IBM Software

Operational Monitoring and Automation for a System z-based Cloud

z/VM and Linux on System z Guests

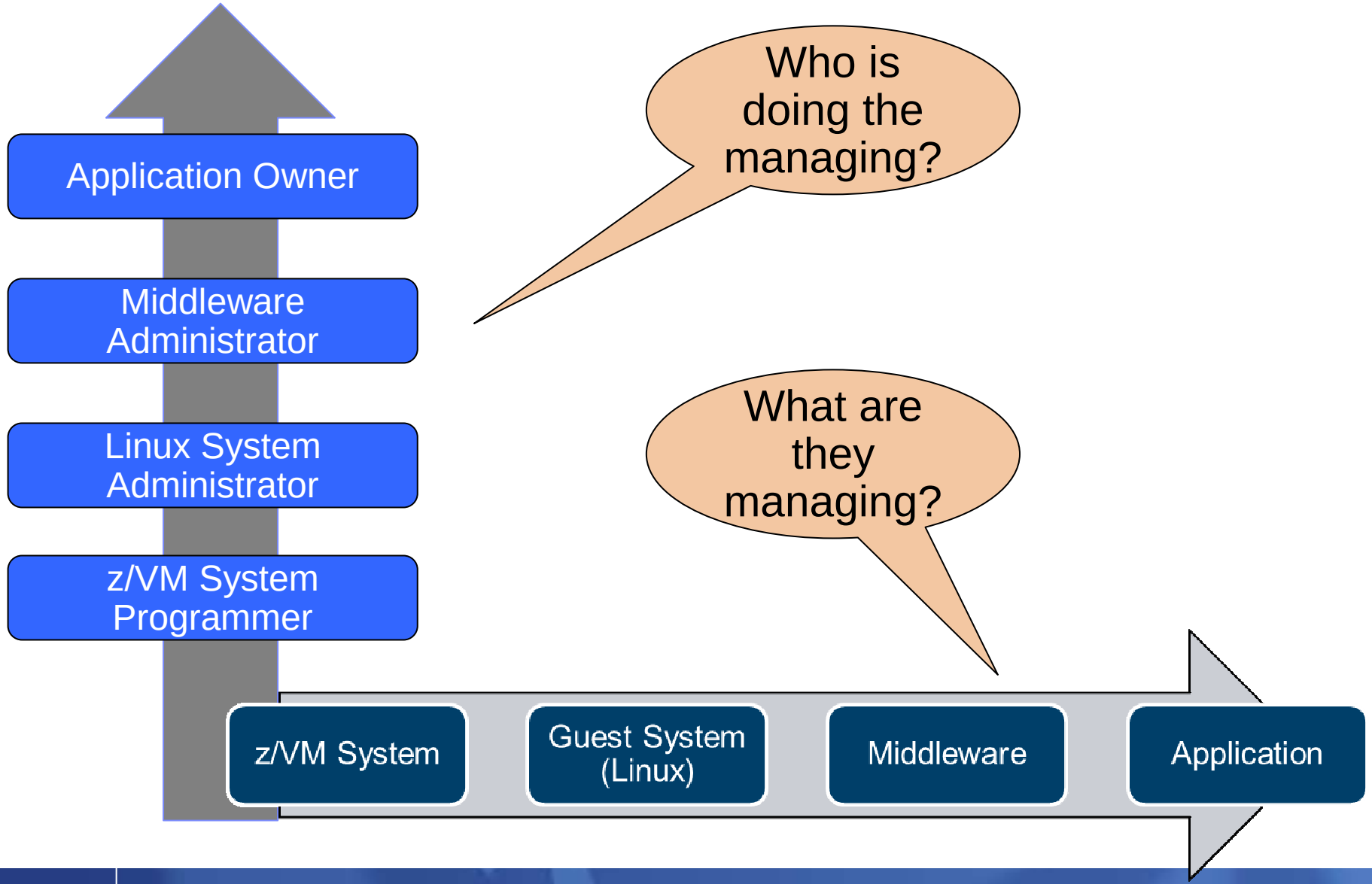
Tracy Dean, IBM
tld1@us.ibm.com

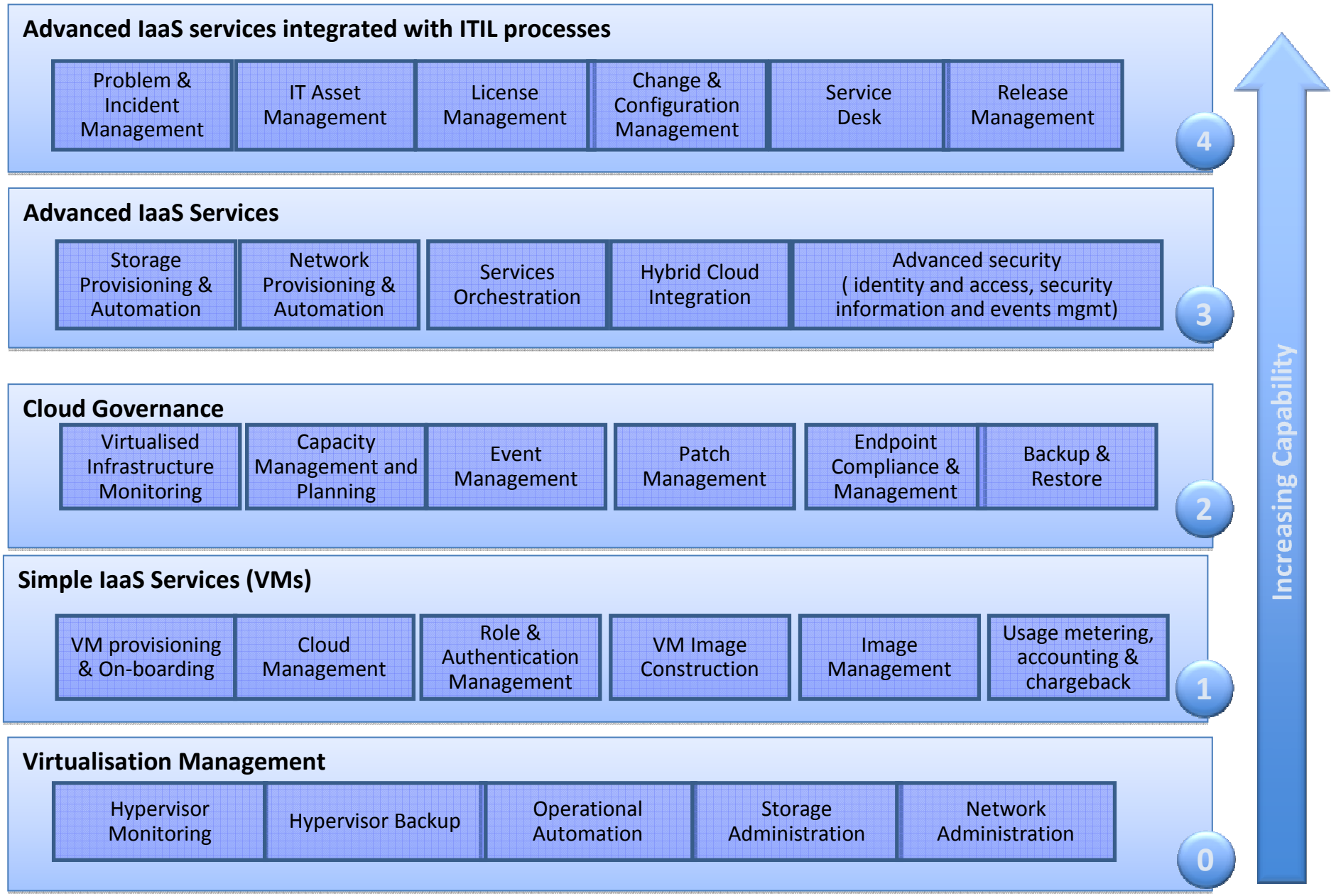
June 2014

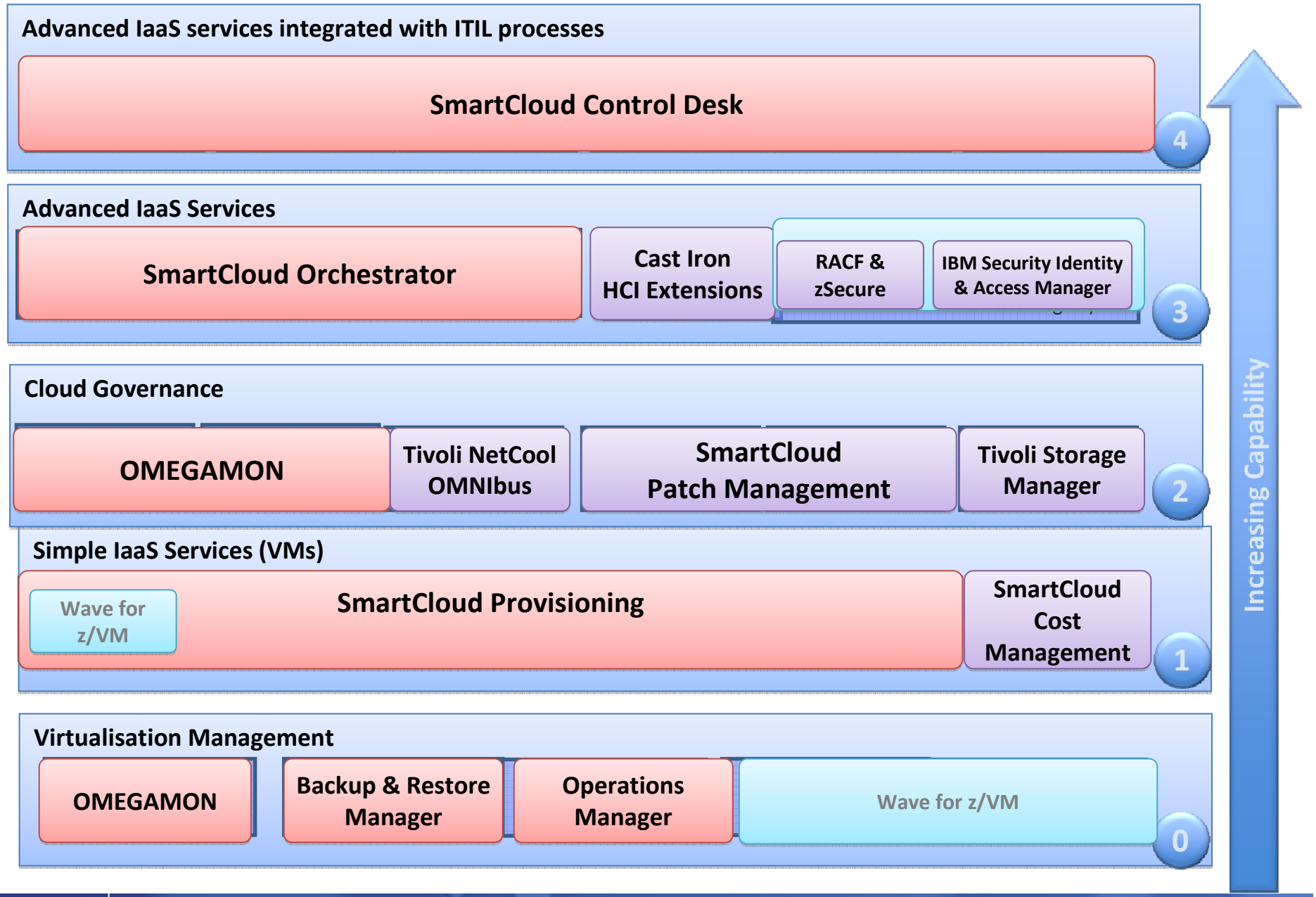
Agenda

- **Where do the IBM products fit**
- **Introduction to recommended practices and examples**
- **Brief overview of product being used**
  - IBM Operations Manager for z/VM
    - What's new in V1.5 – available October 25, 2013
- **Considerations for z/VM Single System Image**
- **Recommended practices in detail**
  - Live demonstrations
    - Configuration and sample code
- **Summary**
  - Reference information
  - Additional demos
    - Configuration options and sample code for all demos

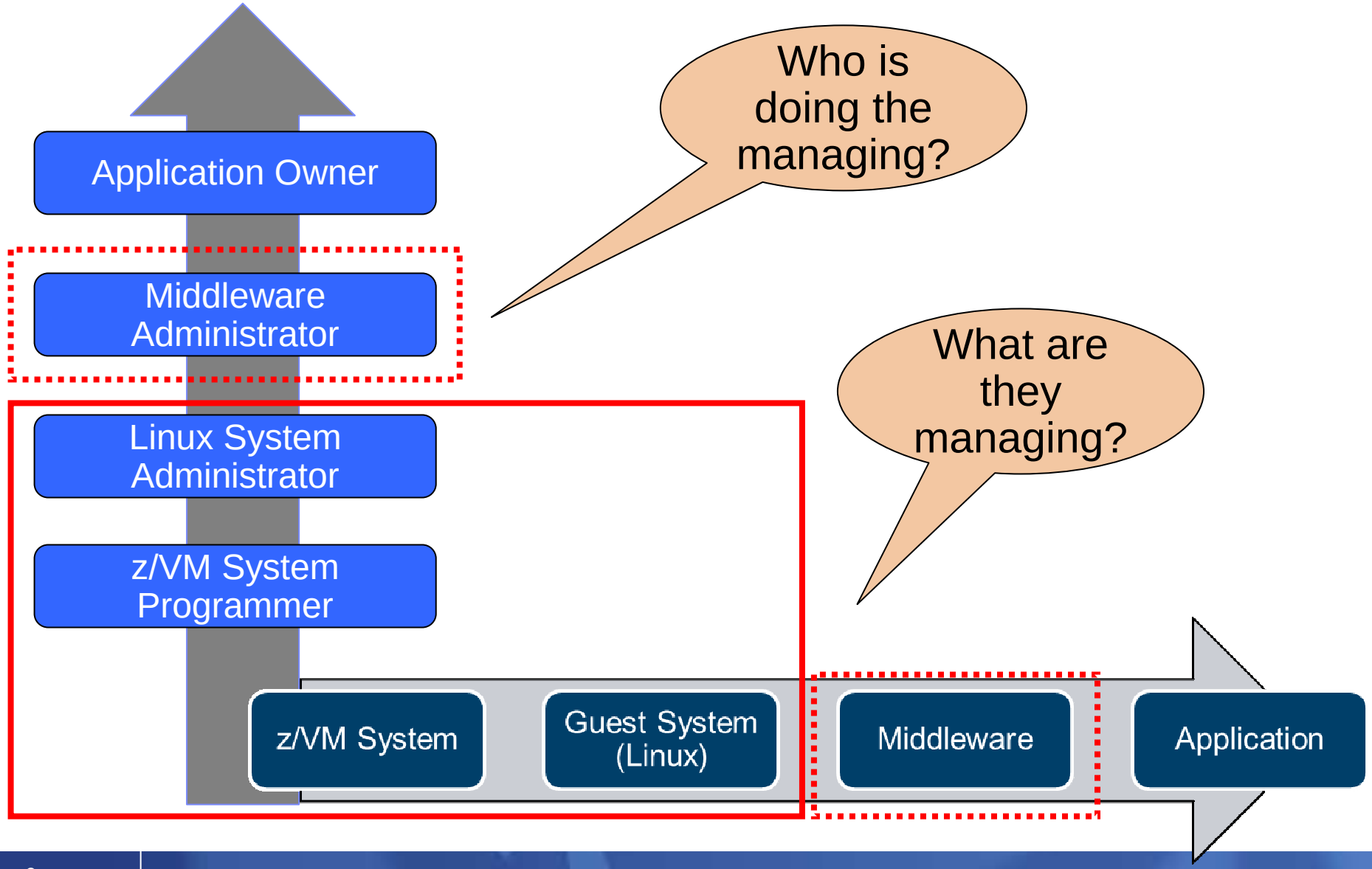
Dimensions of Systems Management







Dimensions of Systems Management



Managing z/VM and Linux on System z

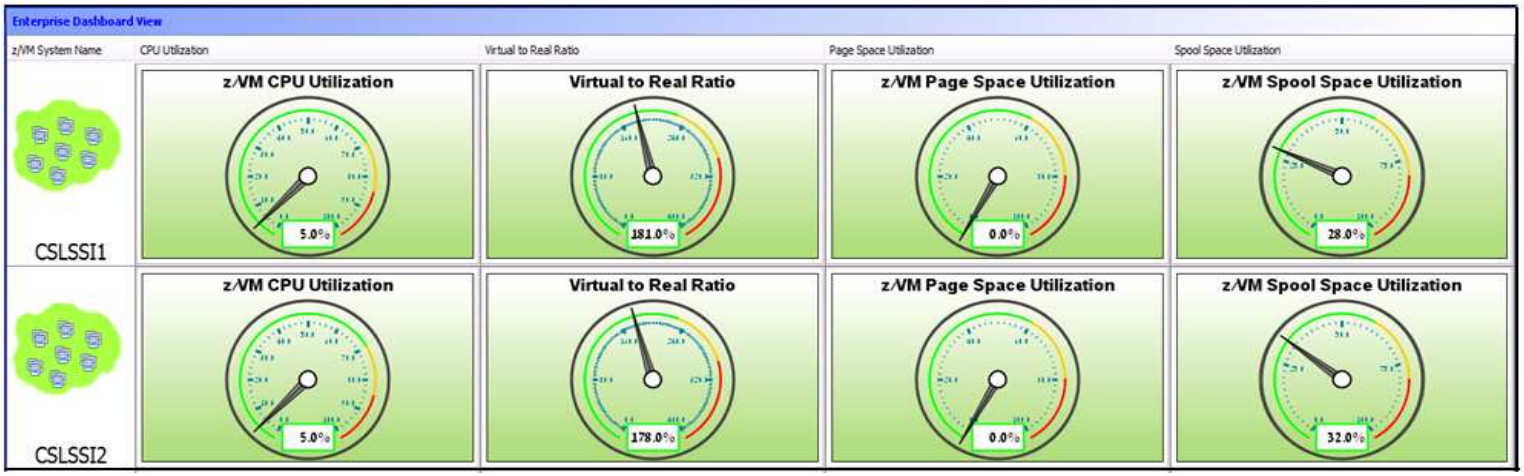
- **Security**
  - RACF and zSecure Manager for z/VM
- **Performance monitoring**
  - OMEGAMON XE on z/VM and Linux
- **Operational monitoring and automation**
  - Operations Manager for z/VM
  - Including integration with existing monitoring and alert systems
- **Backup and recovery**
  - Backup and Restore Manager for z/VM
  - Tape Manager for z/VM
  - Tivoli Storage Manager
- **Interactive provisioning and system resource management**
  - IBM Wave for z/VM

IBM Wave for z/VM and Operations Manager for z/VM

- **IBM Wave for z/VM provides an interactive GUI interface for:**
  - Provisioning of Linux guests
  - Basic performance information
  - Monitoring of virtual server resources
- **Operations Manager for z/VM provides operational monitoring & automation**
  - In the background
    - Monitoring of console messages for z/VM service machines and Linux guests
    - Monitoring “state” information for z/VM service machines and Linux guests
    - Monitoring spool and page space on the z/VM system
    - Automated responses to these monitors when they are triggered
      - Email
      - SNMP alerts
      - Integration with IBM Tivoli Netcool/OMNIbus enterprise alert system
      - Actions that address the problem immediately in addition to or instead of alert notification
  - Interactive when needed
    - View and interact with live service machine and Linux guest consoles
    - View and manage spool files
- **Complementary solutions**
  - Use Operations Manager to monitor Wave service machines
  - Use Operations Manager to automatically initiate tasks in Wave via the Wave CLI

IBM Wave for z/VM and OMEGAMON XE on z/VM and Linux

- **IBM Wave for z/VM** provides point in time monitoring of virtual server resources from a single graphical interface



- **OMEGAMON XE on z/VM and Linux** provides
  - Deeper level monitoring of z/VM
  - Deeper level monitoring of individual Linux guest environments
  - Ability to set service level thresholds and generate events when exceeded
  - Historical view of monitoring data
- Both OMEGAMON XE on z/VM and Linux and IBM Wave can coexist in customer environments
- Both gather the data from the Performance Toolkit for z/VM

Recommended Practices – Operational Management

View and issue commands on consoles of Linux guests and CMS service machines

- Operations staff monitoring multiple consoles or a central console of alerts
- System programmers debugging a problem on a guest or service machine

Generate alerts and/or automatically recover from

- Abend, termination, or error messages
- Service machine disks approaching full
- Critical user IDs or guests being logged off or entering error state
- Spool and/or page space approaching full

Schedule automated system maintenance procedures

- Spool cleanup based on policies
- Minidisk cleanup (from logs), including archiving
- Orderly startup and shutdown
  - Relocation of critical guests to another SSI member
- Backups of z/VM system

Automation Demos Available

1. **Send an e-mail based on a console message**
2. **Send an alert to Netcool/OMNIbus based on a console message, hold and unhold messages**
   a. Using POSTZMSG interface to Netcool/OMNIbus
   b. Using SNMP interface to Netcool/OMNIbus
3. **Send a message or email if spool approaches full**
   a. Send a message if spool usage is too high on any member of an SSI Cluster
   b. Send an email if spool usage is too high on a single system
4. **View and clean up spool files**
5. **Automated spool cleanup**
6. **Archiving DIRMAINT's log files when disk gets full**
7. **Process a file of test messages as a console**
8. **Process Linux syslog data as a console**
9. **Create a central operations console on one z/VM system**
10. **Create a central operations console across multiple z/VM systems**
    a. When the systems are in an SSI cluster
    b. When the systems are not in an SSI cluster
11. **Integration with OMEGAMON XE on z/VM and Linux - take action based on CPU usage of Linux guest**
12. **Monitor service machines for logoff – and autolog them**
13. **Send an email if page space approaches full**
14. **Monitor SSI connectivity between 2 cluster members**
15. **Suppress passwords on Linux consoles**
16. **Autolog a Linux Guest and Send Message if Doesn't Start Successfully**
17. **View consoles of Linux guests, Linux syslog data, and CMS user IDs or service machines**

IBM Software

Product Overview

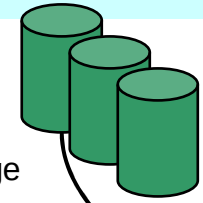
IBM Operations Manager for z/VM

Operations Manager for z/VM

Increase productivity
- Authorized users to view and interact with monitored virtual machines without logging onto them
- Multiple users view/interact with a virtual machine simultaneously

Improve system availability
- Monitor virtual machines and processes
- Take automated actions based on console messages
- Reduce problems due to operator error

Schedule tasks



Monitor page and spool usage

Respond to system events (user state changes)

Console monitoring

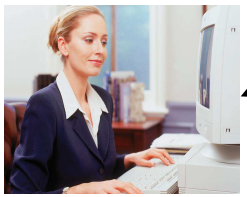
Service Virtual Machine being monitored

Idle monitor

Operations Manager for z/VM

Take action

Service Virtual Machine being monitored



- View & interact with consoles
- View spool files

Console monitoring

Automation
- Routine activities done more effectively with minimal operations staff
- Schedule tasks to occur on a regular basis

Integration
- Fulfill take action requests from performance monitoring products (e.g. OMEGAMON XE on z/VM and Linux)
- Send alerts to email, central event management systems (e.g. Netcool/OMNIbus), etc.

Features and Functions

- **Monitor service machine consoles**
- **Monitor page space and spool usage**
- **Monitor system events**
- **Schedule events/actions**
- **Take actions automatically based on monitoring results**
- **View and interact with monitored consoles from authorized user IDs**
- **Find and view spool files**
- **Dynamic configuration**
- **Separation of access control**

Dynamic Configuration

- **Initial configuration file loaded at startup**
  - May imbed other configuration files
  - Filename can be a substitution variable for the system name
- **Most configuration options can be updated while Operations Manager is running**
  - Add, delete, or change:
    - Rules, actions, monitors, schedules, holidays, groups, user authorization
  - Suspend or resume rules, monitors, schedules
- **Multiple methods**
  - CMS command interface
  - (Re)load a new or updated configuration file
  - Commands in action routines
- **Sample configuration files provided**
  - Includes some of the demos in this presentation
    - Operations Manager configuration statements
    - Sample REXX code

IBM Software

View and Issue Commands on Consoles

Linux Guests and CMS Service Machines

Generate Alerts and/or Automatically Recover From

Abend Messages
Termination Messages
Error Messages

Monitor Service Machine Consoles

Operations Manager

syslog data

```
slog Message 1
slog Message 2
slog Message 3
…
```

OPERATOR

```
OPER Message 1
OPER Message 2
OPER Message 3
…
```

LINUX

```
LNX Message 1
LNX Message 2
LNX Message 3
…
```

TCPIP

```
TCP Message 1
TCP Message 2
TCP Message 3
…
```

Test Data

```
TEST Message 1
TEST Message 2
TEST Message 3
…
```

Data space 1

```
TEST Message 1
TEST Message 2
…
```

Data space 2

```
OPER Message 1
OPER Message 2
…
```

Data space 3

```
LNX Message 1
LNX Message 3
…
```

Data space 4

```
TCP Message 1
TCP Message 2
…
```

Data space 5

```
slog Message 1
slog Message 2
slog Message 3
…
```

Filtered

Data space 6

```
OPER Message 1
LNX Message 1
LNX Message 2
TCP Message 1
slog Message 1
OPM Message 1
TEST Message 1
OPER Message 2
…
```

Daily log

```
OPER Message 1
LNX Message 1
LNX Message 2
TCP Message 1
slog Message 1
OPM Message 1
```

Unfiltered

View and Interact with Consoles

- **Authorized users can view live consoles of monitored service machines & guests**
  - Multiple users can view the same console simultaneously
  - No need to logon to the user ID to see its console
    - No interruption of the user ID
  - No need to create and close console files of disjointed data
  - Test data and Linux syslog data treated as a “console”
  - Views can be defined to look at a group of consoles in one view
  - **Can specify a date and time range for your view within currently available data**
  - Can request a copy of the current console data for a user or set of users
  - **Format of date in the view is based on requestor’s CP DATEFORMAT setting**
- **Full screen mode**
  - Scroll up and down to view and search historical data
  - Auto scroll (on or off) as new output is displayed on the console
  - From command line, issue commands back to the monitored console
- **Amount of data that is visible depends on specified or default data space size**
  - **Or date/time range specified**
- **Rules/actions may modify the view**
  - Suppress messages from the console
  - Hold or highlight messages with color, blinking, etc.
- **Authorized users can view the log file**
  - Can also request a copy of the log file from today or a previous day

Monitor Service Machines

- **Define rules to**
  - Scan console messages for text matching
    - Includes column, wildcard, and exclusion support
    - Optionally restrict to specific user ID(s)
  - Take actions based on matches
- **Multiple rules can apply to one message**
  - Rules processed in order of definition in the configuration file
  - FINAL option available to indicate no additional rules should be evaluated

Executing Actions

- **Define action(s) to be triggered**
  - Specify action to take as part of the console rule definition
    - Action is taken when match is found
  - Types of actions
    - Change color, highlight, hold, or suppress a console message
    - CP or CMS commands
    - REXX EXECs
    - Write data out on a TCP/IP port
      - E.g. send data to a syslog daemon/server
- **Dynamically include data about the triggering event in the action**
  - Available to the action via substitution variables
- **Take multiple actions based on one message**
  - Chain actions together
  - Limit the number of times an action is taken in a specified period of time

IBM Software

Generate Alerts and/or Automatically Recover From

Critical User IDs or Guests Logging Off
Critical User IDs or Guests Enter Error State

Respond to System Events

- **Create monitors for z/VM system events (*VMEVENT) related to user IDs**
  - Class 0
    - 0 - Logon
    - 1 - Logoff
    - 2 - Failure condition (including CP READ and Disabled Wait)
    - 3 - Logoff timeout started
    - 4 - Forced sleep started
    - 5 - Runnable state entered (VM READ)
    - 6 - Free storage limit exceeded
    - 9 - Outbound relocation started
    - 10 - Inbound relocation started
    - 11 - Outbound relocation complete
    - 12 - Inbound relocation complete
    - 13 - Outbound relocation terminated
    - 14 - Inbound relocation terminated
    - 15 – Timebomb exploded
- **Additional classes also supported**
- **Optionally restrict to specific user ID(s)**
- **Specify the action associated with the event**
  - Actions specified are the same as those for schedules, console rules, and other monitors

IBM Software

Generate Alerts and/or Automatically Recover From

Spool Space Approaching Full
Page Space Approaching Full

Monitor Page and Spool Usage, View Spool Files

- **Create page and spool space monitors to trigger actions when**
  - Percent of spool usage falls within a specified range
  - Percent of spool usage increases at a specified rate
  - Percent of page space usage falls within a specified range
  - Percent of page space usage increases at a specified rate
- **Actions triggered can be the same actions used by console monitoring**
- **For spool files, authorized users can**
  - Display a list of spool files based on one or more attributes
    - Owner
    - Size
    - Date created
  - From the list, the user can
    - Sort the list on any of the available columns
    - View the contents of an individual spool file
    - Purge, transfer, or change a spool file

IBM Software

Schedule Automated System Maintenance Procedures

Monitor for Rules, Monitors and Schedules that Were Not Triggered

Spool Cleanup Based on Policies
Backups
Disk Cleanup
Orderly Startup and Shutdown

Schedule Events and Actions

- **Define schedules**
  - Hourly, daily, weekly, monthly, or yearly, nth weekday of the month
  - Once on specified month, day, year, and time
  - Based on ISO week definitions (week number; even, odd, first, last week)
  - At regular intervals
    - Every x hours and y minutes
  - Within a specified window of time
    - Specify start time
    - Specify conflicting schedules
    - Specify maximum time to defer this schedule
  - Within limits
    - Restrict to specific days of the week: Monday through Sunday plus holidays
    - Restrict to certain hours of the day
- **Specify the action associated with the schedule**
  - Actions specified are the same as those for console rules and all other monitors

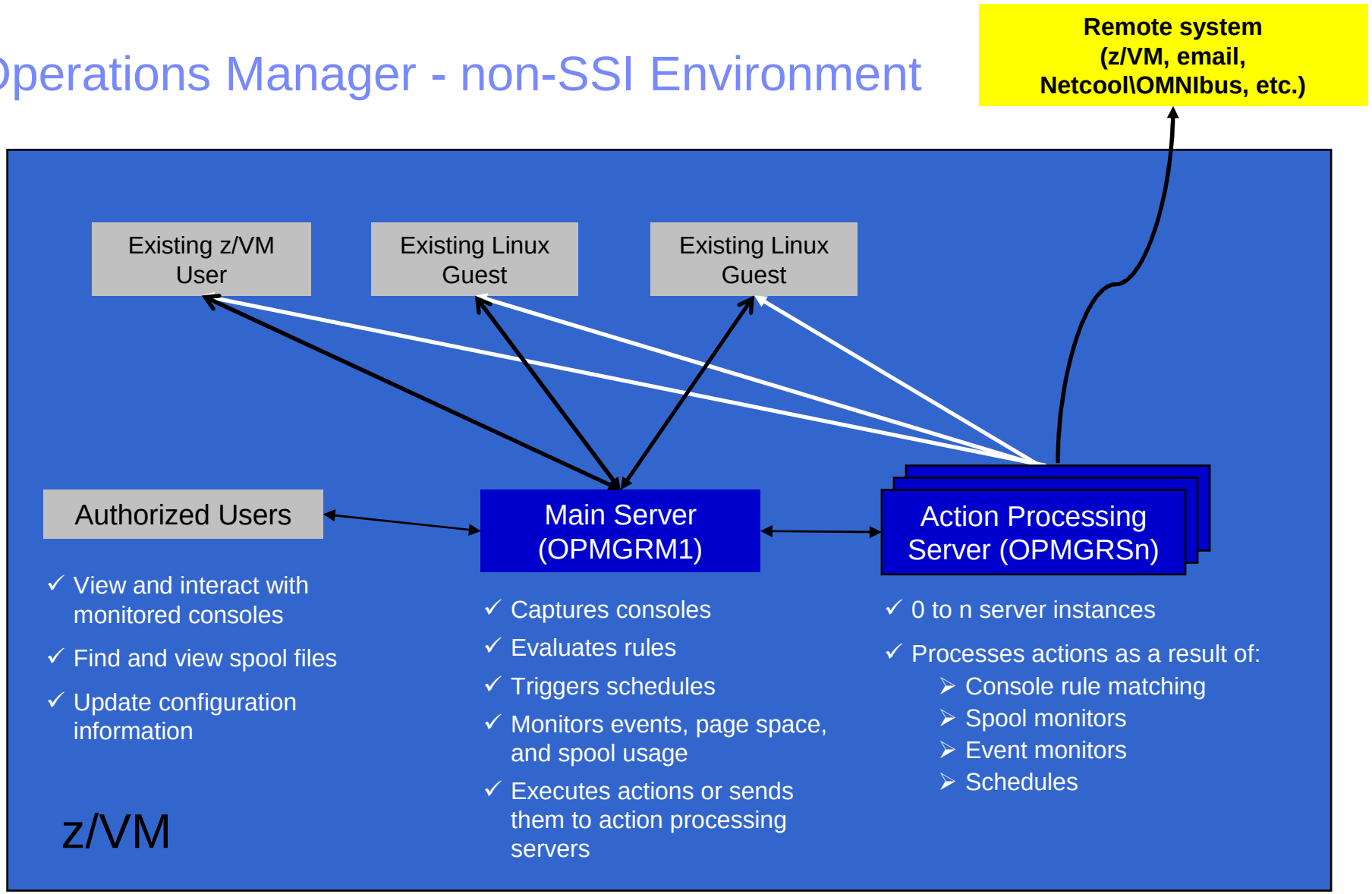
Idle Monitors

- **Define idle monitors**
  - Watch for idle rules, schedules, and monitors
    - Rule, schedule, or monitor **not** triggered ***n*** number of times within specified period of time
- **Specify the action associated with the idle monitor**
  - Actions specified are the same as those for schedules, console rules, other monitors

IBM Software

SSI vs non-SSI Considerations

Operations Manager - non-SSI Environment



IBM Software

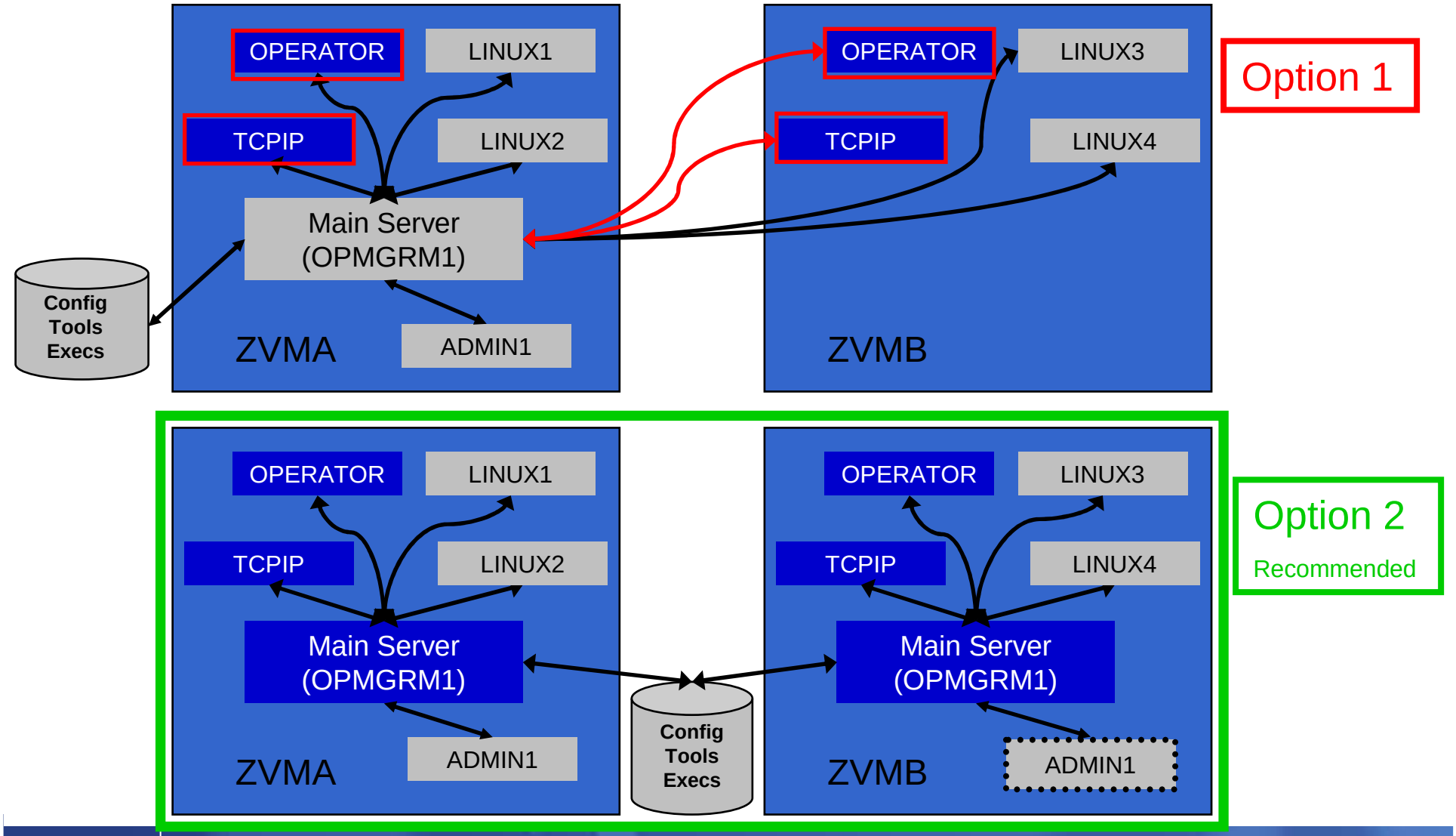
SSI Considerations

Console Monitoring

Single Config User

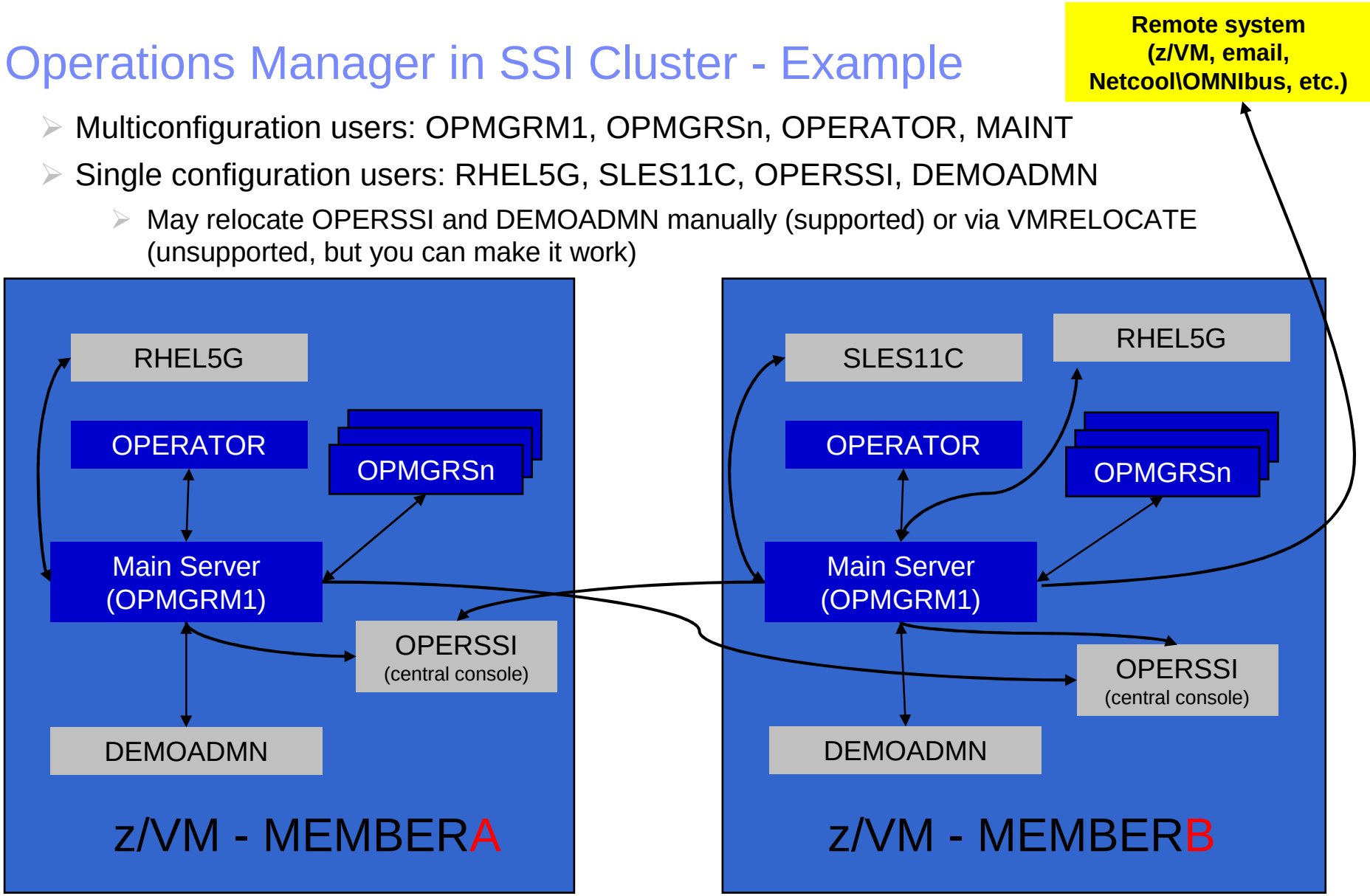
Multiconfig User

SSI Considerations for Console Monitoring



Operations Manager in SSI Cluster - Example

- Multiconfiguration users: OPMGRM1, OPMGRSn, OPERATOR, MAINT
- Single configuration users: RHEL5G, SLES11C, OPERSSI, DEMOADMN
  - May relocate OPERSSI and DEMOADMN manually (supported) or via VMRELOCATE (unsupported, but you can make it work)



