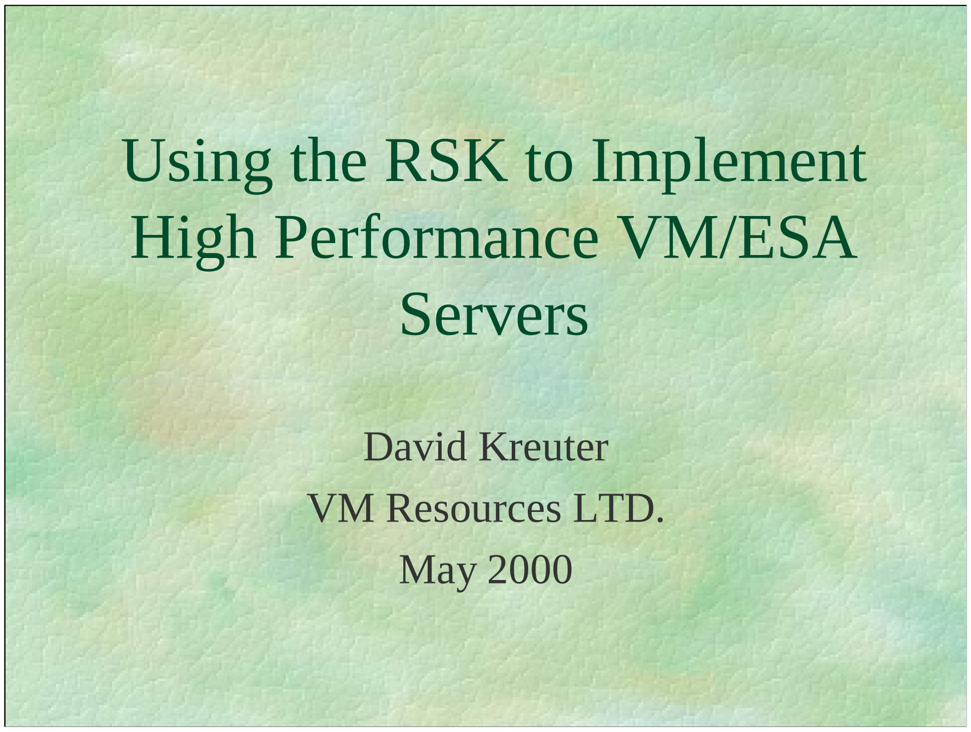


Using the RSK to Implement High Performance VM/ESA Servers

David Kreuter

VM Resources LTD.

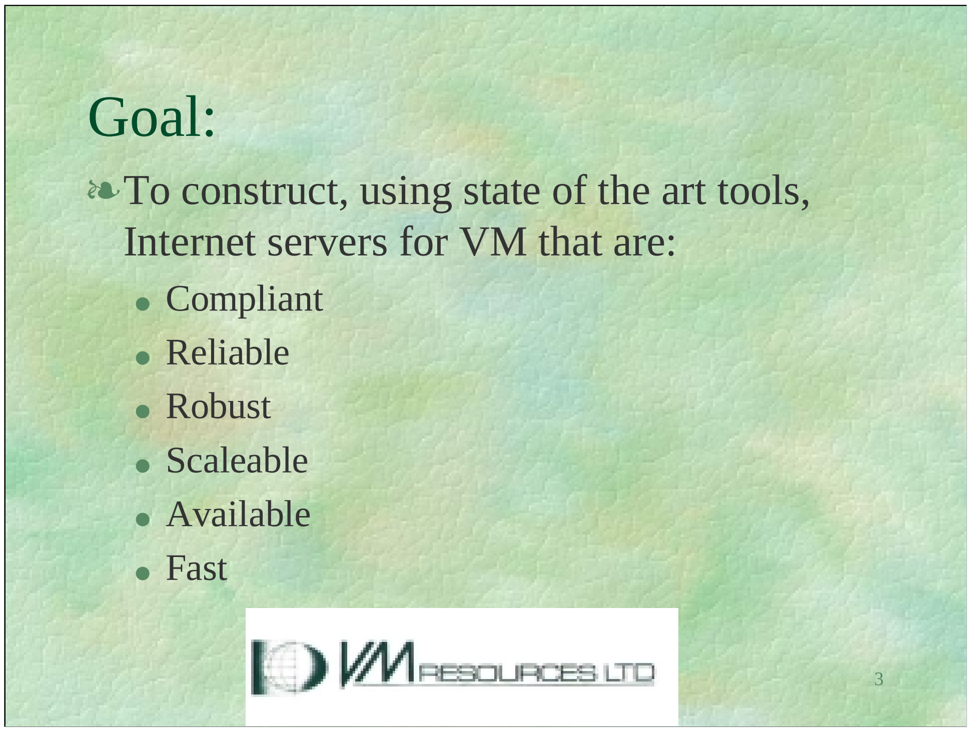
May 2000

RETURN TO INDEX

Today’s Presentation

- Goals for a VM Internet Servers
- Various Internet RFCs
- VM Architecture Advantages
- The CMS Reusable Server Kernel
- How VM Mail Server Works
- Techniques, Debugging, and Tips

VM RESOURCES LTD



Goal:

- To construct, using state of the art tools, Internet servers for VM that are:
  - Compliant
  - Reliable
  - Robust
  - Scaleable
  - Available
  - Fast

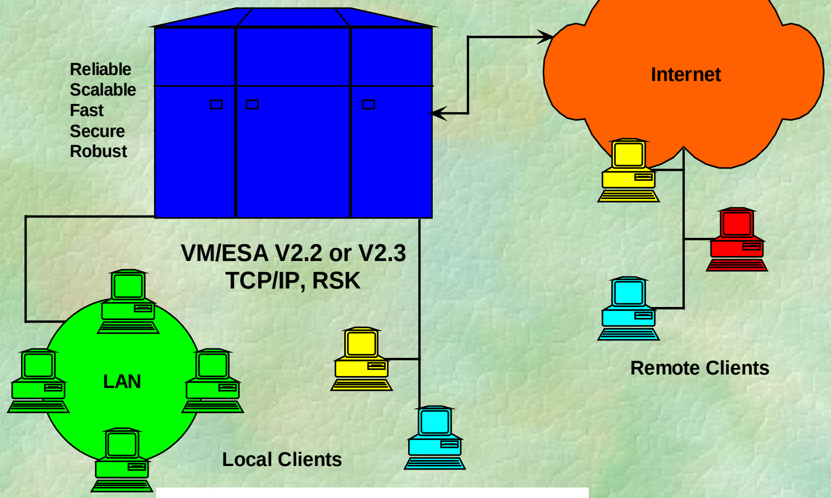
VM servers using the RSK with MAILSRV are serving hundreds of simultaneous requests.

