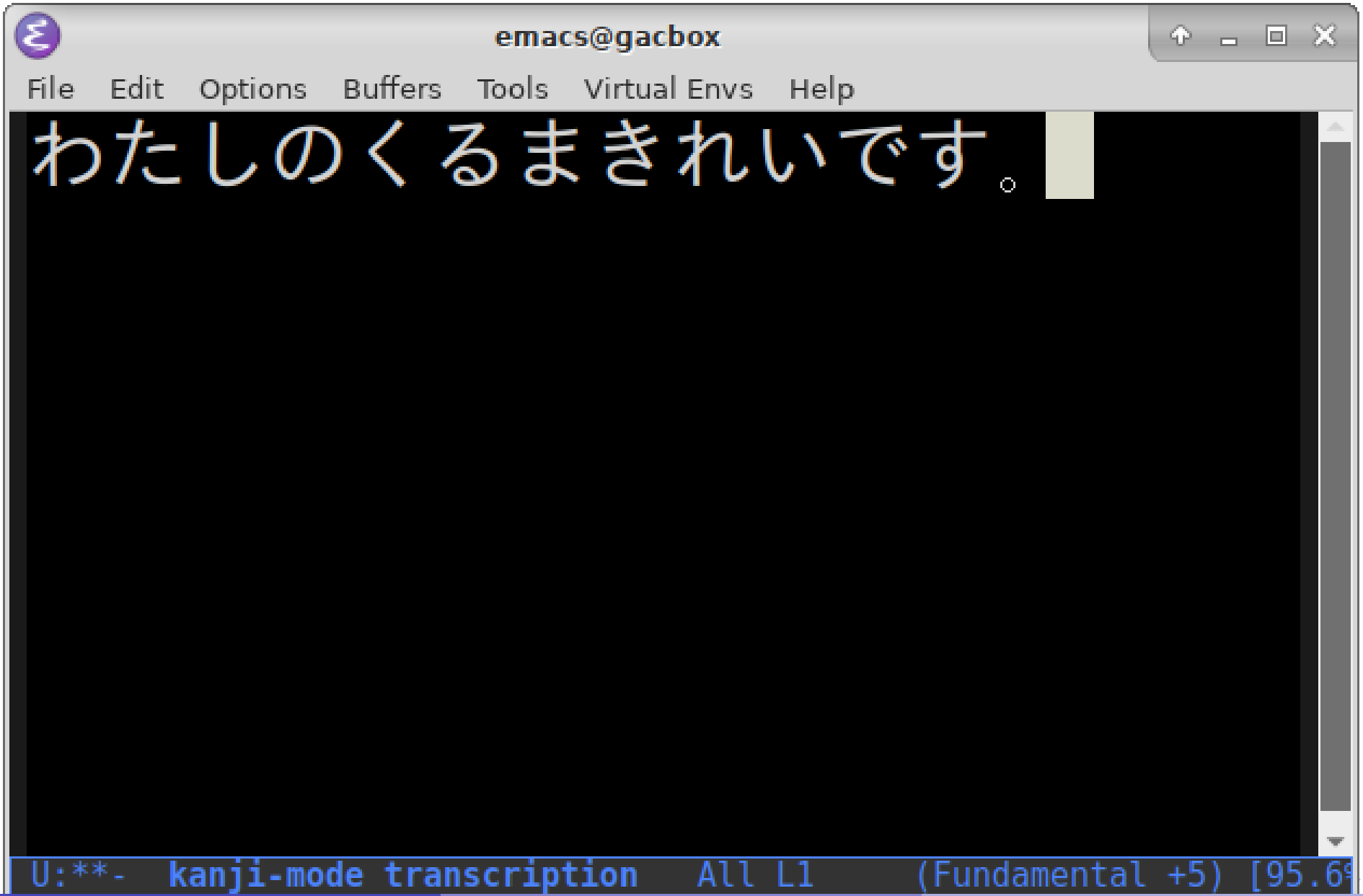
Transcription

Mark text containing Kanji



Transcription

Transcribe it to Hiragana



Oh yeah. The Lisp thing...

```
(defun kanji-mode-stroke-order (point)
  "Take character at point and try to display its
     stroke order."
  (interactive "d")
  (let ((char (char-after point)))
    (km:create-buffer-with-image (km:char-to-hex
       char))))
```