Relocating OPERSSI and DEMOADMN (CMS Users) …

- **VMRELOCATE for CMS user IDs not officially supported**
- **Can be done for some CMS users**
  - Create single configuration user ID for z/VM system disks
  - Copy MAINT 190, 19D, 19E to minidisks owned by this new user ID
  - Relocateable CMS user must IPL from identical NSS (CMS) or minidisk (190)
    - Use SPXTAPE to copy CMS NSS
      - VMRELOCATE uses checksum of NSS to determine if identical
      - CMS NSS includes date/time it was loaded
    - Or, have relocateable CMS users IPL 190 instead of IPL CMS

OPERSSI DIRECT

```
USER OPERSSI …
…
OPTION CHPIDVIRTUALIZATION ONE
…
IPL 190
…
LINK CMAINT 0190 0190 RR
LINK CMAINT 019D 019D RR
LINK CMAINT 019E 019E RR
…
```

PROFILE EXEC

```
/* PROFILE EXEC for OPERSSI */
…
'SET RELPAGE OFF'
…
```

… Relocating OPERSSI and DEMOADMN (CMS Users)

- **Beware**
  - It's worth repeating … VMRELOCATE for CMS user IDs not officially supported
  - All members of the cluster must be kept at same z/VM (or at least CMS) code level
  - If IPL 190, will use more memory as each user ID will have private copy of CMS
  - SET RELPAGE OFF may have a negative impact on overall system performance
  - Only works for "basic" CMS users
    - All relocation rules still apply
    - E.g. user IDs connecting to VMCF or IUCV can't relocate

Monitor Service Machines - Considerations

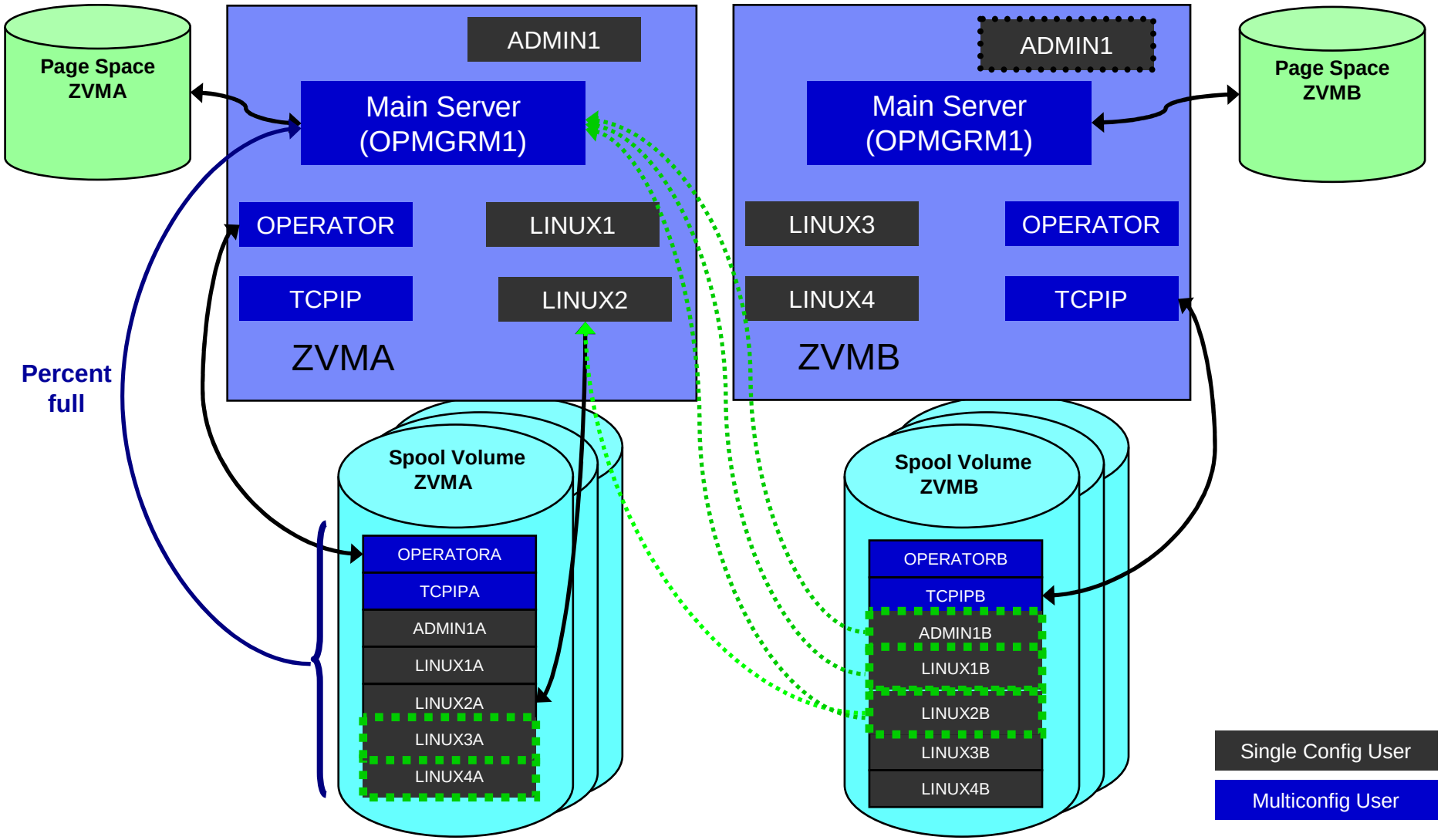
- **Consoles received by Operations Manager via SECUSER or OBSERVER**
  - Prefer SECUSER
    - OBSERVER won't detect CP and VM READ messages
    - Output of actions on OBSERVEd console may not be viewable in console
  - OBSERVER allows Operations Manager to receive console output even if user is logged on
- **Single System Image allows SECUSER and OBSERVER across members of cluster**
  - Content does not contain member name information
  - Rules, actions, and users wouldn't be able to distinguish between IDENTITY users on multiple members
  - Creates single point of failure on one member
- **Recommendation for z/VM V6.2 or V6.3 Single System Image environments**
  - Have all consoles monitored by an Operations Manager server on the same member as the monitored guest (i.e. all Operations Manager servers are IDENTITY users)
    - Requires action processing servers (OPMGRSn) to be on same member as main server
  - Share configuration data on minidisk owned by single configuration user
    - For example: VMTOOLS 198
    - Master configuration file unique to each member
    - Imbed common file(s) used by all members
  - Request a copy of the current console of a remote user
    - `SMSG OPMGRM1 at membername VIEWCON USER(userid),MODE(RDR)`

IBM Software

SSI Considerations

Page Space Monitoring
Spool Space Monitoring
Viewing and Managing Spool Files

SSI Considerations for Spool and Page Space Monitoring



Spool and Page Space Monitoring - Considerations

- **Page space is local**
  - Separate space for each member and only visible to the local member
  - No impact from SSI
- **Spool data**
  - Spool files are placed on spool volumes owned by the member where the spool file was created
  - Users see their own spool data no matter where they are logged on and where the data was created

Spool and Page Space Monitoring - Considerations

- **Users and applications (like Operations Manager) who can see all spool files need to be aware:**
  - Spool data for multiconfiguration users
    - Only spool files owned by the local instance of that user are visible on the local member
    - No visibility to spool files owned by other instances of that user on other members
  - Spool data for single configuration users:

Single configuration user Status	Spool files created on **this** member	Spool files created on **other** members
User logged off	Visible	Not visible
User logged onto **this** member	Visible	Visible (but not on local spool volumes)
User logged onto **another** member	Visible	Not visible

Spool and Page Space Monitoring - Considerations

- **Recommendation**
  - Have an Operations Manager server on each member to monitor spool and page space
  - Be aware of spool files visible in Operations Manager but not resident on this member's spool volumes
    - Indicated with "+" in VIEWSPL

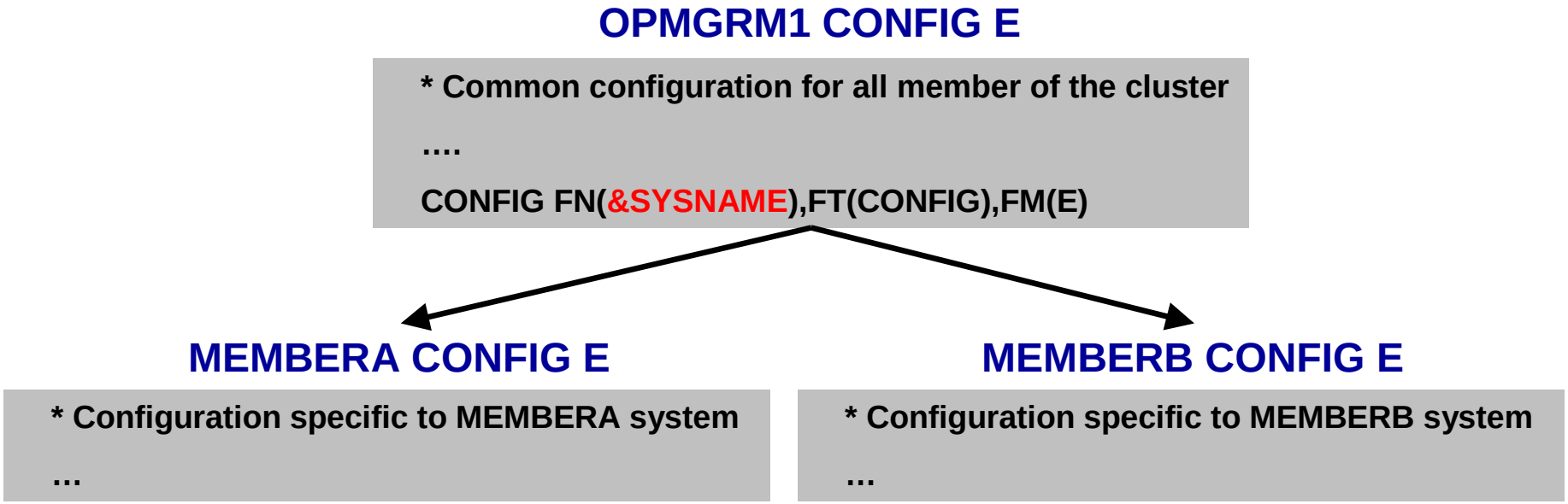
IBM Software

SSI Considerations

Managing Configuration Files

Managing Configuration Files

- **Put all configuration files on a shared disk**
  - Minidisk owned by a single configuration user (not an Operations Manager service machine)
  - SFS
- **Create a common configuration file used by all members**
  - All Operations Manager servers on all members load this file
- **Imbed a unique configuration file based on the system name of this member**
- **Request configuration file reload from user IDs on other members of a cluster**
  - Use `SMSG OPMGR1 at <member> CONFIG …`



Summary

References

Demos – Including Screenshots, Configuration Statements, and REXX

Recommended Practices – Operational Management

View and issue commands on consoles of Linux guests and CMS service machines

- Operations staff monitoring multiple consoles or a central console of alerts
- System programmers debugging a problem on a guest or service machine

VIEWCON

Generate alerts and/or automatically recover from

- Abend, termination, or error messages
- Service machine disks approaching full

Rules

- Critical user IDs or guests being logged off or entering error state
- Spool and/or page space approaching full

Spool/Page monitors

Schedules

Event monitors

Schedule automated system maintenance procedures

- Spool cleanup based on policies

SFPURGER

- Minidisk cleanup (from logs), including archiving
- Orderly startup and shutdown

Rules, monitors

  - Relocation of critical guests to another SSI member
- Backups of z/VM system

Backup Manager

Schedules

Rules, Archive Mgr

Summary

- **Use Operations Manager to**
  - Automate daily operations
  - Integrate your z/VM and Linux on System z environment with existing enterprise monitoring and alerting
  - Prevent problems rather than react to them
  - Automate reactions to problems when they can't be prevented
  - Improve problem determination procedures
  - Increase programmer and operator productivity
  - Continue to monitor locally with improved management of clusters
- **Sometimes several alternatives for monitoring for the same event**
  - Console message (rules)
  - Scheduled healthchecks (schedules)
  - User ID status changes (event monitor)
- **Actions allow integration with other platforms and products**

Reference Information

- **Product Web site**
  - Start at http://www.ibm.com/software/stormgmt/zvm/
  - Product pages include
    - Publications
    - Pre-requisites
    - Presentations
    - White papers
    - Support
- **e-mail**
  - Mike Sine, sine@us.ibm.com, Advanced Technical Skills (ATS)
  - Tracy Dean, tld1@us.ibm.com, Product Manager
- **White papers on Operations Manager website (Resources tab)**
  - Routing Linux syslog data
  - Sending alerts from Operations Manager to Netcool/OMNIbus
  - Using Shared File System to store Operations Manager configuration files and automation EXECs
  - Automatically logging on a user at Linux system boot time for easier console management and action execution

IBM Software

Demonstration Scenarios

Automation Demos Available

1. **Send an e-mail based on a console message**
2. **Send an alert to Netcool/OMNIbus based on a console message, hold and unhold messages**
   a. Using POSTZMSG interface to Netcool/OMNIbus
   b. Using SNMP interface to Netcool/OMNIbus
3. **Send a message or email if spool approaches full**
   a. Send a message if spool usage is too high on any member of an SSI Cluster – see how spool files appear in SSI
   b. Send an email if spool usage is too high on a single system
4. **View and clean up spool files**
5. **Automated spool cleanup**
6. **Archiving DIRMAINT's log files when disk gets full**
7. **Process a file of test messages as a console**
8. **Process Linux syslog data as a console**
9. **Create a central operations console on one z/VM system**
10. **Create a central operations console across multiple z/VM systems**
    a. When the systems are in an SSI cluster
    b. When the systems are not in an SSI cluster
11. **Integration with OMEGAMON XE on z/VM and Linux - take action based on CPU usage of Linux guest**
12. **Monitor service machines for logoff – and autolog them**
13. **Send an email if page space approaches full**
14. **Monitor SSI connectivity between 2 cluster members**
15. **Suppress passwords on Linux consoles**
16. **Autolog a Linux Guest and Send Message if Doesn't Start Successfully**
17. **View consoles of Linux guests and CMS user IDs or service machines specifying a date/time range**

Scenario 1:
Send an Email if Abend or Fatal Message Occurs

- **Watch all monitored consoles for an error message that includes the word “fatal” or “abend”**
  - Message must also contain the word “mail” (for demo purposes only)
- **Send an email if one of the words appears on a console**
- **Dynamically include in the email**
  - Host name of z/VM system where the error occurred
  - User ID that received the error message
  - Indicator of whether the word was fatal or abend
  - Full text of the error message

Scenario 1: Detailed Steps

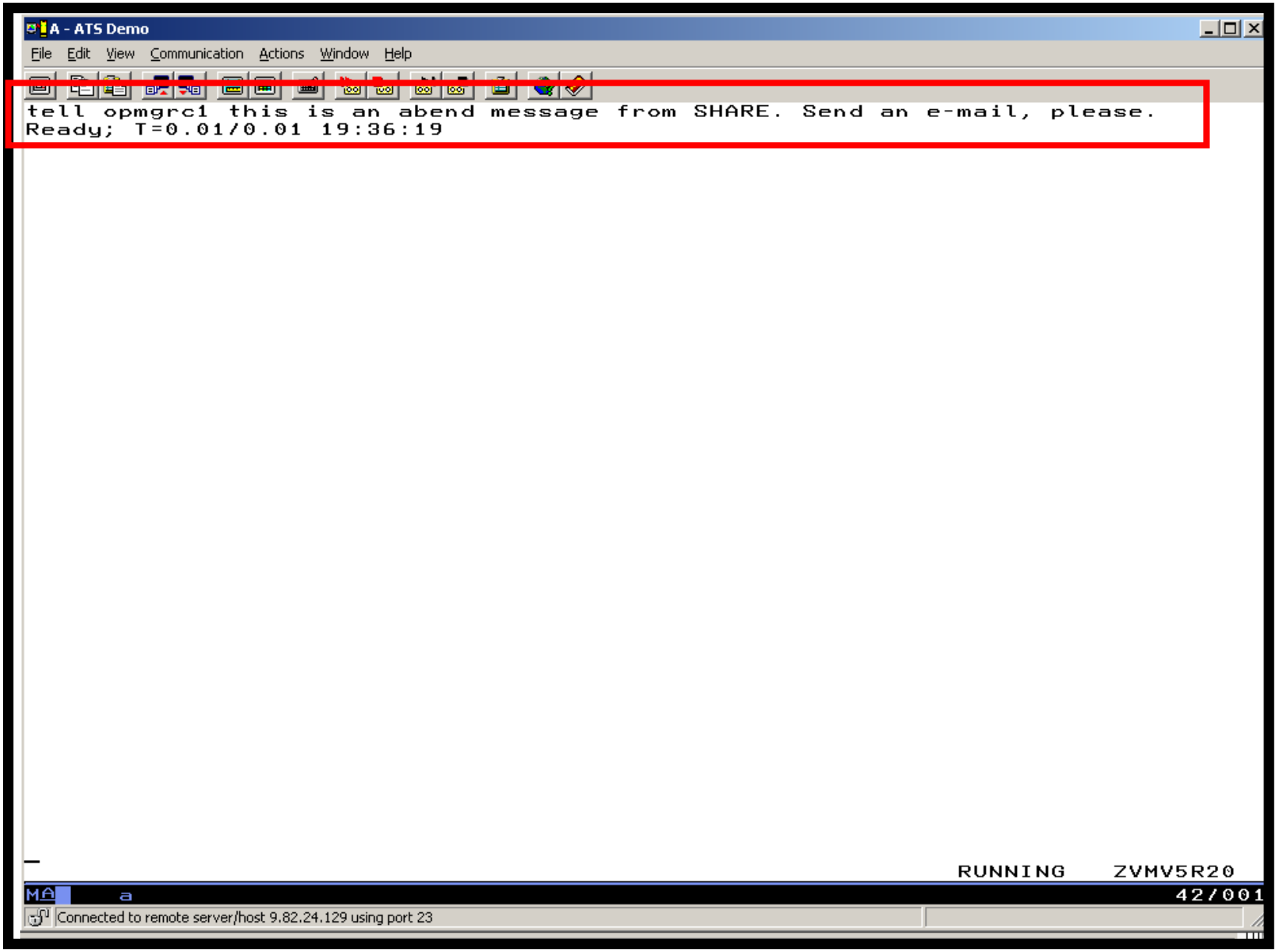
- **From any VM user ID:**

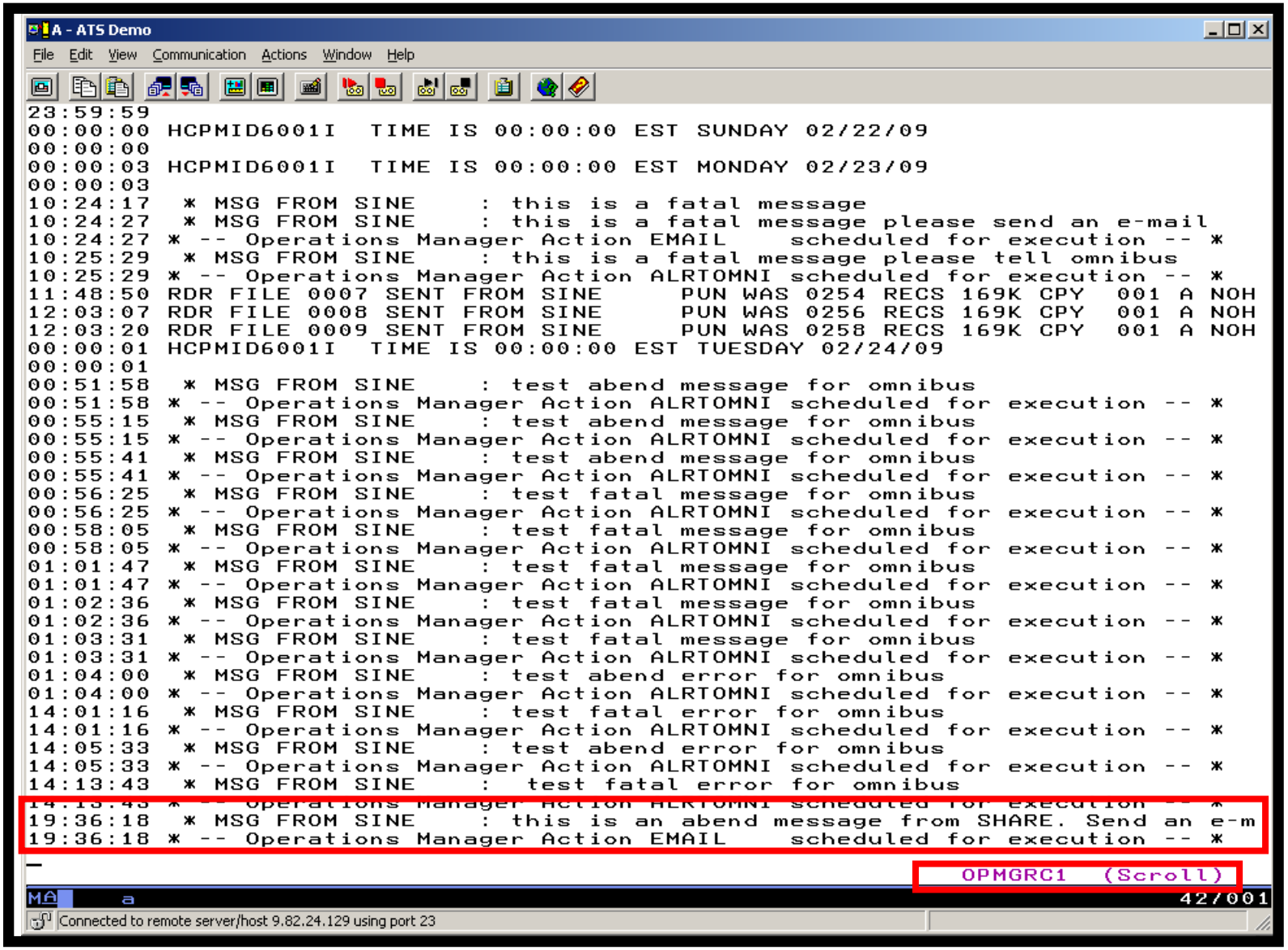
```
tell opmgrc1 this is an abend message from SHARE. Send an e-mail, please.
```

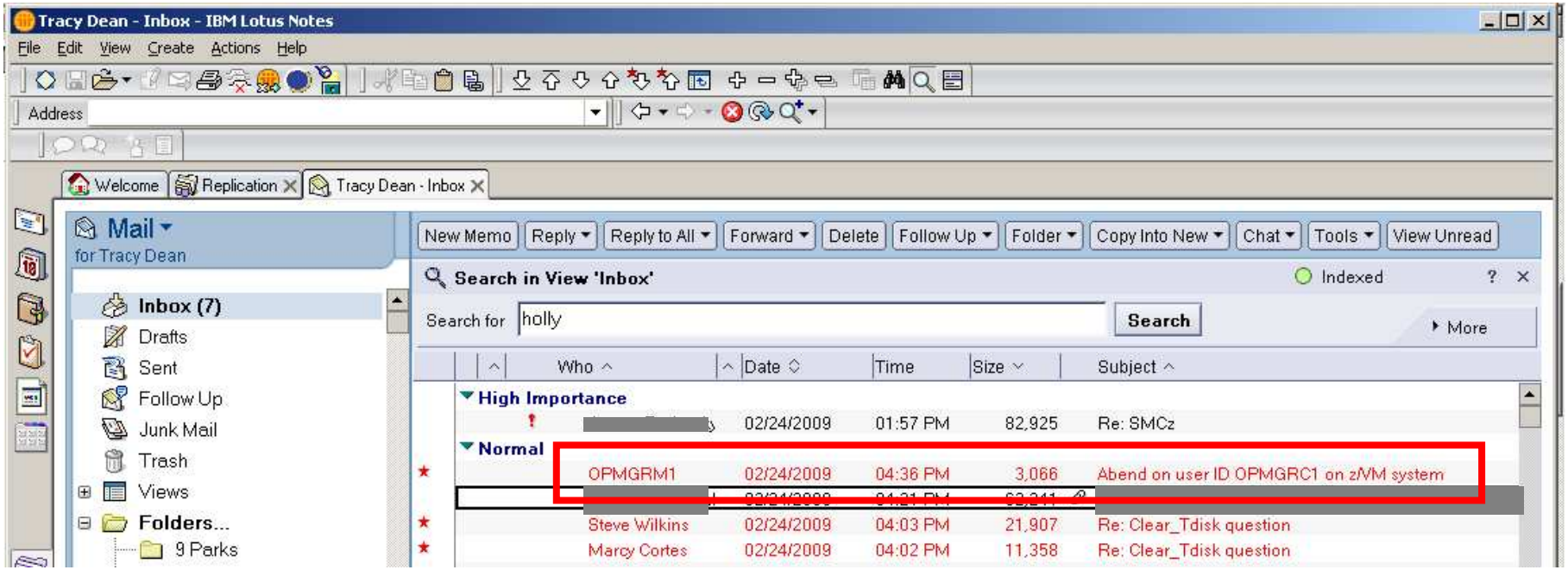
- **From an authorized VM user ID, view the console of OPMGRC1:**

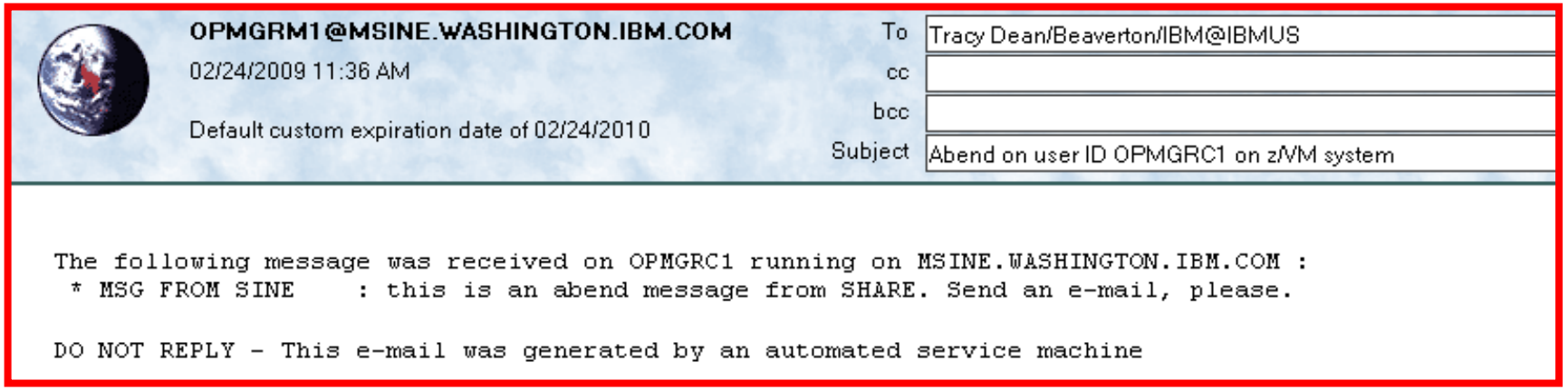
```
gomcmd opmgrm1 viewcon user(opmgrc1)
```

- **Check the inbox of the appropriate person to see the e-mail**









Scenario 1: How Do You Do That?

Rules in Operations Manager:

```
*
* Send an e-mail to someone if I see a message containing the word
* "fatal" on any monitored console
DEFRULE NAME(FATLMAIL),+
  MATCH(*FATAL*mail*),+
  ACTION(EMAIL),+
  PARM(FATAL)
*
* Send an e-mail to someone if I see a message containing the word
* "abend" on any monitored console
DEFRULE NAME(ABNDMAIL),+
  MATCH(*ABEND*mail*),+
  ACTION(EMAIL),+
  PARM(ABEND)
```

Scenario 1: How Do You Do That?

Action in Operations Manager:

```
*
* Replace "tld1 at us.ibm.com" with the e-mail address of the user that
* should receive the e-mail
* Leave &u, &p, and &t as-is. These represent the user ID that had the
* "fatal" message, the parameter passed (fatal or abend), and the
* text of the message.  These will be included in the text of the
* e-mail.
DEFACTN NAME(EMAIL),+
  COMMAND(EXEC SMTPNOTE tld1 at us.ibm.com &u &p &t),+
  OUTPUT(LOG),+
  ENV(LVM)
```

Scenario 1: How Do You Do That?

SMTPNOTE EXEC (excerpts)

```
/*  */
Parse arg mail_user 'AT' mail_node baduser errtype msgtext
if errtype = 'FATAL' then
  errtext = 'Fatal error on user ID' baduser 'on z/VM system'
else
  if errtype = 'ABEND' then
    errtext = 'Abend on user ID' baduser 'on z/VM system'
  else errtext = msgtext
/* Construct the e-mail */
line.1 = 'OPTIONS: NOACK    LOG    SHORT   NONOTEBOOK ALL CLASS A'
line.2 = 'Date: ' Date() ',' Time()
line.3 = 'From: Operations Manager for z/VM'
line.4 = 'To:  ' mail_user 'at' mail_node
line.5 = 'Subject: ' errtext
line.6 = 'The following message was received on' baduser 'running on'
line.7 = msgtext
line.8 = ' '
line.9 = 'DO NOT REPLY - This e-mail was generated by an automated service machine
line.0 = 9
'PIPE stem line. | > TEMP NOTE A'
'EXEC SENDFILE TEMP NOTE A (NOTE SMTP'
```

Scenario 2a: Send an Alert to OMNIbus – Using POSTZMSG

- **Watch all monitored consoles for an error message that includes the word “fatal” or “abend”**
  - Message must also contain the word “omni” (for demo purposes only)
- **Send an alert to OMNIbus if one of the words appears on a console**
  - Use POSTZMSG, running on Linux guest
    - Do not trigger the action if the message is on this guest
- **Dynamically include in the alert**
  - User ID that received the error message
  - Indicator of whether the word was fatal or abend

Scenario 2a: Detailed Steps

- **View “All Events” in OMNIbus**
- **From any VM user ID:**

```
tell opmgrc1 this user is abending at SHARE. Tell OMNIBUS.
```

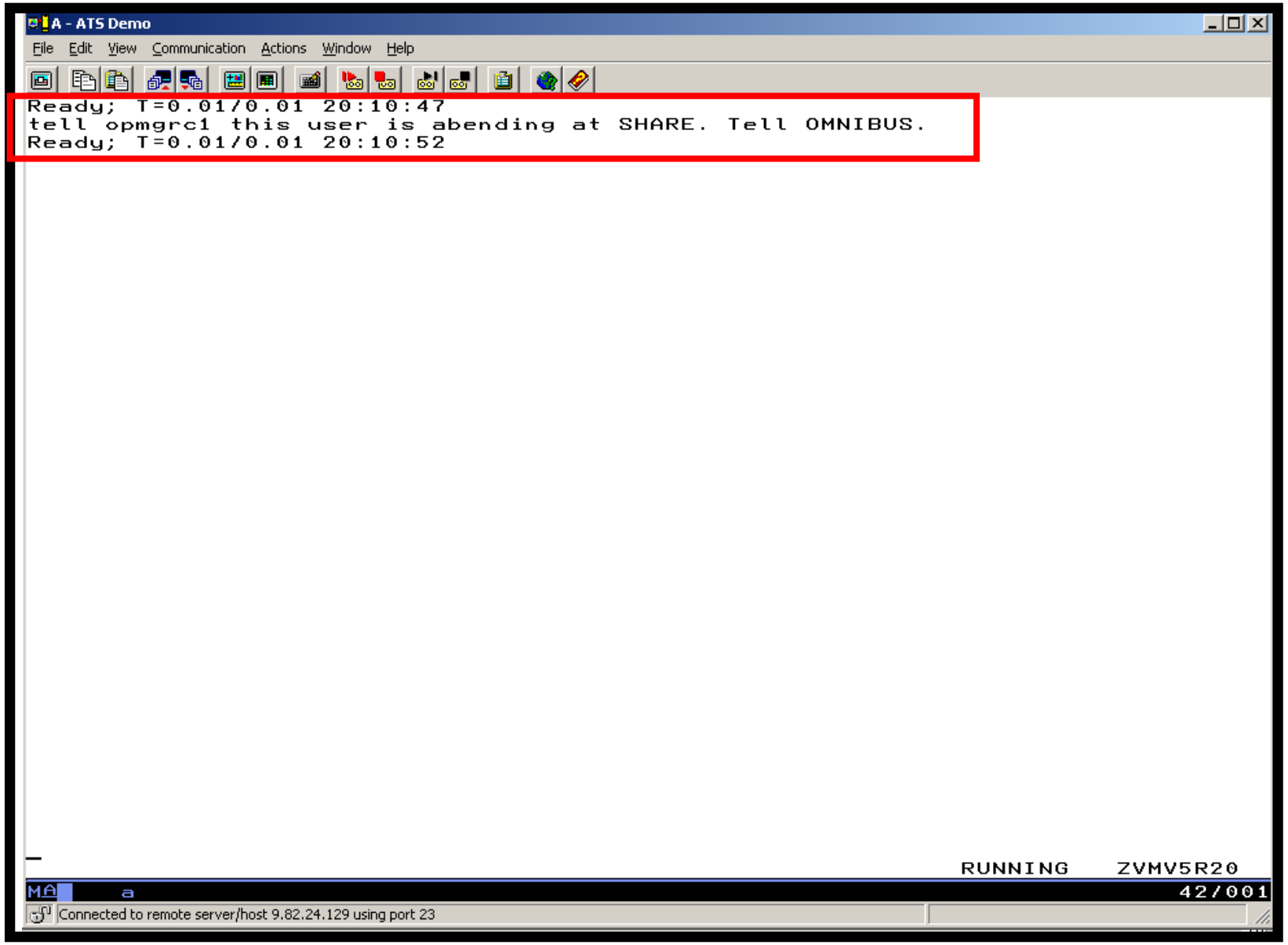
- **From an authorized VM user ID, view the console of OPMGRC1:**

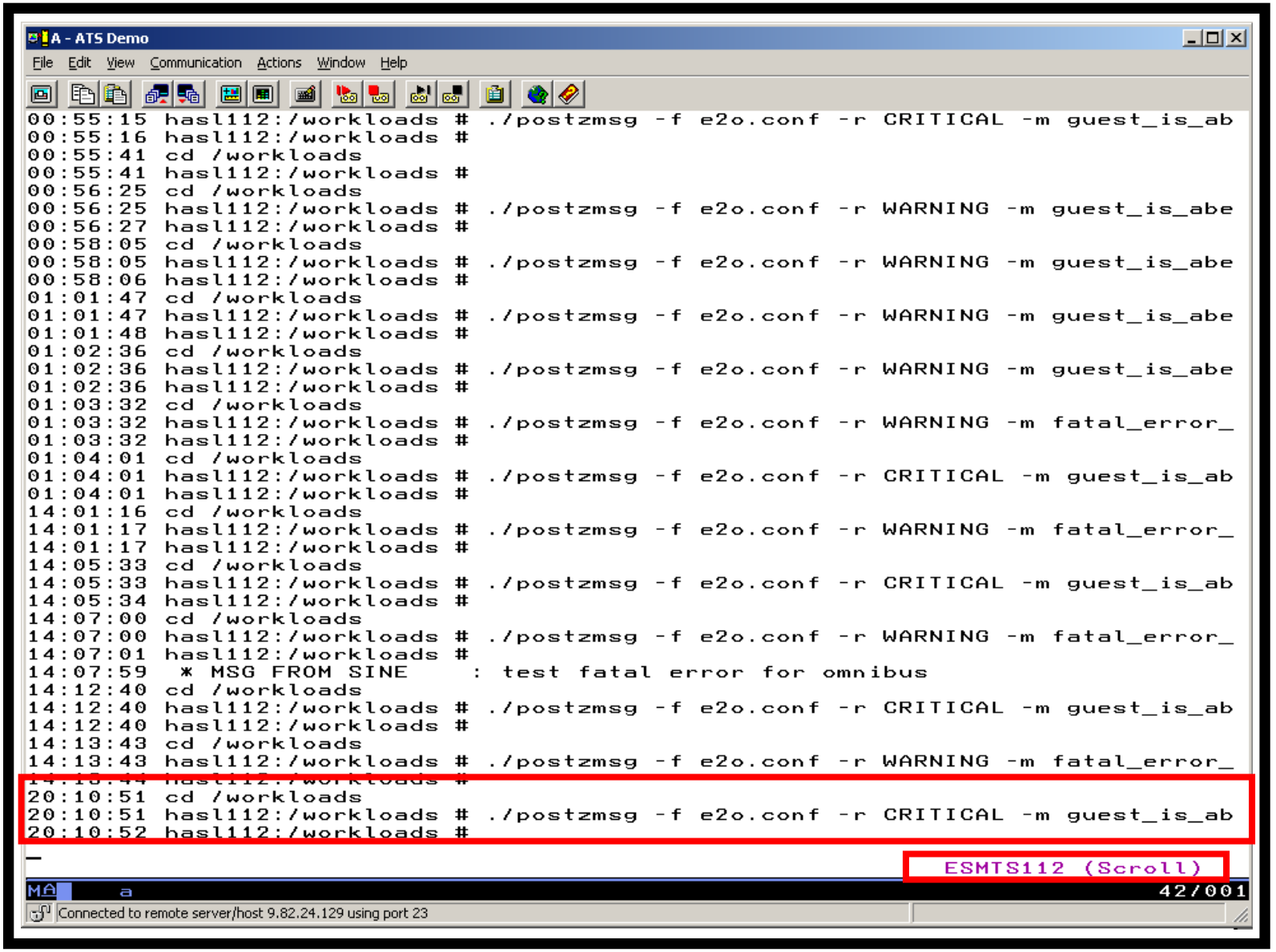
```
gomcmd opmgrm1 viewcon user(opmgrc1)
```

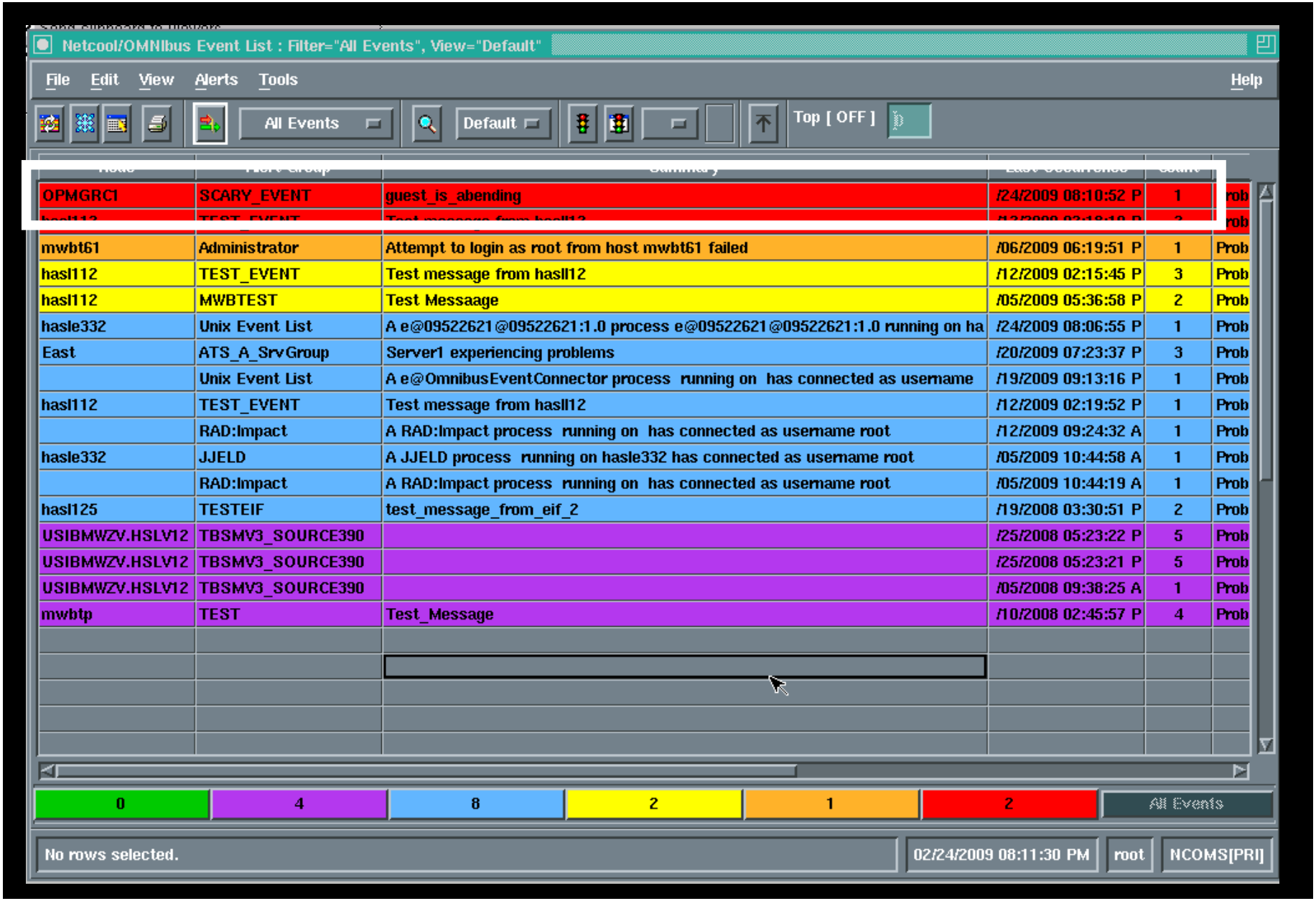
- **From an authorized VM user ID, view the console of the Linux guest that runs POSTZMSG:**

```
gomcmd opmgrm1 viewcon user(esmts112)
```

- **View the OMNIbus console to see the alert**







Scenario 2a: How Do You Do That?