Environment

- VM/ESA has all of the characteristics needed:
  - TCP/IP network and services
  - Reliable, fast, safe, file system (SFS)
  - Very good connectivity options
    - TCP/IP, IUCV, APPC, etc.
  - Access to large memory spaces
  - Scaleable to very large processors
  - Very high availability

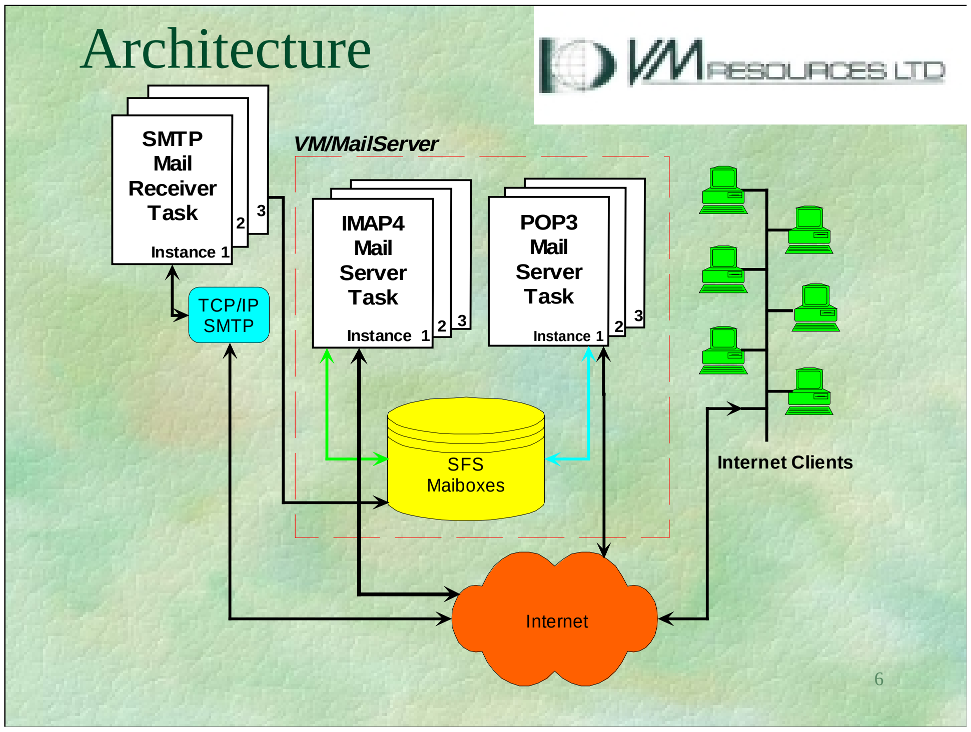
VM RESOURCES LTD

Environment



VM RESOURCES LTD

RSK on VM uses S/390 resources such as main memory, dataspaces, and VM implementations like the SFS, mapped minidisks.



VM is an internet server! The RSK provides, along with TCP/IP, a backbone of high powered server facilities. RSK provides a multithreading environment.

Features

- Full support for POP3 and IMAP4 RFCs.
- One Server for POP3 and IMAP4
- Stores mail boxes and mail items in SFS file space
- ‘Unlimited’ number of mail boxes and mail items
- Mail items stored as EBCDIC files.
- Deleted mail items can be recovered
- Mail user ids and passwords not kept in CP directory

VM RESOURCES LTD

RSK and VM Tools

- Based on Reusable Server Kernel (RSK) technology (CMS 13 or higher).
- RSK provides all communications support, multithreading, memory management.
- Extensive use of Callable Service Library routines for SFS requests.
  - Asynchronous SFS interfaces used.
- RSK is assembler level.
- May drive REXX exits.
- Line drivers provided in RSK for TCP, IUCV, APPC, CONSOLE, etc.

VM RESOURCES LTD

Mail User Information

- Users can be added to or deleted from the server without having to stop it
- Command interface provided.
- User ids and passwords maintained by server
- Enrollment information kept in VM data space for rapid retrieval; CMS File Image Always Available.
- User ids and passwords not limited to 8 or fewer characters (CP directory is not used)

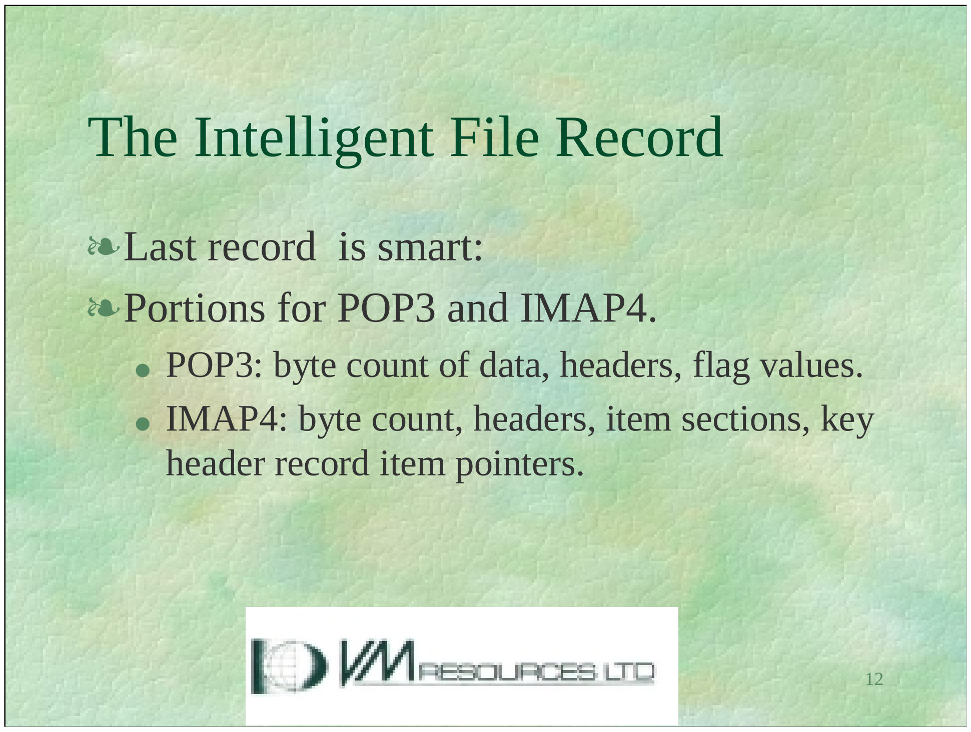
Terminology

Internet or RFC Term	**VM Definition**
Mail box	SFS directory
Mail Item	CMS file
ASCII	EBCDIC
Byte Files	Record Files with "Intelligent" Record
Horizontal Growth	Vertical Growth
Add Another Box	Grow Virtual Machine - or - Add Virtual Machine
Network	Network
Internet	Internet

VM RESOURCES LTD

VM Design

- Server is enrolled in the SFS file pool
- Server does not need file pool administrator authority
- Each user mail box is a separate subdirectory under the server’s top level directory
- CP Class G authority.
- TCP/IP services including SMTP.



The Intelligent File Record

- Last record is smart:
- Portions for POP3 and IMAP4.
  - POP3: byte count of data, headers, flag values.
  - IMAP4: byte count, headers, item sections, key header record item pointers.

VM RESOURCES LTD

By using the intelligent record, information about the file is self contained.

The intelligent record contains POP3 information and IMAP4 information.