Oh yeah. The Lisp thing...

```
(defun km:create-buffer-with-image (name)
  "Create new buffer with relevant image and switch
     to it.
Buffer can be closed by hitting `q'"
  (with-current-buffer (generate-new-buffer name)
    (let ((image (get-svg-for-kanji-code name)))
      (iimage-mode)
      (iimage-mode-buffer t)
      (insert-image image)
      (local-set-key (kbd "q") 'kill-this-buffer)
      (switch-to-buffer (current-buffer)))))
```

Oh yeah. The Lisp thing...

```
(defun get-svg-for-kanji-code (code)
  "Return an image object for the Unicode code
     provided."
  (let ((image-path (concat (expand-file-name code *
     kanji-svg-path*) ".svg")))
    (create-image image-path)))
```

Shoulders of Giants

Shoulders of Giants

- KanjiVG (https://kanjivg.tagaini.net/)

Shoulders of Giants

- KanjiVG (https://kanjivg.tagaini.net/)
- Kakasi (http://kakasi.namazu.org/)

Thank you

- GitHub - https://github.com/wsgac/kanji-mode
- MELPA - kanji-mode

I needed a tool for defining and describing protocols and test cases

I needed a tool
for defining and describing
protocols and test cases

I needed a tool
for defining and describing
protocols and test cases

I needed a tool
for defining and describing
protocols and te

Next year

PRO

Next year

Protocols

„Don’t get so obsessed over protocols.”

Protocols

„Don’t get so obsessed over protocols.”

~ Robert Strandh, 16th April 2018

Protocols

- Protocol: a set of bindings between operations and data types

Protocols

- Protocol: a set of bindings between operations and data types
- Operations: generic functions, macros, ...

Protocols

- Protocol: a set of bindings between operations and data types
- Operations: generic functions, macros, ...
- Data types: classes, condition types, ...

Protocols

- Protocol: a set of bindings between operations and data types
- Operations: generic functions, macros, ...
- Data types: classes, condition types, ...
- See http://metamodular.com/protocol.pdf

Example protocol

Example protocol

`fuelable`

Example protocol

`fuelable`

Example protocol

`fuelable`

Example protocol

```
fuelable
```

Example protocol

`fuelable`

- `(fuel fuelable)`
- `(setf (fuel fuelable) new-value)`

Example protocol

fuelable

```
(define-protocol fuelable ()
  (:function fuel
             (object))
  (:function (setf fuel)
             (new-value object)))
```

- `(fuel fuelable)`
- `(setf (fuel fuelable)
        new-value)`

Example protocol

fuelable

```
(define-protocol fuelable ()
  (:function fuel
             (object))
  (:function (setf fuel)
             (new-value object)))
```

```
(DEFGENERIC FUEL (OBJECT))
(DEFGENERIC (SETF FUEL) (NEW-VALUE OBJECT))
```

- `(fuel fuelable)`
- `(setf (fuel fuelable) new-value)`

Example protocol

fuelable

- (fuel fuelable)
- (setf (fuel fuelable) new-value)

```
(define-protocol fuelable ()
  (:function fuel
             ((object fuelable))
             real)
  (:function (setf fuel)
             ((new-value real) (object fuelable))
             real))

(DEFGENERIC FUEL (OBJECT))
(DECLAIM (FTYPE (FUNCTION (FUELABLE) REAL)
         FUEL))
(DEFGENERIC (SETF FUEL) (NEW-VALUE OBJECT))
(DECLAIM (FTYPE (FUNCTION (REAL FUELABLE) REAL)
         (SETF FUEL)))
```

Example protocol

fuelable

- (fuel fuelable)
- (setf (fuel fuelable) new-value)

```
(define-protocol fuelable (:export (fuel))
  (:function fuel
             ((object fuelable))
             real)
  (:function (setf fuel)
             ((new-value real) (object fuelable))
             real))

(DEFGENERIC FUEL (OBJECT))
(DECLAIM (FTYPE (FUNCTION (FUELABLE) REAL)
         FUEL))
(DEFGENERIC (SETF FUEL) (NEW-VALUE OBJECT))
(DECLAIM (FTYPE (FUNCTION (REAL FUELABLE) REAL)
         (SETF FUEL)))
(EXPORT '(FUEL))
```