Rules in Operations Manager:

```
*
* Send an alert to OMNIBUS for fatal errors on consoles
DEFRULE NAME(FATLOMNI),+
  MATCH(*fatal*omni*),+
  EXUSER(ESMTS112),+
  ACTION(ALRTOMNI),+
  PARM(FATAL)
*
* Send an alert to OMNIBUS for abends on consoles
DEFRULE NAME(ABNDOMNI),+
  MATCH(*abend*omni*),+
  EXUSER(ESMTS112),+
  ACTION(ALRTOMNI),+
  PARM(ABEND)
```

Scenario 2a: How Did You Do That?

Action in Operations Manager:

```
*
* Call POSTZMSG on a Linux guest to send alert to OMNIBUS
DEFACTN NAME(ALRTOMNI),+
  COMMAND(EXEC POSTZMSG &u &p),+
  OUTPUT(LOG),+
  ENV(LVM)
```

Scenario 2a: How Did You Do That?

POSTZMSG EXEC (excerpts)

```
/*  */
Parse arg baduser errtype
if errtype = 'ABEND' then
  do
    zerrtype = 'CRITICAL'
    cmdpart2 = '-m guest_is_abending hostname='baduser
    cmdpart4 = 'sub_origin=tcp SCARY_EVENT OpsMgr'
  end
else
  do
    zerrtype = 'WARNING'
    cmdpart2 = '-m fatal_error_on_guest hostname='baduser
    cmdpart4 = 'sub_origin=tcp WARN_EVENT OpsMgr'
  end
cmdpart1 = './postzmsg -f e2o.conf -r' zerrtype
cmdpart3 = 'sub_source=postzmsg origin='baduser
'CP SEND ESMTS112 cd /workloads'
'CP SEND ESMTS112' cmdpart1 cmdpart2 cmdpart3 cmdpart4
```

Scenario 2b: Send an Alert to OMNIbus – Using SNMP

- **Watch all monitored consoles for an error message that includes the word “abend”**
  - Message must also contain the word “snmp” (for demo purposes only)
- **If this word appears on a console**
  - Change the message to red and hold it
  - Send an alert to OMNIbus, using SNMPTRAP command on z/VM
  - Automatically unhold the message after 4 minutes
- **Dynamically include in the alert**
  - IP address of the z/VM system where the error occurred
  - User ID that received the error message
  - Text of the abend message

Scenario 2b: Detailed Steps

- **View “All Events” in OMNIbus**
- **From any VM user ID:**

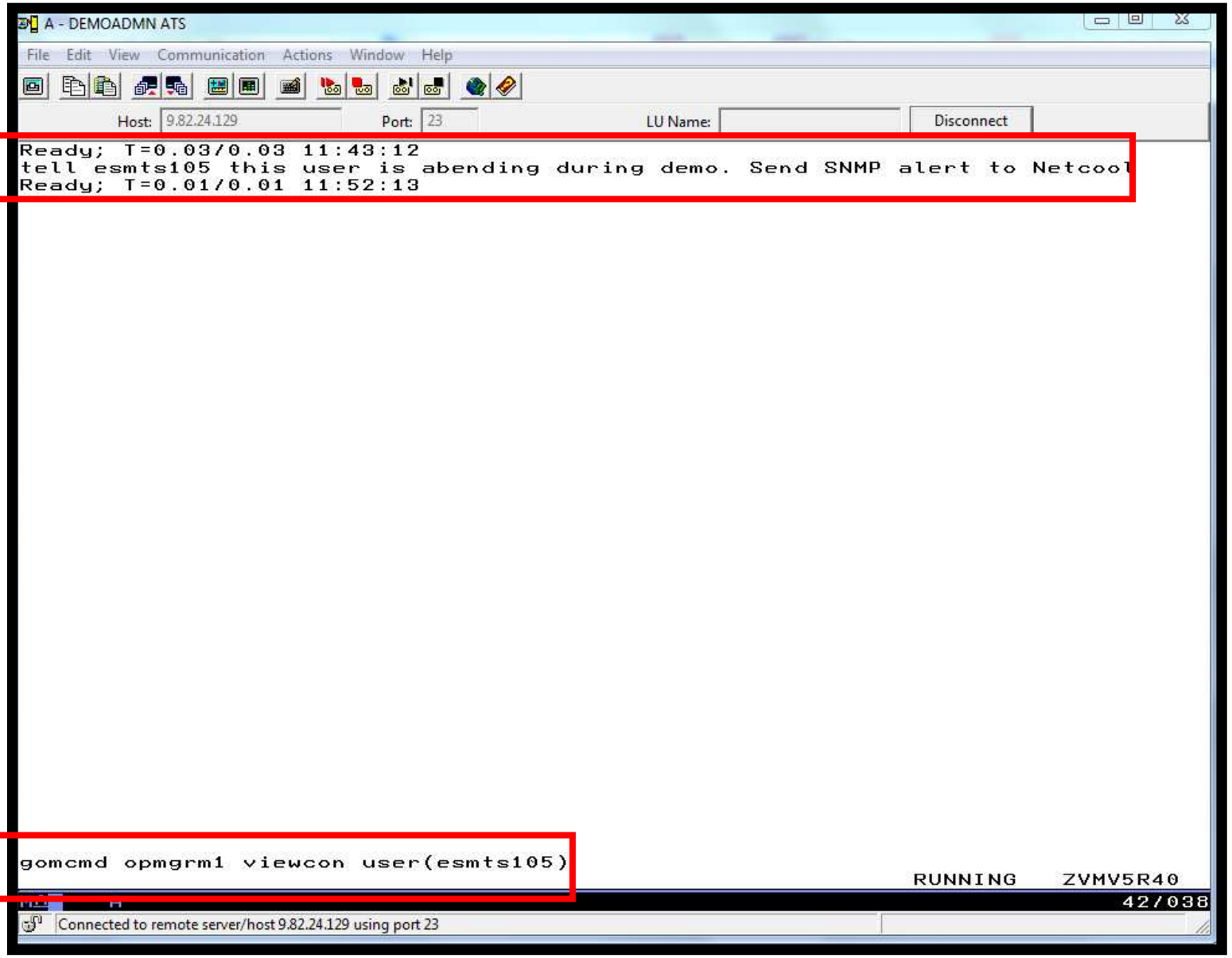
```
tell esmts105 this user is abending during demo. Send SNMP alert to Netcool
```

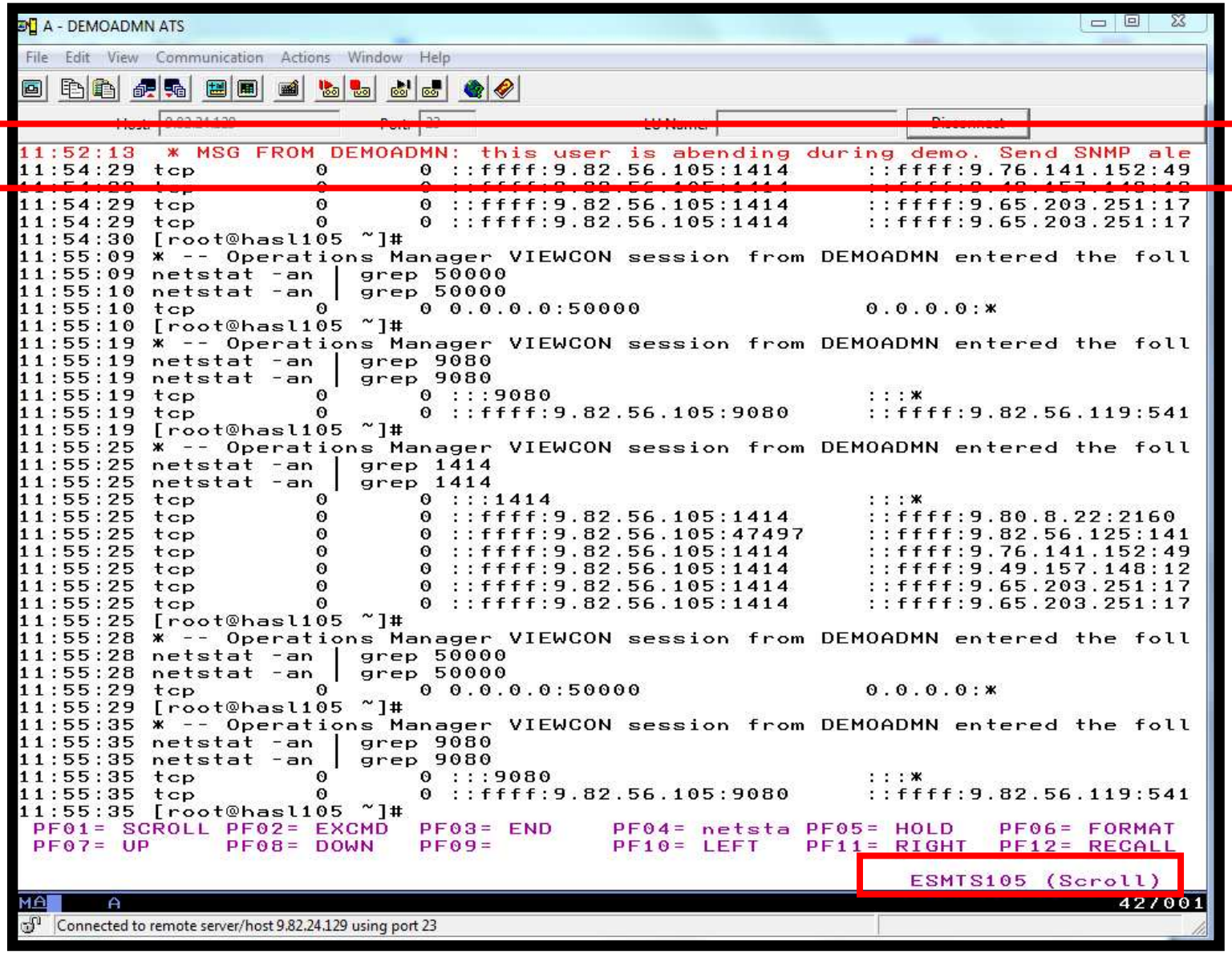
- **From an authorized VM user ID, view the console of ESMTS105 (a Linux guest):**

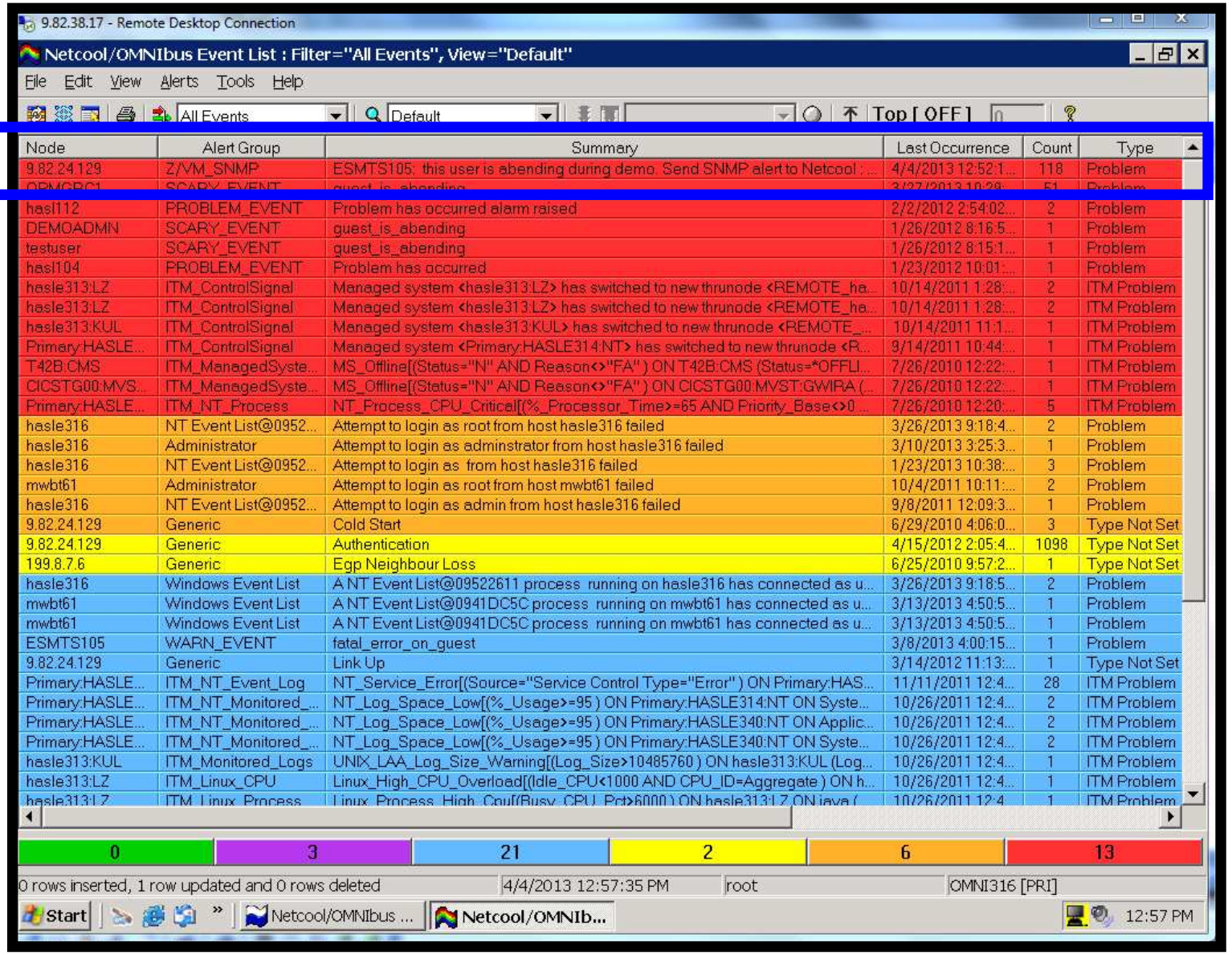
```
gomcmd opmgrm1 viewcon user(esmts105)
```

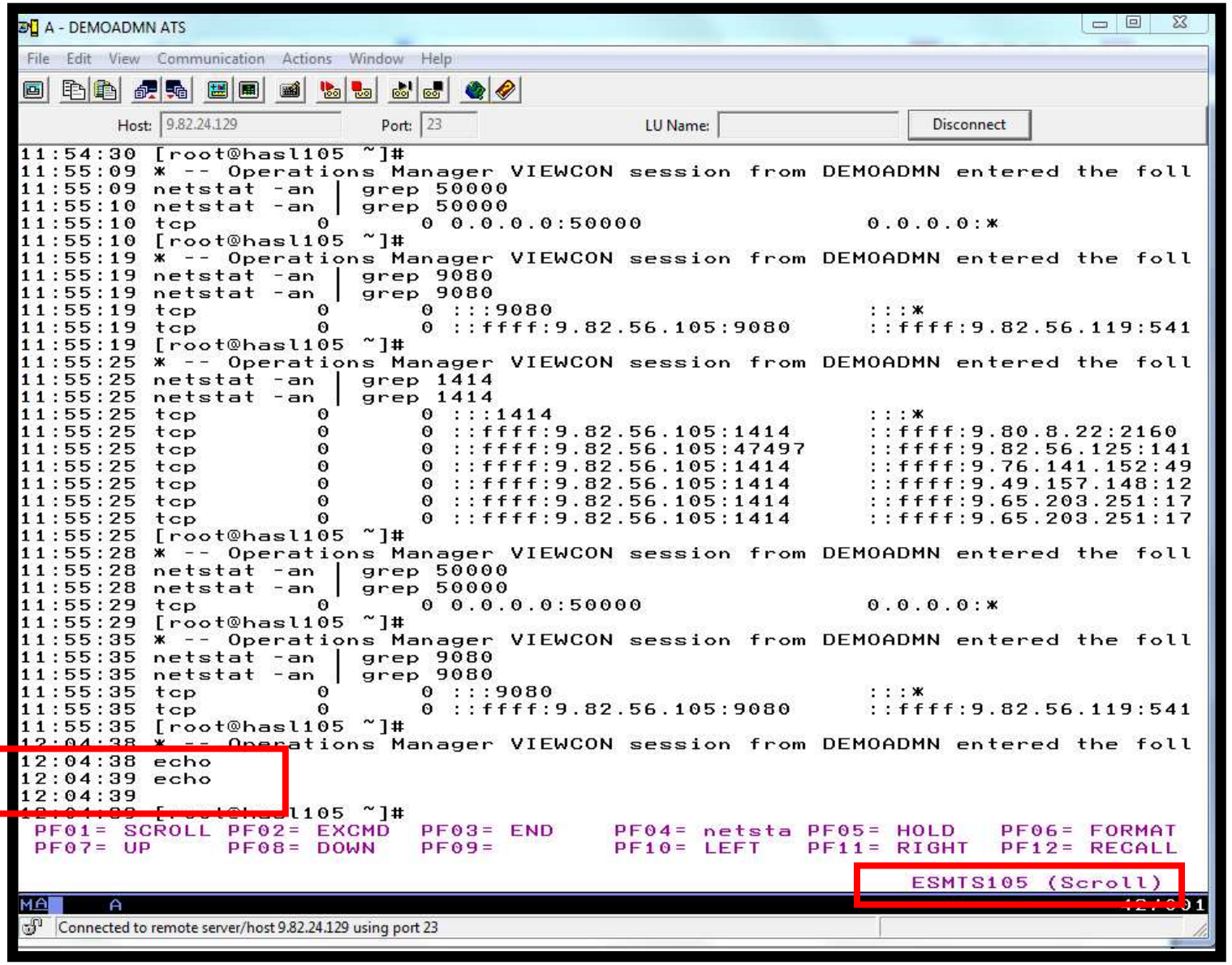
- **Issue some Linux commands so the held message moves to the top of the screen**
- **View the OMNIbus console to see the alert**
- **After 4 minutes, view the console of ESMTS105 again and notice the held message has moved off the screen**

```
gomcmd opmgrm1 viewcon user(esmts105)
```









Scenario 2b: How Do You Do That?

Rule and actions in Operations Manager:

```
*
* Send an alert to OMNIbus using SNMP for abend msgs on consoles
DEFRULE NAME(ABNDSNMP),+
  MATCH(*abend*snmp*),+
  ACTION(SNMPALRT)
*
DEFACTN NAME(SNMPALRT),+
  COMMAND(EXEC SNMP2OMN &T),+
  INPUT(CRE,HLD),+
  ENV(SVM),+
  NEXTACTN(UNHOLD),+
  NEXTDELY(03:30)
*
DEFACTN NAME(UNHOLD),+
  COMMAND('ALTRCON USER(ESMTS105),MATCH(*abend*snmp*),ELAPSED(180),HLD(N)'),+
  ENV(GOM)
```

Scenario 2b: How Did You Do That?

SNMP2OMN EXEC

```
/* SNMP2OMN action routine for Operations Mgr */
address command
parse arg ":" msgtext
msgtext2 = '"'msgtext '"'
/* Send message  */
snmptrap trape 1.1 number 30 1.2 text "UXZVM001" 1.3 text msgtext2 ent 1.3.6.1.4.1.9545.6
exit
```

Scenario 2b: Additional Steps Required on z/VM

- **SNMPD user ID configured and running**
- **Update files on TCPMAINT 198 disk**
  - Add OMNIbus IP address to SNMPTRAP DEST file
  - Open SNMPD and SNMPQE ports in PROFILE TCPIP
  - Update SNMPMIBX TEXT section of MIB_EXIT DATA
- **Give OPMGRM1 and OPMGRSn access to SNMPTRAP command**
  - On TCPMAINT 592 disk

Scenario 2b: Additional Steps Required on OMNIbus

- **Install the IBM Tivoli Netcool/OMNIbus SNMP Probe**
  - Install it on same platform as target OMNIbus server
- **Customize operational information in the probe properties (mttrapd.props)**
  - Listening port, heartbeat interval, mibs and mibs locations, etc.
- **Customize the probe rules (mttrapd.rules)**
  - Map variables created by the probe (from data extracted from the SNMP trap) into the desired OMNIbus event fields
    - Default mappings for the SNMP generic traps (trap types 0-5)
    - Enterprise-specific traps (trap type 6) require customization
- **Documentation for installation and customization**
  - IBM Tivoli Netcool/OMNIbus SNMP Probe Reference Guide (SC23-6003-04)

Scenarios 2a and 2b – POSTZMSG vs SNMP

- **Using POSTZMSG**
  - Can direct the alert to only the IP address(es) you specify
  - Need a Linux guest running and logged on that can run POSTZMSG and must be on the same z/VM system
    - Can be overcome by using a socket interface to send POSTZMSG command to the guest
  - Limit of 160 characters on POSTZMSG command sent to Linux guest (using CP SEND)
    - Can't always send full text of message
    - Can be overcome by using a socket interface to send POSTZMSG command to the guest
- **Using SNMP**
  - No requirement for a Linux guest. SNMP runs on z/VM.
  - No limit on message size
  - All SNMP alerts on z/VM go the same set of IP addresses

Scenario 3a:
Send a Message if Spool Usage is Too High on Any Member in an SSI Cluster

- **Operations Manager monitors the spool usage (percent full) on each member of a cluster**
  - For demo purposes, spool monitor is currently suspended
  - We'll dynamically resume (re-activate) the spool monitor
    - Must reactivate on each member of a cluster
  - Demo monitor requires spool to only be 5% full
- **Usage exceeds the specified limit**
- **Automatically send a message to a central console for the entire cluster**
  - Send a maximum of 3 messages per hour
- **Message includes the member name and % full**
- **For demo purposes, suspend (de-activate) the spool monitors when complete**
- **Demonstrate which spool files are visible on each member**

Scenario 3a: Detailed Steps

- **From an authorized VM user ID, see the spool usage on local member TEST7SSI:**

```
gomcmd opmgrm1 viewspl
```

- **From a user ID with Operations Manager privileges:**

```
gomcmd opmgrm1 resume spool(splfull)
smsg opmgrm1 at testcssi resume spool(splfull)
```

- **Check the Operations Manager log to see the spool monitor triggered on local member:**

```
gomcmd opmgrm1 viewlog
```

- **View the central console for the cluster to see warning messages from each member:**

```
gomcmd opmgrm1 viewcon user(operssi)
```

- **From a user ID with Operations Manager privileges:**

```
gomcmd opmgrm1 suspend spool(splfull)
smsg opmgrm1 at testcssi suspend spool(splfull)
```

Scenario 3a: Detailed Steps

- **From member TEST7SSI, send a spool file to a single configuration and a multiconfiguration user:**

```
sendfile test7 file a demoadmn op1
```

- **From member TEST7SSI, send a spool file to a multiconfiguration user on another member:**

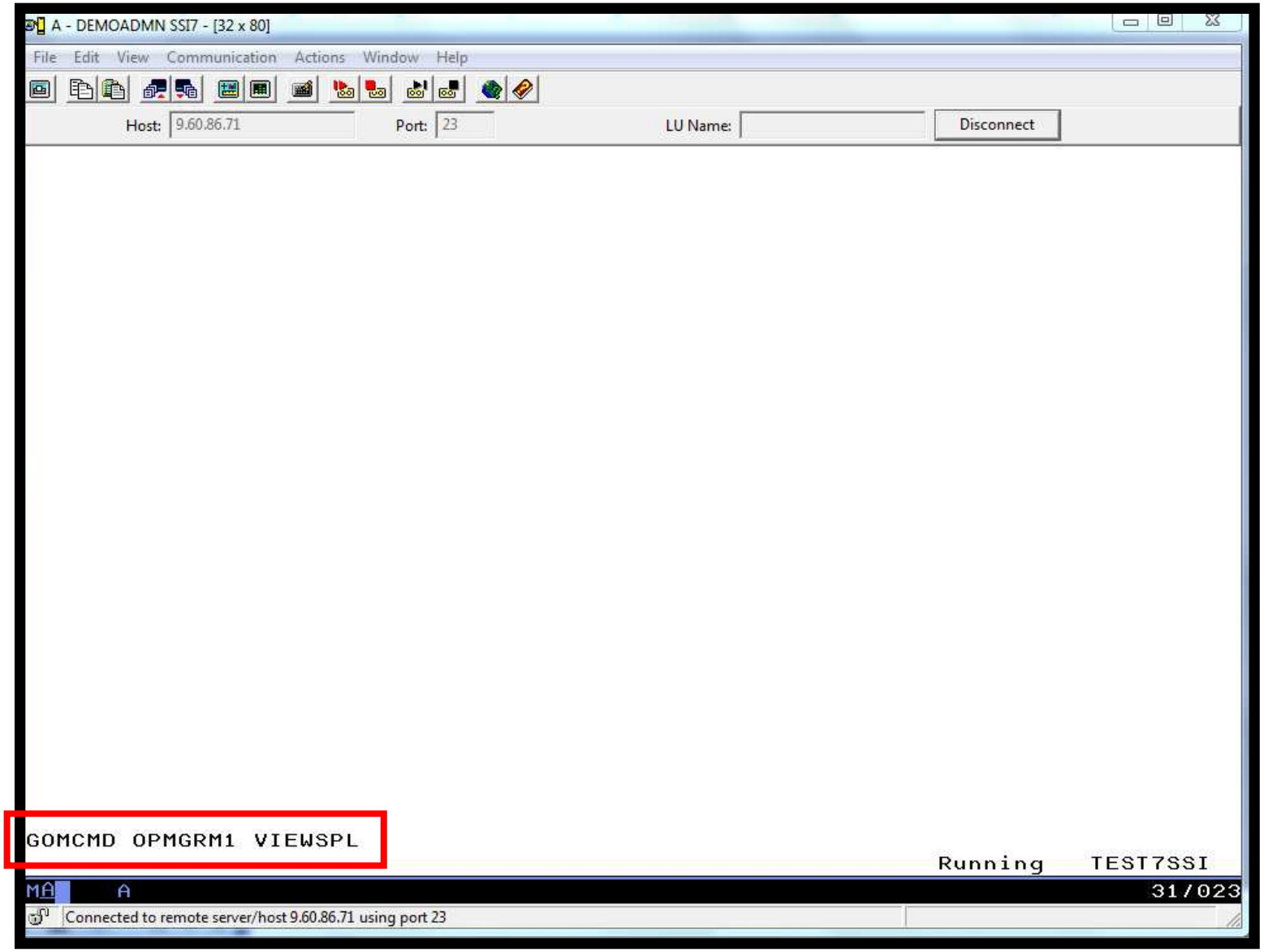
```
sendfile testc file a op1 at testcssi
```

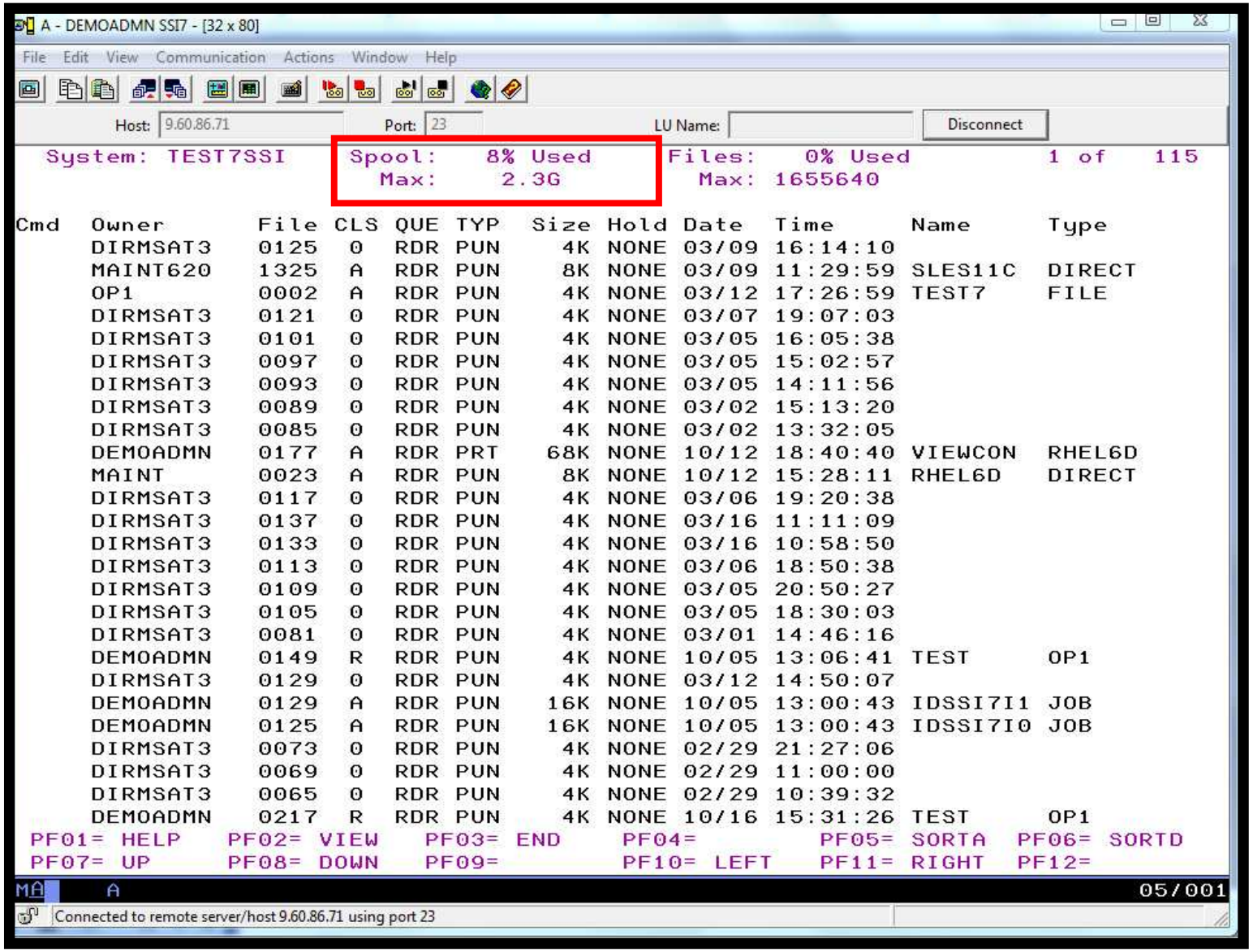
- **From a user ID with Operations Manager privileges on TEST7SSI, view spool files on TEST7SSI:**

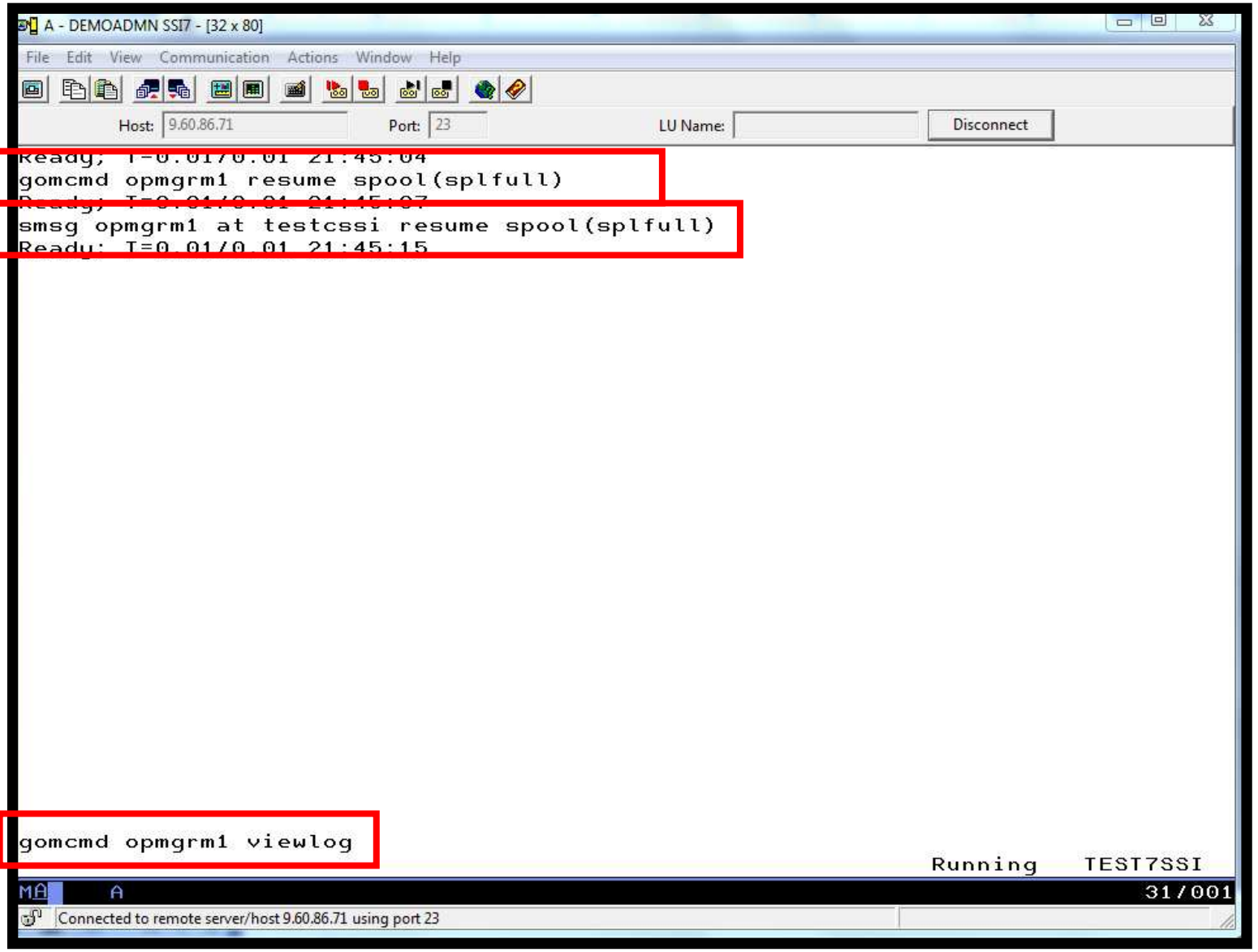
```
gomcmd opmgrm1 viewspl
```

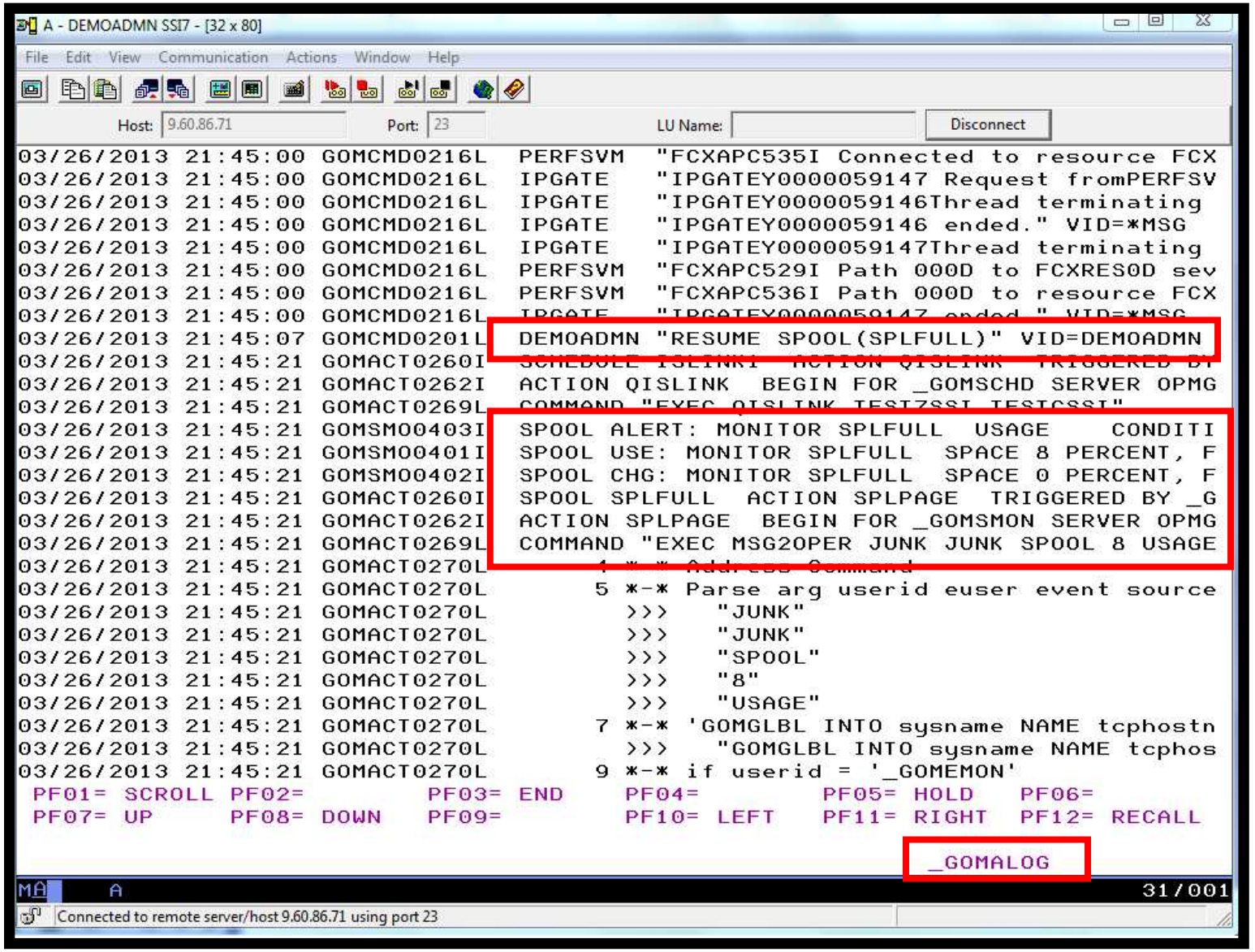
- **From a user ID with Operations Manager privileges on TESTCSSI, view spool files on TESTCSSI:**