So far, using the last record instead of the first has not shown itself to be

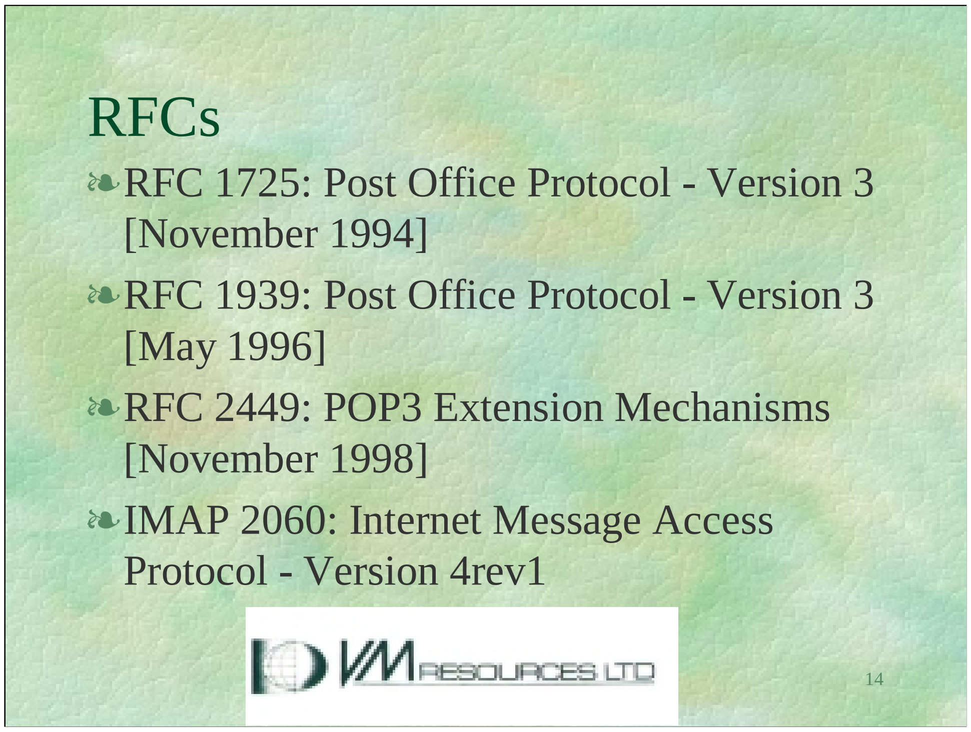
a performance bottleneck. This may be due to the in storage buffering techniques we use in our code.

Also, all the SFS file processing is done using the asynchronous facilities provided by the CSL interface to the SFS.

Client Support

- Supports all standard POP3 and IMAP4 clients.
- Support popular client command extensions
- Tested mail clients
  - Netscape 2.02 On OS/2
  - Netscape 4.7 On Windows NT
  - Eudora Pro 4.1 On Windows NT

VM RESOURCES LTD



RFCs

- RFC 1725: Post Office Protocol - Version 3 [November 1994]
- RFC 1939: Post Office Protocol - Version 3 [May 1996]
- RFC 2449: POP3 Extension Mechanisms [November 1998]
- IMAP 2060: Internet Message Access Protocol - Version 4rev1

VM RESOURCES LTD

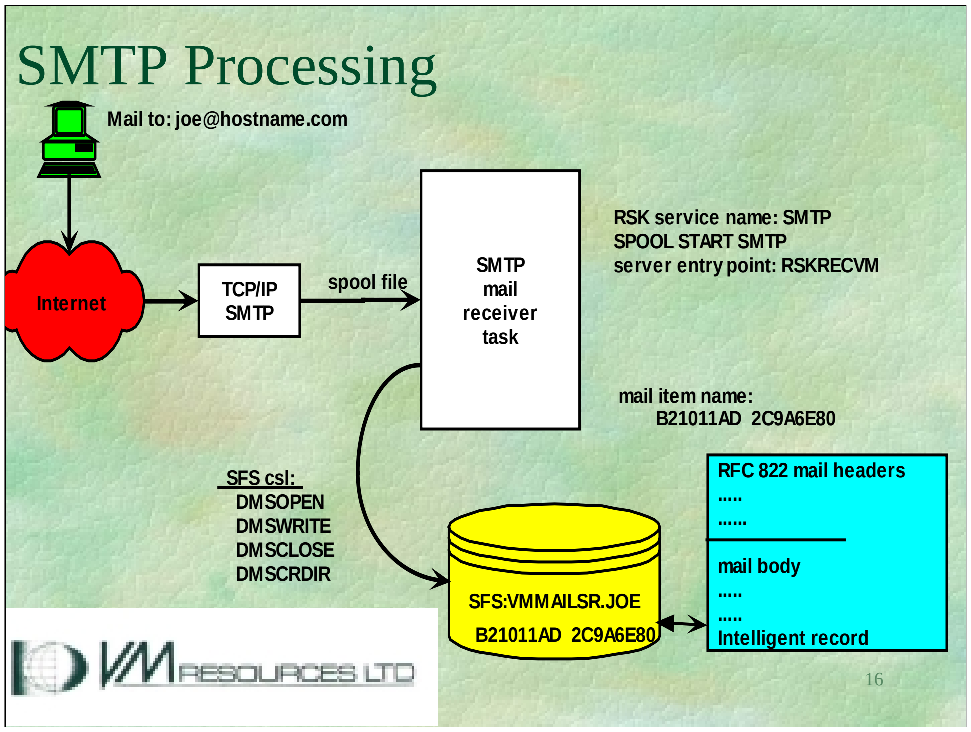
14

Most RFCs require a lot of reading to understand the intentions of the items being discussed!

CP Data

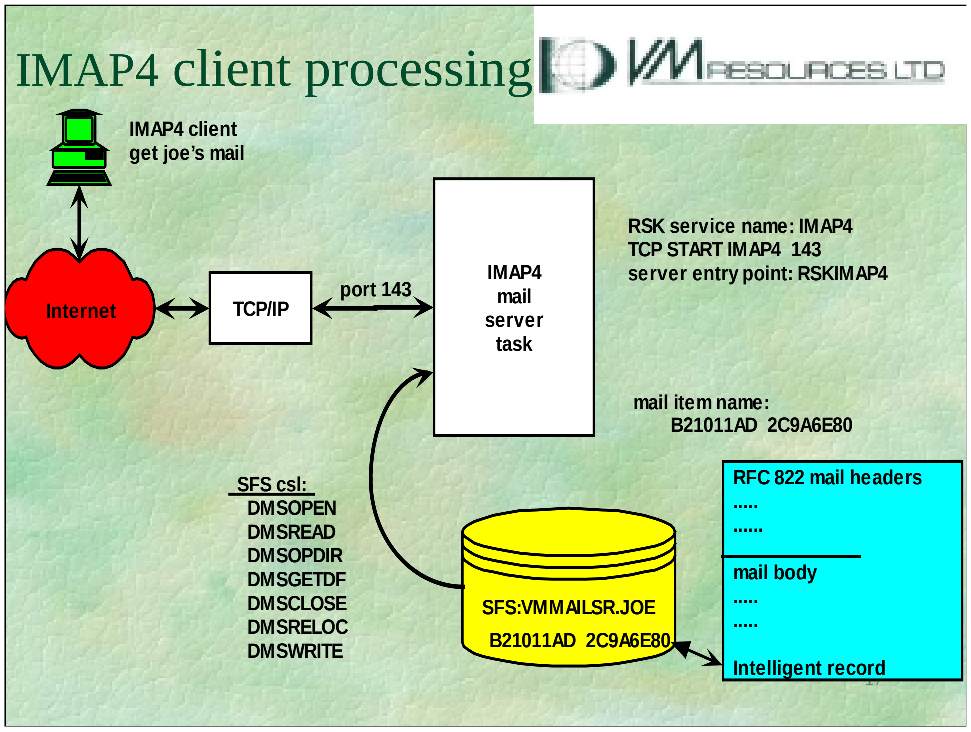
- Support for CP Monitor Data,(RSK specific and Mailsrv produced).
- Accounting Records.