Example protocol

fuelable

- (fuel fuelable)
- (setf (fuel fuelable) new-value)

```
(define-protocol fuelable
    (:export (fuel)
     :documentation „Describes objects which
require some sort of fuel in order to function.

Fuel is understood as a numerical value.”)
  (:function fuel ((object fuelable)) real)
  „Retrieves the amount of fuel in the fuelable.”
  (:function
   (setf fuel) ((new-value real) (object fuelable))
   real)
  „Sets the amount of fuel in the fuelable.”)
```

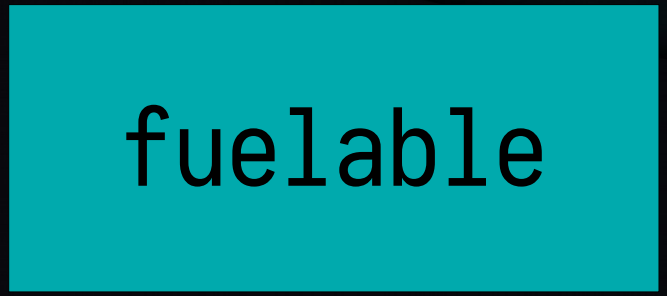
Example protocol

fuelable

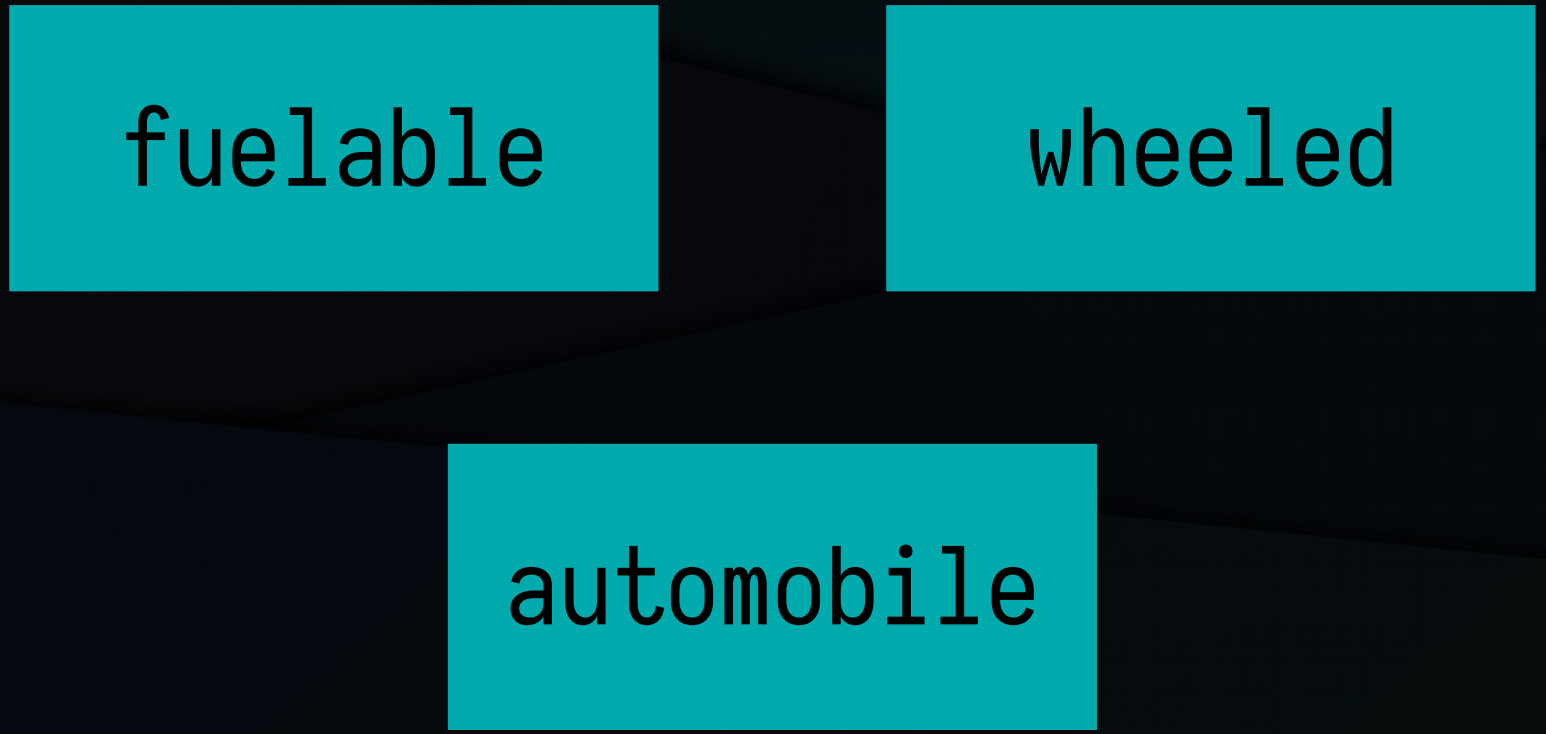
- (fuel fuelable)
- (setf (fuel fuelable) new-value)

```
(define-protocol fuelable
    (:export (fuel)
     :documentation „Describes objects which
require some sort of fuel in order to function.

Fuel is understood as a numerical value.”)
  (:function fuel ((object fuelable)) real)
  „Retrieves the amount of fuel in the fuelable.”
  (:function
   (setf fuel) ((new-value real) (object fuelable))
   real)
  „Sets the amount of fuel in the fuelable.”)

(find-protocol ‘fuelable)
;; ⇒ #<PROTOCOL FUELABLE (2 elements)>
```

