THE REXX FOR UNIX DISCUSSION PANEL

MODERATOR: ED SPIRE (TWG)
SPEAKERS: IAN COLLIER
MARK HESSLING
NEIL MILSTED (IX)
RICK MCGUIRE (IBM)
STEVE BACHER (DRAPER LABS)
"TRANSCRIBER:" F. SCOTT OPHOF (CONSULTANT)

The REXX for Unix discussion panel

4th annual REXX Symposium
La Jolla, California
May 18, 1993

Moderator: Ed Spire (TWG)
Speakers: Ian Collier
Mark Hessling
Neil Milsted (iX)
Rick McGuire (IBM)
Steve Bacher (Draper Labs)

"Transcriber": F. Scott Ophof (cmsultant)

Note from the "transcriber":

Due to an unfortunate circumstance, there was no recording of the discussion on which to base a transcription for the Proceedings. The following thus consists of a reconstruction pieced together from memories and comments.
Appended you will find the list of subjects planned for discussion, depending on the time available and the time used by the actually discussed subjects.

Thus, though the accuracy in questionable, the intent is present. And we (speakers, moderator and "transcriber") hope the subjects specified in the "List of Subjects" will receive more attention.

Introduction:

Moderator Ed Spire introduced the speakers, probably saying something close to:

Steve Bacher, commonly known as "Batchman", is a Technical Staff person in the Computer Support group at Draper Laboratory in Cambridge, Massachusetts. He has moved over the years from straight MVS system programming to supporting both mainframe users and Unix workstation users. Steve is a very enthusiastic user of REXX and a contributor of the odd TSO/REXX utility. The Draper Consulting "CLOG" help desk facility, which Steve implemented, is written entirely in REXX. Steve is the author of the MVS NNTP and Gopher clients (not written in REXX). and was the creator of the "Zil" Lisp system for MVS (also not written in REXX).

Ian Collier worked for a year at IBM Hursley (near Winchester in the South of England), which is where his interest in Rexx originates. Ian then went to Oxford University and took a Ba (first class) from Oxford University in 1990 in "Mathematics and Computation". He is presently in the third year of a D.Phil. at Oxford in the field of "correctness-preserving transformations in action systems", which is a branch of parallel computing. Irritated for the last two years by the lack of a Rexx interpreter (or indeed any "decent" interpreter) on their SunOS system, Ian decided to write one. Unfortunately, academic work has often for long periods prevented him from working on it. So although the interpreter has been working for some time it was summer 1992 before Ian added the file I/O functions and finally released REXX/imc to the public.

Mark Hessling, currently working as Oracle DBA for Griffith University in Brisbane in Australia, started on ICLs, moved to DEC-10s and then to VM/CMS. From there he went to Unix via VMS. Mark had about 4 years REXX experience on VM/CMS including contract work in the UK. He has been working on SunOS for about the past 3 years. Mark is here at the REXX Symposium to present "The Hessling Editor", which is based on Kedit and CMS-XEDIT.

Neil Milsted, of iX Corporation, implemented a REXX for UNIX known as uni-REXX. Neil is highly active as Vice Chairman in the X3J18 REXX Standards Committee, where the effort of standardizing REXX is taking place.

Rick McGuire is well-known as lead developer/designer of the REXX products (like the OS/2 interpreter) from IBM Endicott in New York. Rick, due to time pressures, will lead off the discussion.

Discussion:

Rick McGuire led off with an entertaining speech about the history of REXX's introduction to new platforms (starting with VM). He also spoke on common misconceptions about why Rexx is not useful on Unix. The consensus is that Rick had enough good material for a complete sessior of his own.

On the question of what kind of people will be using REXX in Unix, Ian Collier said:
I think that one thing we should do is give the Rexx programmer access to basic Unix system calls, such as those for dealing with sockets. If Unix hackers see that Rexx is a powerful tool for controlling Unix then they might start to pay some attention. It would be quite neat if we could implement an NNTP newsreader for Unix in Rexx - especially if it is better than "rn", or whatever. That way people will start to see how useful Rexx is.