VM RESOURCES LTD

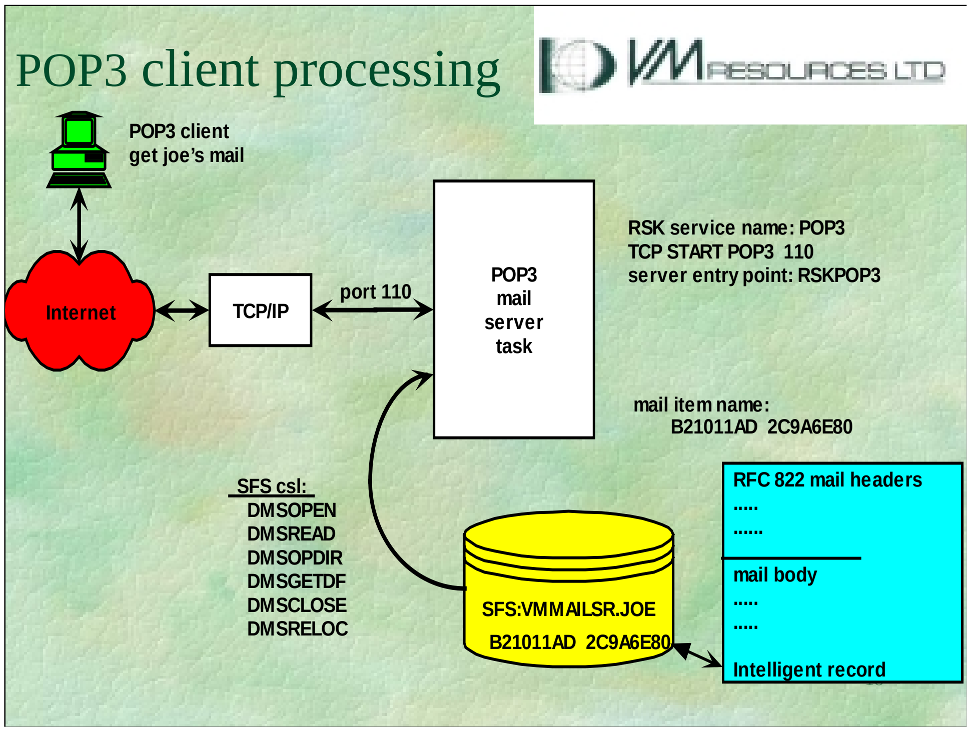


SMTP Processing

The SMTP task's responsibility is to retrieve the data sent from SMTP; parse it; form the intelligent record; and "lay it down" to the SFS. Since the RSK provides a SPPOL driver, and an SMTP listening port, we bind our code to those services. The SMTP task is multithreading.



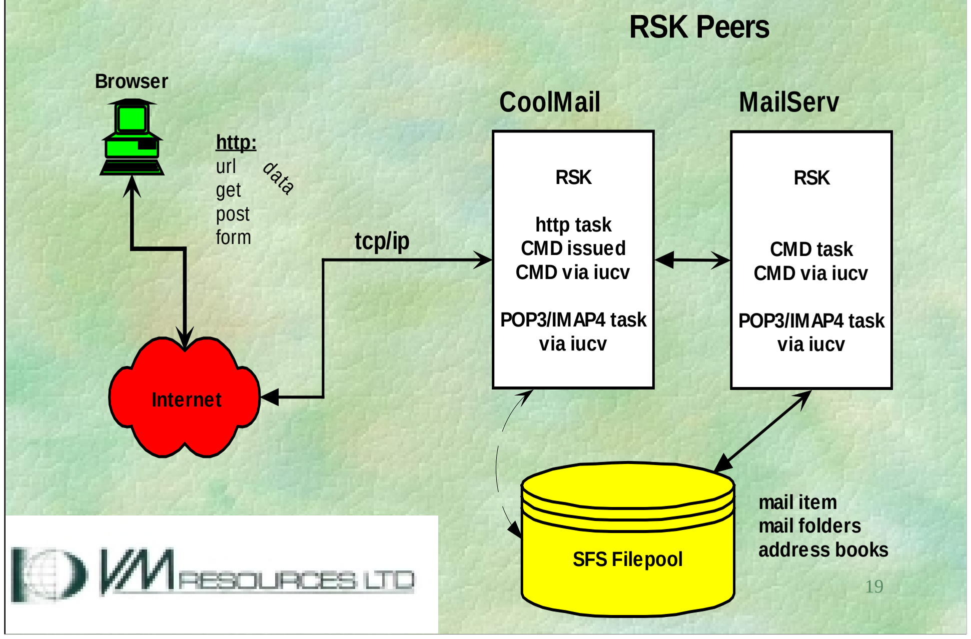
IMAP4 task uses RSK facilities to listen on TCP port 143, and handle the events. Multithreading. Reads and even writes files. IMAP4 task is created by us, and binds to the RSK service for TCP listening on 143.



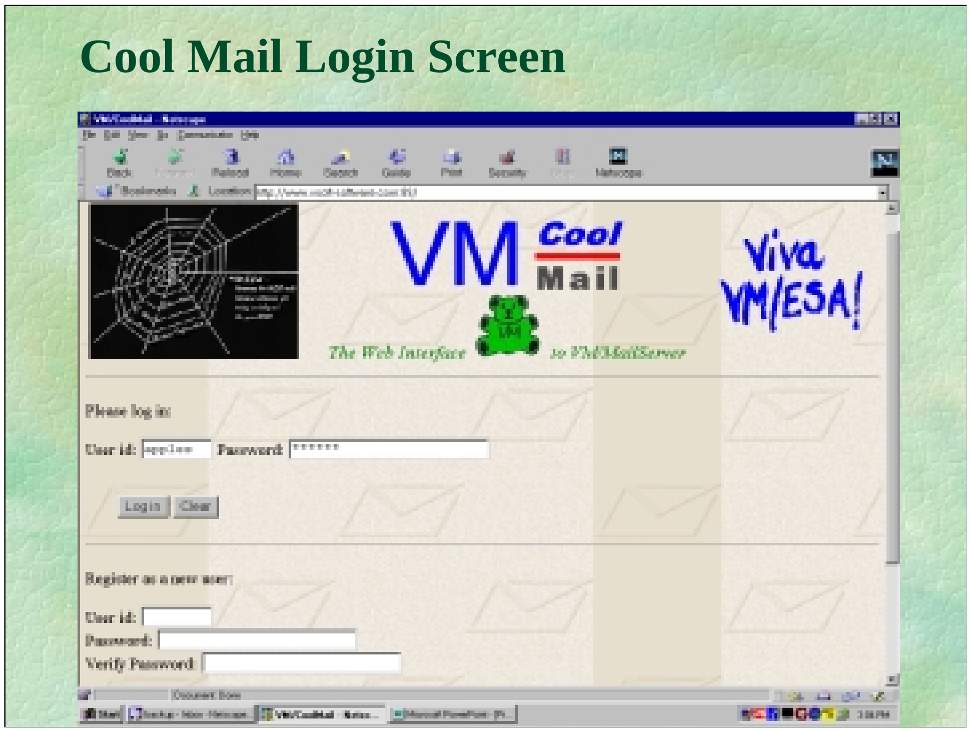
POP3 client processing

POP3 task is created by us; binds to TCP listening on port 110. Reads files. Multithreading.