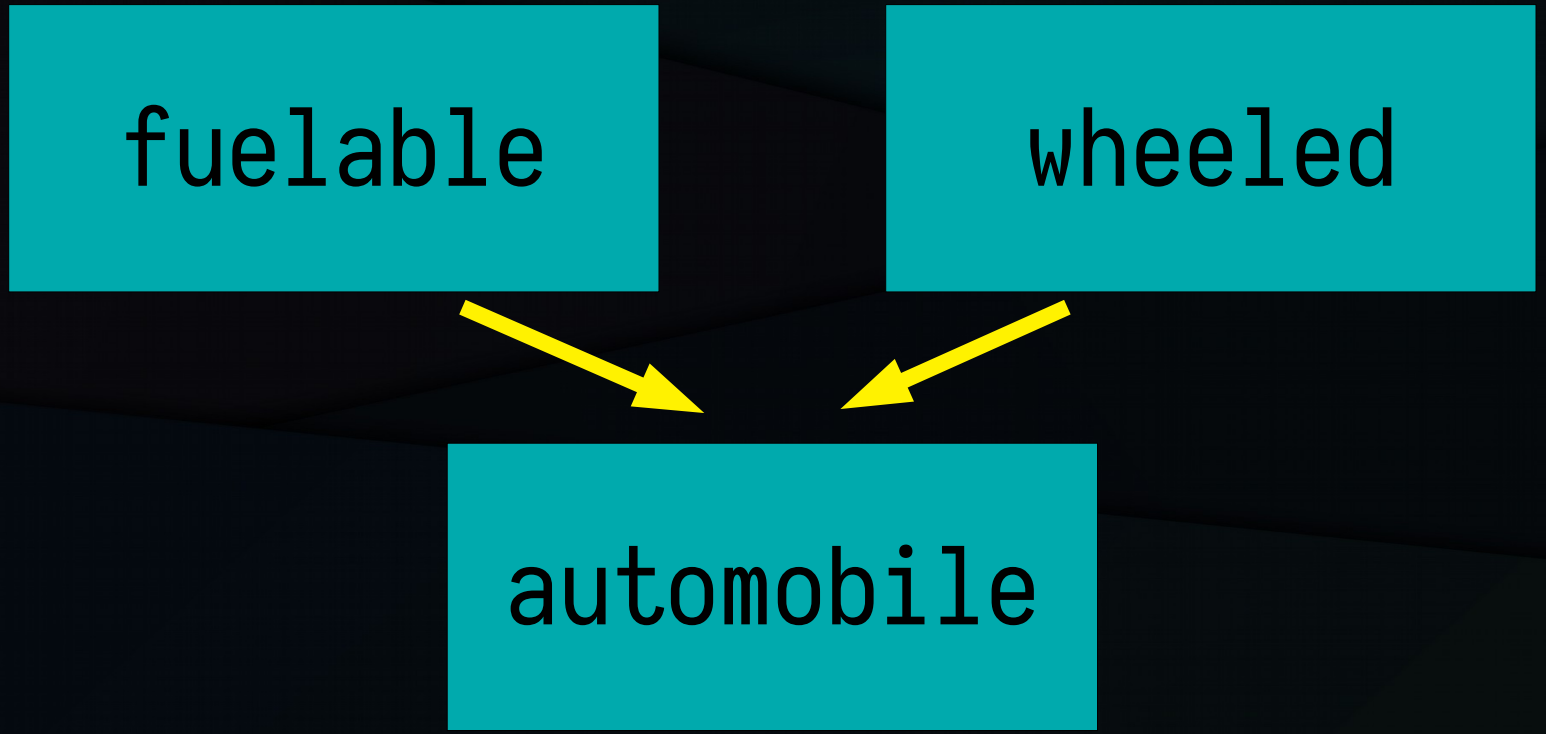
Inheritance



Inheritance



Inheritance



Future work

PROTEST is a work in progress

Future work

PROTEST is a work in progress

- Does class FOO participate in protocol BAR?

Future work

PROTEST is a work in progress

- Does class FOO participate in protocol BAR?
- *How* does class FOO participate in protocol BAR?

Future work

PROTEST is a work in progress

- Does class FOO participate in protocol BAR?
- *How* does class FOO participate in protocol BAR?
- Output HTML/TeX – via CCLDoc

Future work

PROTEST is a work in progress

- Does class FOO participate in protocol BAR?
- *How* does class FOO participate in protocol BAR?
- Output HTML/TeX – via CCLDoc
- Broaden the documentation

Future work

PROTEST is a work in progress

- Does class FOO participate in protocol BAR?
- *How* does class FOO participate in protocol BAR?
- Output HTML/TeX – via CCLDoc
- Broaden the documentation
- Write a paper

Future work

PROTEST is a work in progress

- Does class FOO participate in protocol BAR?
- *How* does class FOO participate in protocol BAR?
- Output HTML/TeX – via CCLDoc
- Broaden the documentation
- Write a paper
- Check if we have any time for questions

Reddit 1.0

(and some table flips)

reddit-archive / **reddit1.0**

Watch 34 | Unstar 971 | Fork 98

Code | Pull requests 0 | Projects 0 | Insights

No description, website, or topics provided.

Common Lisp 100.0%

Branch: **master** | Create new file | Upload files | Find file | Clone or download

spez and **Steve Huffman** Initial commit — Latest commit `bb4fbdb` 19 days ago