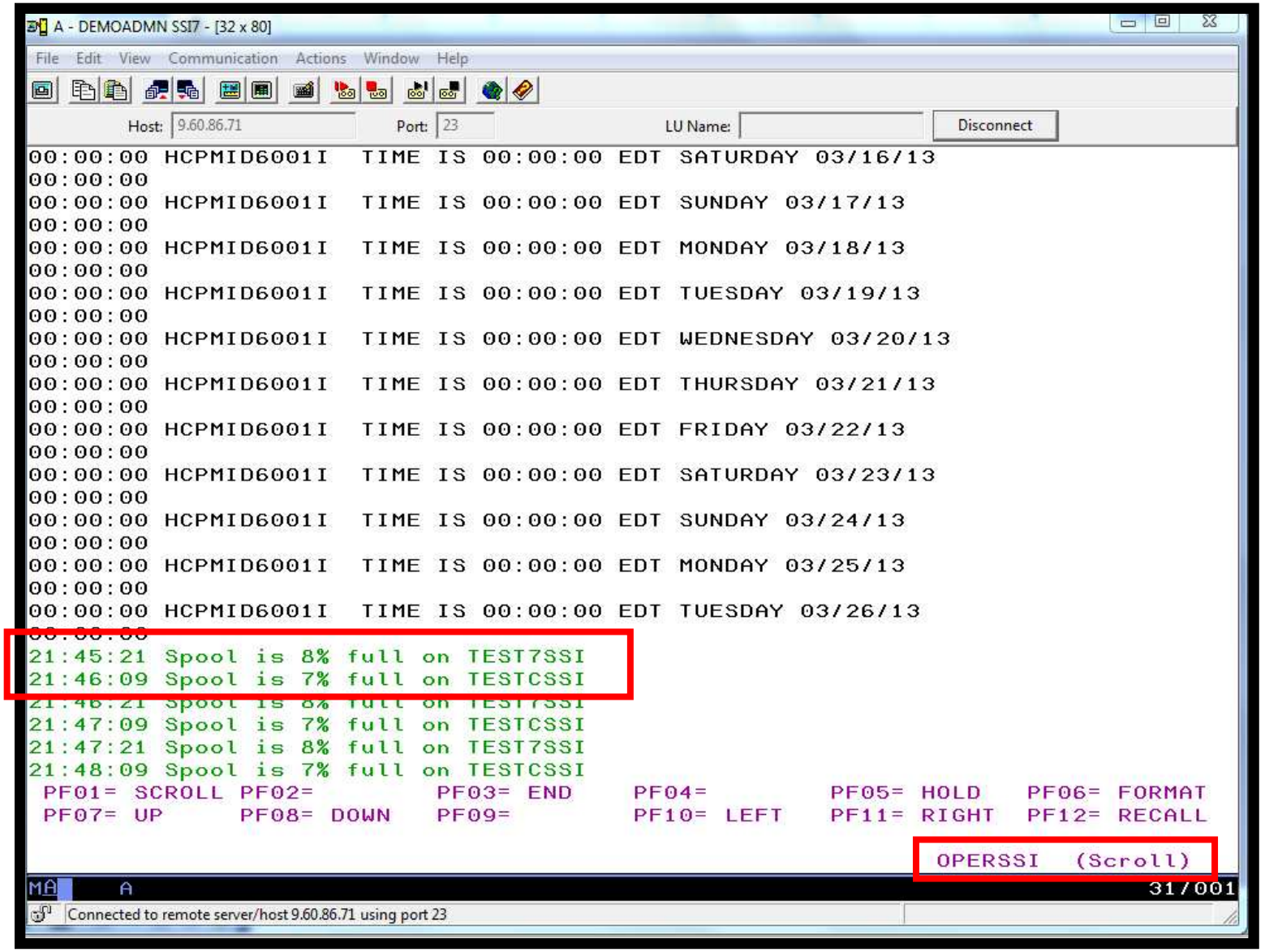
```
gomcmd opmgrm1 viewspl
```

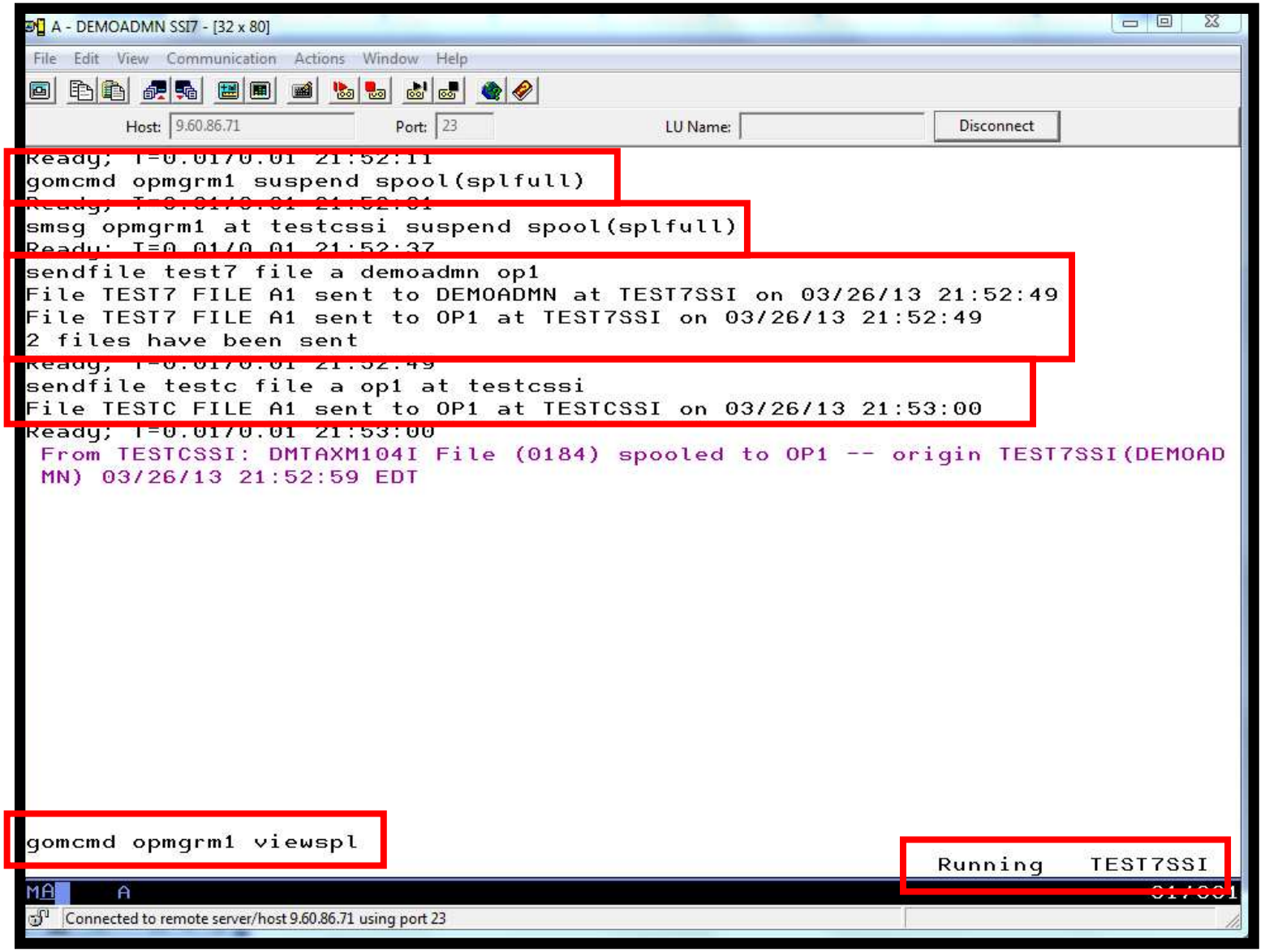


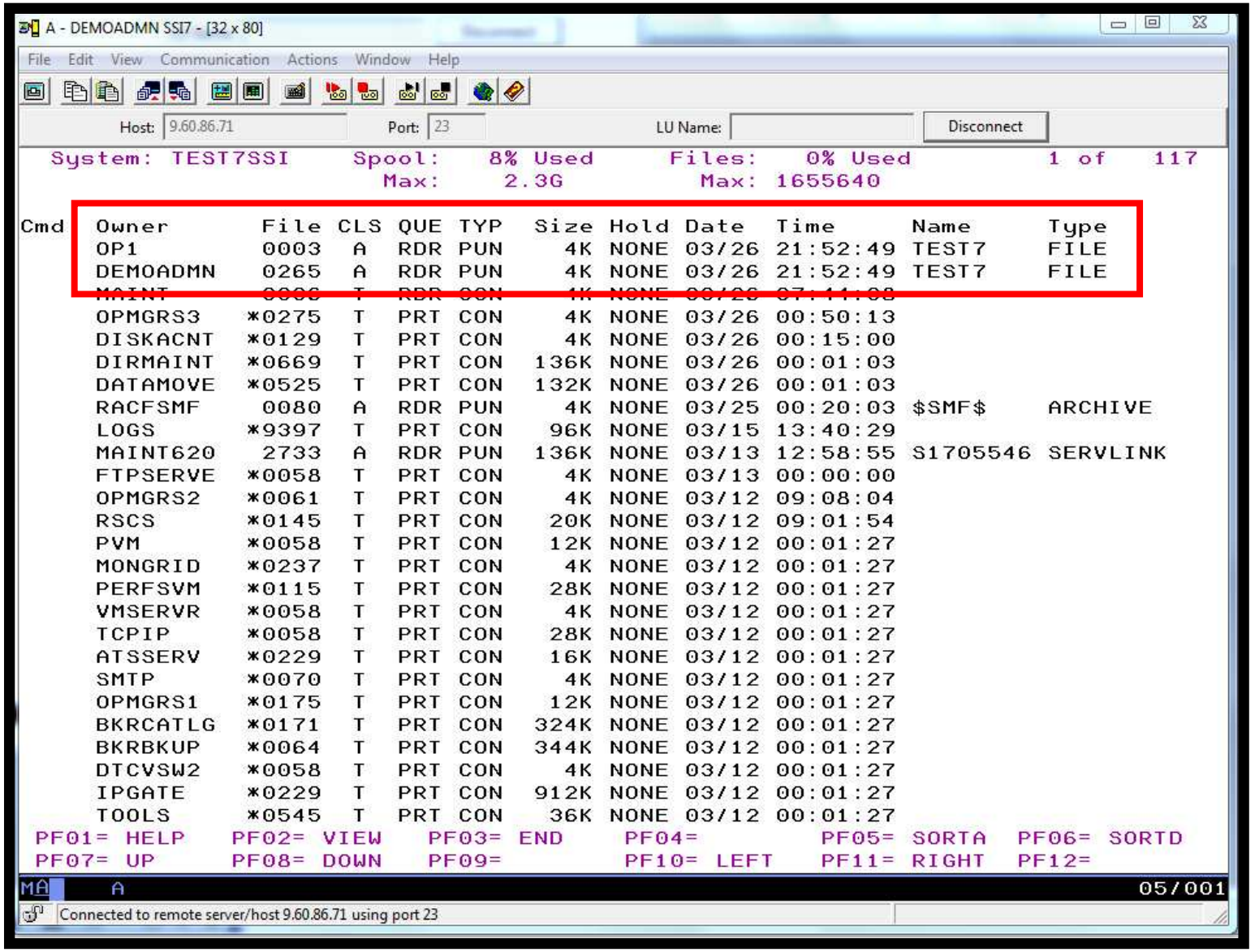


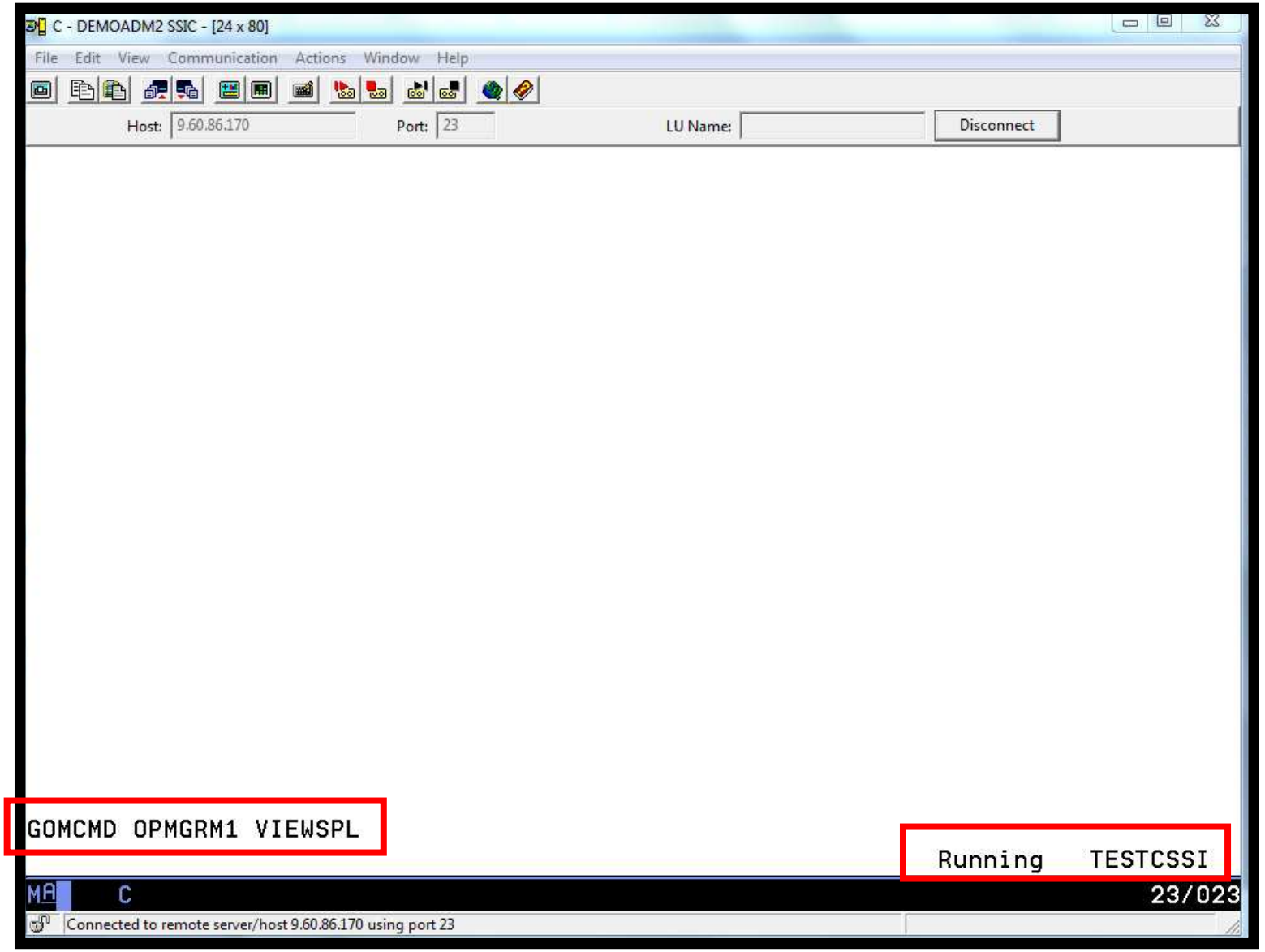


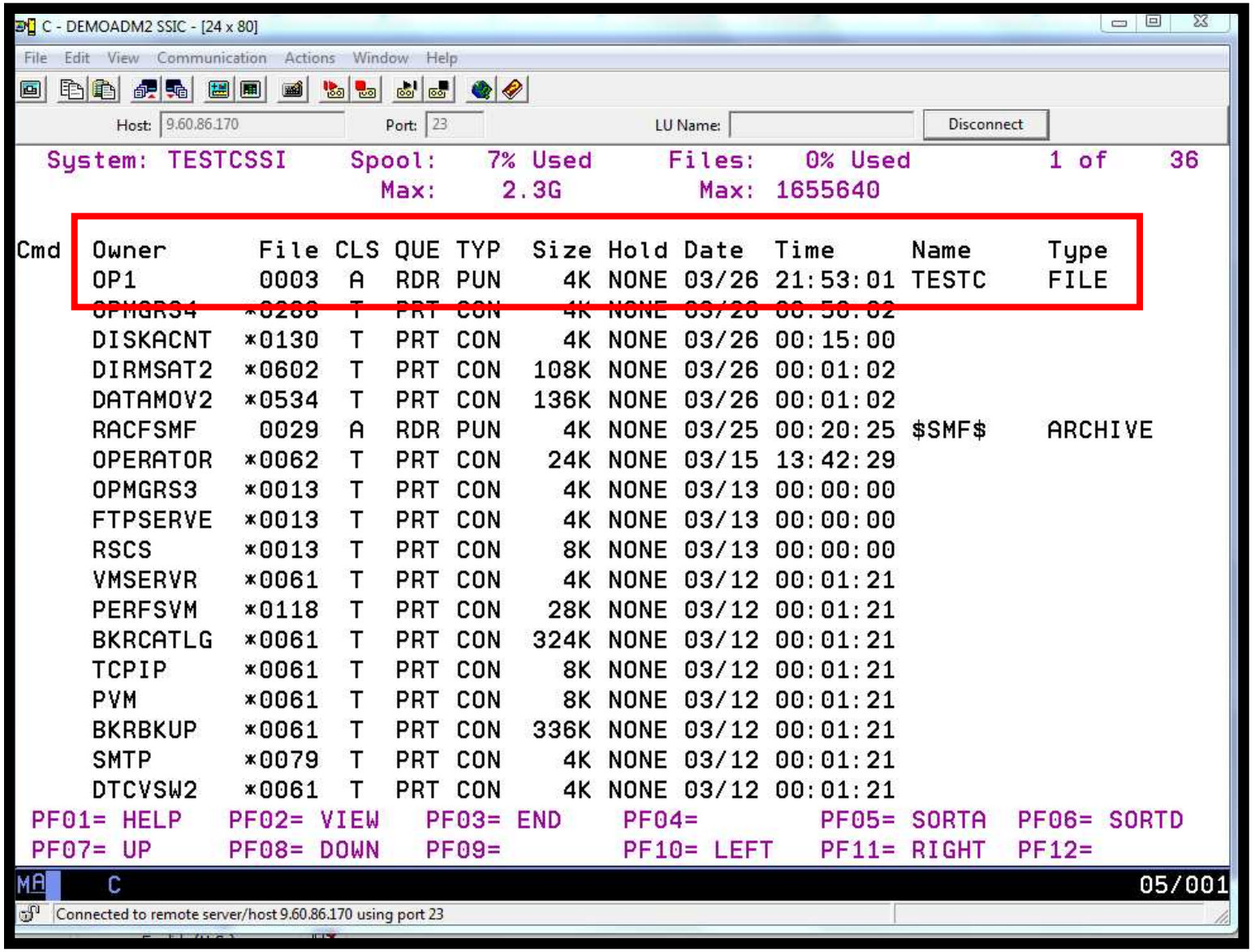












Scenario 3a: How Do You Do That?

Rule and action in Operations Manager:

```
*
* Send an alert to OPERSSI console if spool too full
DEFSMON NAME(SPLFULL),+
  USAGE(005-100),+
  INTERVAL(1),+
  LIMIT(3,3600),+
  ACTION(SPLPAGE)
*
DEFACTN NAME(SPLPAGE),+
  COMMAND(EXEC MSG2OPER junk junk &0 &4 &3),+
  ENV(LVM)
*
SUSPEND SPOOL(SPLFULL)
```

Scenario 3a: How Do You Do That?

MSG2OPER EXEC

```
Address Command
Parse arg userid euser event sourcesys msgtext

'GOMGLBL INTO sysname NAME tcphostname'

if userid = '_GOMEMON' then
  do
    if event = 9 then
      msgtext = 'Outbound relocation for' euser 'on' sourcesys 'started'
    else
      msgtext = 'Inbound relocation for' euser 'on' sourcesys 'started'
    'CP MSGNOH OPERSSI AT ALL From' sysname ':' msgtext
  end
else
  if event = 'SPOOL' then
    'CP MSGNOH OPERSSI AT ALL Spool is' sourcesys'% full on' sysname
  else
  if event = 'PAGE' then
    'CP MSGNOH OPERSSI AT ALL Page space is' sourcesys'% full on' sysname
  else
    'CP MSGNOH OPERSSI AT ALL From' userid 'on' sysname ':' msgtext

Exit rc
```

Scenario 3b: Send an Email if Spool Usage is Too High

- **Operations Manager monitors the spool usage (percent full)**
  - For demo purposes, spool monitor is currently suspended
  - We'll dynamically resume (re-activate) the spool monitor
  - Demo monitor requires spool to only be 5% full or higher
- **Usage exceeds the specified limit**
- **Automatically send an e-mail to someone who can evaluate and take action**
- **For demo purposes, suspend (de-activate) the spool monitor when complete**

Scenario 3b: Detailed Steps

- **From an authorized VM user ID, see the spool usage:**

```
gomcmd opmgrm1 viewspl
```

- **From a user ID with Operations Manager privileges:**

```
gomcmd opmgrm1 resume spool(splfull)
```

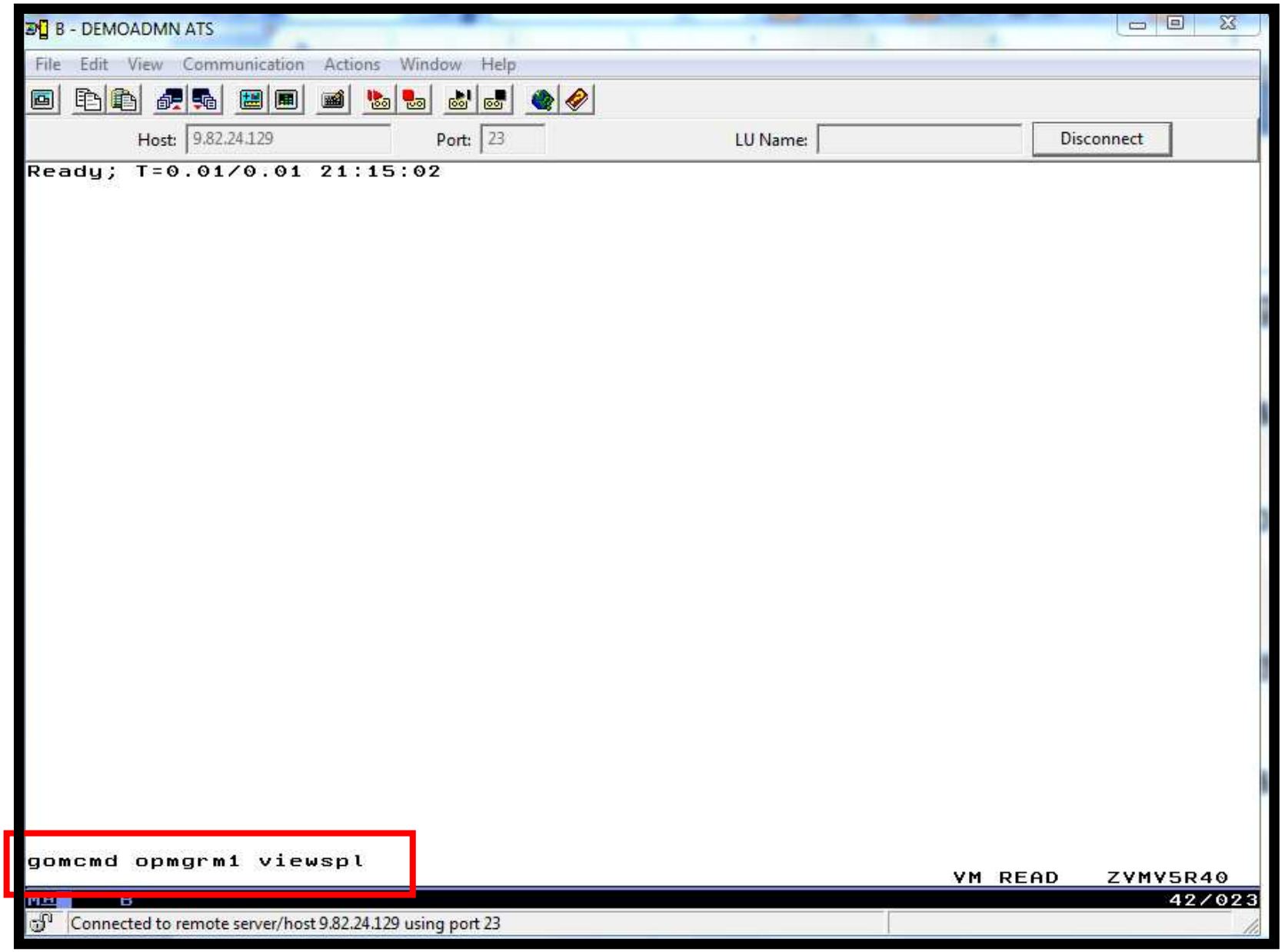
- **Check the Operations Manager log to see the spool monitor triggered:**

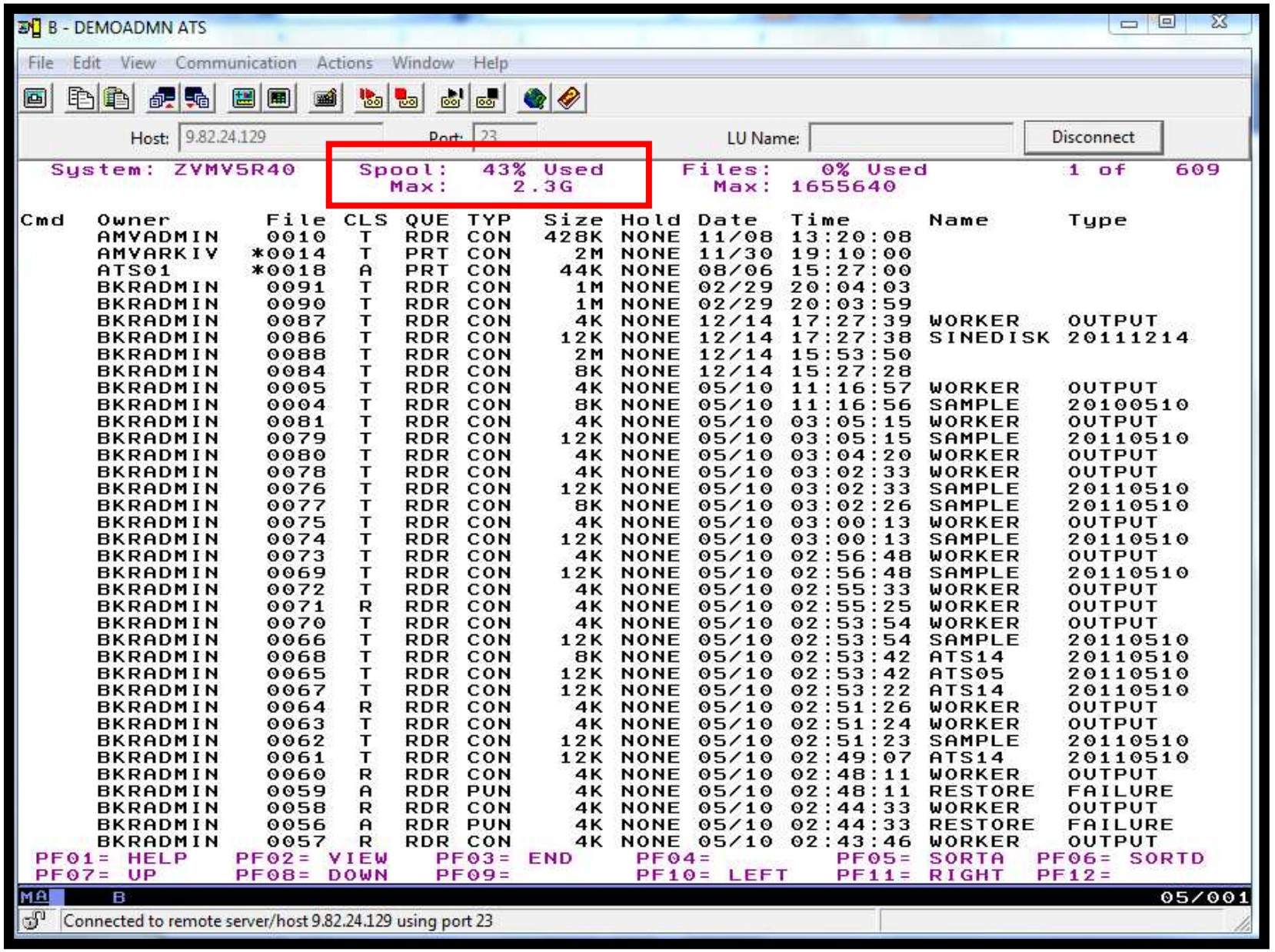
```
gomcmd opmgrm1 viewlog
```

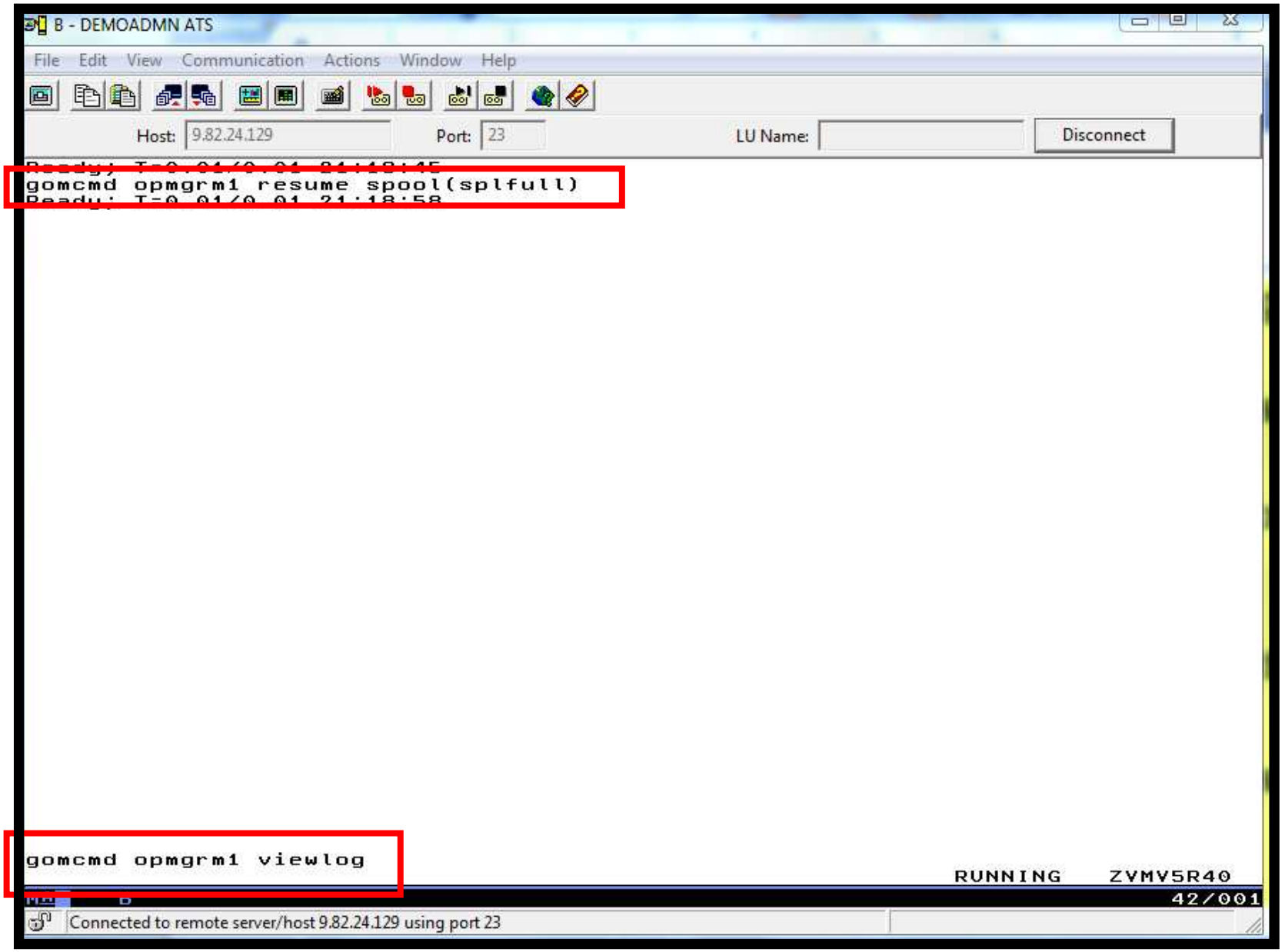
- **Check the inbox of the appropriate person to see the e-mail**

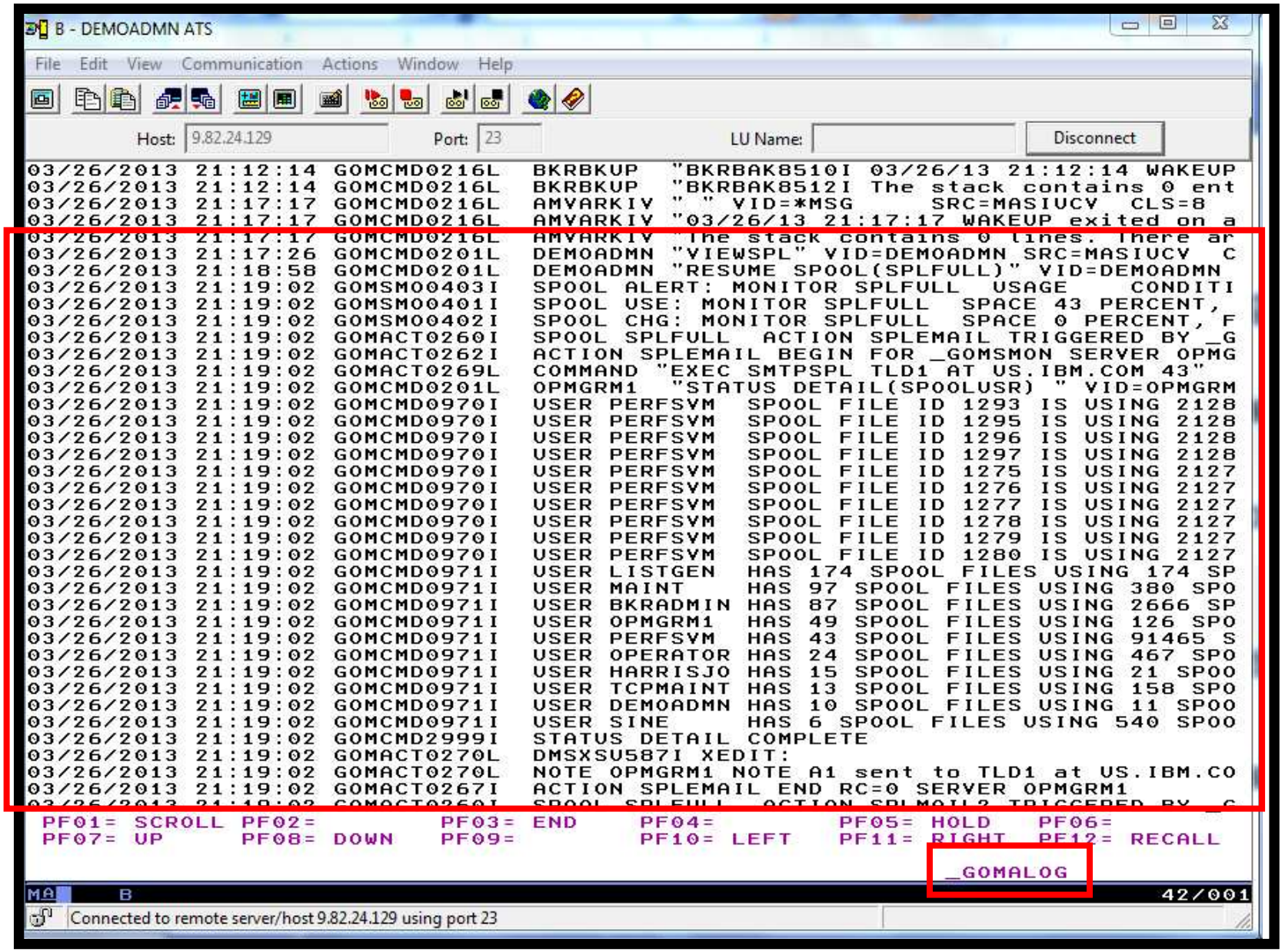
- **From a user ID with Operations Manager privileges:**

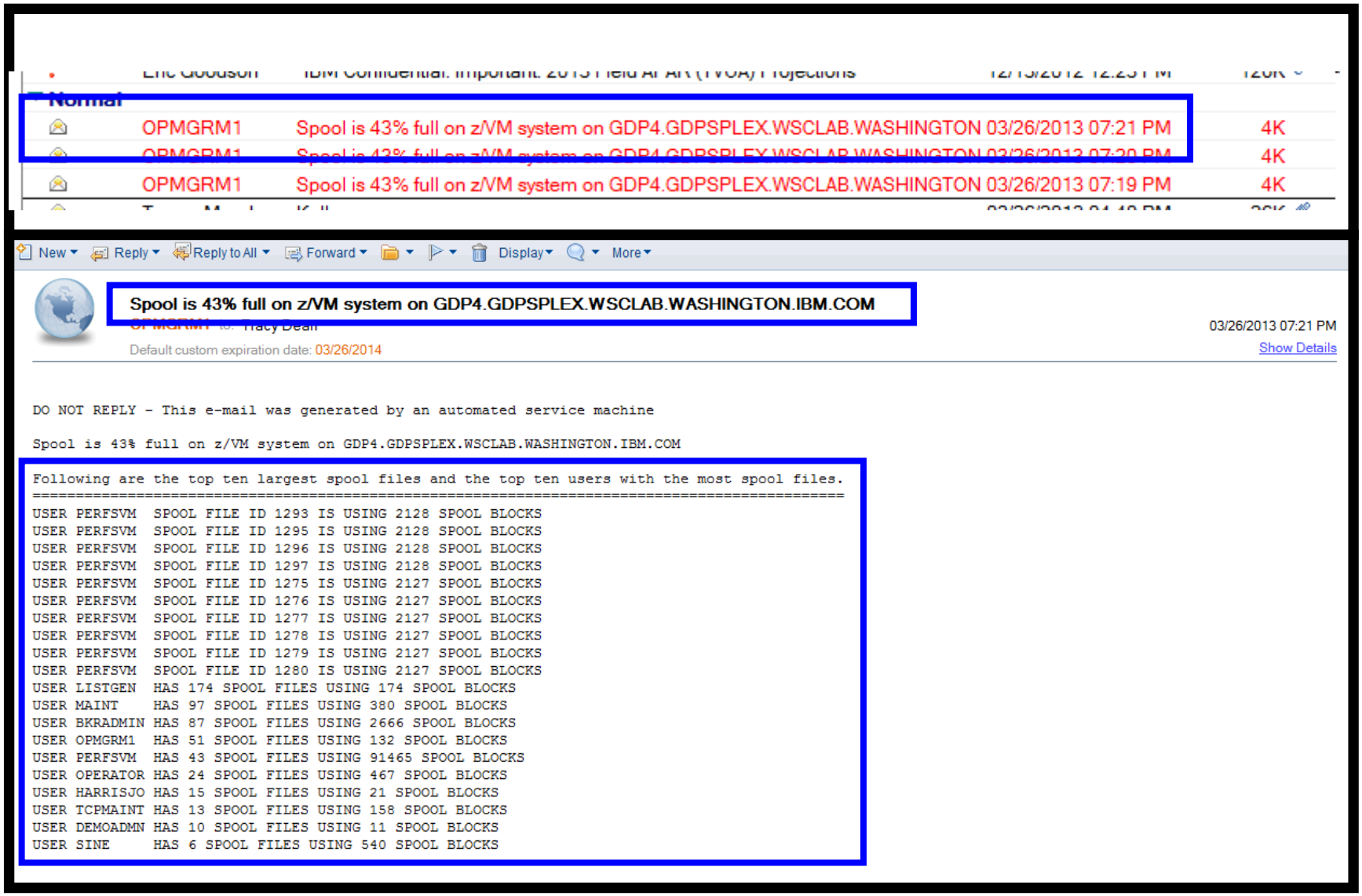
```
gomcmd opmgrm1 suspend spool(splfull)
```











Scenario 3b: How Do You Do That?

Spool monitor and action in Operations Manager:

```
*
*
DEFSMON NAME(SPLFULL),+
  USAGE(025-100),+
  INTERVAL(1),+
  LIMIT(3,3600),_
  ACTION(SPLEMAIL)
*
DEFACTN NAME(SPLEMAIL),+
  COMMAND(EXEC SMTPSPL tld1 at us.ibm.com &4),+
  ENV(LVM)
```

Scenario 3b: How Do You Do That?