Browser E-mail



VM/CoolMail Browser Based Email:
Requires VM/MailSrv
Controlled by your Site
Uses WebServe Lite in a RSK machine to handle HTTP task(s).
RSK to RSK peer communication.
Version with direct file interface.
Address books.
Version in progress with mail agent (CoolMail server acts like POP3/IMAP4 client.)



Cool Mail Login Screen

Please log in:

User id: Password: ******

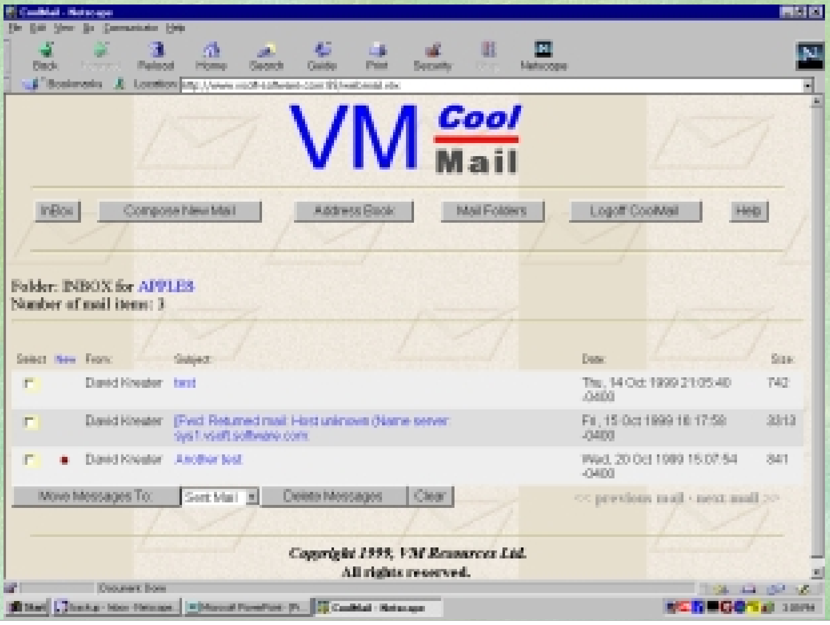
Login Clear

Register as a new user:

User id:
Password:
Verify Password:

The LOGIN screen as presented at our development site. Interfaces with MailSrv machine.

Mail Item Screen


Folder: INBOX for APPLES
Number of mail items: 3

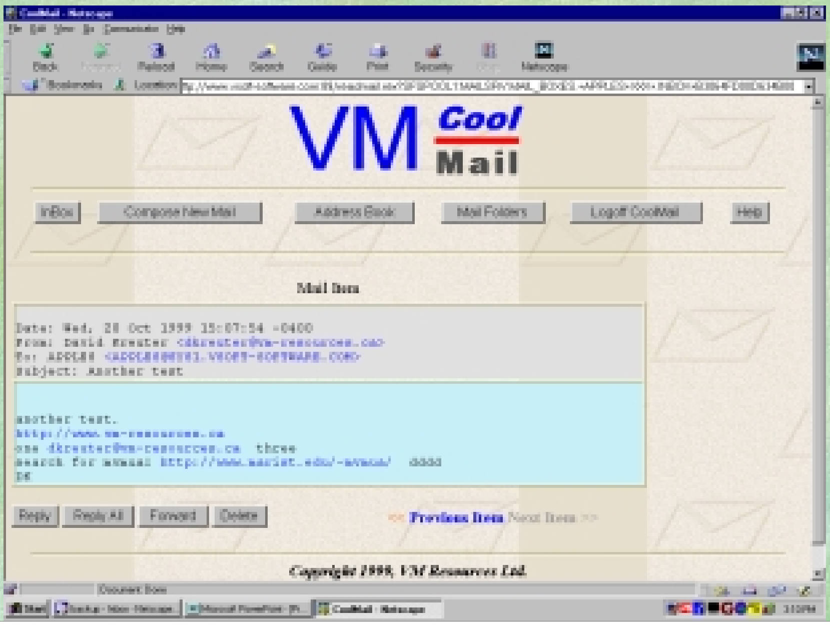
Select	New	From:	Subject:	Date:	Size:
		David Kreuter	test	Thu, 14 Oct 1999 21:05:40 -0400	742
		David Kreuter	[Fwd: Returned mail: Host unknown (Name server: sys1.vsoft.software.com	Fri, 15 Oct 1999 18:17:58 -0400	3313
	●	David Kreuter	Another test	Wed, 20 Oct 1999 15:07:54 -0400	841

<< previous mail · next mail >>

Copyright 1999, VM Resources Ltd.
All rights reserved.

Mail list screen as presented at our development site.

Mail Item Screen



Mail Item

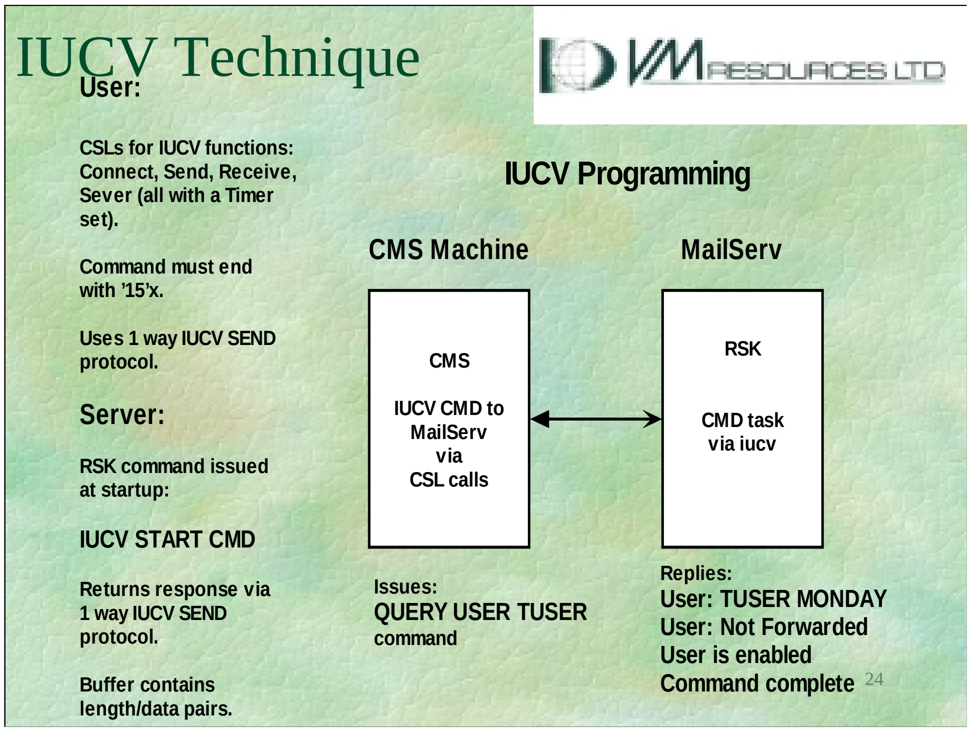
```
Date: Wed, 28 Oct 1999 15:07:54 -0400
From: David Kreuter <dkreuter@vm-resources.ca>
To: APPL88 <APPL88@VSI.VSOFT-SOFTWARE.COM>
Subject: Another test


another test.
http://www.vm-resources.ca
one dkreuter@vm-resources.ca  three
search for armeas http://www.marist.edu/~armeas/   dddd
DK
```

Copyright 1999, VM Resources Ltd.