LICENSE	Initial commit	19 days ago
autocompute.lisp	Initial commit	19 days ago
classify.lisp	Initial commit	19 days ago
conditions.lisp	Initial commit	19 days ago
cookiehash.lisp	Initial commit	19 days ago
crc.lisp	Initial commit	19 days ago
data.lisp	Initial commit	19 days ago
frame.lisp	Initial commit	19 days ago

reddit-archive / **reddit1.0**

Watch 34 | Unstar 971 | Fork 98

Code | Pull requests 0 | Projects 0 | Insights

No description, website, or topics provided.

Common Lisp 100.0%

Branch: **master** | Create new file | Upload files | Find file | Clone or download

spez and **Steve Huffman** Initial commit — Latest commit `bb4fbdb` 19 days ago

LICENSE	Initial commit	19 days ago
autocompute.lisp	Initial commit	19 days ago
classify.lisp	Initial commit	19 days ago
conditions.lisp	Initial commit	19 days ago
cookiehash.lisp	Initial commit	19 days ago
crc.lisp	Initial commit	19 days ago
data.lisp	Initial commit	19 days ago
frame.lisp	Initial commit	19 days ago

- CMUCL-specific code

- CMUCL-specific code
- No documentation or tests

- CMUCL-specific code
- No documentation or tests
- Rough code quality

- CMUCL-specific code
- No documentation or tests
- Rough code quality
- Missing database schema

- CMUCL-specific code
- No documentation or tests
- Rough code quality
- Missing database schema
- Plaintext passwords in database

(ノಠ益ಠ)ノ彡┻━┻

- Somewhat understandable code

- Somewhat understandable code
- Very few calls to CMUCL primitives (sockets and threads)

- Somewhat understandable code
- Very few calls to CMUCL primitives (sockets and threads)
- It worked!