SMTPSPL EXEC (excerpts)

```
/*  */
Parse arg mail_user dummyat mail_node spoolpct

errtext = 'Spool is' spoolpct'% full on z/VM system'

/* Get TCP hostname and domain from Ops Mgr global variables */
…
line.1 = 'OPTIONS: NOACK    LOG    SHORT   NONOTEBOOK ALL CLASS A'
line.2 = 'Date: ' Date() ',' Time()
…
line.8 = errtext 'on' fqdomain_name
line.9 = ' '
line.10 = 'Following are the top ten largest spool files and the top ten u
ith the most spool files.'
line.11 = '============================================================'
line.0 = 11
'PIPE stem line. | > TEMP NOTE A'
'PIPE command GOMCMD OPMGRM1 STATUS DETAIL(SPOOLUSR) | specs words 4-* 1 |
spooldata.'
spooldata.0 = 20
'PIPE stem spooldata. | >> TEMP NOTE A'
'EXEC SENDFILE TEMP NOTE A (NOTE SMTP'
```

Scenario 4:
Find and View Spool Files – Clean up the Spool

- **Authorized user specifies spool search criteria**
  - By user ID
  - By date
  - By file size
- **Result list presented**
  - Sort
  - Open/view a specific spool file
  - Purge, modify metadata, or transfer a file

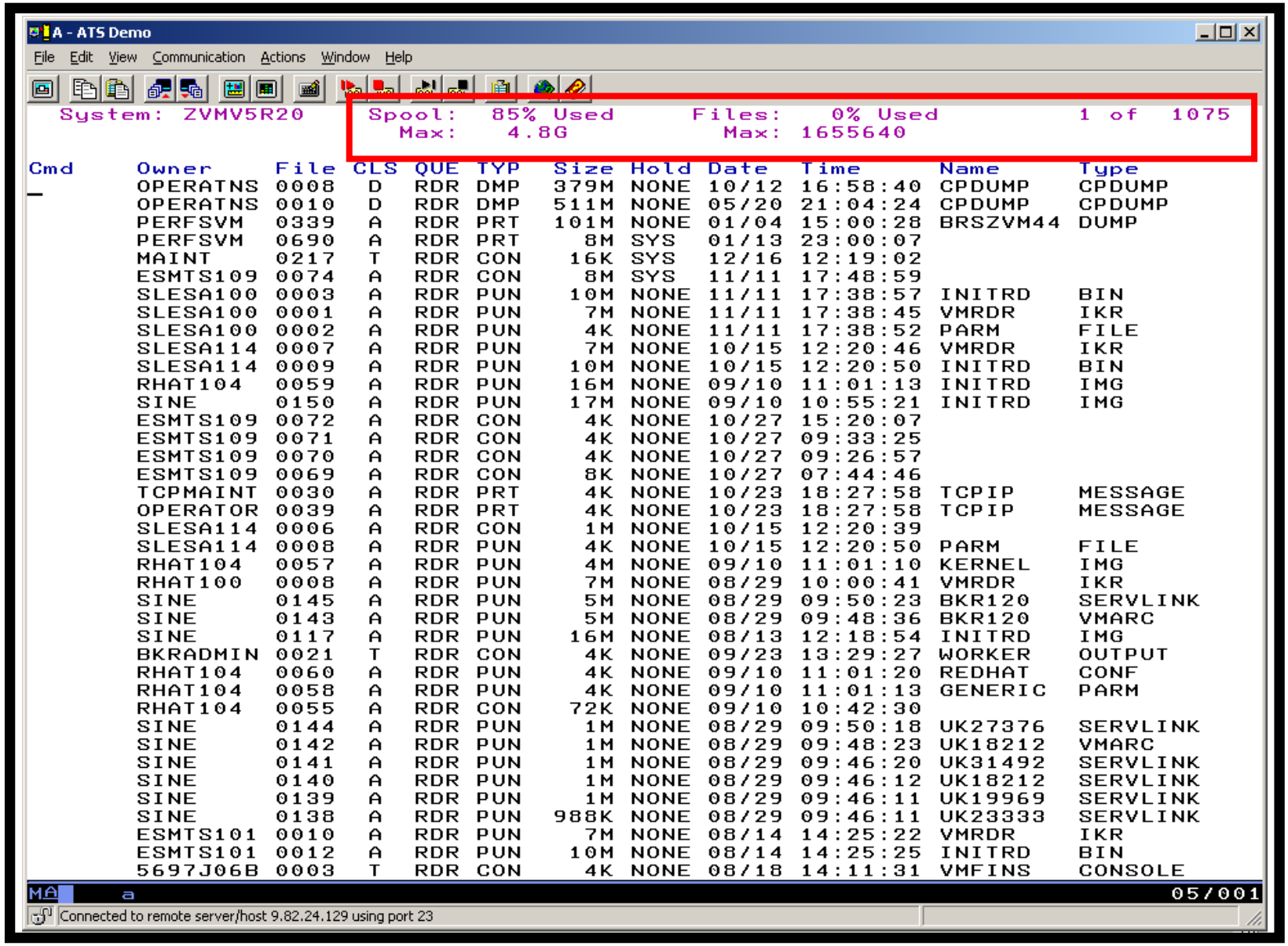
Scenario 4: Detailed Steps

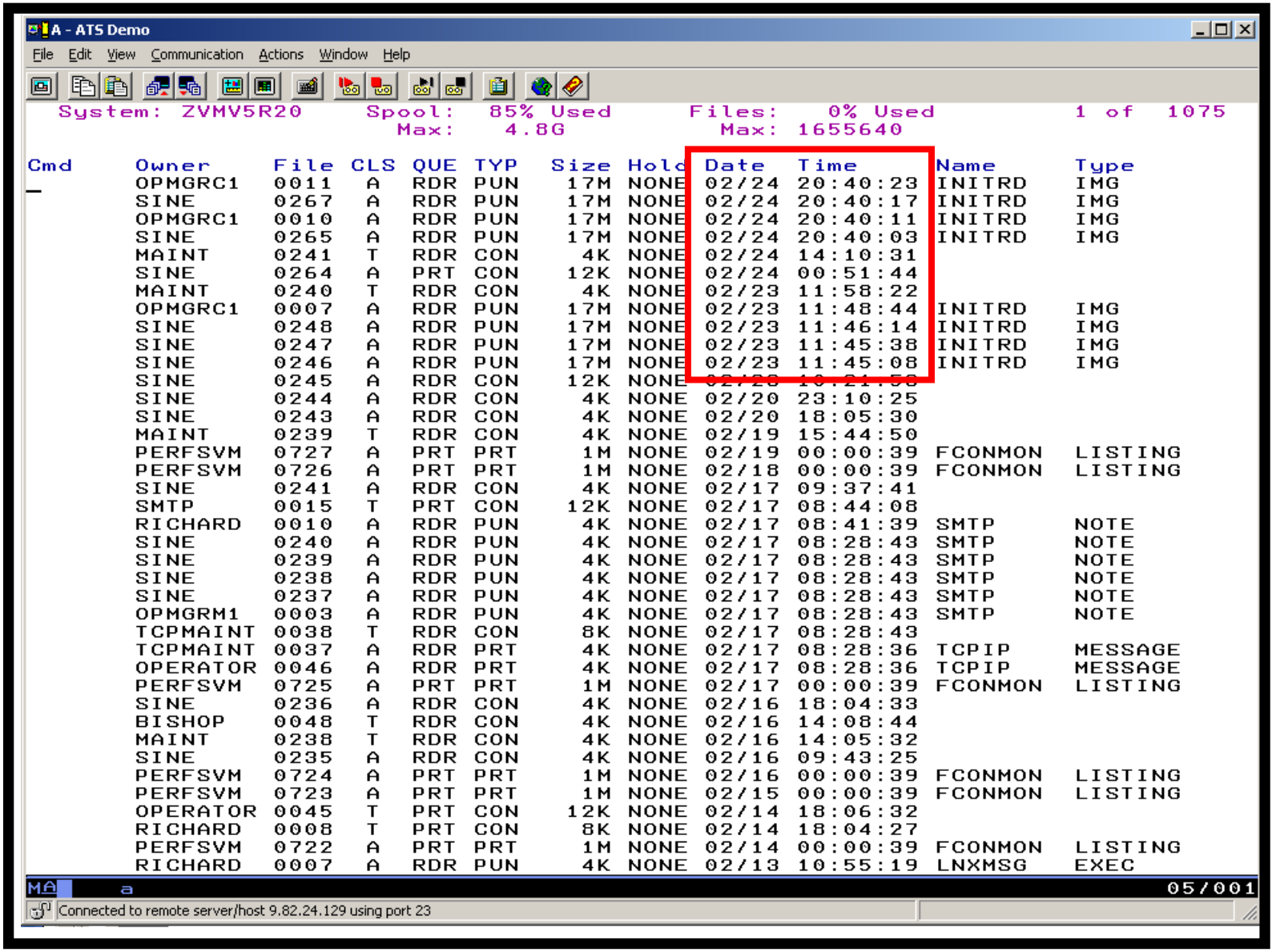
- **From an authorized VM user ID, view the spool files:**

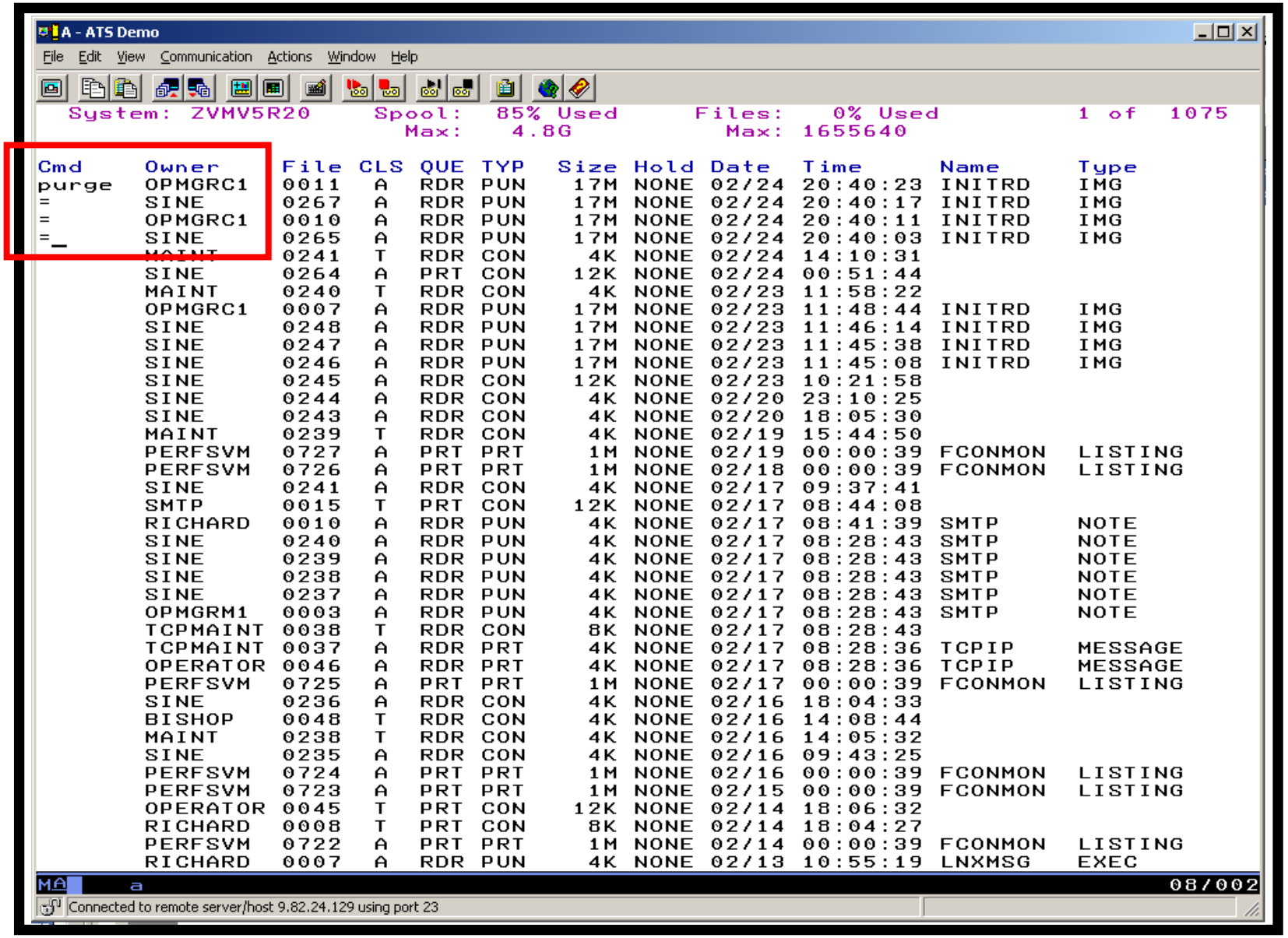
```
gomcmd opmgrm1 viewspl
```

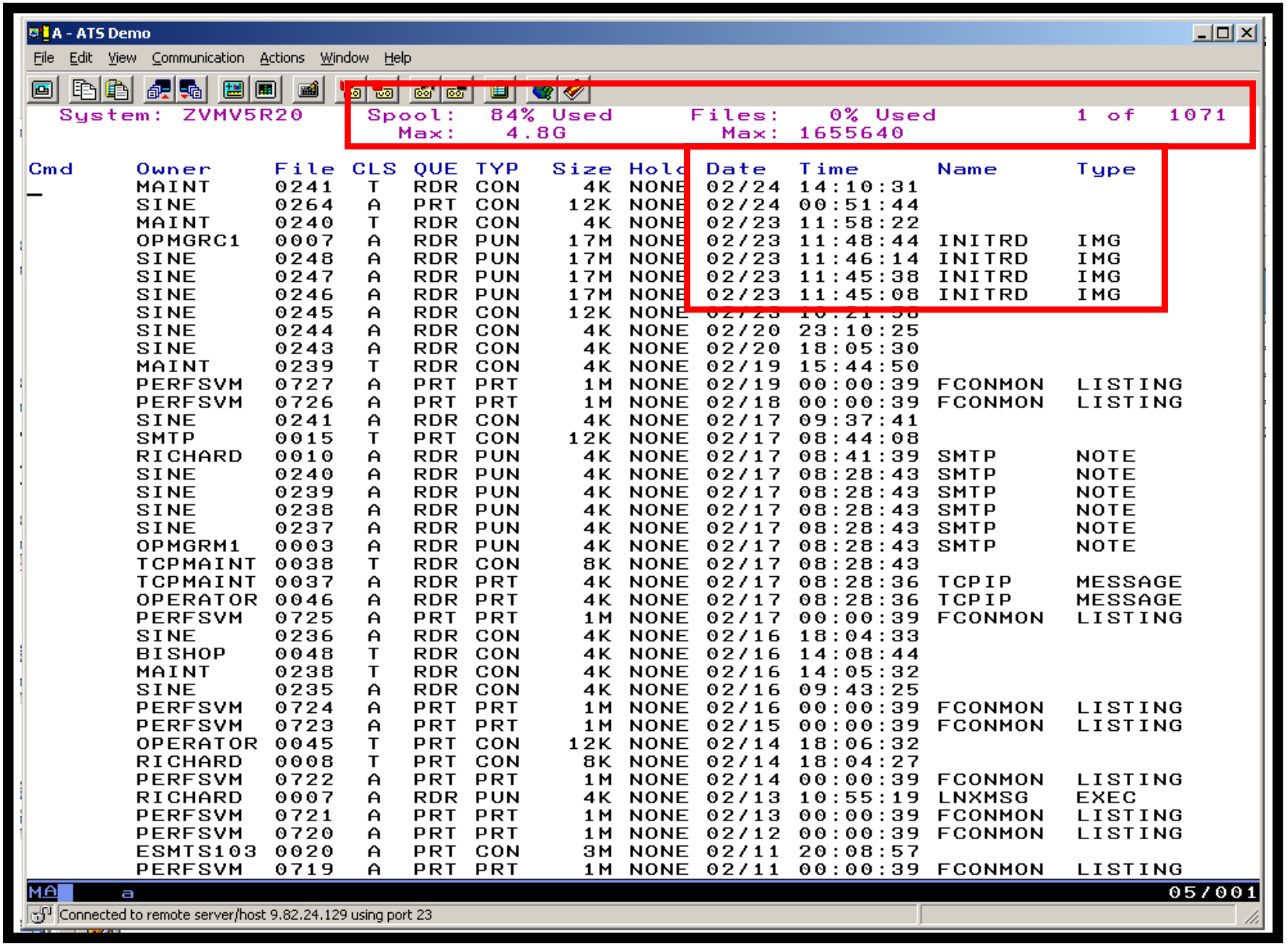
- **Sort by date**
  - Put cursor on date column header and hit F6
- **Find the spool files just sent and type PURGE next to them**
- **From an authorized VM user ID, view the log to see that the spool monitor is no longer triggered:**

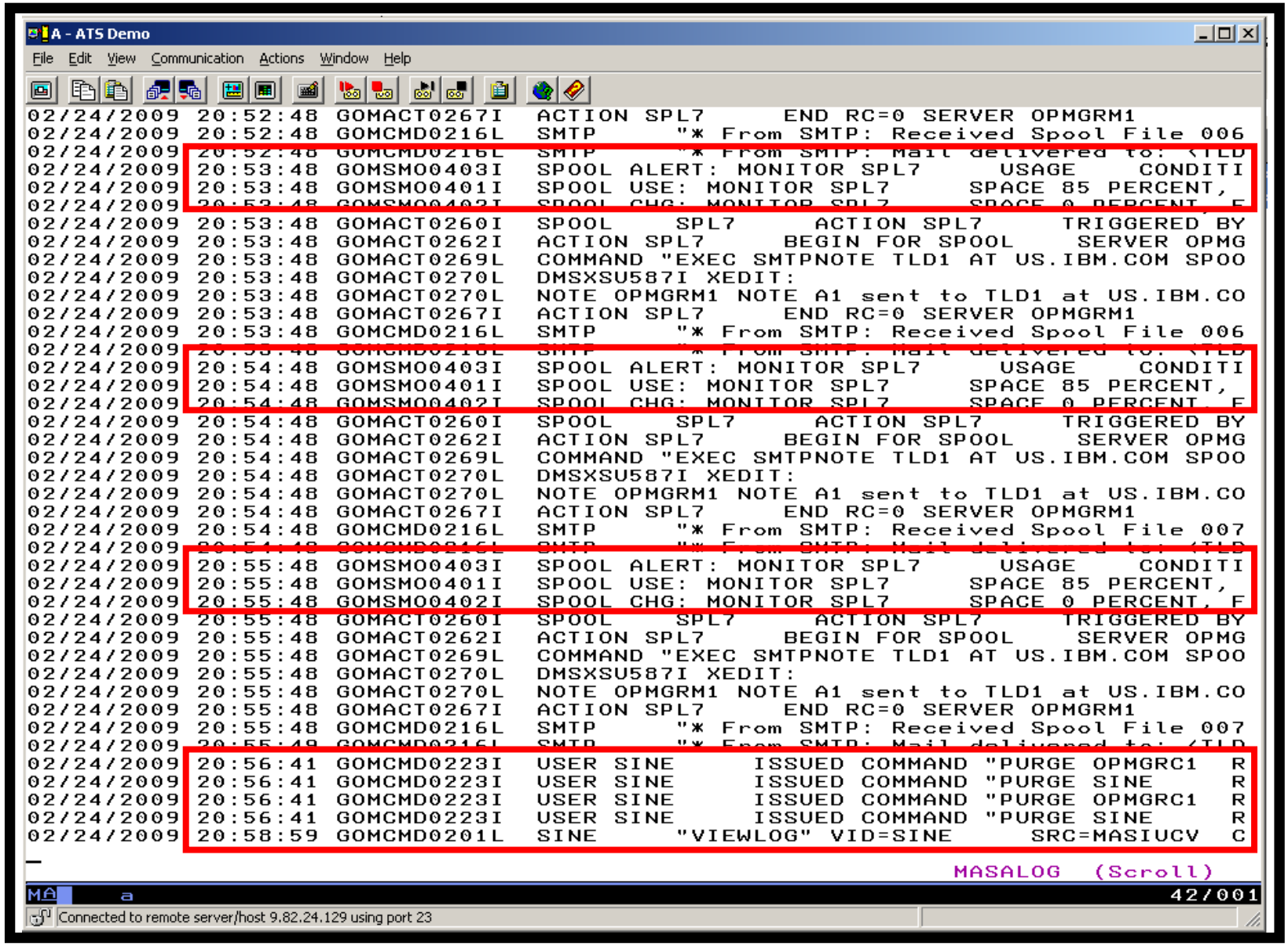
```
gomcmd opmgrm1 viewlog
```











Scenario 5:
Automated Spool Clean Up

- **Use z/VM SFPURGER utility to manage spool files based on criteria, e.g.**
  - User ID
  - Days in spool
  - Class
  - Number of records
- **Automate SFPURGER execution**
  - Regularly scheduled using Operations Manager
  - Triggered by Operations Manager spool monitor

Scenario 5: Detailed Steps

- **From an authorized VM user ID, view the spool files for a specific user:**

```
gomcmd opmgrm1 viewspl user(tstadmn2)
```

- **Send a file to this user as class Z**

```
sendfile profile exec a tstadmn2 (class z
```

- **View spool files for this user again to see the new file**

```
gomcmd opmgrm1 viewspl user(tstadmn2)
```

- **Delete any existing schedules called DEMO**

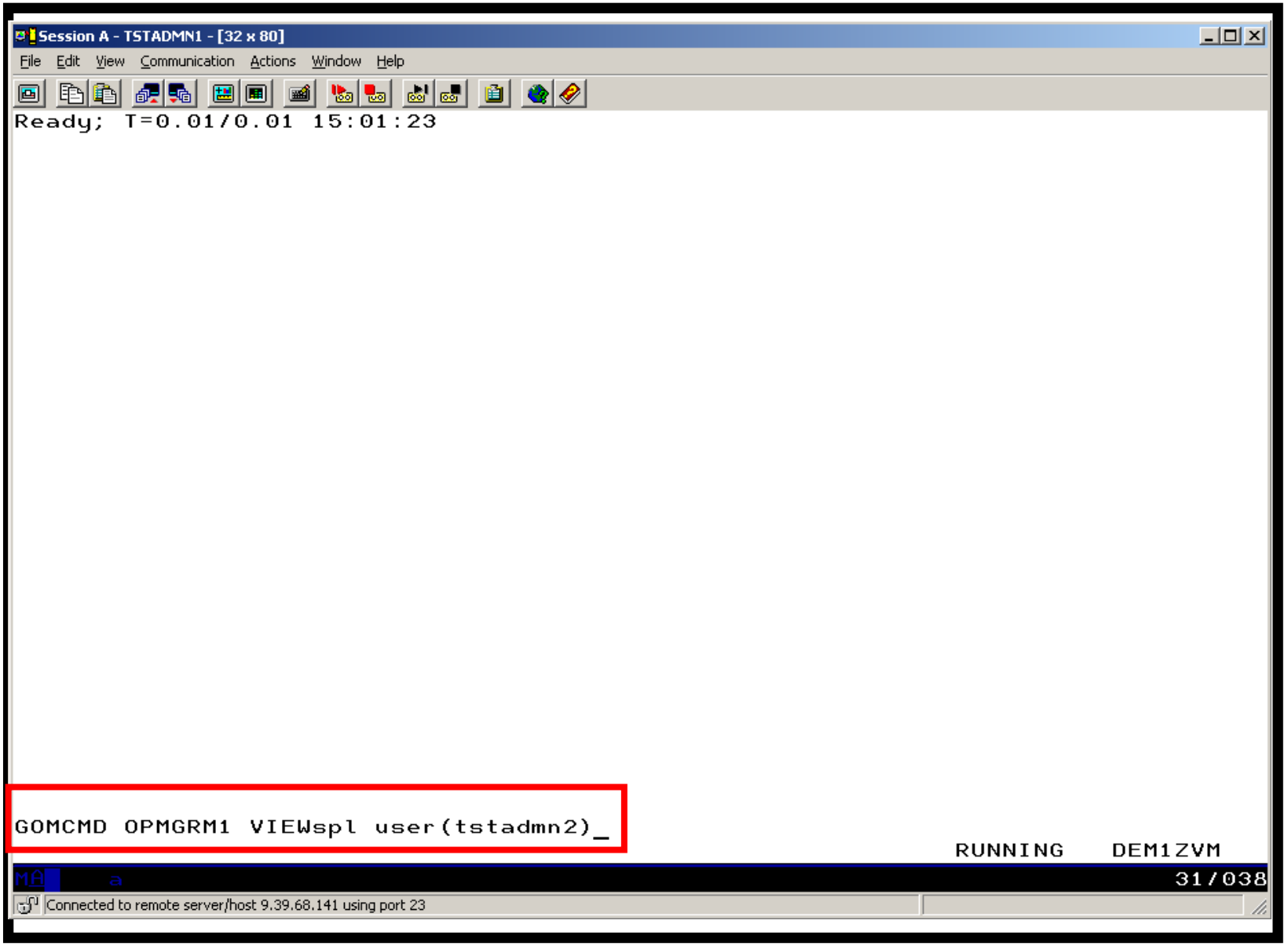
```
gomcmd opmgrm1 delschd name(demo)
```

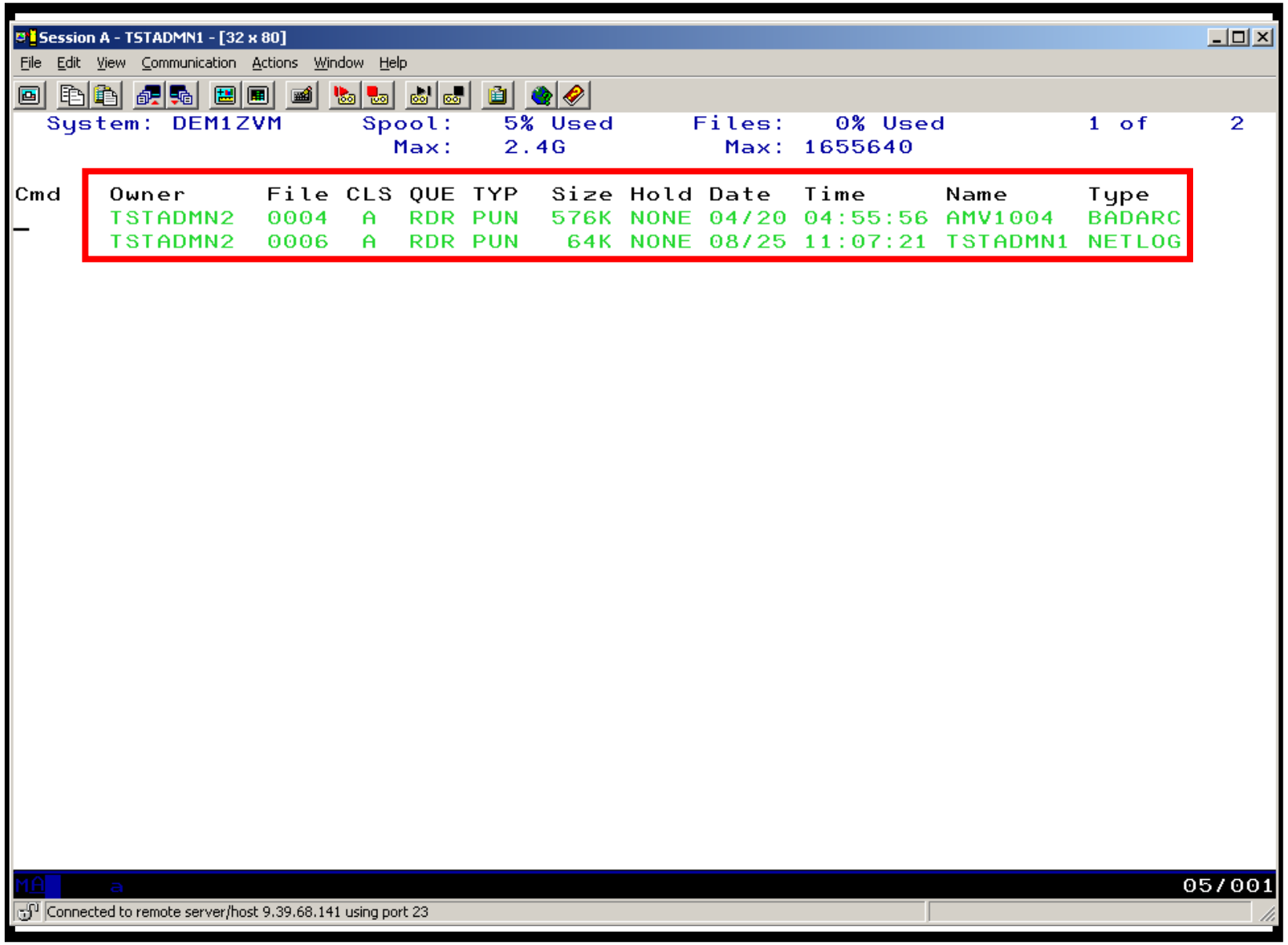
- **Schedule SFPURGER for execution**
  - It will purge any files of class Z

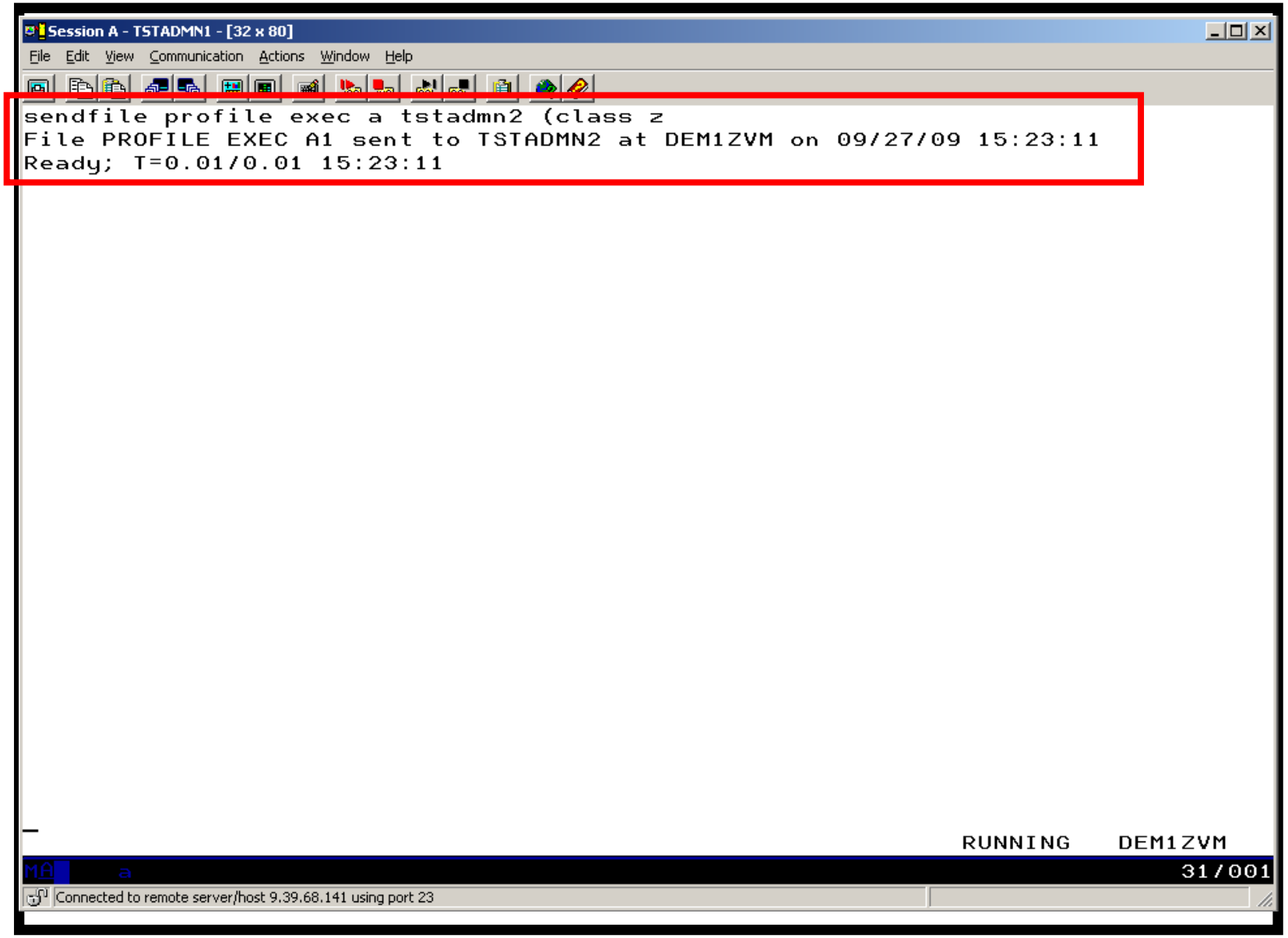
```
gomcmd opmgrm1 defschd name(demo),action(sfpurger),WHEN(now)
```

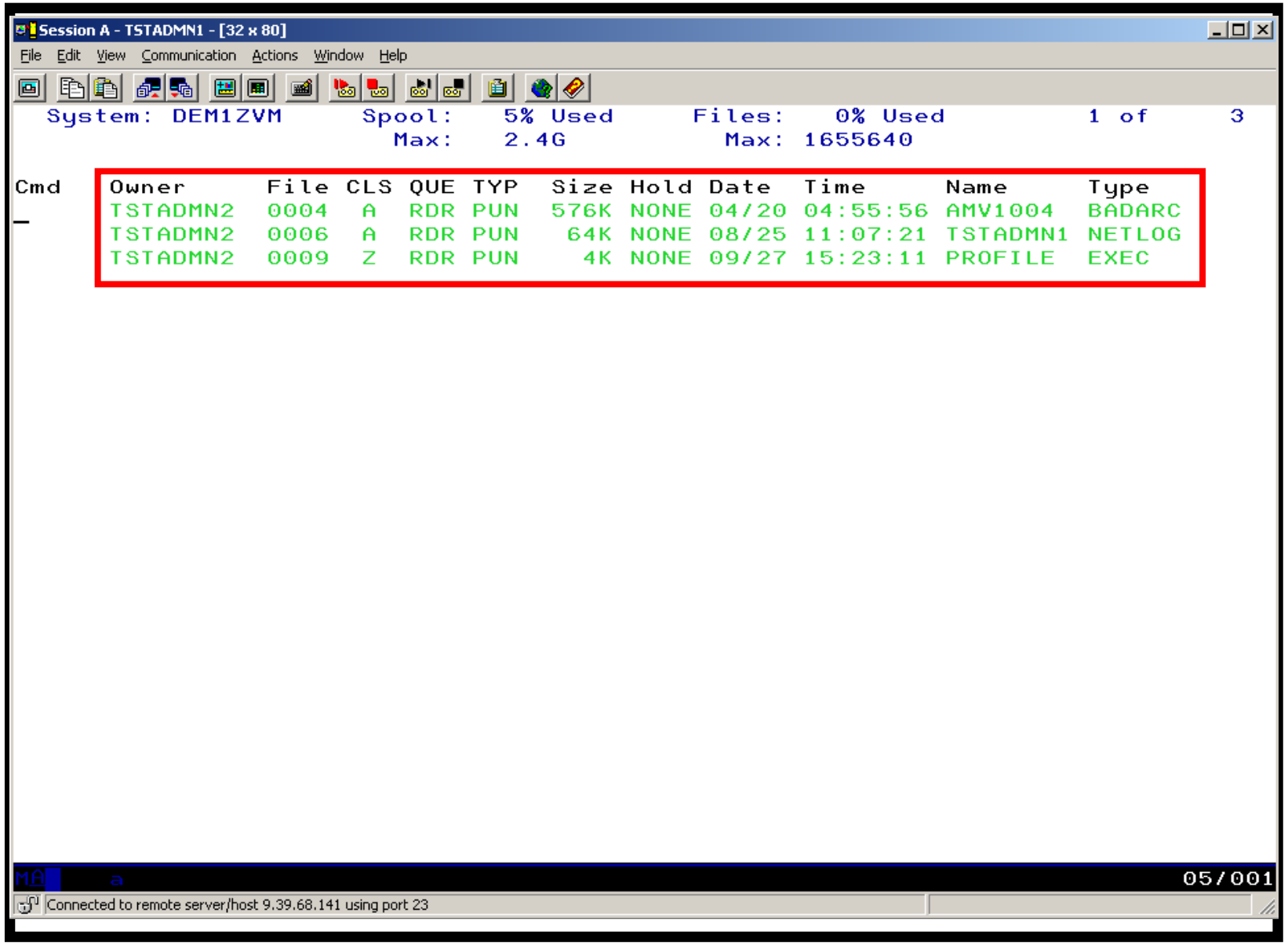
- **View spool files for this user again to see the new file is gone**

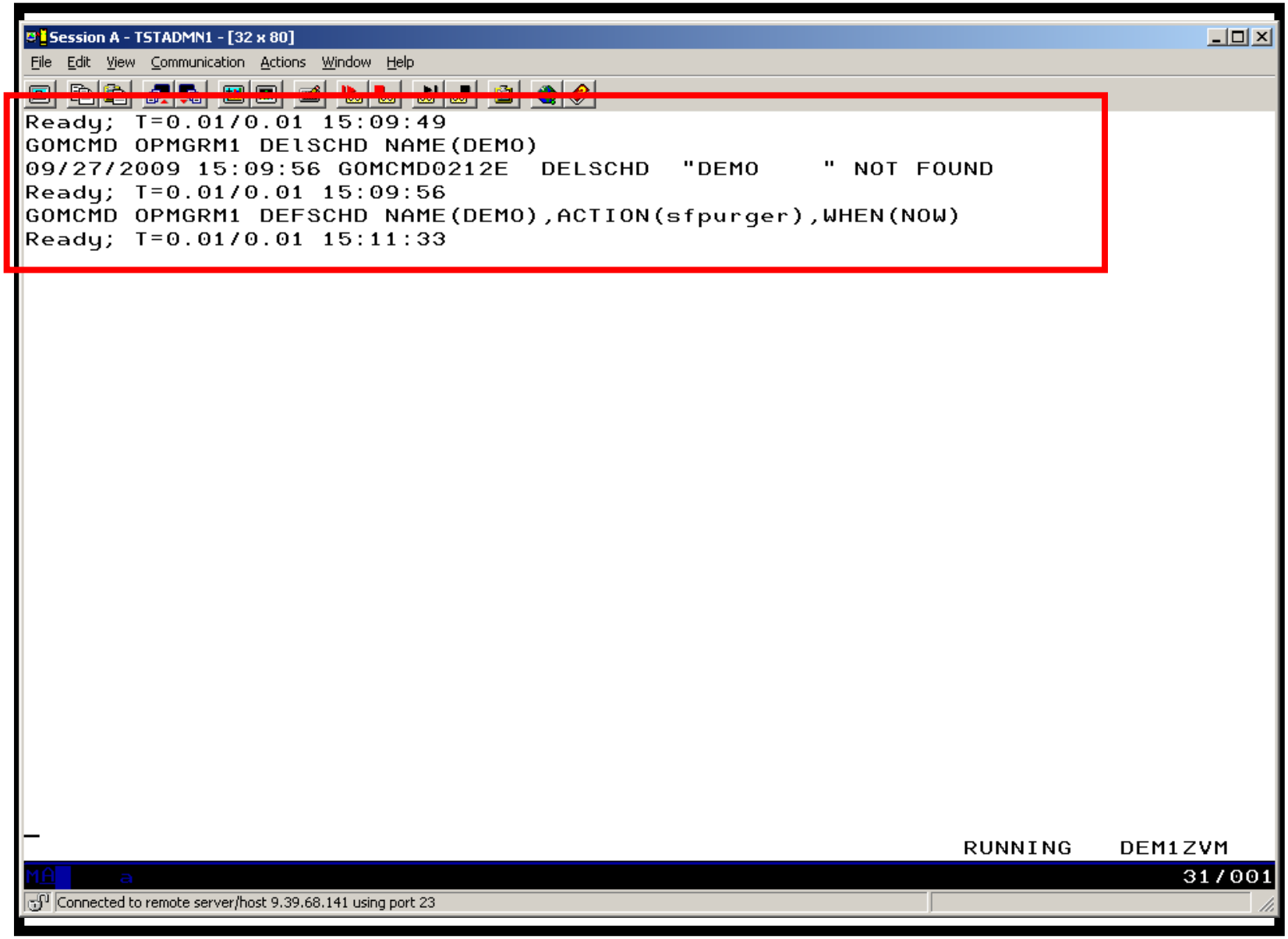
```
gomcmd opmgrm1 viewspl user(tstadmn2)
```

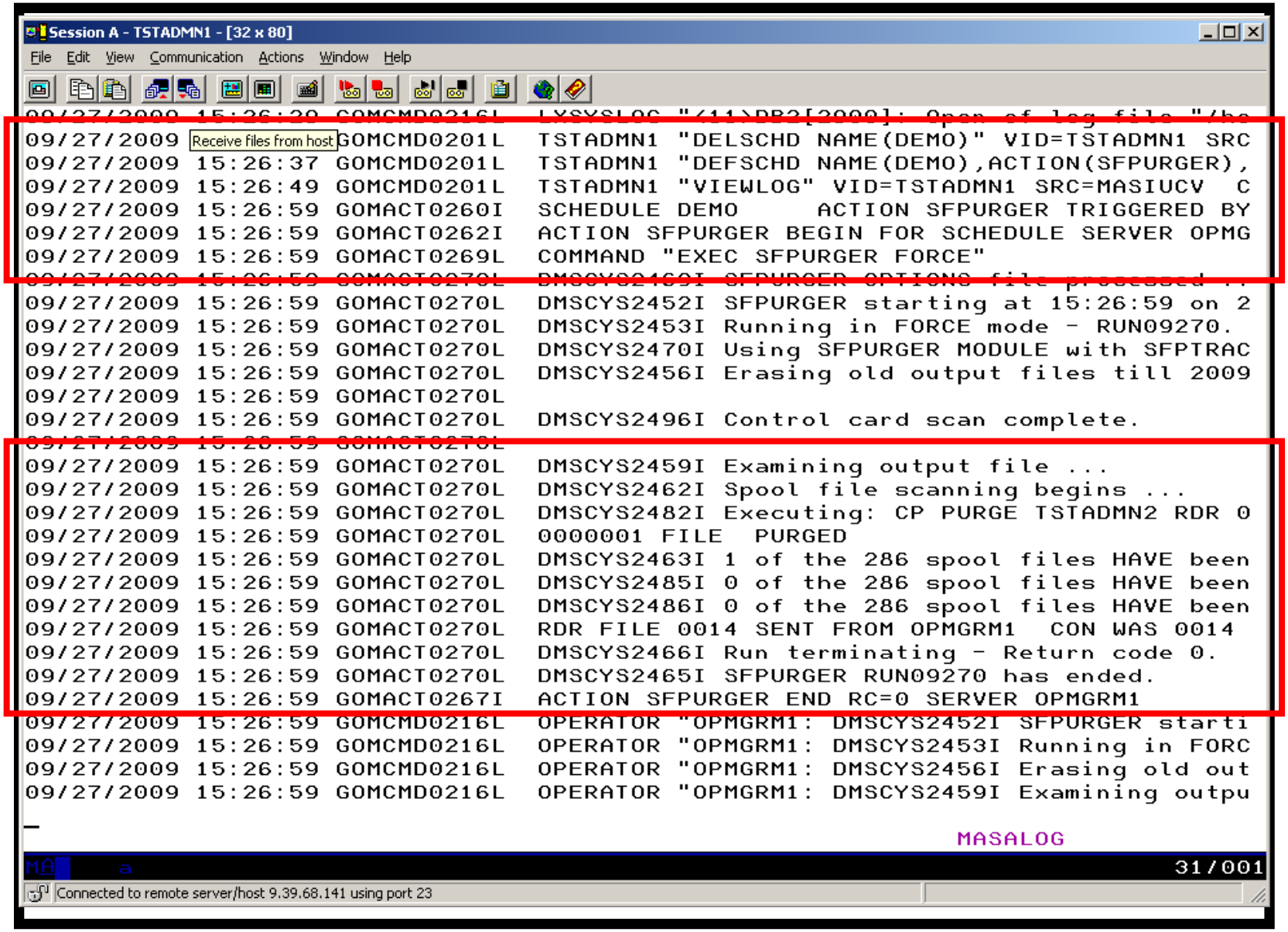


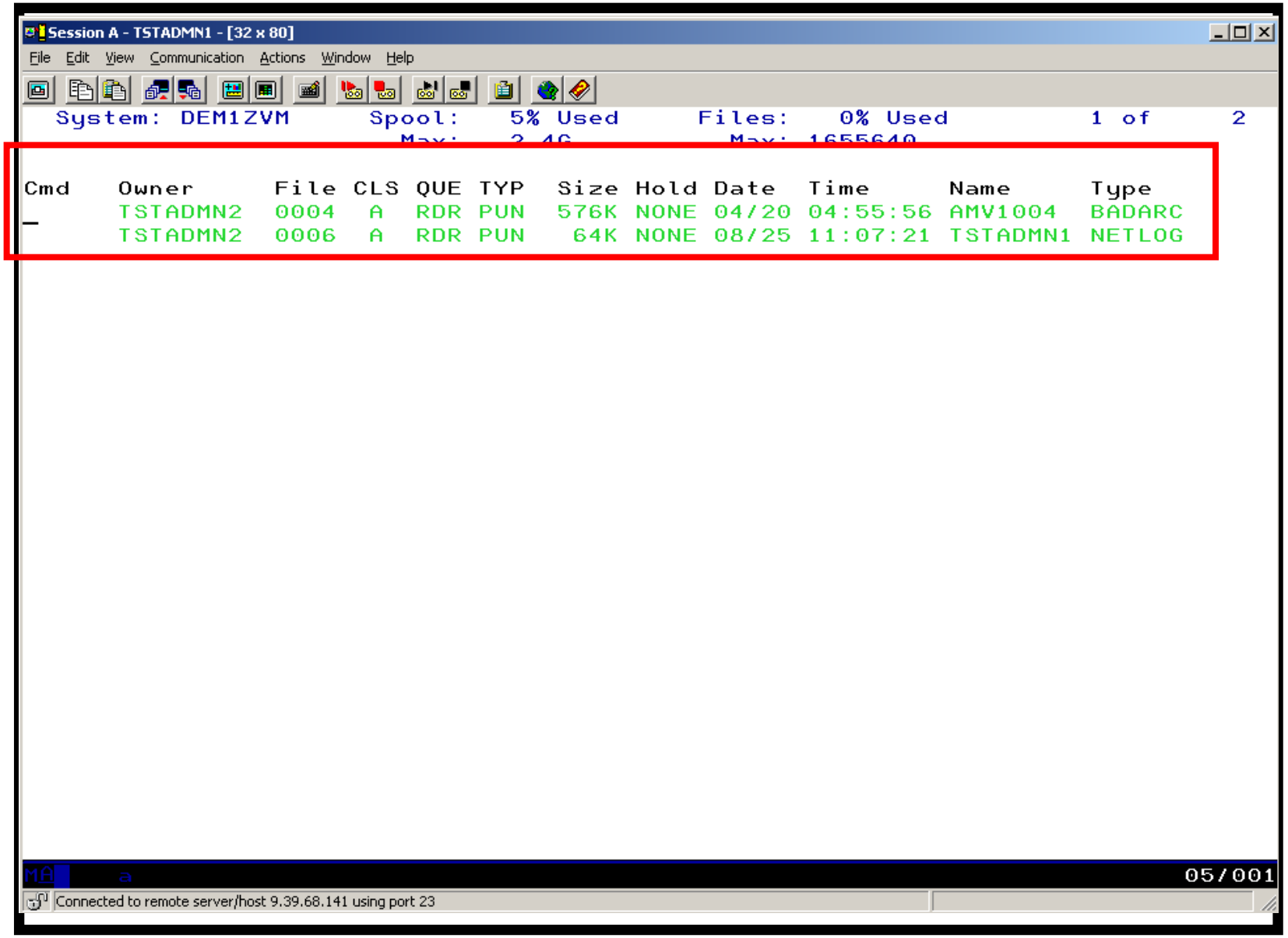












Scenario 5: How Do You Do That?