Contents of mail item, with “links” activated as presented at our development site.

Neat RSK Techniques

- Non-RSK virtual machine communicates via IUCV to issue MailSrv commands.
- Traditional asynchronous timed IUCV requests (WAITECB/POST) stored in CSL.
- Tell RSK in CONFIG or CONSOLE command: IUCV START CMD



IUCV Technique

User:

CSLs for IUCV functions: Connect, Send, Receive, Sever (all with a Timer set).

Command must end with '15'x.

Uses 1 way IUCV SEND protocol.

Server:

RSK command issued at startup:

IUCV START CMD

Returns response via 1 way IUCV SEND protocol.

Buffer contains length/data pairs.

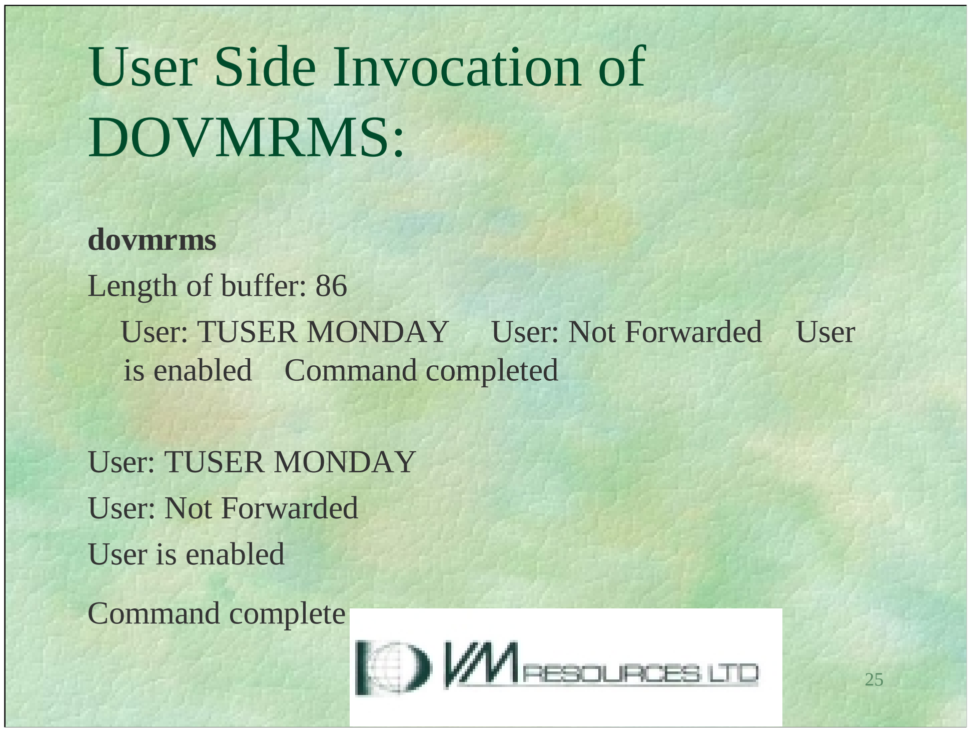
Issues:
QUERY USER TUSER
command

Replies:
User: TUSER MONDAY
User: Not Forwarded
User is enabled
Command complete

24

type dovmrms exec

```
/**/
parse source . . xcnm xctyp . . how .

serv = 'CMD'
lserv = length(serv)
VMID ='DKREUTER'
VMRC    ='VMRCONN '
Call CSL ('VMRC RETC REAS SERV LSERV VMID
TOKEN')
if retc <> '0' then do
  say date('s') time() xcnm 'Error from:' vmrc
  say date('s') time() xcnm 'Return code:' retc 'Reason
code:' reas
  return reas

end
```



User Side Invocation of DOVMRMS:

dovmrms

Length of buffer: 86

User: TUSER MONDAY User: Not Forwarded User is enabled Command completed

User: TUSER MONDAY

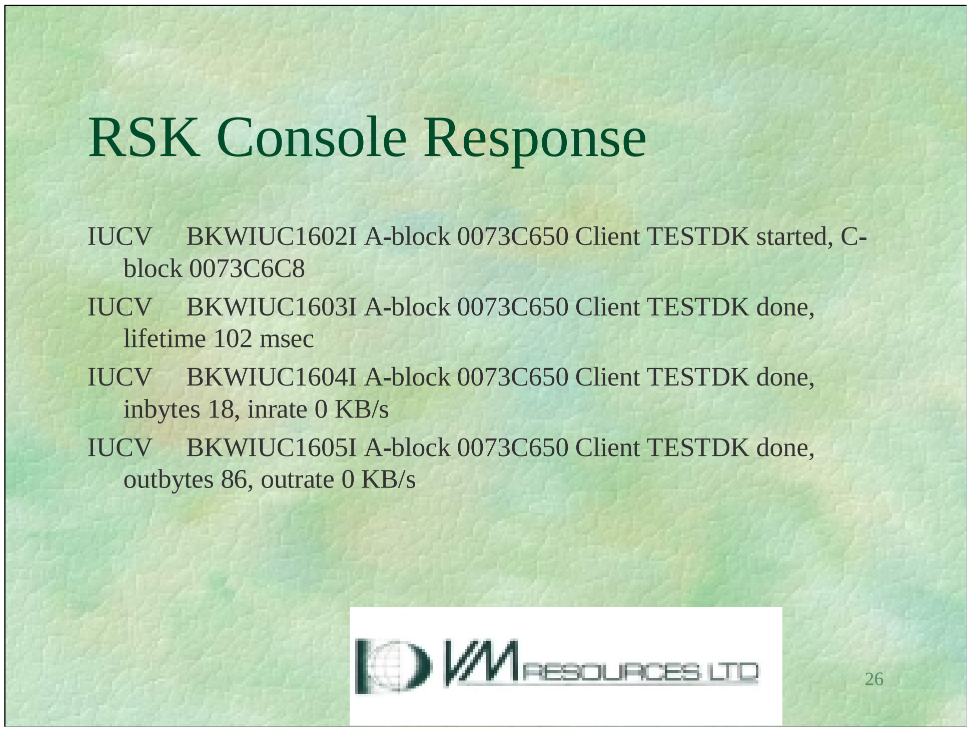
User: Not Forwarded

User is enabled

Command complete

25

```
…
 vmrc = ’VMRSEND ’
data = ’QUERY USER TUSER ’
data = data||’15’x
ldata = length(data)
Call CSL (’VMRC RETC REAS TOKEN DATA
LDATA’)
if retc <> ‘0’ then … < snip>

vmrc = ’VMRRECV’
recbuf = copies(’ ’,2560)
lrecbuf = length(recbuf)
Call CSL (’VMRC RETC REAS TOKEN RECBUF
LRECBUF ALRECBUF’)
if retc <> '0' then do     … < snip >

say ’Length of buffer:’ alrecbuf
say substr(recbuf,1,alrecbuf)
  …
```



RSK Console Response

```
IUCV     BKWIUC1602I A-block 0073C650 Client TESTDK started, C-
    block 0073C6C8
IUCV     BKWIUC1603I A-block 0073C650 Client TESTDK done,
    lifetime 102 msec
IUCV     BKWIUC1604I A-block 0073C650 Client TESTDK done,
    inbytes 18, inrate 0 KB/s
IUCV     BKWIUC1605I A-block 0073C650 Client TESTDK done,
    outbytes 86, outrate 0 KB/s
```