- Somewhat understandable code
- Very few calls to CMUCL primitives (sockets and threads)
- It worked!
- It worked well enough to boost Reddit into its current size

┬──┬ノ(º _ ºノ)

tamurashingo / **reddit1.0**

forked from reddit-archive/reddit1.0

Code | Issues 1 | Pull requests 0 | Projects 0 | Insights

No description, website, or topics provided.

28 commits | **4** branches | **0** releases

Branch: **master** | **New pull request**

This branch is 27 commits ahead of reddit-archive:master.

tamurashingo change startup, shutdown function ...

tamurashingo / **reddit1.0**

forked from reddit-archive/reddit1.0

No description, website, or topics provided.

28 commits | **4** branches | **0** releases

Branch: **master** | **New pull request**

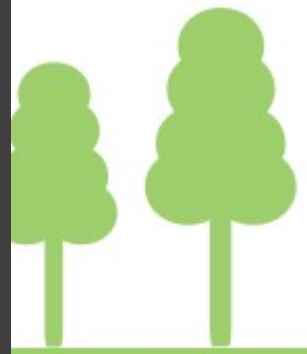
This branch is 27 commits ahead of reddit-archive:master.

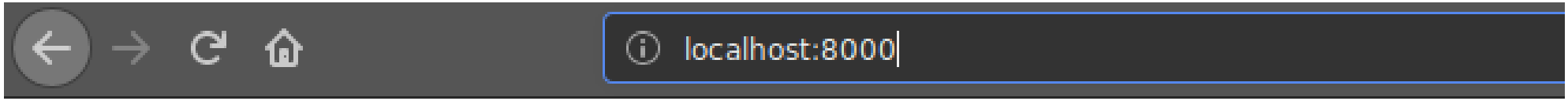
tamurashingo change startup, shutdown function ...

```
CL-USER> (ql:quickload :clsql-cffi)
To load "clsql-cffi":
  Load 1 ASDF system:
    clsql-cffi
; Loading "clsql-cffi"
To load "cffi-uffi-compat":
  Load 1 ASDF system:
    cffi-uffi-compat
; Loading "cffi-uffi-compat"
.
.
(:CLSQL-CFFI)
CL-USER> (ql:quickload :reddit)
To load "reddit":
  Load 1 ASDF system:
    reddit
; Loading "reddit"
.....................
(:REDDIT)
CL-USER> (reddit:startup-reddit)
#<HUNCHENTOOT:EASY-ACCEPTOR (host *, port 8000)>
CL-USER>
```

```
CL-USER> (ql:quickload :clsql-cffi)
To load "clsql-cffi":
  Load 1 ASDF system:
    clsql-cffi
; Loading "clsql-cffi"
To load "cffi-uffi-compat":
  Load 1 ASDF system:
    cffi-uffi-compat
; Loading "cffi-uffi-compat"
.
.
(:CLSQL-CFFI)
CL-USER> (ql:quickload :reddit)
To load "reddit":
  Load 1 ASDF system:
    reddit
; Loading "reddit"
.....................
(:REDDIT)
CL-USER> (reddit:startup-reddit)
#<HUNCHENTOOT:EASY-ACCEPTOR (host *, port 8000)>
CL-USER>
```

*actually not that easy

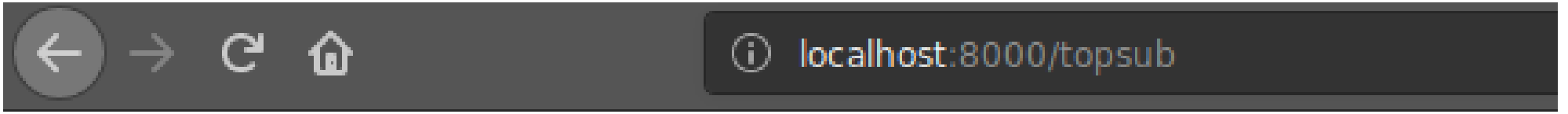




Welcome

When you're reading this message, Hunchentoot has been properly installed.

Please read the documentation.