Action in Operations Manager to call z/VM's SFPURGER EXEC

```
*
DEFACTN NAME(SFPURGER),+
  COMMAND(EXEC SFPURGER FORCE),+
  OUTPUT(LOG),+
  ENV(LVM)
```

SFPURGER OPTIONS file

```
* Send console log to user ID TSTADMN1 at demo node
CONSOLE   TSTADMN1 DEM1ZVM
* Erase LOG and RUN files that are more than 3 days old
KEEPDAY  21
* Set prime shift start and end times
PRIMSHFT  07:30:00 16:30:00
* Use defaults for the following:
*   MSGTYPE  SORTMOD  SFPCNTL  SOSCNTL  SFPMOD  APPEND
SFPCNTL SFPTRACY
```

Scenario 5: How Do You Do That?

SFPTRACY CONTROL

```
* Ignore any spool files found in the NSS queue (privilege class E)
QUEUE NSS                                              ACTION IGNORE
*
* Purge any spool files found in class Z
CLASS Z                                                ACTION PURGE
```

Make sure OPMGRM1 links and accesses MAINT 193 disk for access to SFPURGER functions

Scenario 6:
Detecting Disk Full Conditions of Logging IDs

- **Operations Manager monitors the console of a user ID that does logging**
  - DIRMAINT, for example
- **Disk full or early warning message triggers a rule/action in Operations Manager**
  - Quiesce or shut down DIRMAINT
  - Send the log files to a separate service machine
  - Erase the log files from DIRMAINT's logging disk
  - Restart DIRMAINT
  - Separately, other service machine automatically archives all files it receives (in Archive Manager for z/VM)
  - Log files are safely archived in Archive Manager and DIRMAINT is running with a clean log disk
- **Get a copy of the console for further review/debugging**

Scenario 6: Detailed Steps

- **From an authorized VM user ID, view the DIRMAINT console:**

```
gomcmd opmgrm1 viewcon user(dirmaint)
```

- **In the console view**
  - Issue CMS commands to copy old (large) log files to DIRMAINT's log disk

```
cms copyfile dirmaint tlog0914 t = tlog0912 h
```

  - Verify the logging disk is more than 75% full

```
cms q disk
```

  - Run DIRMAINT's hourly processing now

```
exec dvhourly
```

  - Verify the logging disk is less than 75% full

```
cms q disk
```

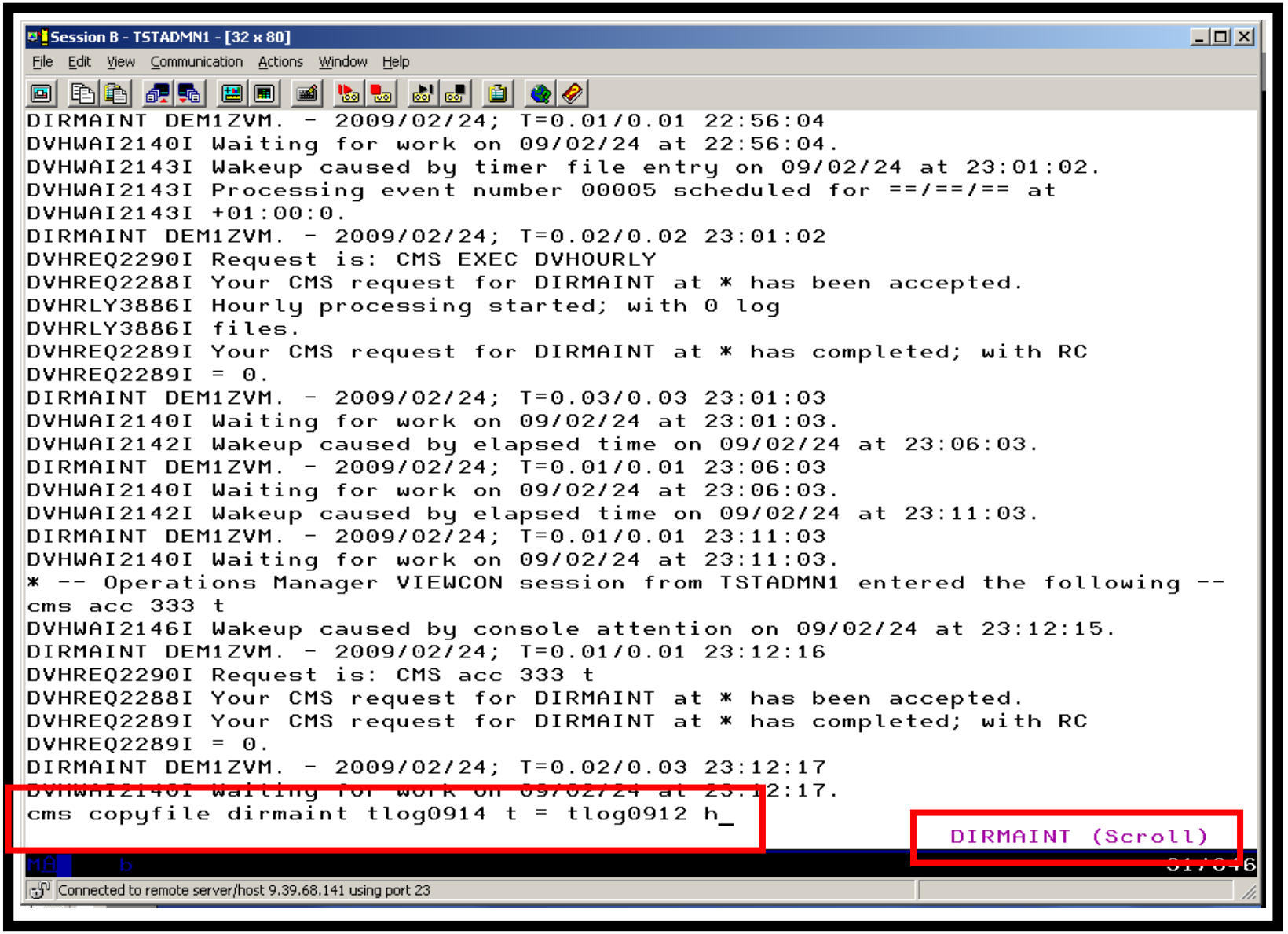
- **Exit the console view and find the files in the archive**

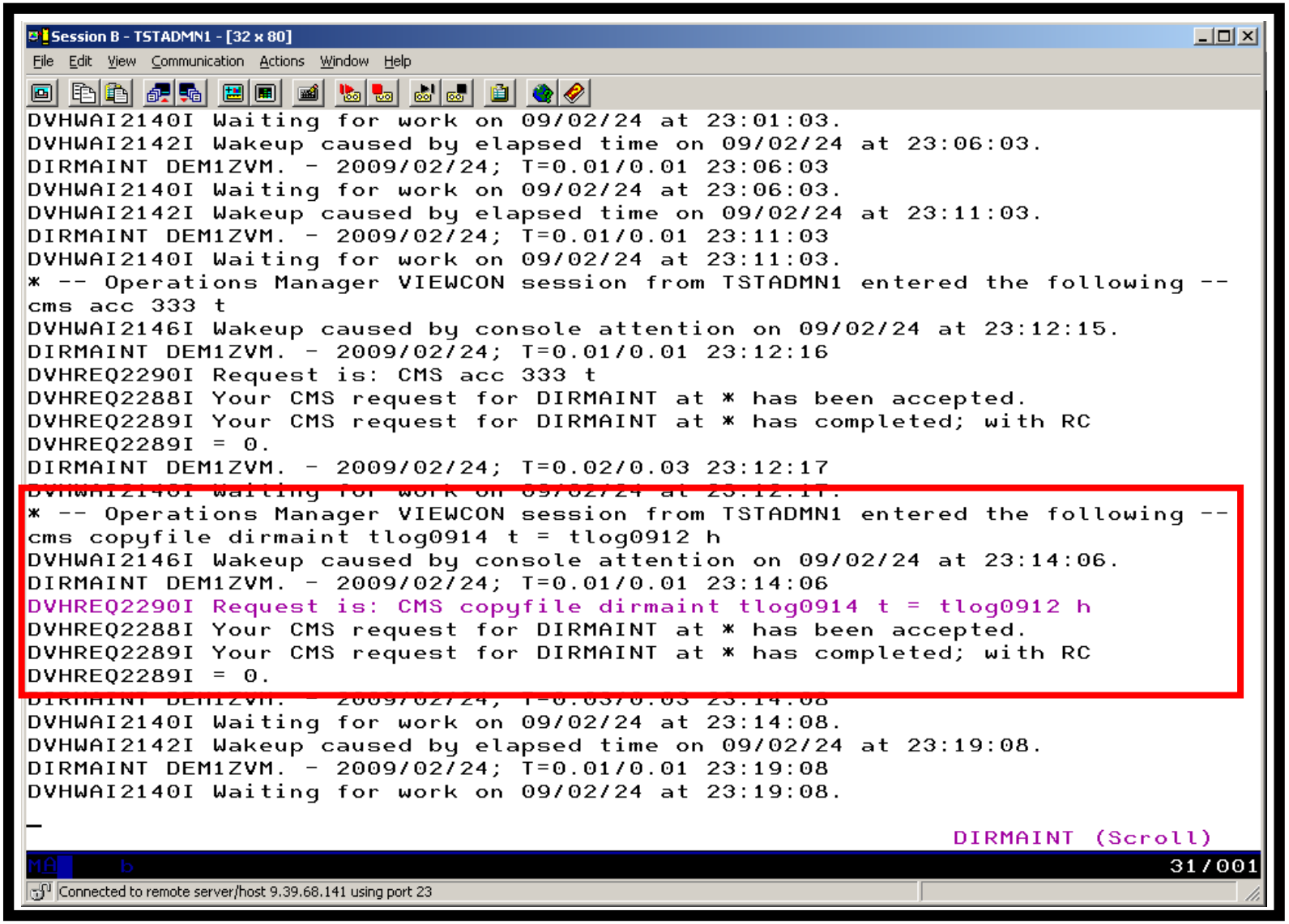
```
amvlist
```

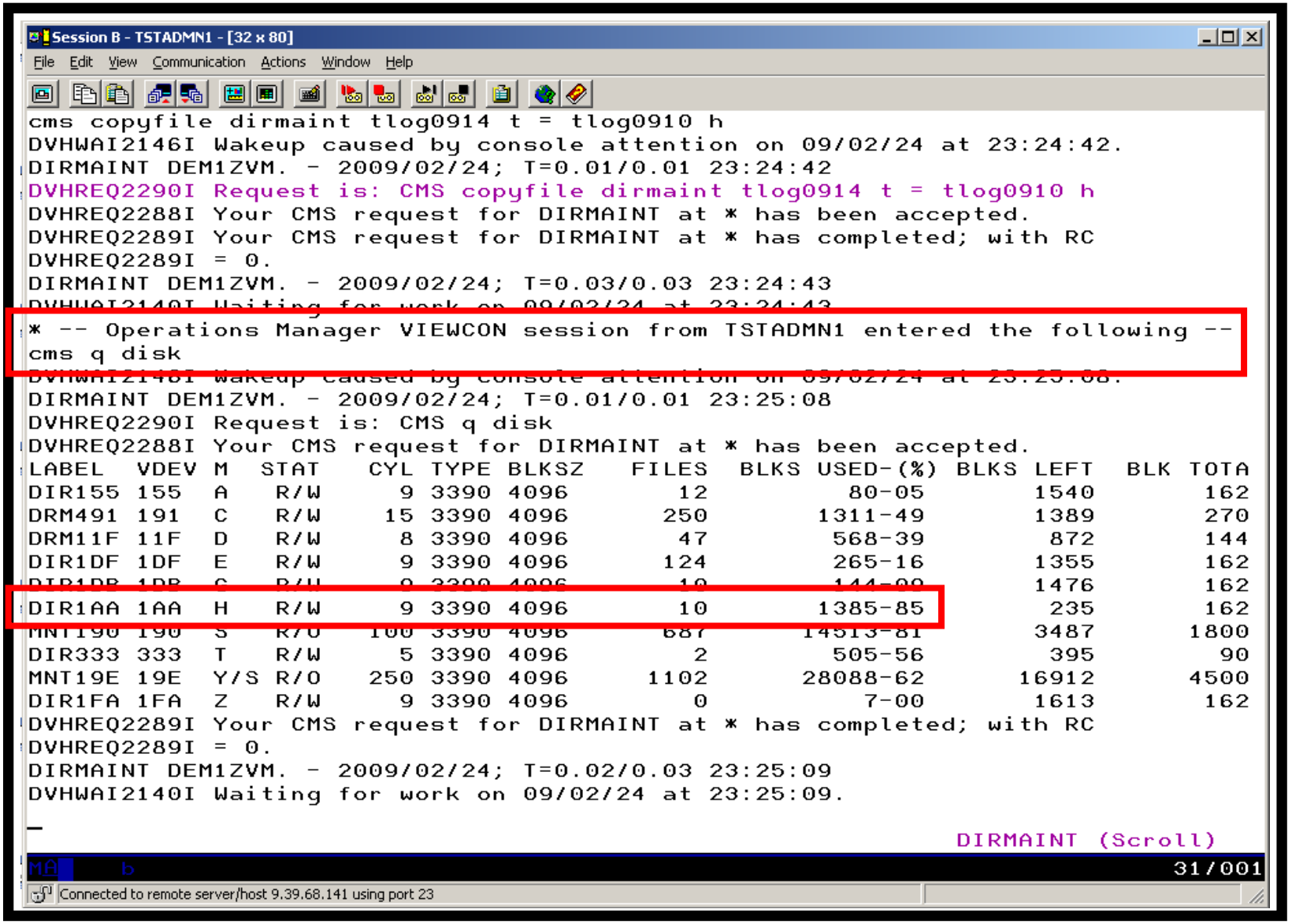
  - Type "archlogs" in the owner field and press ENTER

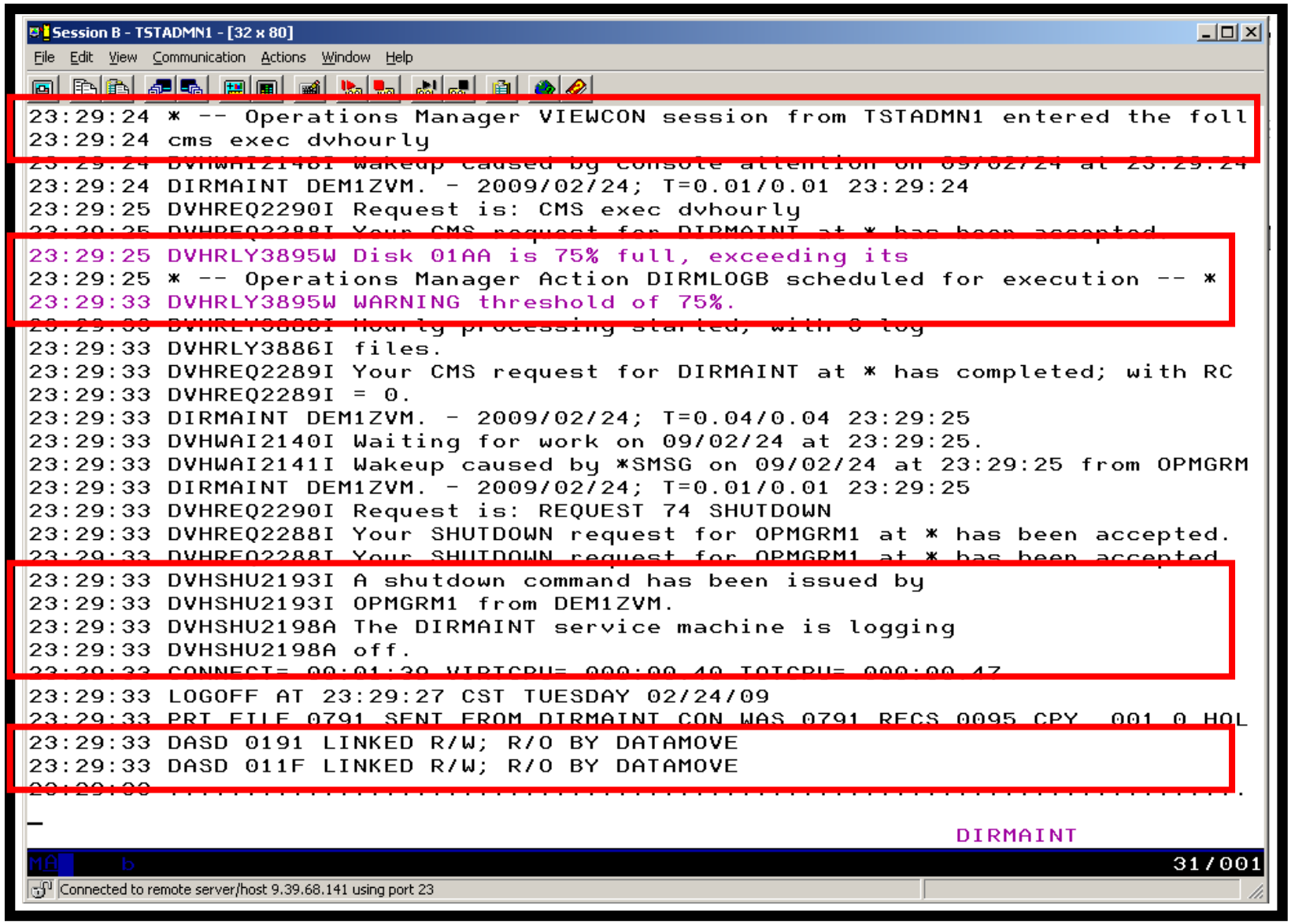
- **Request a copy of the console for further review/debugging**

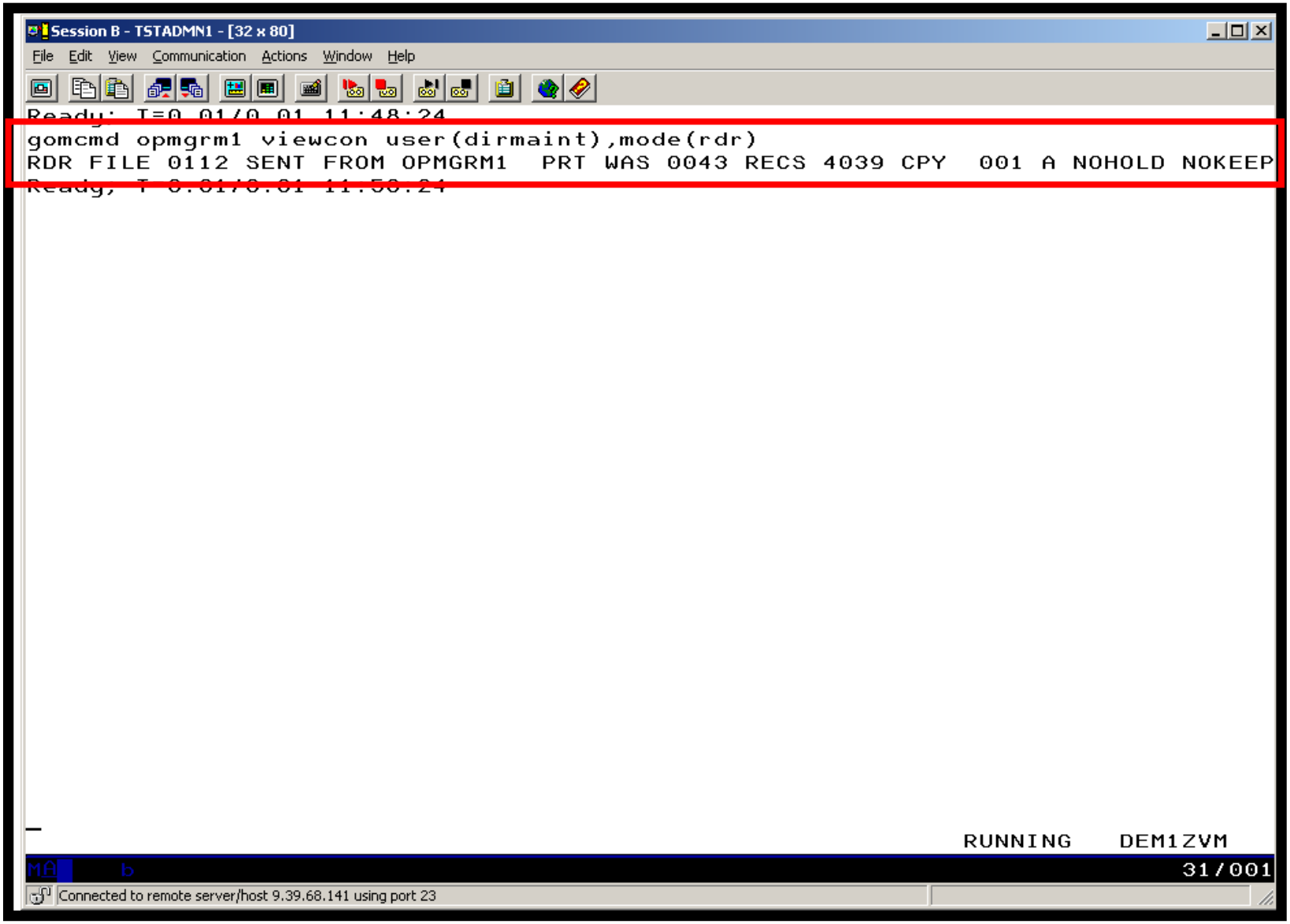
```
gomcmd opmgrm1 viewcon user(dirmaint),mode(rdr)
```

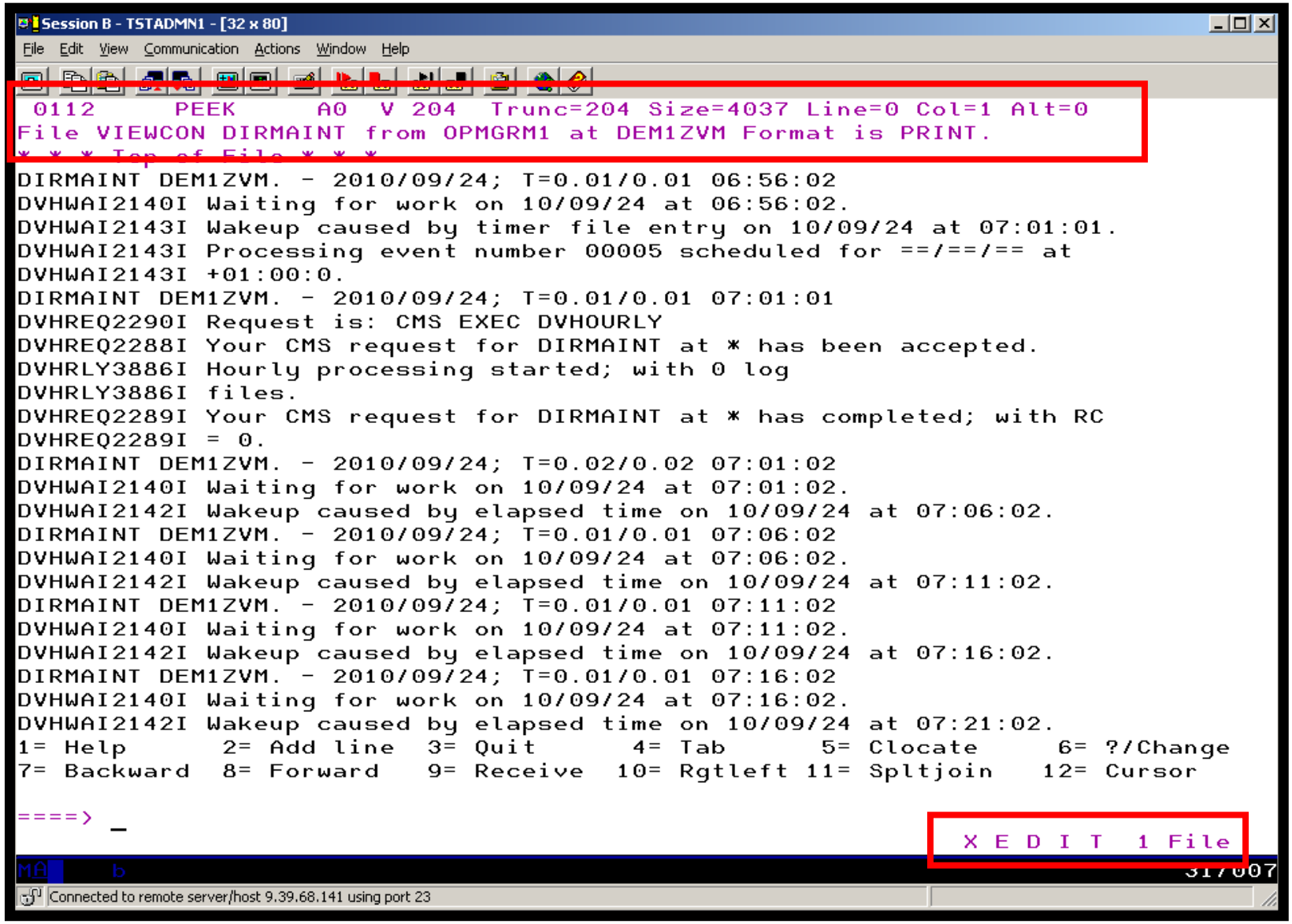












Scenario 6: How Do You Do That?

Console rule and action in Operations Manager:

```
DEFRULE NAME(DIRMLOG),+
  MATCH(*DVHRLY3895*01AA*),+
  USER(DIRMAINT),+
  ACTION(DIRMLOG)
*
DEFACTN NAME(DIRMLOG),+
  INPUT(AHI),+
  NEXTACTN(DIRMLOGB)
*
DEFACTN NAME(DIRMLOGB),+
  COMMAND(EXEC DIRM1AA &U),+
  ENV(LVM)
```

Authorize Operations Manager to issue DIRM SHUTDOWN – from MAINT issue

```
DIRM AUTHFOR OPMGRM1 CMDLEVEL 150A CMDSET O
```

Scenario 6: How Do You Do That?

DIRM1AA EXEC (excerpts):

```
Parse Upper Arg Tuser . ;
/* Try to shut DIRMAINT down. */
Say 'DIRM1AA - Issuing DIRM SHUTDOWN ....';
Address CMS 'DIRM SHUTDOWN';

Address Command 'CP LINK' Tuser '1AA' Dev 'MR';
Address CMS 'ACCESS' Dev Fm;

Address Command 'PIPE CMS LISTFILE DIRMAINT *LOG*' Fm '( NOHEADER',
  '| STEM FILES.';
Do I = 1 to Files.0;
  Parse Upper Var Files.I Fn Ft .;
  Address CMS 'SENDFILE' Fn Ft Fm 'TO ARCHLOGS';
  If Rc = 0 then Do;
    Sent = Sent+1;
    Address CMS 'ERASE' Fn Ft Fm;
  End

Address Command 'CP XAUTOLOG' Tuser;
```

Scenario 7:
Process a File of Test Messages as a Console

- **Create a file containing lines of test messages**
  - Test rules and actions without creating critical conditions
- **Use Operations Manager to send the file for processing**
  - Treat it as the console of one user
  - Send it again treating it as the console of another user
  - Notice triggered rules and actions are different
- **View the “consoles” of these two users**

Scenario 7: Detailed Steps

- **Create or view a file of test messages**

```
xedit test consdata a
```

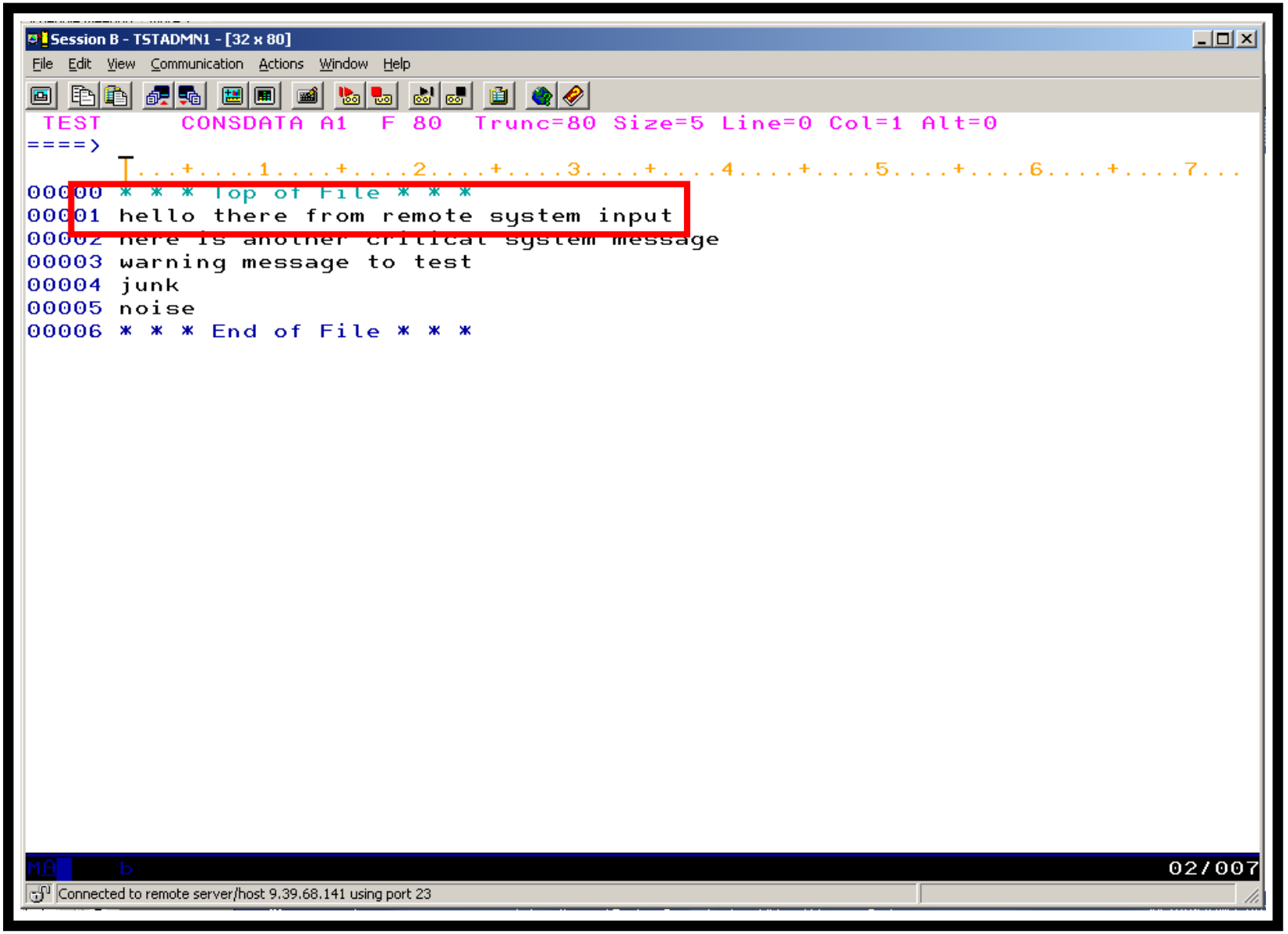
  - Notice the “hello” message in the file

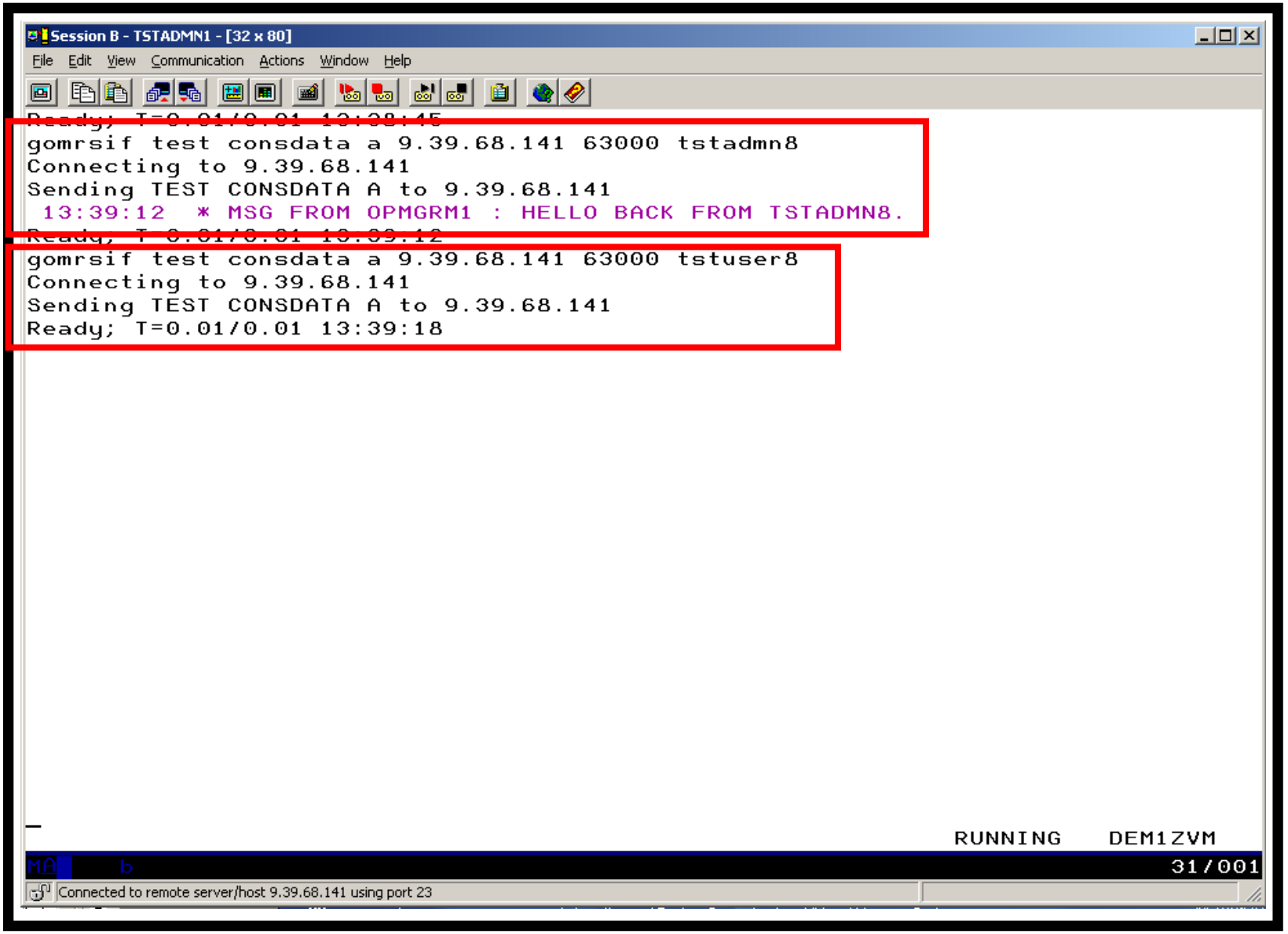
- **From a z/VM user ID, send the test file to Operations Manager**
  - Send it twice, specifying two different “owning” user IDs. One generates a message and one doesn’t:

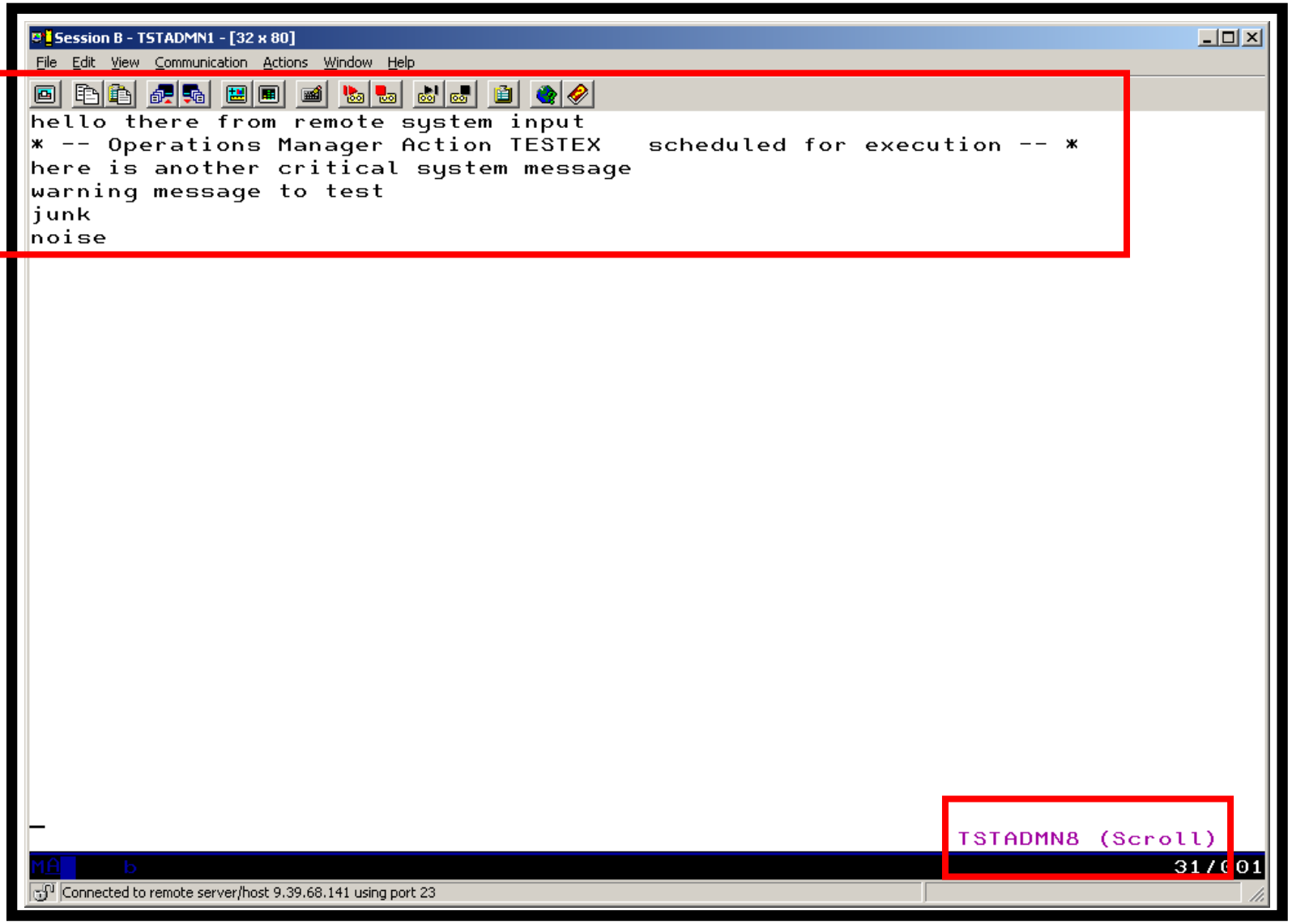
```
gomrsif test consdata a 9.39.64.72 63000 tstadmn8
gomrsif test consdata a 9.39.64.72 63000 tstuser8
```

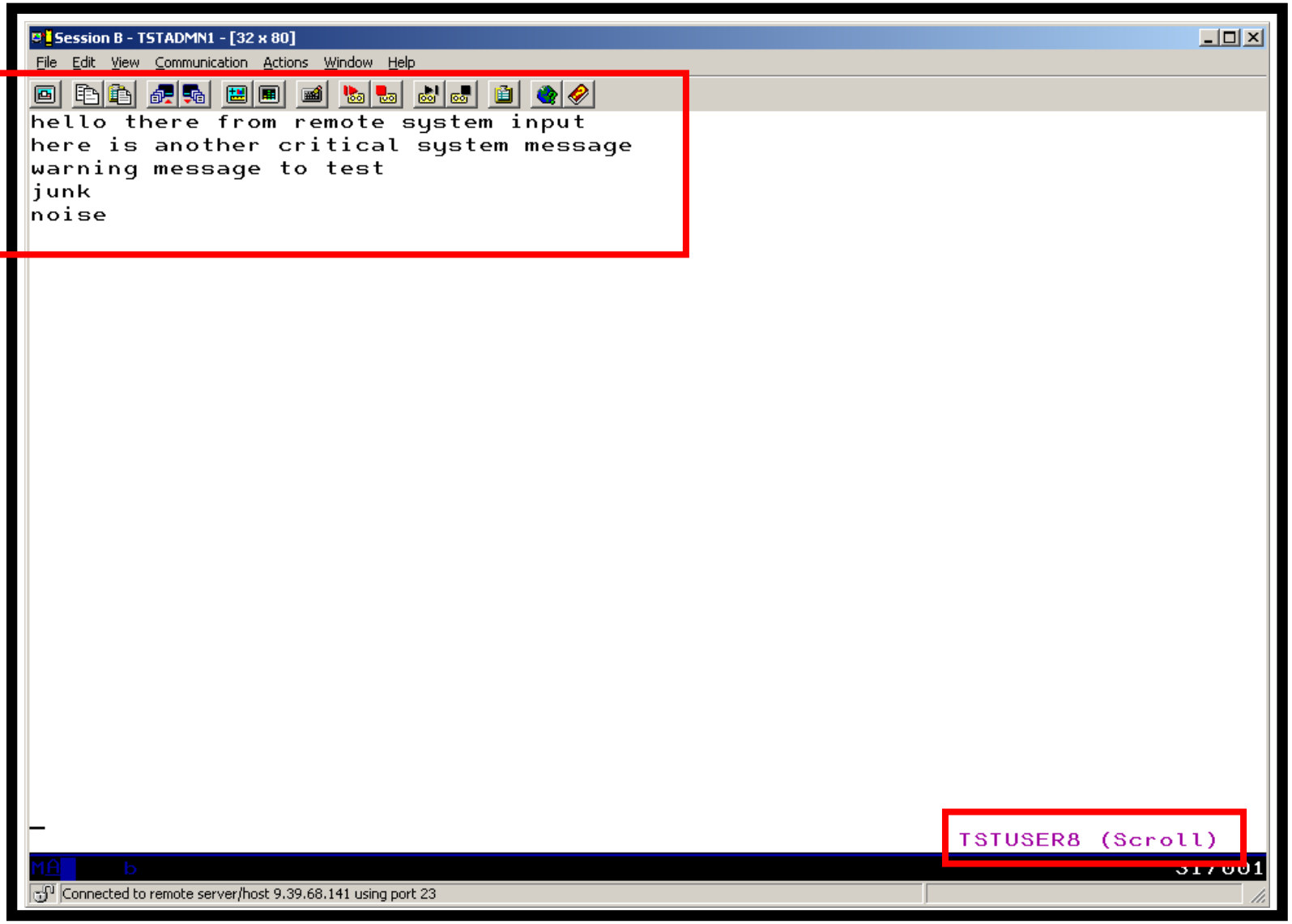
- **From an authorized z/VM user ID, view the consoles of the owning user IDs:**

```
gomcmd opmgrm1 viewcon user(tstadmn8)
gomcmd opmgrm1 viewcon user(tstuser8)
```









Scenario 7: How Do You Do That?

