

Rexinda

**Stephen Rondeau
AugmenTek**

Pages 236-251

Rexinda™

Stephen Rondeau
AugmenTek

3606 S. 180th St. C-22
SeaTac, WA 98188
Phone: 206-246-6077
augmentek@acm.org

Rexinda is a trademark of AugmenTek.

Agenda

-- Linda®

-- Rexinda

-- Applications

-- Future

If enough time...

-- Availability

-- Parallelization

Linda: Background

- David Gelernter, early 1980s dissertation
- Parallel programming model
 - > coordinate execution and data sharing to solve common problem
 - > simple to use
 - > shared memory model
 - > "tuple space" (global data area) managed by a server
- C and FORTRAN implementations
- Several companies
 - > Scientific Computing Associates, Inc.
 - > Torque Systems, Inc.
 - > Others

Linda: Terminology

-- Tuple: like a database record

(field1, field2, field3, ...)

> Examples: ("ball", "color", "red")
("list", {2,4,6,8,10})

-- Tuple Space: unordered collection of tuples,
possibly distributed over many
processors

> Example:

("ball", "color", "red") ("box", "size", 10)
("list", {2,4,6,8,10}) ("ball", "color", "green")
(27) (4989, 67, 828763) ("box", "size", 10)

-- Matching: uses number of fields, data types,
field order, and user values

Linda: Functions

-- Six functions: out(), rd(), in(), eval(),
rdp(), inp()

> Examples use Rexinda's syntax

-- out(): put tuple into tuple space

Call out "ball", "color", "red"
> ("ball", "color", "red")

Do i=1 to 5
 number.i = i*2
End
number.0 = 5

Call out "list", "@S number."
> ("list", {2,4,6,8,10})

Linda: Functions (continued)

-- rd(), in(), rdp(), inp(): get values from tuple space by matching

rd() -- wait for match, copy values

in() -- wait, copy values, remove tuple

rdp() -- match not found, return 0; else rd()

inp() -- match not found, return 0; else in()

> Examples given later

-- eval(): start a new process

> Example:

Call eval "sort_result", "C:\SORTRXI"

> ("sort_result", 0) after completion

Linda: Matching Examples

Given the tuple space (TS):

("ball", "color", "red") ("box", "size", 10)
("list", {2,4,6,8,10}) ("ball", "color", "green")
(27) (4989, 67, 828763) ("box", "size", 10)

In order of execution:

Call rd "ball", "color", "? color"

> color="green" -- or "red", TS unchanged

Call in ".C", "size", 10

> One of ("box", "size", 10) is removed

If rdp("box", "size", "?N size") = 0

then Call out "box", "size", 20

else Say "size="size /* 10 in this case */

Call rd "ball", "size", "?N size"

> waits for matching tuple to appear

Rexinda: Rationale

- Popularize parallel programming
 - > models world
 - > requires modularity
 - > allows recoverability
 - > offers scalability

- Follows REXX fairly well
 - > functions are easy to remember
 - > associativity similar to stems
 - > tuple space is global, like default scope

- Leverage REXX's fast prototyping

- Extend REXX to handle user-defined events (data appearing in tuple space)

Rexinda: Goals

-- Goals

- > C Linda-like (conversion, reference)
- > Avoid preprocessing source code
- > Extendable
- > Automatic datatyping with overrides
- > Progressive disclosure philosophy
- > Handle errors

-- Basic method: prefix string argument with "markers"

Rexinda: Syntax Markers

-- Needed on out() only to force data type

> "@d" string

where data type d is:

- "C" or missing for character
- "N" for a valid REXX number
- "S" for a stem

-- Input functions really need it:

> "?d varname"

get value of type d and put it in varname

> ".d"

ignore field that has data type d

> "@d" string

force data type d for this string

Applications: Simple Email

Email:

My program:

Parse arg name message

Call out "mailto", name,,
"from", "stephen",,
message

Everyone is running this program:

Parse arg my_name .

Do Forever

Call in "mailto", my_name, ,
"from", "? sender",,
"? message"

Say "Mail from" sender": " message

End

Applications: Simple Print Spooler

-- Print spooler client:

```
Parse arg file_name
If rdp("spooler") = 0
  then Call eval "C:\SPOOLER"
```

```
Call in "id", "?N id"
Call out "id", id+1
```

```
Call out "print", id, file_name
Call in "done", id
Say "Job" id "has printed."
```

-- Print spooler ("C:\SPOOLER"):

```
Call out "spooler"
id = 1
Call out "id", id
```

```
Do while rdp("spooler", "quit") = 0
  Select
    When inp("print", id, "? file_name") then do
      Address CMD "@COPY" file_name "/B LPT1"
      Call out "done", id
      id = id+1
    end
    Otherwise Call Delay 1 /* Every inactive second */
  End
End
```

Future: Enhancements

-- Matching:

- > Aggregates: match and return values for more than one tuple per call (1.0)
- > Counting: count number of matches (1.0)
- > Inequalities: allow matches based on $<$, $>$, \neq , \leq , \geq a value
- > Patterns: allow matches within a field to cause match of tuple

-- Persistence

-- Security

-- Recoverability

-- Transparent data/object access

Availability

- Rexinda Base (version 0.1): Now
 - > Source code
 - > No user support
 - > Inconvenient and slow
 - > Cannot distribute server source code
 - > US\$20. plus \$3. S&H, US Destinations
(WA residents add 8.2% tax)
 - Price subject to change without notice

- Rexinda 1.0: if sufficient interest
 - > Function library (DLL) and fast server
 - > No source code, compatible with Base
 - > Some enhancements (TBD)

- Rexinda n.0, $n > 1$: success of v1.0

- Rexinda Net 1.0: if sufficient interest
 - > Network (TCP/IP) version
 - > Some enhancements for networking

Parallelization

-- Carriero and Gelemtner:

How to Write Parallel Programs, MIT Press,
1991, ISBN 0-262-03171-X

-- Three approaches:

> Result -- the shape of the problem

Example: SQRT(elements of matrix A)

> Specialist -- the makeup of the workforce

Example: send requests to servers

> Agenda -- the tasks to do

Example: many capable workers, list of
tasks