

toRuby 31st

関将俊

poken...

- play with dRuby.
- exchange your nick and url.

transcribe #1

- server / client style

transcribe #1

```
require 'drb/drb'  
  
ro = DRbObject.new_with_uri(ARGV.shift)  
  
name = 'your_nick'  
url = 'http://your_page'  
desc = 'your comment'  
  
p ro.exchange(name, url, desc)
```

server #1

```
require 'rinda/tuplespace'

module Rinda
  class Njet
    def initialize(value)
      @value = value
    end

    def ==(other)
      @value != other
    end
  end
end

class CardHolder
  def initialize
    @ts = Rinda::TupleSpace.new
    @ts.write(['m_seki', 'http://d.hatena.ne.jp/m_seki', 'Hello, Again'])
  end

  def exchange(name, url, desc)
    @ts.write([name, url, desc])
    @ts.take([Rinda::Njet.new(name), nil, nil])
  end
end

DRb.start_service('druby://:56789', CardHolder.new)
puts DRb.uri
DRb.thread.join
```

transcribe #2

- peer to peer
- meets in the tuplespace

transcribe #2

```
require 'thread'
require 'drb/drb'
require 'rinda/rinda'

class Agent
  def initialize(name, url, desc)
    @tuple = [name, url, desc]
    @inbox = Queue.new
    @renewer = Rinda::SimpleRenewer.new(15)
  end

  def pop
    @inbox.pop
  end

  def hello(name, url, desc)
    @inbox.push([name, url, desc])
    return @tuple
  end

  def meet(who)
    tuple = who.hello(*@tuple)
    @inbox.push(tuple)
  end

  def enter(place)
    place.write([:agent, @tuple[0], self],
               @renewer)
  end

  def broadcast(place)
    place.read_all([:agent, String,
                   DRbObject]).each do
      |k, name, ro|
        next if name == @tuple[0]
        ro.meet(self) rescue nil
      end
    end
  end
end

DRb.start_service
place = DRbObject.new_with_uri(ARGV.shift)

name = 'your_nick'
url = 'http://your_page'
desc = 'your comment'

agent = Agent.new(name, url, desc)
agent.enter(place)
agent.broadcast(place)

while true
  p agent.pop
end
```

server #2

```
require 'rinda/tuplespace'

ts = Rinda::TupleSpace.new(15)
DRb.start_service('druby://:56788', ts)

while true
  puts
  puts DRb.uri

  ts.read_all([nil, nil, nil]).each do |t|
    p [t[0], t[1], t[2].__drburi]
  end
  sleep 10
end
```