Console rule and action in Operations Manager:

```
*
DEFRULE NAME(TESTEX),+
  MATCH(*HELLO*),+
  MCOL(001:030),+
  ACTION(TESTEX),+
  EXGROUP(TSTUSERS)
*
DEFACTN NAME(TESTEX),+
  COMMAND(CP MSG TSTADMN1 HELLO BACK FROM &U.),+
  OUTPUT(LOG),+
  ENV(LVM)
```

Scenario 7: How Do You Do That?

Set up TCP/IP listener for test data and define group of consoles:

```
*
DEFTCPA NAME(TESTDATA),+
  TCPUSER(TCPIP),+
  TCPAPPL(GOMRSIF),+
  TCPADDR(000.000.000.000),+
  TCPPORT(63000)
*
DEFGROUP NAME(TSTUSERS),+
  USER(TSTUSER*)
```

Update TCP/IP configuration to allow Operations Manager to listen on the specified port

Scenario 8:
Process Linux Syslog Data as a Console

- **Route syslog data from a Linux guest to Operations Manager for z/VM**
  - Supports syslogd, syslog-ng, rsyslog
  - syslog-ng and rsyslog include hostname or IP address in message
- **Treat it as the console of a “fake” user ID**
- **Trigger rules and actions based on syslog data**
- **View the “console” containing syslog data**
- **Option to create one console per syslog or combine multiple syslogs into one console**

Scenario 8: Detailed Steps

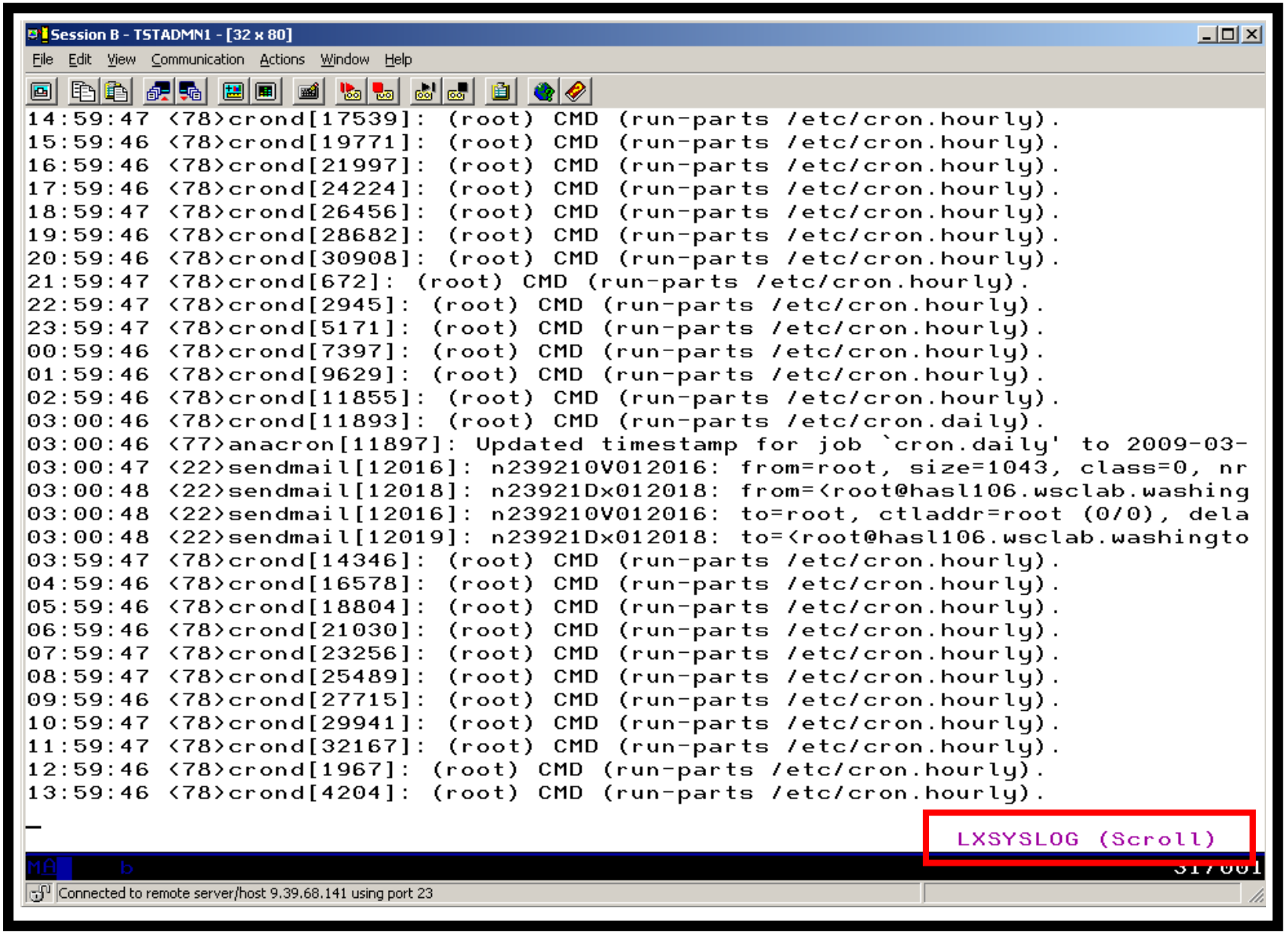
- **From an authorized z/VM user ID, view any syslog data already received**

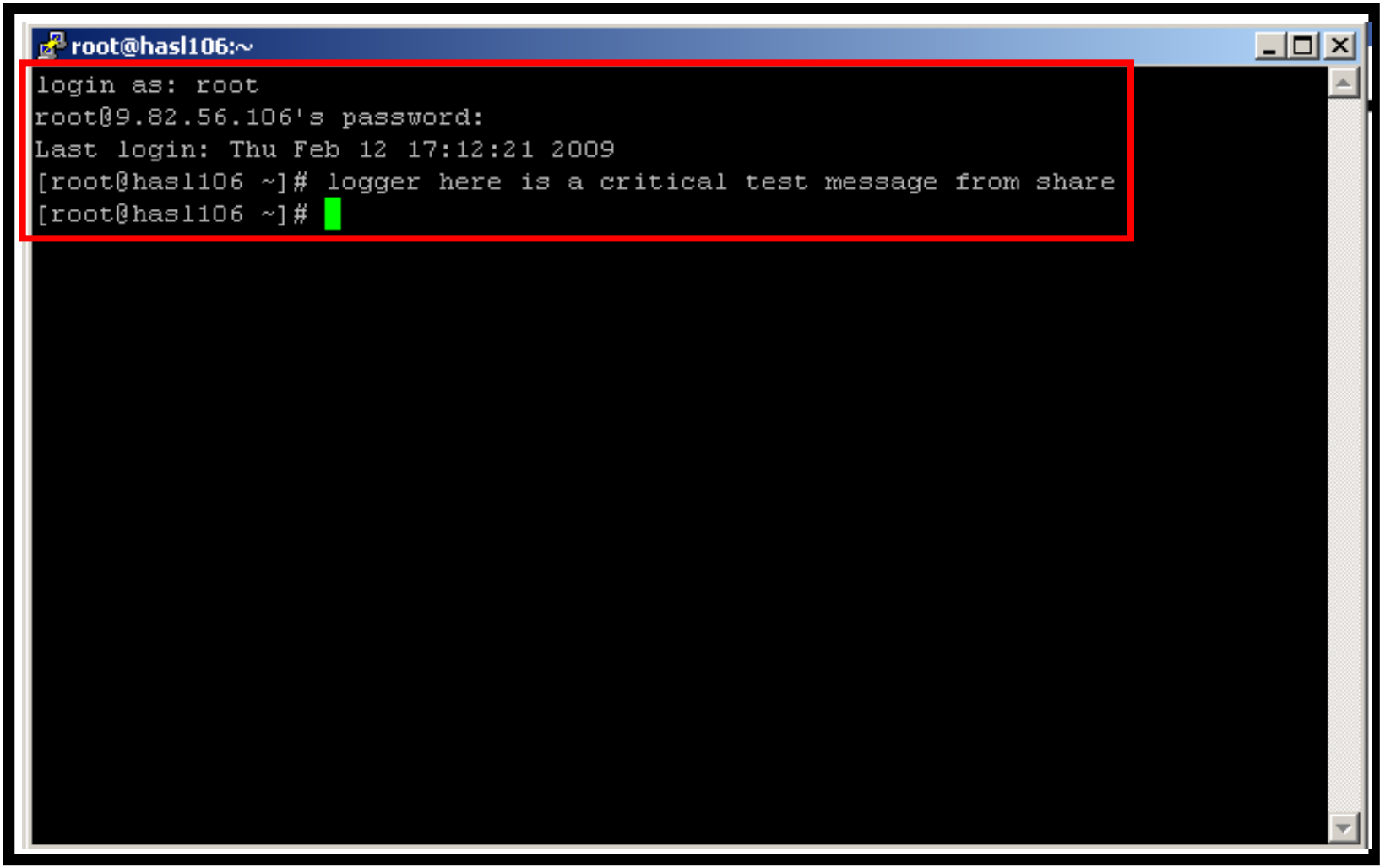
```
gomcmd opmgrm1 viewcon user(lxsyslog)
```

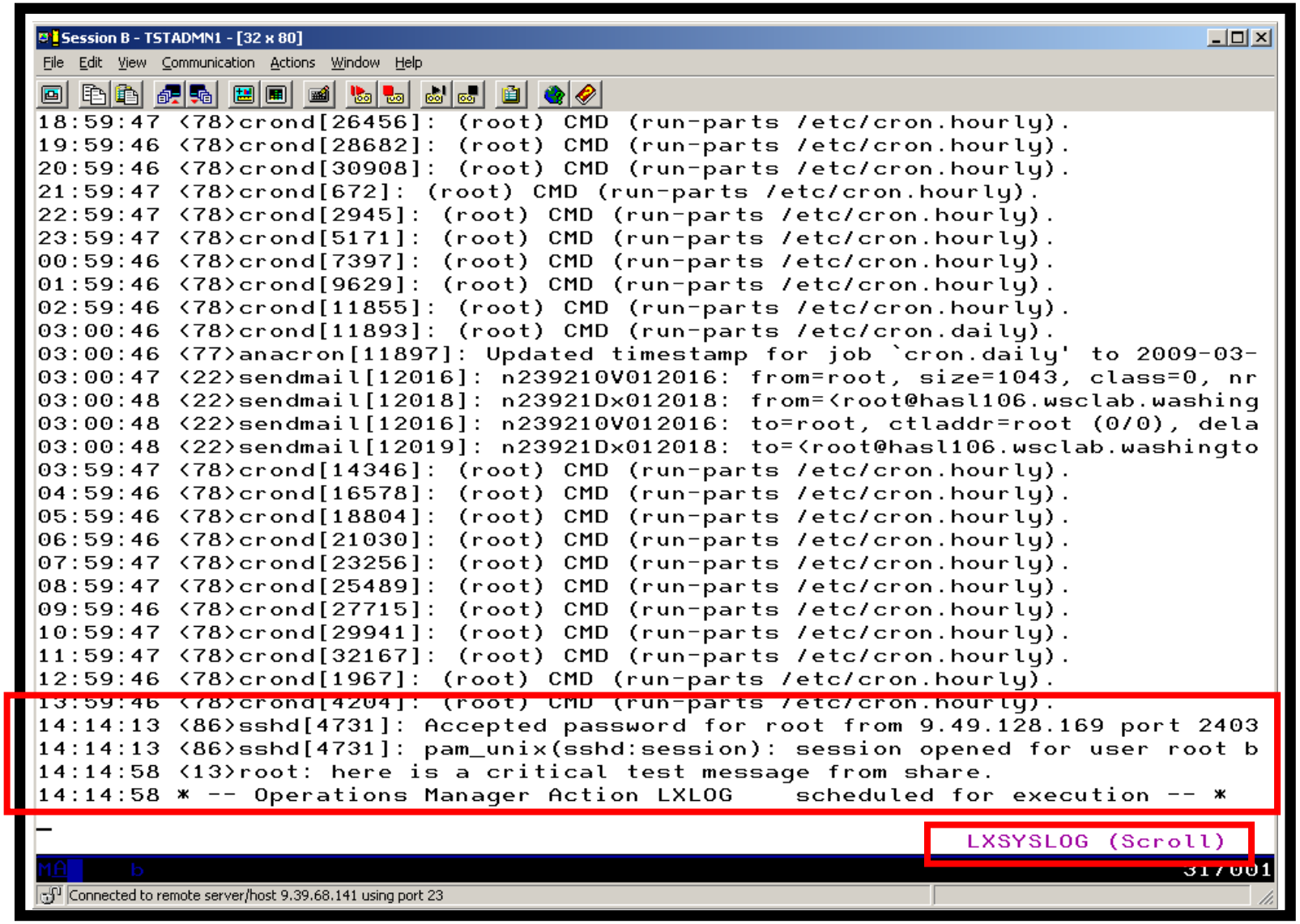
- **Use PUTTY to connect to a Linux guest**
- **Login as root and issue the command**

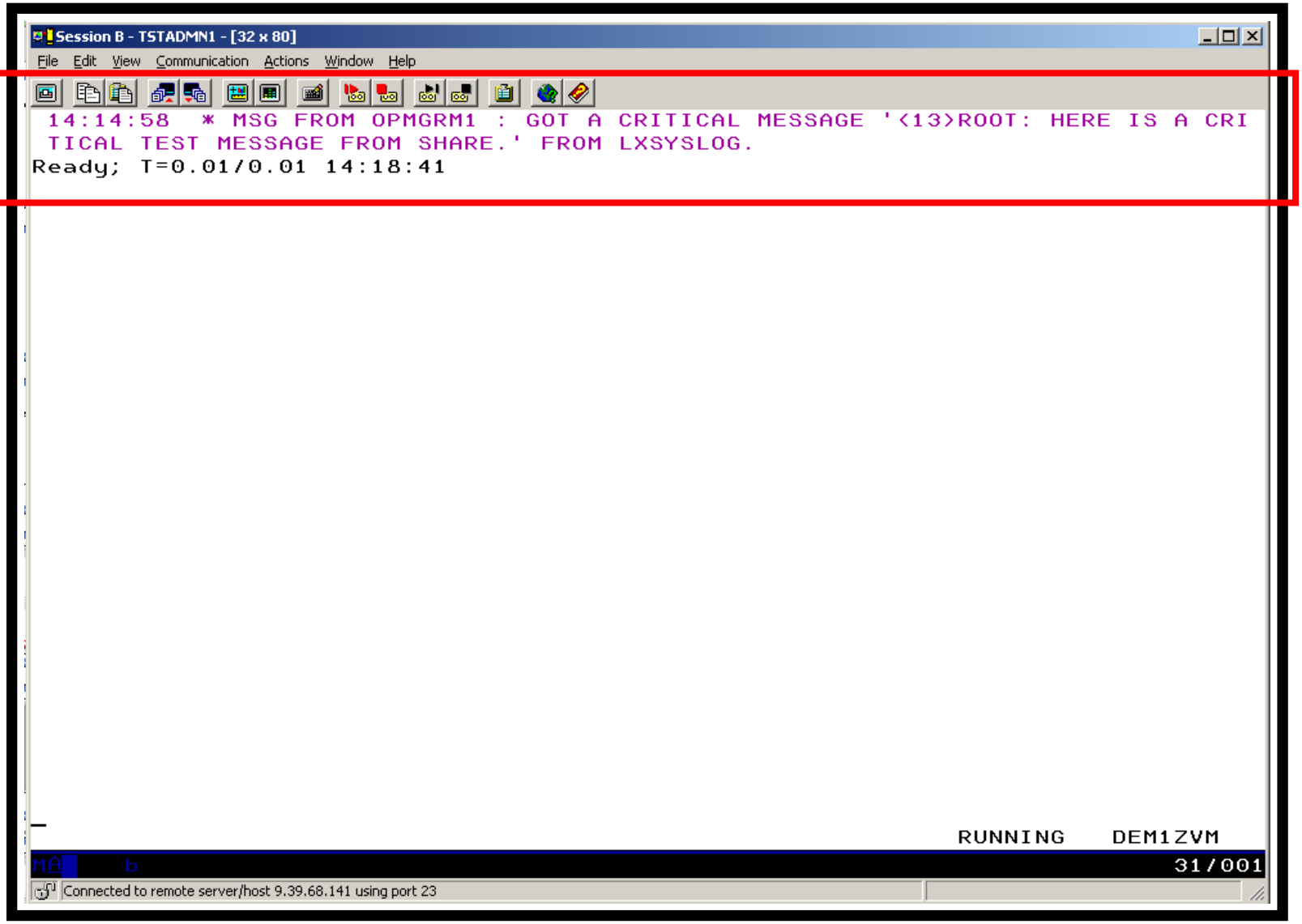
```
logger here is a critical test message from SHARE
```

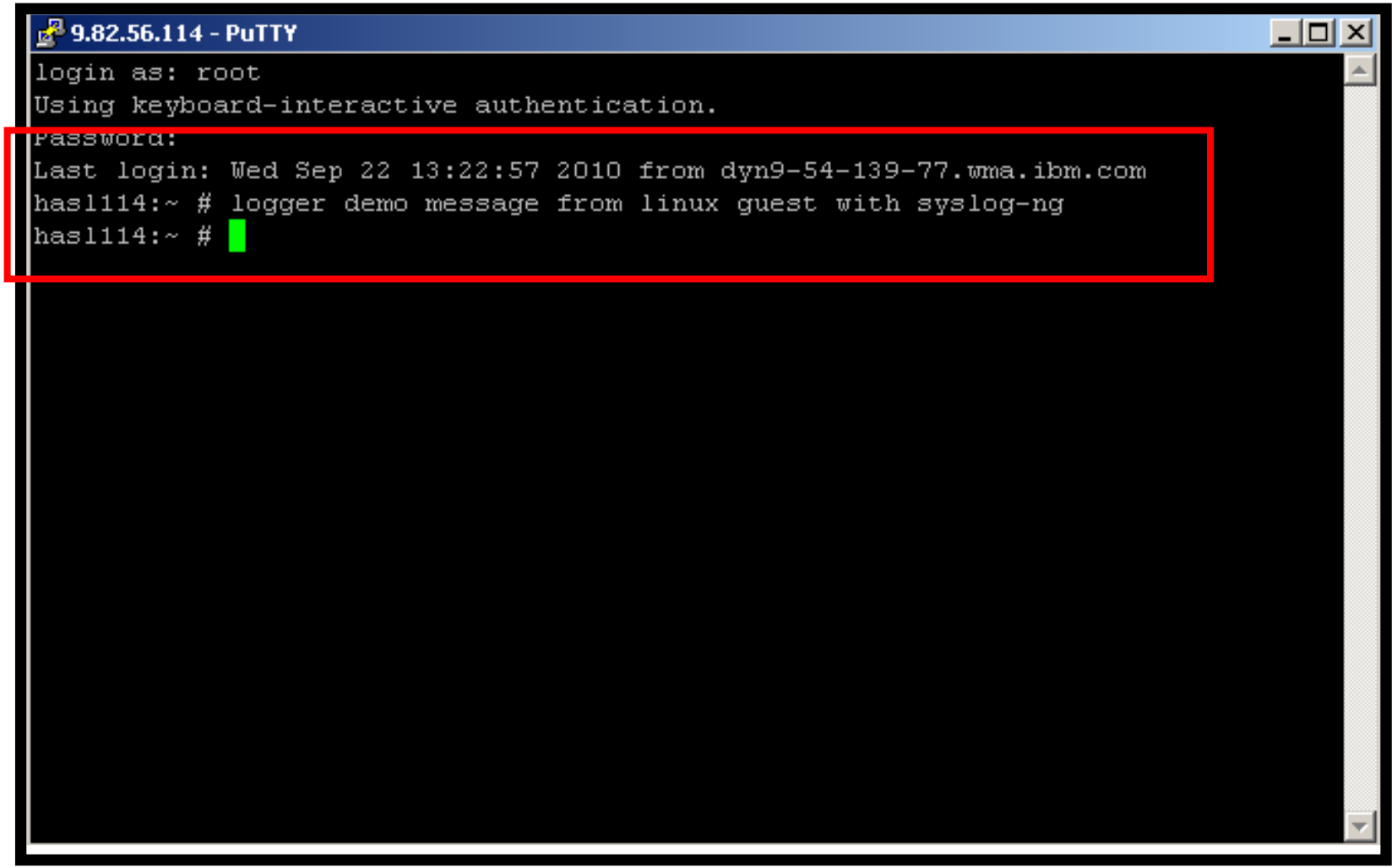
- **Return to the VIEWCON session**
  - See the message in the syslog “console”
  - Using syslog, so no hostname or IP address
- **Repeat from a different Linux guest that uses syslog-ng**

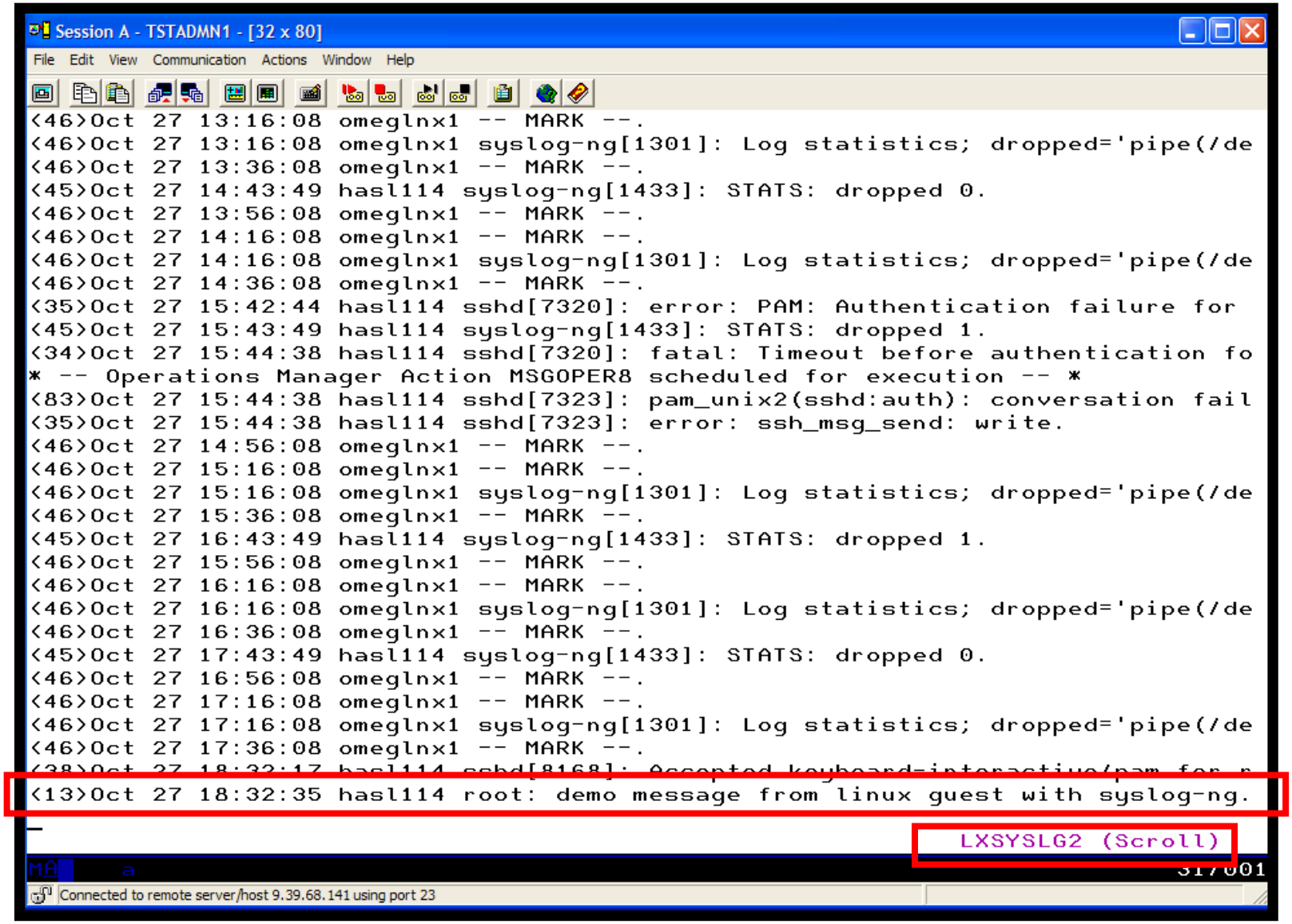


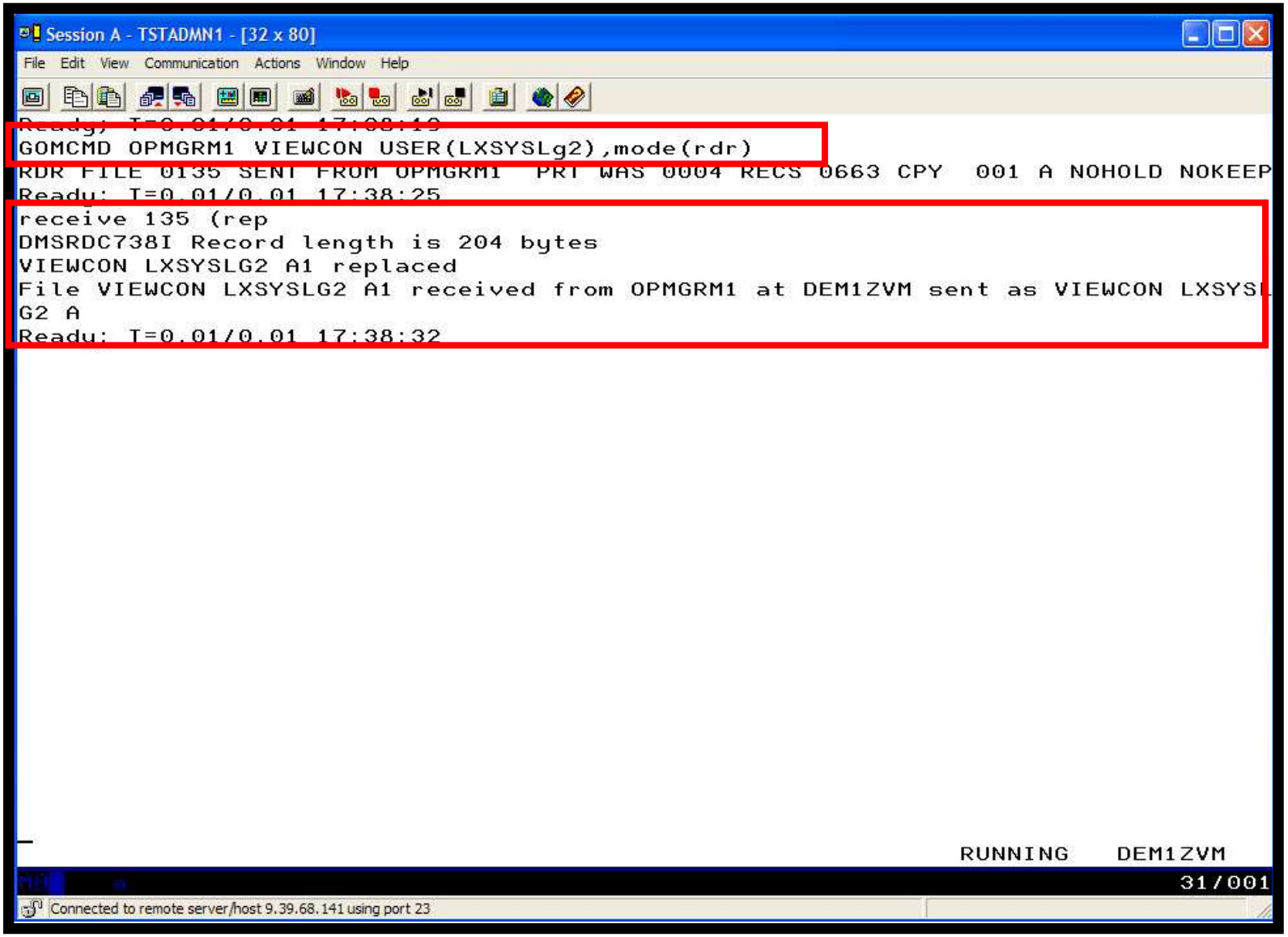


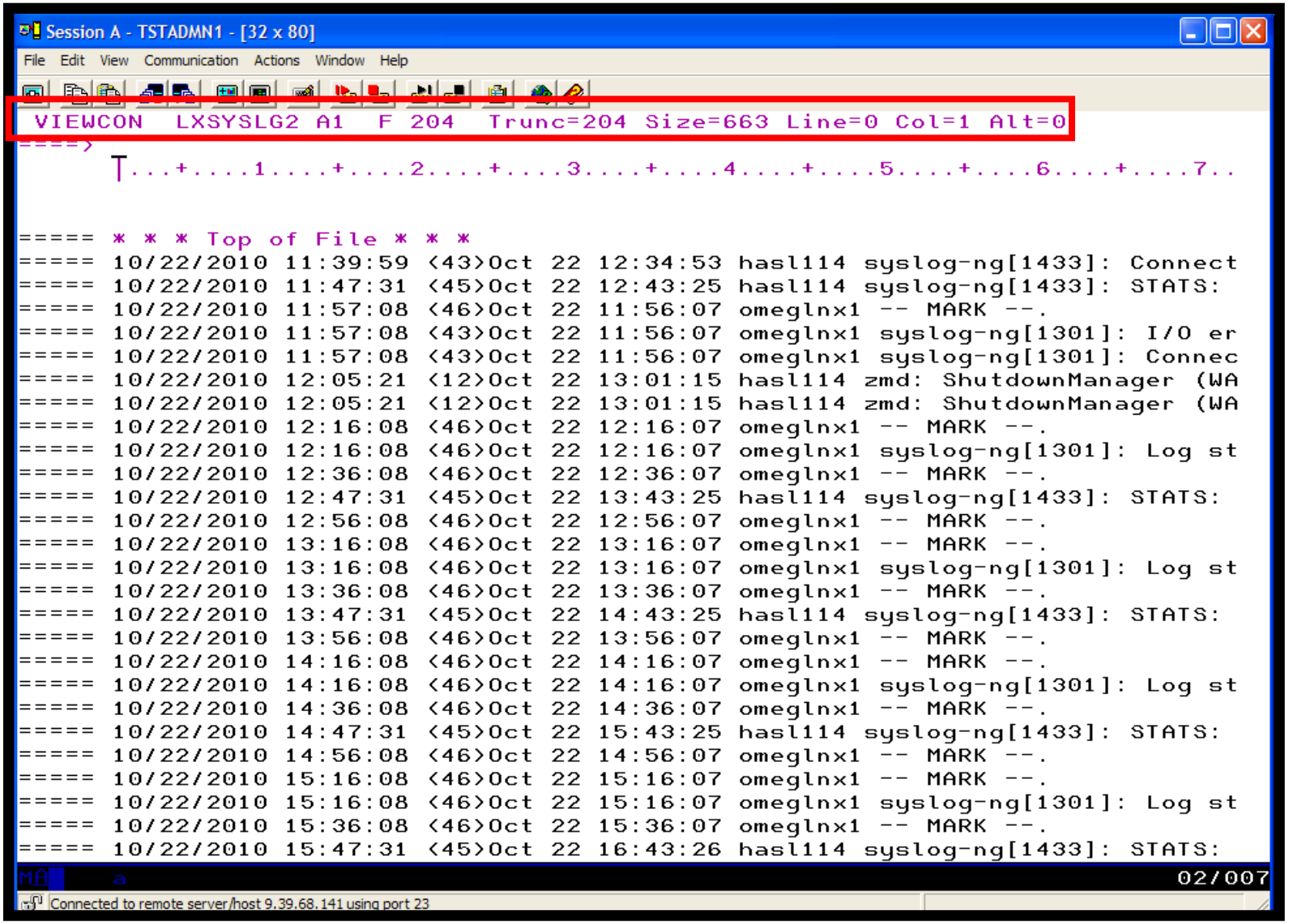












Scenario 8: How Do You Do That?

Console rule and action in Operations Manager:

```
*
DEFRULE NAME(LXLOG),+
  MATCH(*critical test message*),+
  ACTION(LXLOG),+
  USER(LXSYSLOG)
*
DEFACTN NAME(LXLOG),+
  COMMAND(CP MSG TSTADMN1 Got a critical message '&T' from &U.),+
  OUTPUT(LOG),+
  ENV(LVM)
```

Scenario 8: How Do You Do That?

