SLAC-R-95-464
CONF-9505198--

PROCEEDINGS OF THE REXX SYMPOSIUM FOR DEVELOPERS AND USERS

May 1-3, 1995
Stanford, California

Convened by
STANFORD LINEAR ACCELERATOR CENTER
STANFORD UNIVERSITY, STANFORD, CALIFORNIA 94309

Program Committee

Cathie Dager of SLAC, Convener
Forrest Garnett of IBM
Pam Taylor of The Workstation Group
James Weissman

Prepared for the Department of Energy
under Contract number DE-AC03-76SF00515

Printed in the United States of America. Available from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, Virginia 22161.

Proceedings of the 6th International REXX Symposium

Table of Contents

6th International REXX Symposium

Program

Pages 1-6

Monday, 1 May 1995

8:45 **Welcome and Announcements**

Cathie Dager, Stanford Linear Accelerator Center

9:00 **Introduction to REXX Tutorial**

Chip Davis, Aresti Systems

10:00 **Break**

10:15 **Intermediate Programming in REXX Tutorial**

Chip Davis, Aresti Systems

11:15 **Advanced REXX Programming Tutorial**

Chip Davis, Aresti Systems

12:15 **Lunch**

1:30 **Keynote Address: REXX 1995 – The Growth of a Language**

M. F. Cowlishaw, IBM Fellow

Much of the character of REXX today was determined during the first year of its development. In this talk, Mike will take highlights from that first year, and show how the design decisions and user feedback of 1979 have let to steady growth since then and the world-wide use that we see today.

2:30 **Break**

2:45 **The Future of REXX**

Tim Browne, IBM

This presentation discusses IBM's plans for Object REXX including making REXX more pervasive in the industry and aligning with key industry standards. Includes an opportunity for Q&A with the IBM REXX Product Manager.

3:45 **REXX in PC/DOS 7.0**

Dave Gomberg, Experimenta

Another operating system acquired a built-in REXX when IBM shipped DOS 7.0. Although DOS is no longer a cutting edge OS, it now has the latest and greatest shell language available at no extra cost. And legacy users can take advantage of REXX as a shell and prototyping language. This new offering will enhance REXX's claim to be the universal language.

4:15 **Issues and Problems Writing REXX Compilers**

Markus Pelt-Layman, Pelt Industries

This session will cover issues and problems writing REXX compilers.

9:00 **Writing World Wide Web CGI Scripts in REXX**

Les Cottrell & Bebo White, Stanford Linear Accelerator Center

The Common Gateway Interface (CGI) is an interface for running external programs, or gateways, under an information server such as the World Wide Web (WWW). Gateway programs, or CGI Scripts, are executable programs designed to enhance the functionality of a server by providing non-native services. Les and Bebo will describe the operation of CGI and demonstrate how CGI scripts may be written in REXX. In addition, they will point out some of the "gotchas" that SLAC has encountered when using REXX with WWW. This talk will primarily focus on the use of REXX in a Unix environment with the CERN and NCSA WWW servers. Some mention will be made of the VM WWW server written in REXX by Rick Troth.

9:45 **Object REXX Demo**

Rick McGuire, IBM

A demonstration of the latest features in Object REXX, including support for the Workplace Shell, persistent objects, shared objects, and more. See how Object REXX can be used to enhance your OS/2 desktop.

10:45 **Break**

11:00 **Object REXX: Open Doc Support**

Tom Brawn, IBM

11:45 **Report from the X3J18 Committee**

Brian Marks, IBM

The first REXX symposium produced enthusiasm for the idea of a REXX language standard, and offers to participate in the development. The effort started in 1991 and fifteen committee meetings later there is now a proposal for what the standard should say. It is being reviewed by the public. This presentation will cover: choices, corrections, and extensions made by the committee; what is new, what is not, and why; what happens to the proposal next; how you can get a copy; how to understand its more formal parts; what you can do about flaws you detect or perceive in the proposal.

12:15 **Lunch**

1:15 **CenterPiece - An Object-Oriented REXX Development and Runtime Environment**

Sandy Syx, Mantissa Corporation

CenterPiece is a modern, graphical, object-oriented, programmable, distributed, multi-platform, multi-user, interpretive, interactive environment. CenterPiece is suitable for both rapid development and delivery of complex, multi-user, pseudo-realtime applications. CenterPiece was built to be the foundation for developing all types of monitoring and control applications. Mantissa's specific area of interest is in datacenter management solutions. This talk will explore CenterPiece specifically focusing on the object-oriented REXX aspects of CenterPiece.