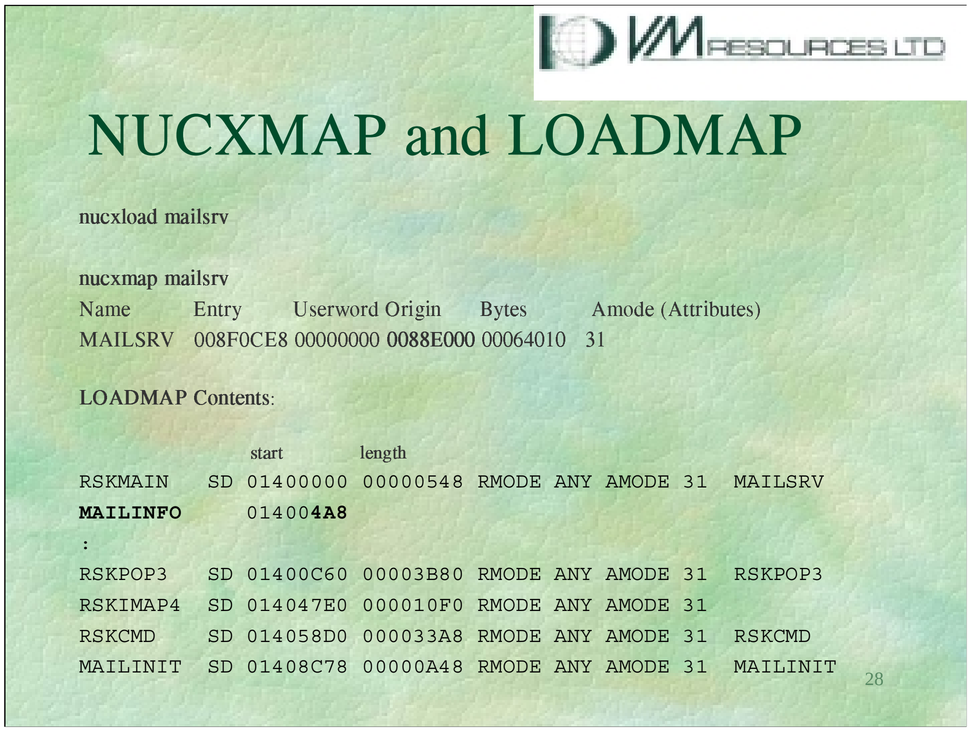
26

```
...
vmrc = 'VMRDISC '
Call CSL ('VMRC RETC REAS TOKEN')
if retc <> '0' then do
 ....  < snip>

tot = 0
do until tot >= alrecbuf
   buffl = x2d(c2x(substr(recbuf,1,4)))
   thisb = substr(recbuf,5,buffl)
   say thisb
   recbuf  = substr(recbuf,buffl+5,alrecbuf-buffl-5)
   tot = tot + buffl+4
end
exit
```

RSK Debugging Techniques

- Traditional CMS debugging with NUCXLOAD, NUCXMAP, CP TRACE, CP DISPLAY.

VM RESOURCES LTD



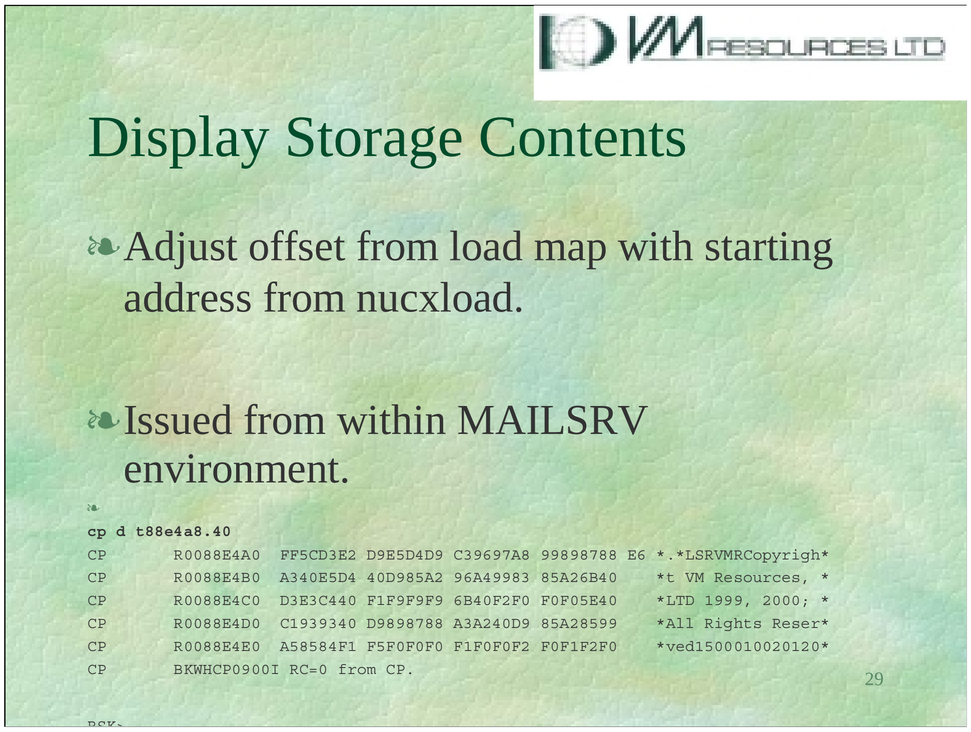
VM RESOURCES LTD

NUCXMAP and LOADMAP

```
nucxload mailsrv

nucxmap mailsrv
Name       Entry     Userword Origin   Bytes      Amode (Attributes)
MAILSRV    008F0CE8 00000000 0088E000 00064010   31

LOADMAP Contents:

                start      length
RSKMAIN    SD 01400000  00000548 RMODE ANY AMODE 31   MAILSRV
MAILINFO      014004A8
:
RSKPOP3    SD 01400C60  00003B80 RMODE ANY AMODE 31   RSKPOP3
RSKIMAP4   SD 014047E0  000010F0 RMODE ANY AMODE 31
RSKCMD     SD 014058D0  000033A8 RMODE ANY AMODE 31   RSKCMD
MAILINIT   SD 01408C78  00000A48 RMODE ANY AMODE 31   MAILINIT
```

A NUCXLOAD of MAILSRV is performed as first step in debugging.
NUCXMAP displays the address. Entry 8F0CE8 is where MAILSRV starts, but initial entry point for rsk startup routine is at 88E000. Page aligned address will make for easy hex arithmetic!

The load map was created at load time with the LOAD … (FULLMAP option. The original load address was 1400000. Of course, the NUCXLOAD will relocate to a different address, but the offsets of the routines are valid.
Since both origin points are page aligned, adding the offsets is straight forward.