Ed Spire asked:
Should we really give the user all the Unix system calls, including fork()?

Steve Bacher replied:
This might not be a good idea, especially for non-hackers since once you call fork() you become two people and you've got to figure out who you really are.
[Steve might also have said: "Right now, only ex-mainframers use REXX on Unix. The RS/6000 market is like that. This isn't good enough. We MUST reach the native Unix hackers."]

[Note: The next day however Ed showed that he had already given the user a fork() call in Uni-Rexx].

Mark Hessling commented:
There are 3 kinds of potential REXX users on Unix - ex-mainframers and hackers, plus folks told to use it with no significant computer training, a la PC/DOS/Windows users. This might be the largest potential upcoming market.

The stack and GUI issues were discussed in some detail.
Someone seems to have said that it would be too much work to implement the Rexx stack with sockets.

Ian Collier replied to this:
I did implement the stack in REXX/imc with sockets; it allows the programmer to type "ls -al | rxstack" (where "rxstack" is a program which communicates with the stack via the socket) and have the output stacked. This mimics OS/2 behaviour, and is also vaguely similar to the CMS method. This has another benefit: you can start off a stack before calling Rexx, and then any data on the stack will persist

between invocations of Rexx.

Charles Daney asked a question about whether there was interest in REXX as an embedded macro language under Unix, and Neil said it had become easier to do. But the impression seemed to be that there wasn't a lot of interest.

Even though the panel was accorded extra time, there just wasn't enough for the issues planned. A number of points on the prepared list thus didn't get a mention, including the questions for Mike Cowlishaw... The above is all that we could piece together.

List of Subjects for Discussion:

Included are some things various speakers wanted to say on specific subjects, and indeed may even have said if that subject actually was discussed at the Symposium.

1: REXX still seems to be associated with CMS.

2 : Old world vs. New world (text mode vs. graphical interfaces)

3: A natural universal notation for capturing the output of a command.

4: Extensions to Rexx unique to Unix (regexps, RXSOCKET, . ..).
Ian Collier: It might be useful to discuss whether these extensions are to be keyword instructions, built-in functions, or library functions. I propose the latter. In this case you might not call them "extensions" at all. However, it is clearly necessary to make sure that the library for each interpreter contains the same functions.

5: Redirect/pipe into the REXX environment (vars, stack, etc.).

6: Redirect/pipe into another command (ADDRESS SHELL/EXEC/PERL/SH/...).

7: How to best integrate Rexx into the Unix environment.
Steve Bacher might have said:
We have to find those things that keep people from trying REXX (like the IBM association) and also those things that keep people from continuing to use it once they've tried it (lack of features, performance, etc.).

8: Are the (free) REXXes robust enough?

9: Way(s) of interfacing to Unix.
Steve Bacher: "We need to discuss the passing of arguments to REXX from a typical Unix shell environment. This is critical."

10: Is there any persistence?
To this Ian Collier might have said "Not in Unix there isn't. End of story...".
Steve Bacher: "Imagine providing Unix users a way to write a script that will "cd" or set environmental variables in the main shell. Of course, this requires people with smarts about

Unix internals to get involved in the REXX implementation process."

11: Macros and scripting.

12: Have any issues regarding integrating Rexx into the shell been addressed?
Steve BAcher: "Mainly the issue of argument passing, which has been ignored for far too long. (No, PARSE QUOTED won't help.) Other thoughts of a "rexxsh" are probably not viable now - REXX may not be the most pleasant INTERACTIVE environment for people."

13: What direction would the user community have for us on these issues?
Steve Bacher: "Ask them! But more to the point, have something to sell them Show them what REXX can give them that perl can't. Also show them that anything they can do in the other shell scripting languages can be done in REXX - if it can't it's probably too arcane to be bothered with. If there is anything that can't be done in REXX that is important to Unix users (like regexps), then by all means start working on integrating it into REXX."

14: Is the ANSI effort holding REXX useage back, and could this also be relevant to other platforms?

15: Value of conversion utilities (sh2rexx, csh2rexx, perl2rexx...)?