2:15 **Break**

2:30 **Beyond Client Server**

John Tibbetts, Kinexis

3:30 **Getting Ready for Object REXX**

Rick McGuire, IBM

4:30 SOM: Present and Future

Simon Nash, IBM

IBM's System Object Model (SOM) was introduced in 1992 and is now in its third release, running on eight platforms. A major component of IBM's object strategy, SOM provides a language-neutral object model that allows class libraries to be developed and used in a number of object-oriented and procedural languages (both compiled and interpreted). SOM defines interfaces that allow class libraries to be distributed in binary form and used from other languages than the implementation language, thus enhancing their reusability. Other SOM features include CORBA-compliant object distribution (allowing remote location of objects, transparent to client code) and release-to-release compatibility (allowing new versions of class libraries to be used by unmodified and unrecompiled client code). This talk gives an overview of SOM today, and looks at possible future directions, particularly relating to support for Object REXX and similar dynamic (non-compiled) languages.

8:30 **Rexinda, A REXX Implementation of the Linda Parallel Programming Model**

Stephen Rondeau, AugmenTek

The Linda® parallel programming model was conceived by David Gelernter at Yale University in the 1980's to simplify programming parallel applications. Linda extends a language with four basic functions and two variations on them. These functions put data into and get data from a global, content-addressable data area ("tuple space"). User functions can access that global data and execute in parallel, via multitasking or multicomputing. Since the tuple space and REXX's compound variables are associative, a REXX user may be "comfortable with" Linda. REXX makes it easier to prototype and experiment with hot topics such as data mining and software agents, which can exploit Linda. This presentation will describe Linda and Rexinda in more detail, briefly show how to parallelize a program, elaborate on a simple example and comment on future directions.

9:30 **REXX for CICS/ESA**

Bob Vogel, IBM

10:15 **Break**

10:30 **REXX Changes in OS/2 Warp Version 3**

Dick Goran, C F S Nevada, Inc.

11:15 **S/Rexx for Unix**

David Salthouse, Open Direct

S/Rexx is an implementation of REXX 4.00 for Unix systems with some extensions developed by Robert Benaroya. The presentation discusses the design objectives and gives examples of the benefits. S/Rexx is completely integrated with a Unix Xedit-like editor SEDIT. The principle characteristics are: an interpreter free from size or shape limitations; support for dynamic loading of external procedures which can share global variables with the main procedure; enhanced debugging facilities including a more detailed trace output as well as a Motif based debugger; simplified interface to the Unix platform via a number of additional built-in functions; new ADDRESS environments; extended syntax on a number of instructions.

12:15 **Lunch**

1:15 **A REXX-based Stock Exchange Realtime Client-Server Environment for Research, Educational and Public Relations Purposes: Implementation and Usage Issues**

Martin Misseyer, Free University of Amsterdam

This paper presents the design, development, and implementation of C/S systems from both developer and user views and from both technical and non-technical points of view. Questions addressed include: the difference between quasi-C/S and full C/S; how to develop quasi or full C/S environments in REXX using APIs; REXX portability in C/S environments (designing applications for portability); which REXX programming techniques should be used developing a full C/S environment; which performance and programming techniques should be used in a full C/S environment having large scale database operations and I/O (designing applications for performance); how to create C/S GUIs (specific applications as well as monitors) in REXX.

2:00 **Break**

2:15 **REXX/370 Compiler and Library: What, Why, and How ****

Rick McGuire, IBM

What can the IBM REXX/370 Compiler and Library do for you? What's new with Release 3? This presentation covers performance, other advantages, compatibility, enhancements, supported systems, and tips.

2:45 **REXX and How I Hit the Ground Running in Unix**

Lois White, Stanford Linear Accelerator Center

Possible subtitles for this talk might be something like "uni-REXX, Rx for Making the Transition from VM/CMS to Unix a Little Less Painful" or "How I Learned to Stop Worrying and Love to Type in Significant Mixed Case". Using REXX in Unix really did make it possible for me to move to and, dare I say, "thrive" in the Unix environment. This presentation will show how I use REXX in Unix and how being able to use it there made it possible for me to hit the round running instead of barely crawling.

3:15 **Closing Remarks**

Cathie Dager, Stanford Linear Accelerator Center

**** Session replaced by a general session. Presentation included in the Proceedings.**