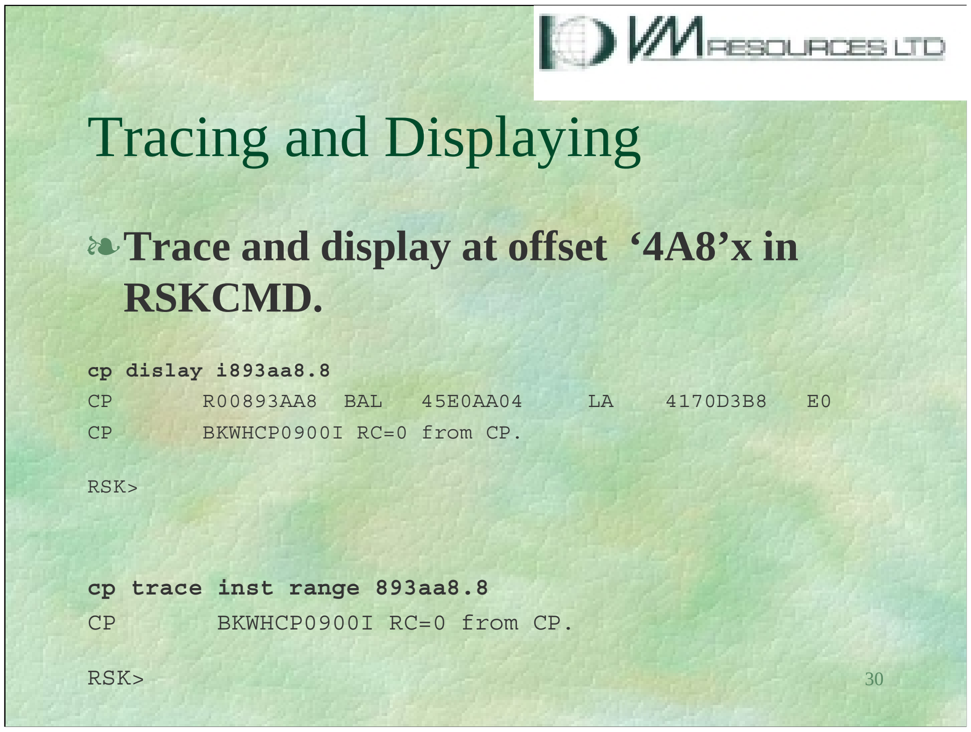


Display Storage Contents

- Adjust offset from load map with starting address from nucxload.
- Issued from within MAILSRV environment.

```
cp d t88e4a8.40
CP        R0088E4A0  FF5CD3E2 D9E5D4D9 C39697A8 99898788 E6 *.*LSRVMRCopyrigh*
CP        R0088E4B0  A340E5D4 40D985A2 96A49983 85A26B40    *t VM Resources, *
CP        R0088E4C0  D3E3C440 F1F9F9F9 6B40F2F0 F0F05E40    *LTD 1999, 2000; *
CP        R0088E4D0  C1939340 D9898788 A3A240D9 85A28599    *All Rights Reser*
CP        R0088E4E0  A58584F1 F5F0F0F0 F1F0F0F2 F0F1F2F0    *ved1500010020120*
CP        BKWHCP0900I RC=0 from CP.
```

Given the nucleus extension load address of 88E000, and the offset of MAILINFO as 4A8, adding together results in: 88E4A8.   This address is displayed for 64 bytes as shown. The command is issued with an RSK environment. RSK environment was created when MAILSRV was invoked (not shown).



Tracing and Displaying

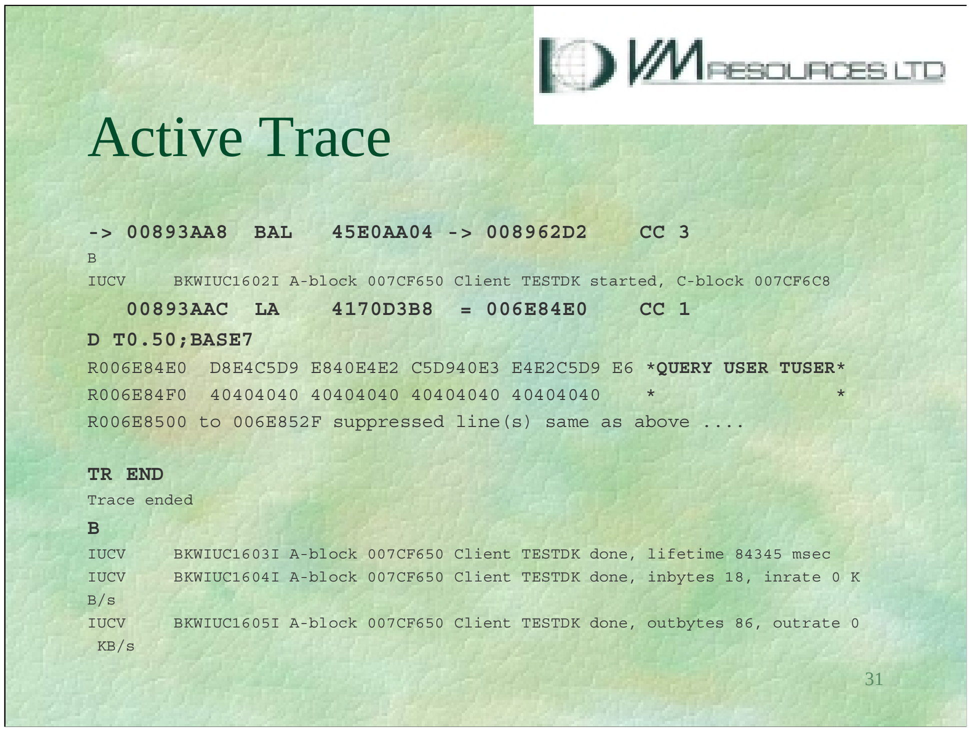
- **Trace and display at offset ‘4A8’x in RSKCMD.**

```
cp dislay i893aa8.8
CP        R00893AA8  BAL   45E0AA04      LA    4170D3B8   E0
CP        BKWHCP0900I RC=0 from CP.

RSK>


cp trace inst range 893aa8.8
CP        BKWHCP0900I RC=0 from CP.

RSK>
```

A debugging session was conducted in RSKCMD by studying it’s listing. The offset in RSKCMD that is of interest is 1D8. To derive this address, that offset is added to the offset of RSKCMD from the starting point of MAILSRV, as reported by NUCXLOAD.

So, 88E000 + 58D0 + 1D8 = 893AA8.

A display and then a trace is issued on that range for a few bytes.



Active Trace

```
-> 00893AA8  BAL   45E0AA04 -> 008962D2    CC 3
B
IUCV     BKWIUC1602I A-block 007CF650 Client TESTDK started, C-block 007CF6C8
   00893AAC  LA    4170D3B8  = 006E84E0    CC 1
D T0.50;BASE7
R006E84E0  D8E4C5D9 E840E4E2 C5D940E3 E4E2C5D9 E6 *QUERY USER TUSER*
R006E84F0  40404040 40404040 40404040 40404040    *                *
R006E8500 to 006E852F suppressed line(s) same as above ....

TR END
Trace ended
B
IUCV     BKWIUC1603I A-block 007CF650 Client TESTDK done, lifetime 84345 msec
IUCV     BKWIUC1604I A-block 007CF650 Client TESTDK done, inbytes 18, inrate 0 K
B/s
IUCV     BKWIUC1605I A-block 007CF650 Client TESTDK done, outbytes 86, outrate 0
 KB/s
```

The TRACE trap is triggered when the “client” issues the DOVMRMS EXEC, causing the IUCV path through the CMD handler to be issued.
The contents of register 7 is displayed which contains the command buffer, QUERY USER TUSER.

Today’s Presentation

- Goals for a VM Internet Servers
- Various Internet RFCs
- VM Architecture Advantages
- The CMS Reusable Server Kernel
- How VM Mail Server Works
- Techniques, Debugging, and Tips

VM RESOURCES LTD