"

want to join? register in seconds |submit|faq|blog|feedback

hottestnewesttop all-timetop submitters

login/register

username:

password:

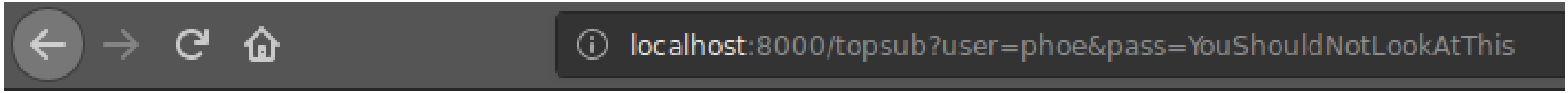
remember me

Login Register

what's my password?

top submitters today
top submitters this week
top submitters all-time

"



"

want to join? register in seconds |submit|faq|blog|feedback

hottestnewesttop all-timetop submitters

login/register

username:

password:

remember me

Login Register

what's my password?

top submitters today
top submitters this week
top submitters all-time

"

localhost:8000/topsub?user=phoe&pass=YouShouldNotLookAtThis

"

want to join? register in seconds |submit|faq|blog|feedback

hottestnewesttop all-timetop submitters

login/register

username:

password:

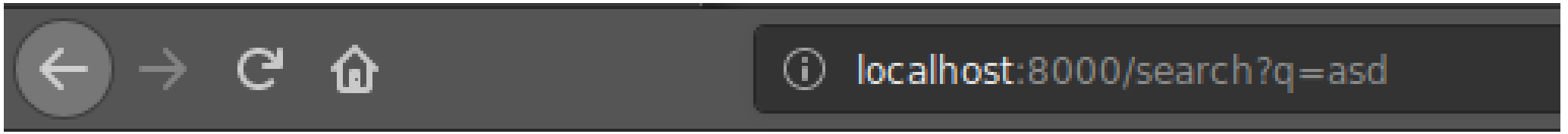
remember me

Login Register

what's my password?

top submitters today
top submitters this week
top submitters all-time

"



Internal Server Error

An error has occurred

Hunchentoot 1.2.38 (SBCL 1.4.6) at localhost:8000

```
[2018-04-17 09:54:46 [ERROR]] The variable REDDIT.SEARCH::[ARTICLES_SN is
                              unbound.
Backtrace for: #<SB-THREAD:THREAD "hunchentoot-worker-127.0.0.1:60766" RUNNING {1008337953}>
0: (REDDIT.SEARCH::SEARCH-SITES "asd" 25 0)
1: (REDDIT.SITES::GET-SEARCH-SITES #<unused argument> "asd" 25 0)
2: (REDDIT.WEB::SEARCH-SITE-TABLE)
3: (REDDIT.WEB::PAGE-SEARCH)
4: ((:METHOD HUNCHENTOOT:ACCEPTOR-DISPATCH-REQUEST (HUNCHENTOOT:EASY-ACCEPTOR T)) #<HUNCHENTOOT:E
ASY-ACCEPTOR (host *, port 8000)> #<HUNCHENTOOT:REQUEST {1007D4AF23}>) [fast-method]
5: ((:METHOD HUNCHENTOOT:HANDLE-REQUEST (HUNCHENTOOT:ACCEPTOR HUNCHENTOOT:REQUEST)) #<HUNCHENTOOT
:EASY-ACCEPTOR (host *, port 8000)> #<HUNCHENTOOT:REQUEST {1007D4AF23}>) [fast-method]
6: ((:METHOD HUNCHENTOOT:PROCESS-REQUEST (T)) #<HUNCHENTOOT:REQUEST {1007D4AF23}>) [fast-method]
7: ((LAMBDA NIL :IN HUNCHENTOOT:PROCESS-CONNECTION))
8: (HUNCHENTOOT::DO-WITH-ACCEPTOR-REQUEST-COUNT-INCREMENTED #<HUNCHENTOOT:EASY-ACCEPTOR (host *,
port 8000)> #<CLOSURE (LAMBDA NIL :IN HUNCHENTOOT:PROCESS-CONNECTION) {1007D4A2DB}>)
9: ((:METHOD HUNCHENTOOT:PROCESS-CONNECTION (HUNCHENTOOT:ACCEPTOR T)) #<HUNCHENTOOT:EASY-ACCEPTOR
 (host *, port 8000)> #<USOCKET:STREAM-USOCKET {1008332813}>) [fast-method]
10: ((FLET CALL-NEXT-METHOD :IN "/home/phoe/.roswell/lisp/quicklisp/dists/quicklisp/software/hunc
```

┻━┻ ‿‿ ¯\(ツ)/¯ ‿‿ ┻━┻

FIN