- **Set up TCP/IP listener for syslog data**

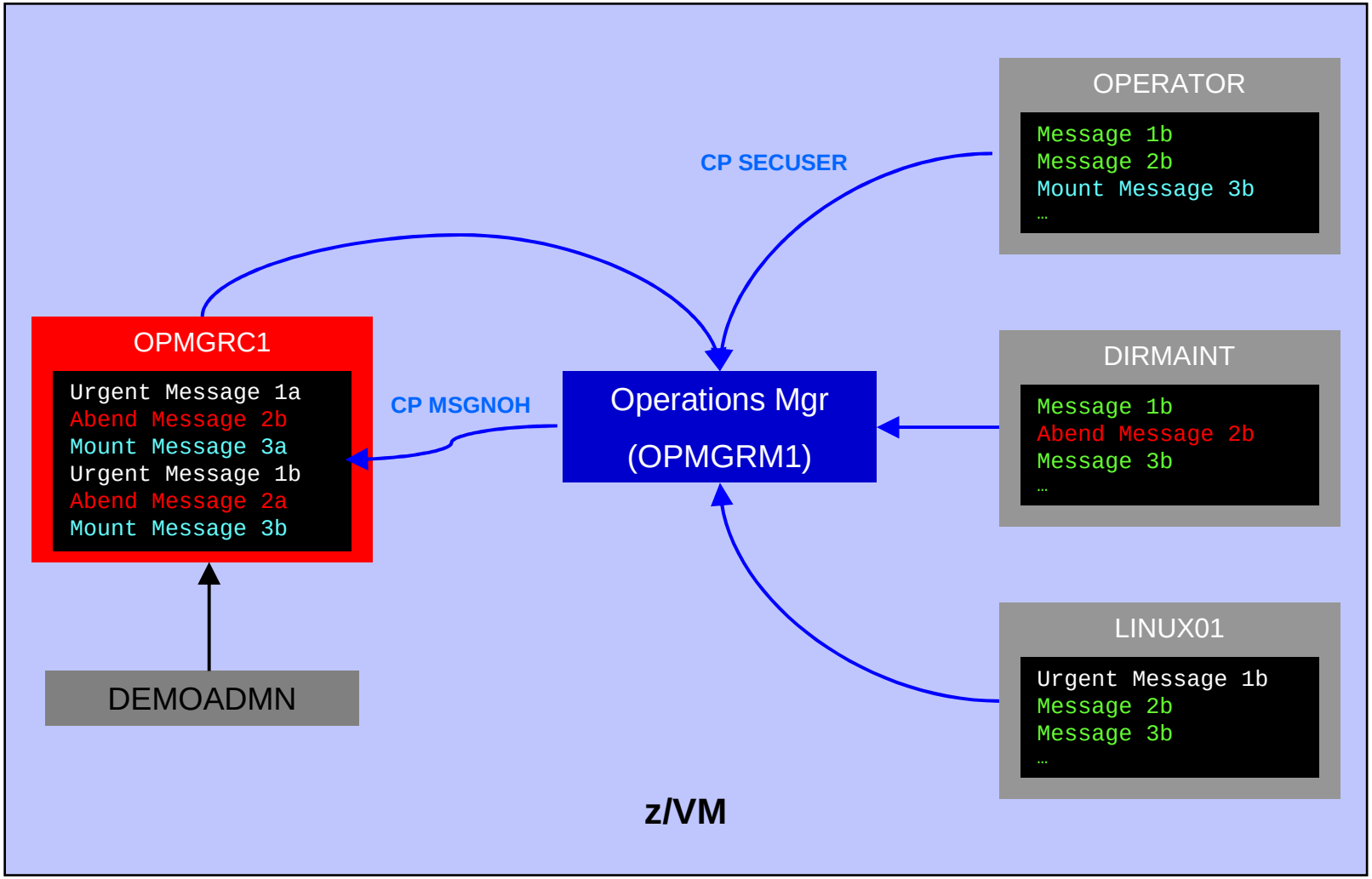
```
*
DEFTCPA NAME(LNXSYSLG),+
  TCPUSER(TCPIP),+
  TCPAPPL(GOMRSYL),+
  TCPADDR(000.000.000.000),+
  TCPPORT(00514),+
  PARM(LXSYSLOG03330417UTF8)
*
DEFTCPA NAME(LNXSYSL2),+
  TCPUSER(TCPIP),+
  TCPAPPL(GOMRSYL),+
  TCPADDR(000.000.000.000),+
  TCPPORT(00515),+
  PARM(LXSYSLG203330417UTF8)
```

- **Update TCP/IP configuration to allow Operations Manager to listen for UDP traffic on the specified port(s)**
  - Ports 514 and 515 used here
- **Update the Linux guest to send its syslog data to the IP address and port of your z/VM system**

Scenario 9:
Create a Central Operations Console on One z/VM System

- **Use Operations Manager to watch for error, warning, fatal messages on service machine consoles**
  - DIRMAINT, TCP/IP, RACF, etc.
  - Linux guests
  - Linux syslog
- **Route these messages to a central operations console**
- **Operations staff watches operations console for signs of trouble**
  - View individual service machine consoles for more details when needed

Creating a Central Console on One z/VM System



Scenario 9: Detailed Steps

- **From an authorized z/VM user ID, put “abend”, “fatal”, and error messages on DIRMAINT console**

```
msgnoh dirmaint this is a test abend message
msgnoh dirmaint this is a fake fatal message
msgnoh dirmaint DMSxxxxxxxE here is a made-up CMS error msg
```

- **View the “Operations Console” to see the messages**

```
gomcmd opmgrm1 viewcon user(oper8)
```

- **Note the fatal message is red and abend message is highlighted and will be held when other messages come in**

Scenario 9: Detailed Steps

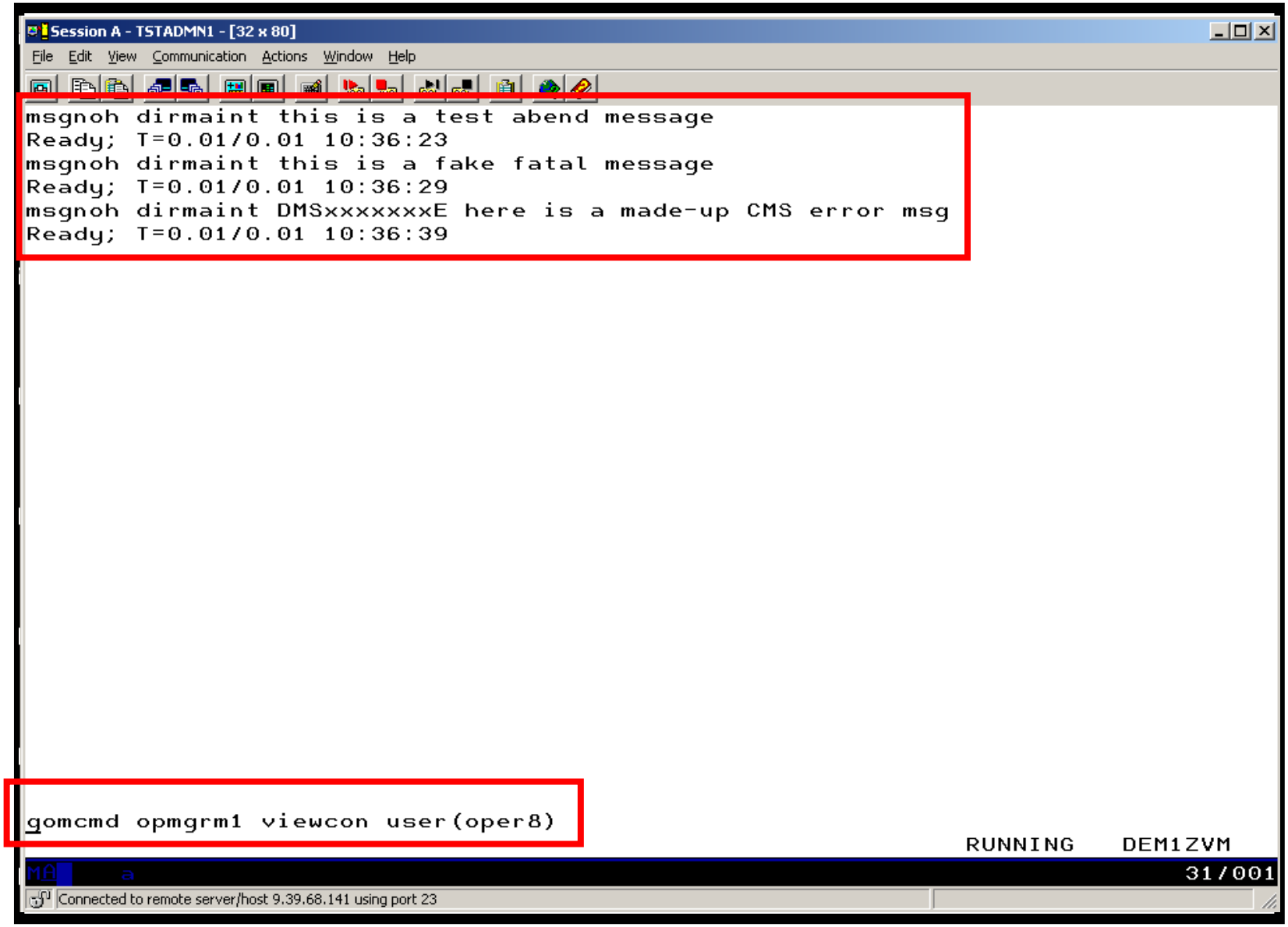
- **From another user ID, run an EXEC to send multiple messages to the Operations Console**

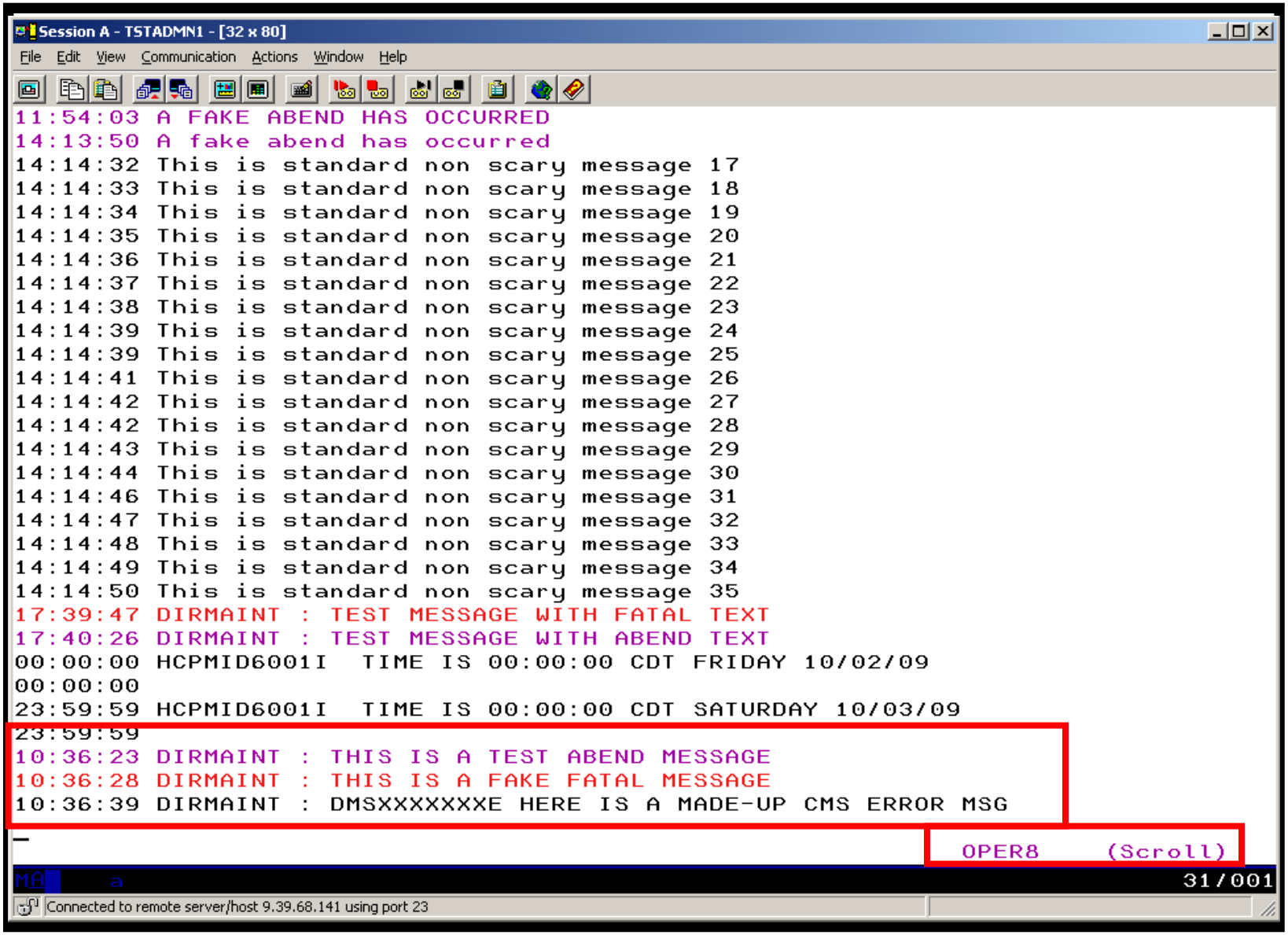
```
lotsmsgs
```

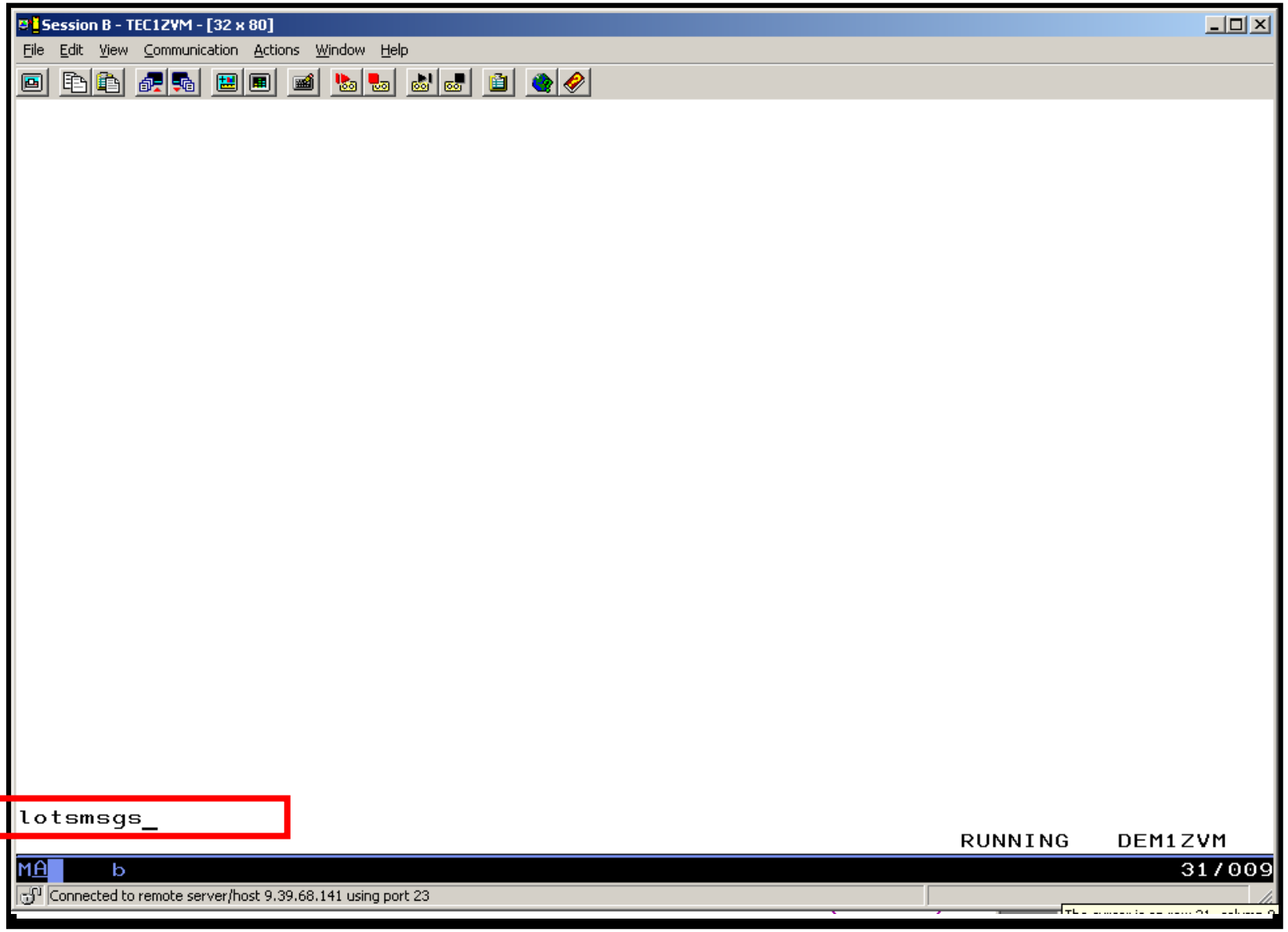
- **View the “Operations Console” to see the messages**

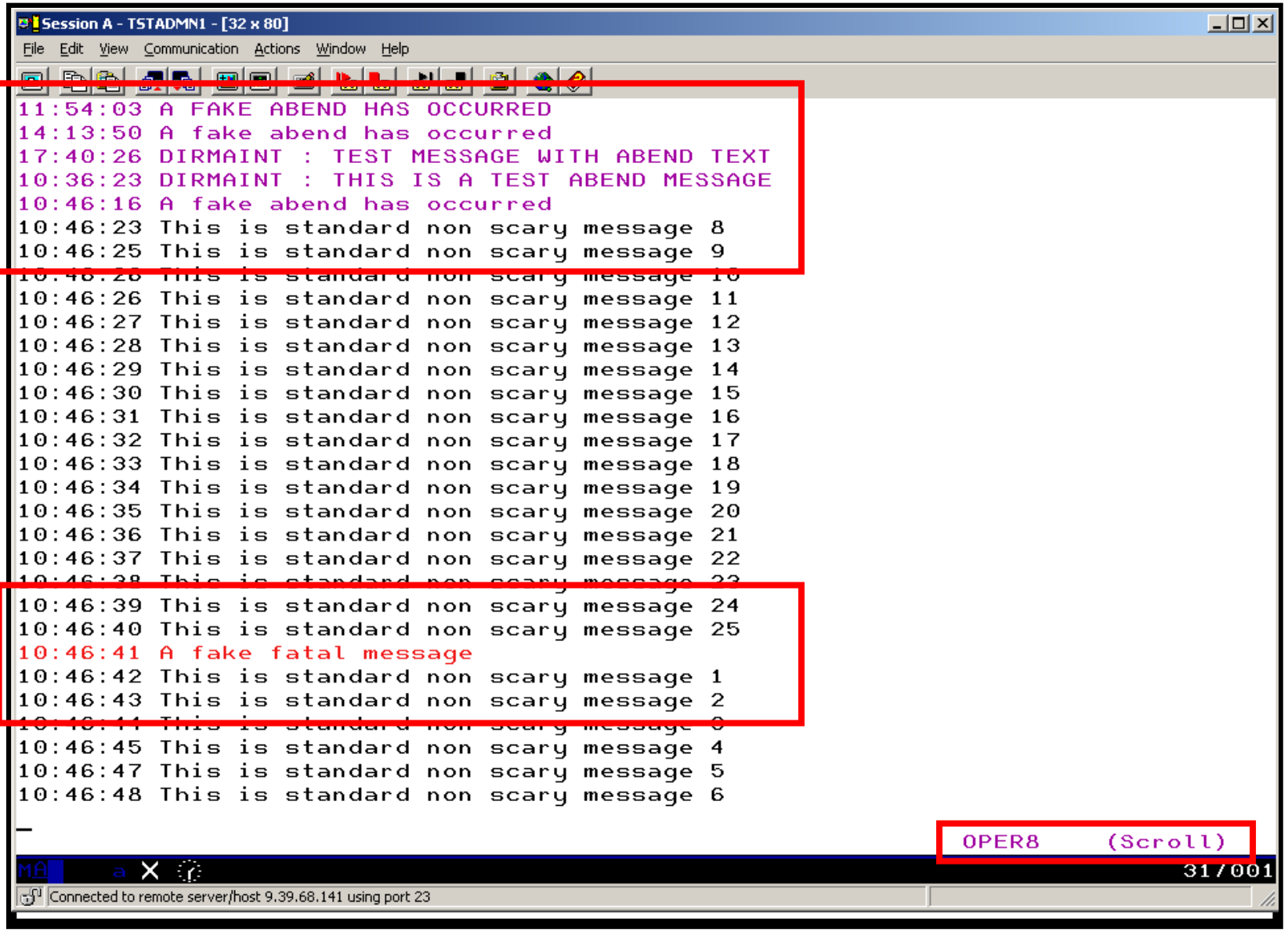
```
gomcmd opmgrm1 viewcon user(oper8)
```

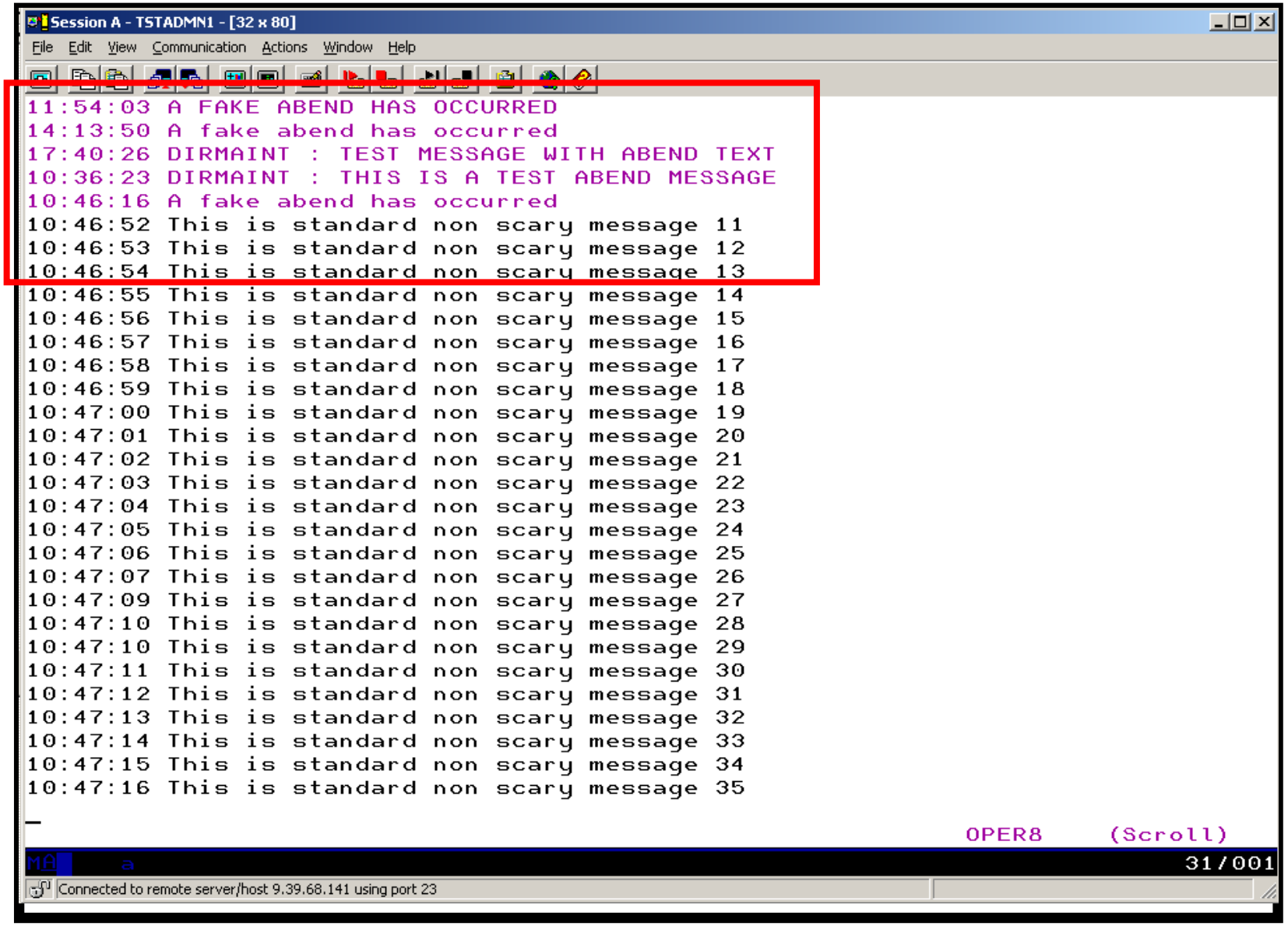
- **Watch the scrolling, held messages, etc.**











Scenario 9: How Do You Do That?

Console rules in Operations Manager:

```
*
DEFRULE NAME(ABEND),+
  MATCH(*abend*),+
  EXUSER(OPER8),+
  ACTION(MSGOPER8)
*
DEFRULE NAME(FATAL),+
  MATCH(*fatal*),+
  EXUSER(OPER8),+
  ACTION(MSGOPER8)
*
DEFRULE NAME(EMSGS),+
  MATCH(DMS*E),+
  MCOL(001:011),+
  EXUSER(OPER8),+
  ACTION(MSGOPER8)
```

Action in Operations Manager:

```
*
DEFACTN NAME(MSGOPER8),+
  COMMAND(CP MSGNOH OPER8 &U : &T),+
  OUTPUT(LOG),+
  ENV(LVM)
```

Scenario 9: How Do You Do That?

Console rules in Operations Manager:

```
*
DEFRULE NAME(ABENDHLT),+
  MATCH(*abend*),+
  USER(OPER8),+
  ACTION(HLTHOLD)
*
DEFRULE NAME(FATALRED),+
  MATCH(*fatal*),+
  USER(OPER8),+
  ACTION(RED)
```

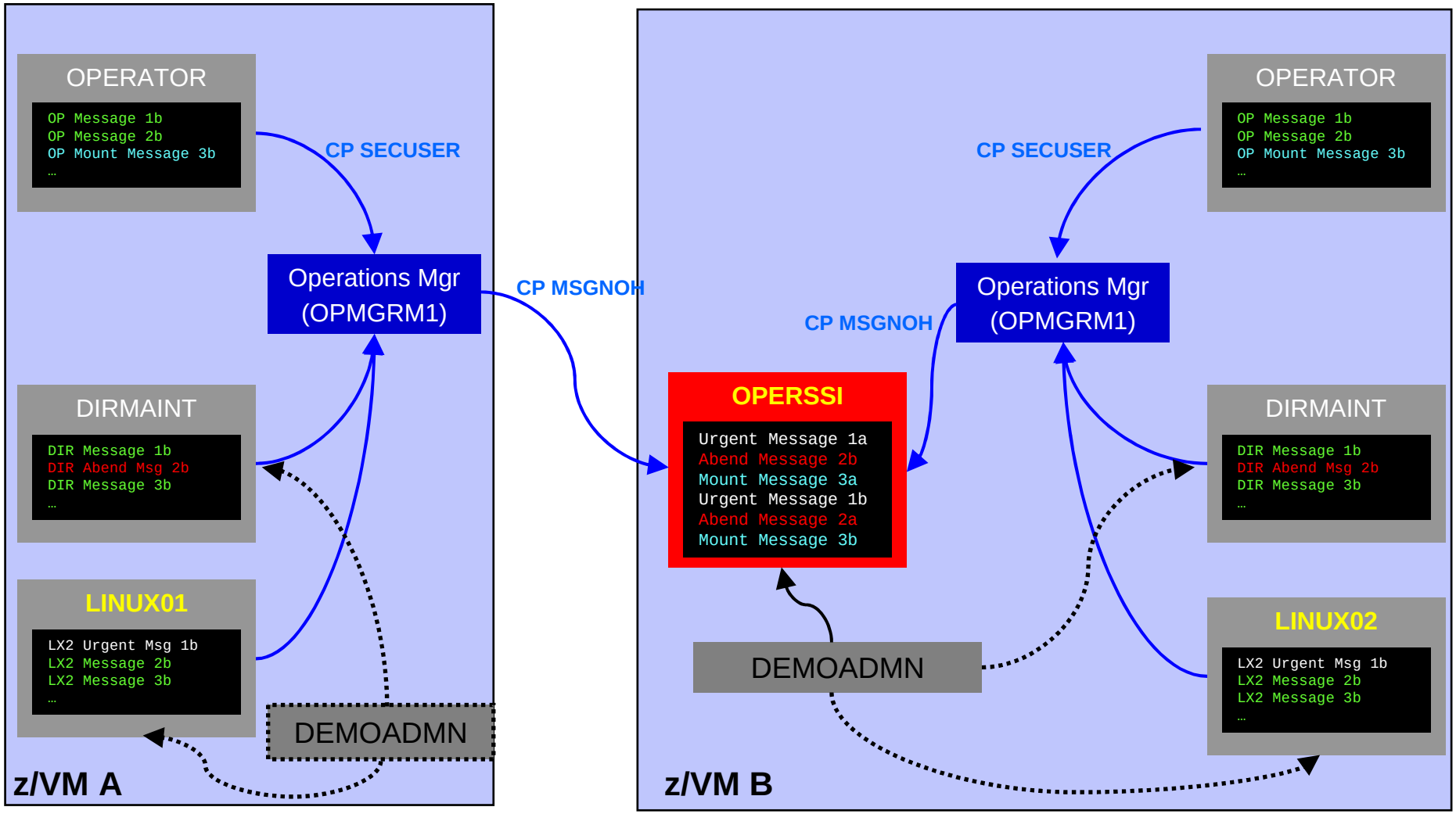
Actions in Operations Manager:

```
*
DEFACTN NAME(HLTHOLD),+
  INPUT(AHI,HLD)
*
DEFACTN NAME(HILITE),+
  INPUT(AHI)
*
DEFACTN NAME(RED),+
  INPUT(CRE)
```

Scenario 10a: Create a Central Operations Console across multiple z/VM systems **in an SSI cluster –** Includes relocation of Linux and CMS guests

- **Use Operations Manager to watch for error, warning, fatal messages on service machine consoles on one or more systems in an SSI cluster**
  - OPERATOR, DIRMAINT, TCPIP, RACF, etc.
  - Linux guests
  - Linux syslog
- **Route these messages to a central operations console on one of the z/VM systems**
- **Operations staff watches one operations console for signs of trouble across multiple z/VM systems**
  - View individual service machine consoles for more details when needed

Creating a Central Console Across Multiple Members of SSI Cluster



Single Configuration Users: LINUX01, LINUX02, OPERSSI, DEMOADMN
Multiconfiguration (IDENTITY) Users: OPERATOR, DIRMAINT, OPMGRM1

Scenario 10a: Detailed Steps

- **On System B (TEST7SSI), view the "Operations Console" (user ID OPERSSI)**

```
gomcmd opmgrm1 viewcon user(operssi)
```

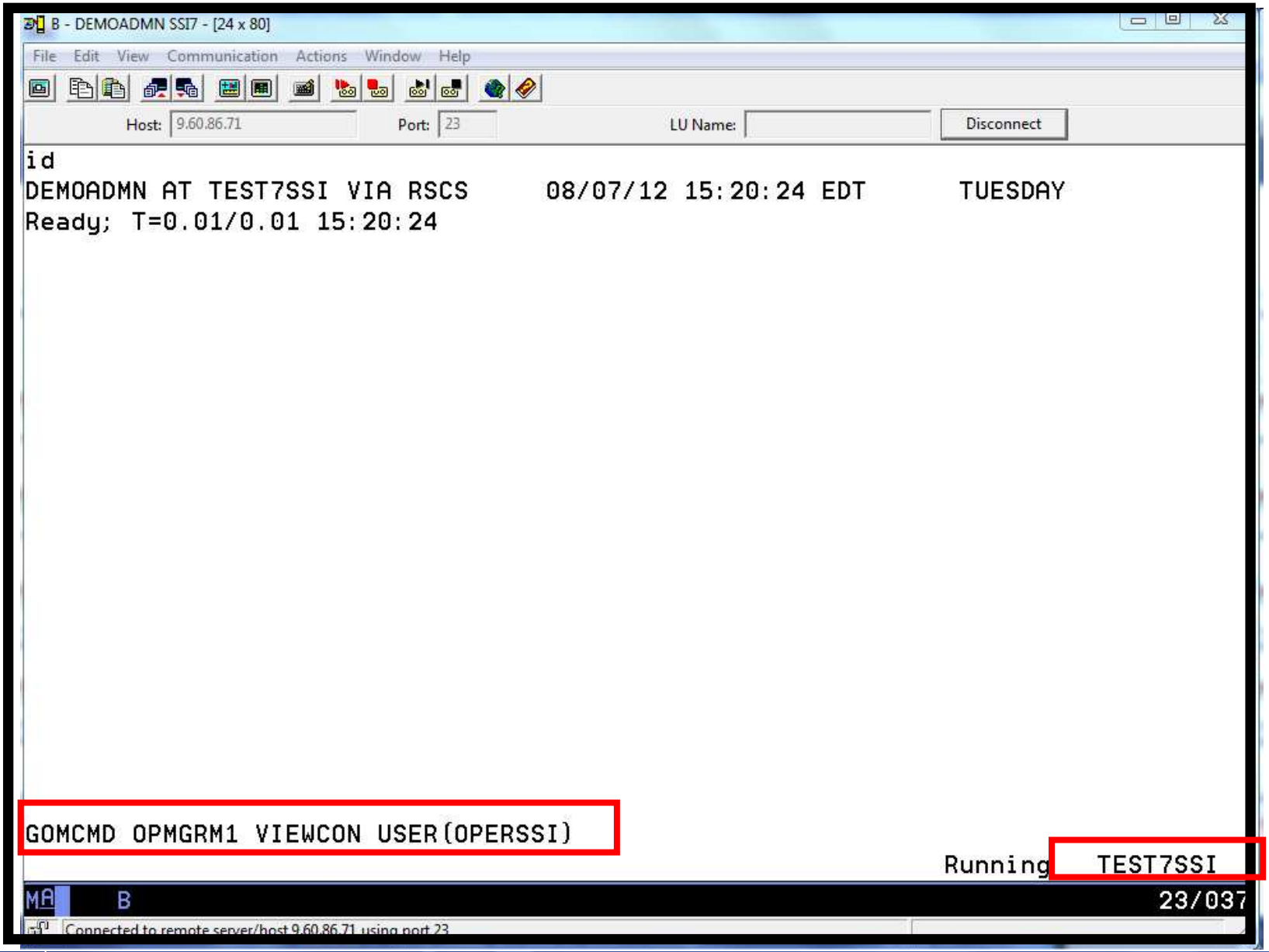
- **On System A (TEST7SSI), find a Linux guest running disconnected locally and relocate it**

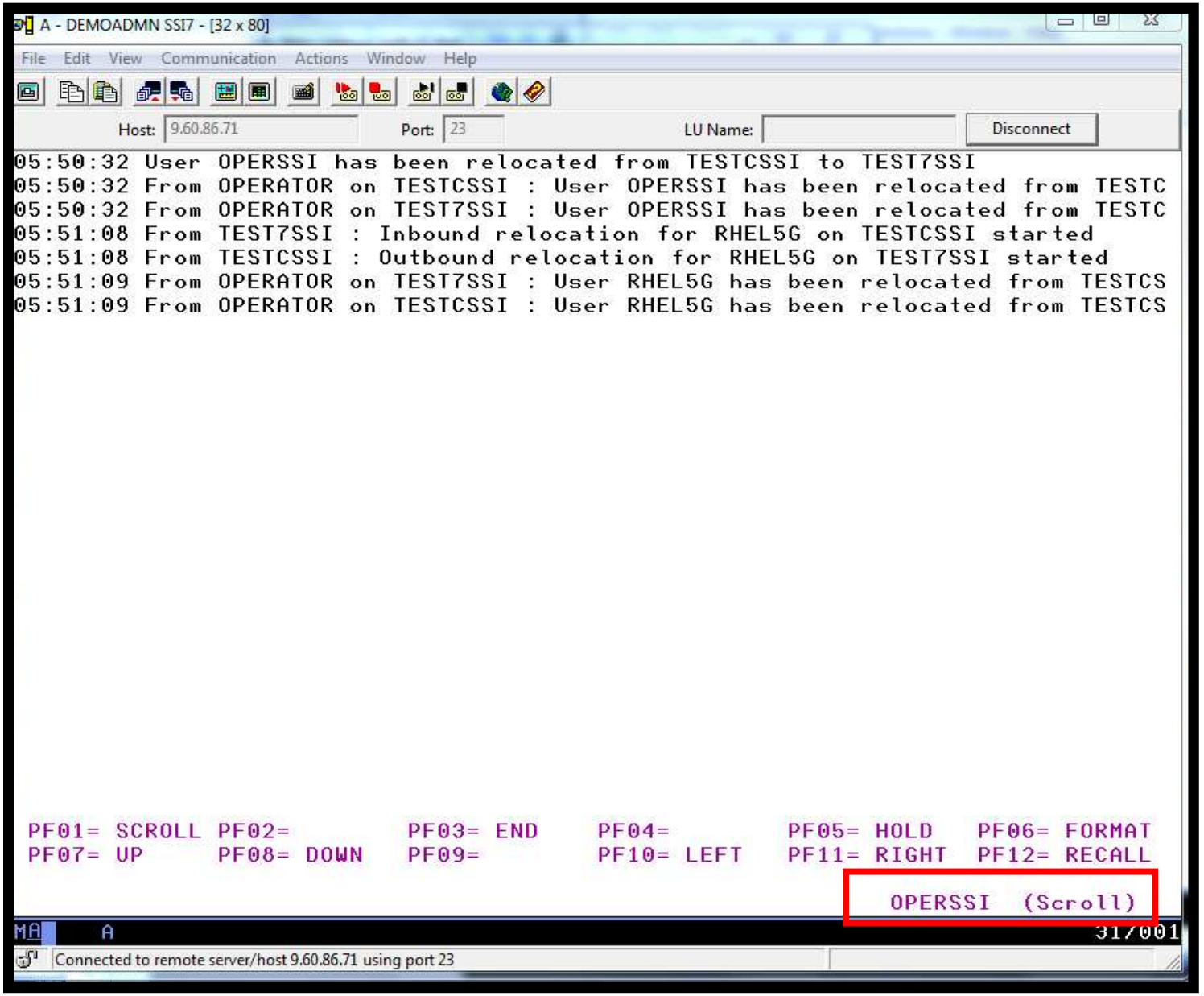
```
q names

VMRELOCATE MOVE USER RHEL5G TO TESTCSSI
```

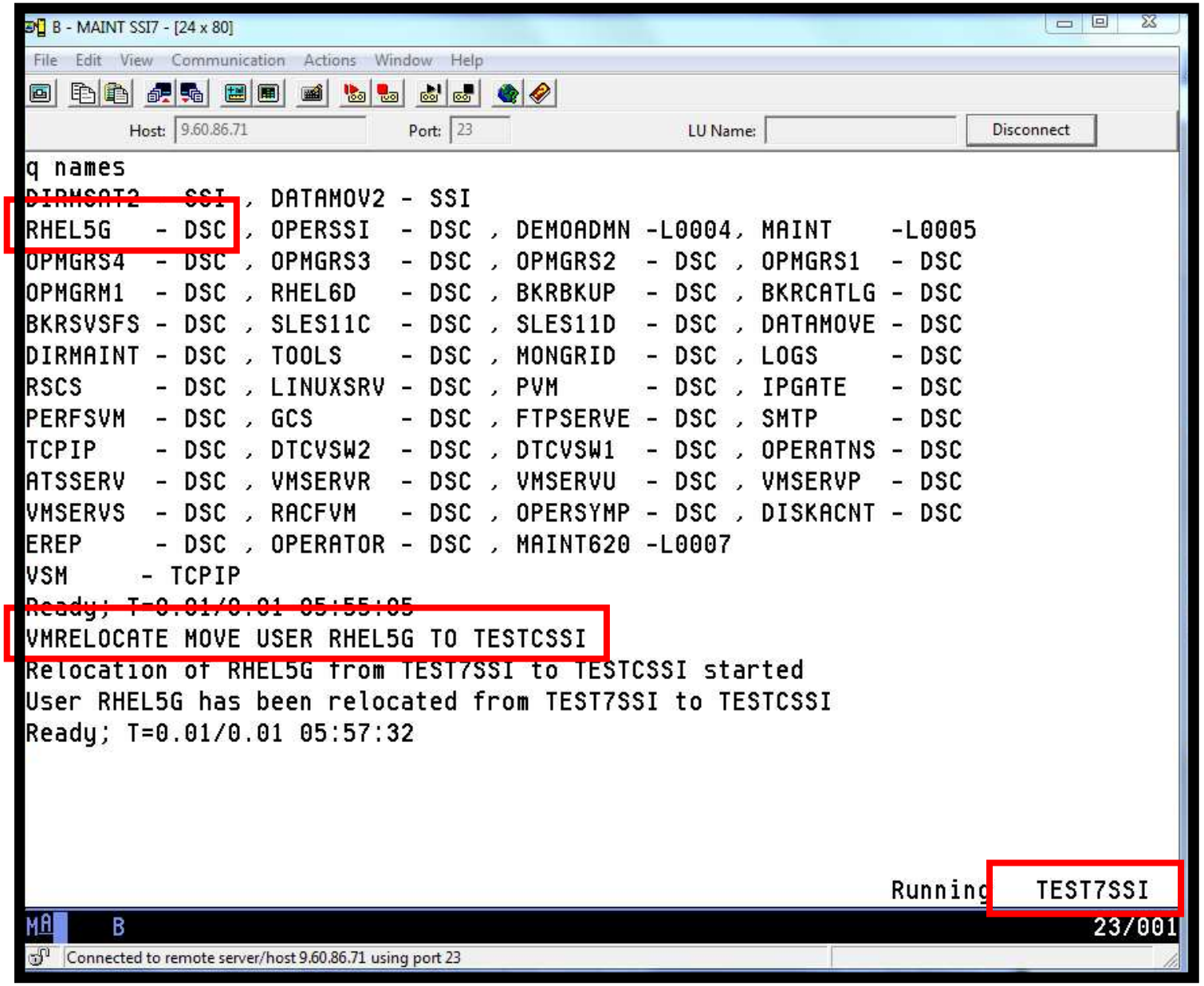
- **On System B (TEST7SSI), prepare for planned shutdown by relocating the central operations console (OPERSSI)**

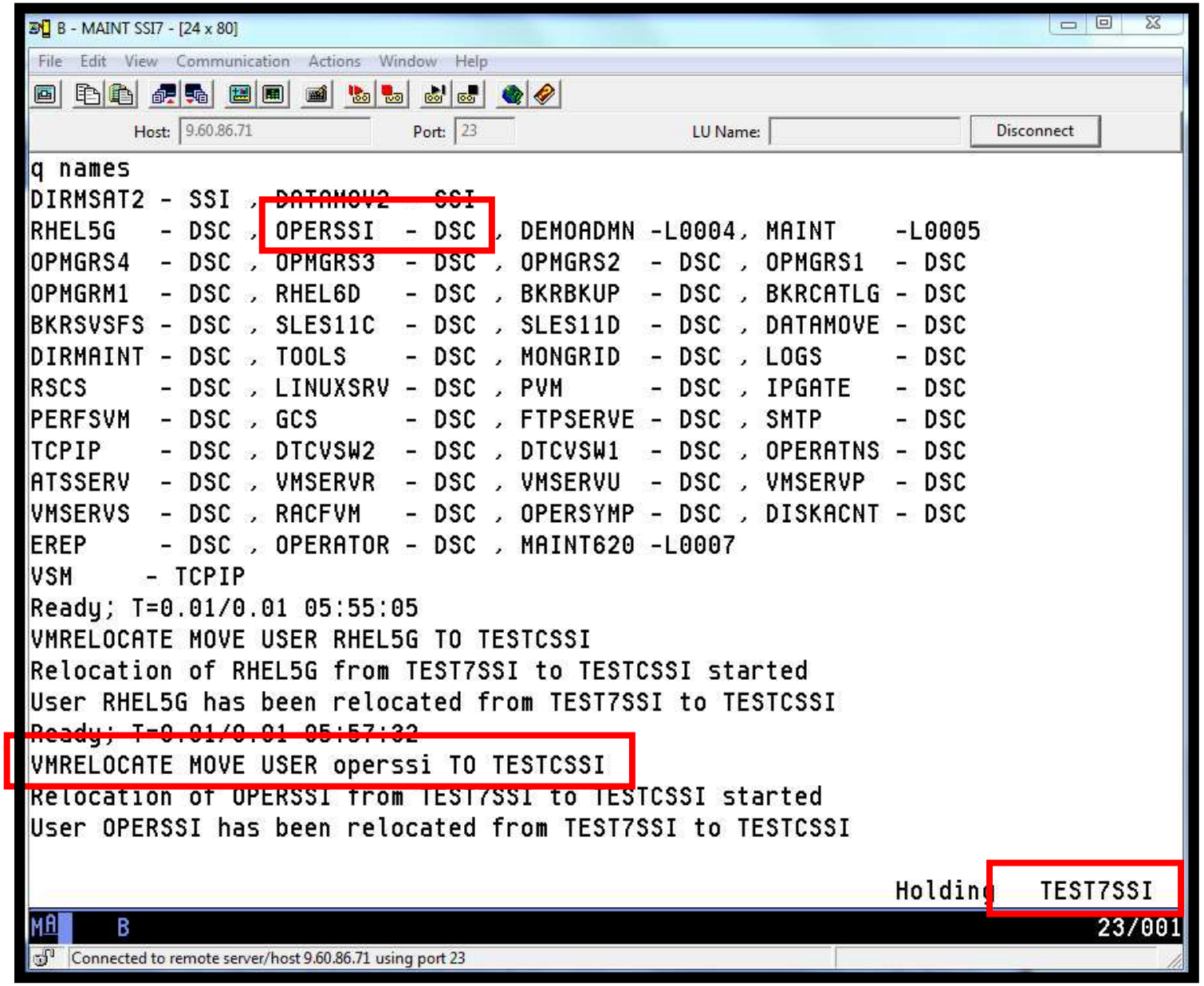
```
VMRELOCATE MOVE USER OPERSSI TO TESTCSSI
```

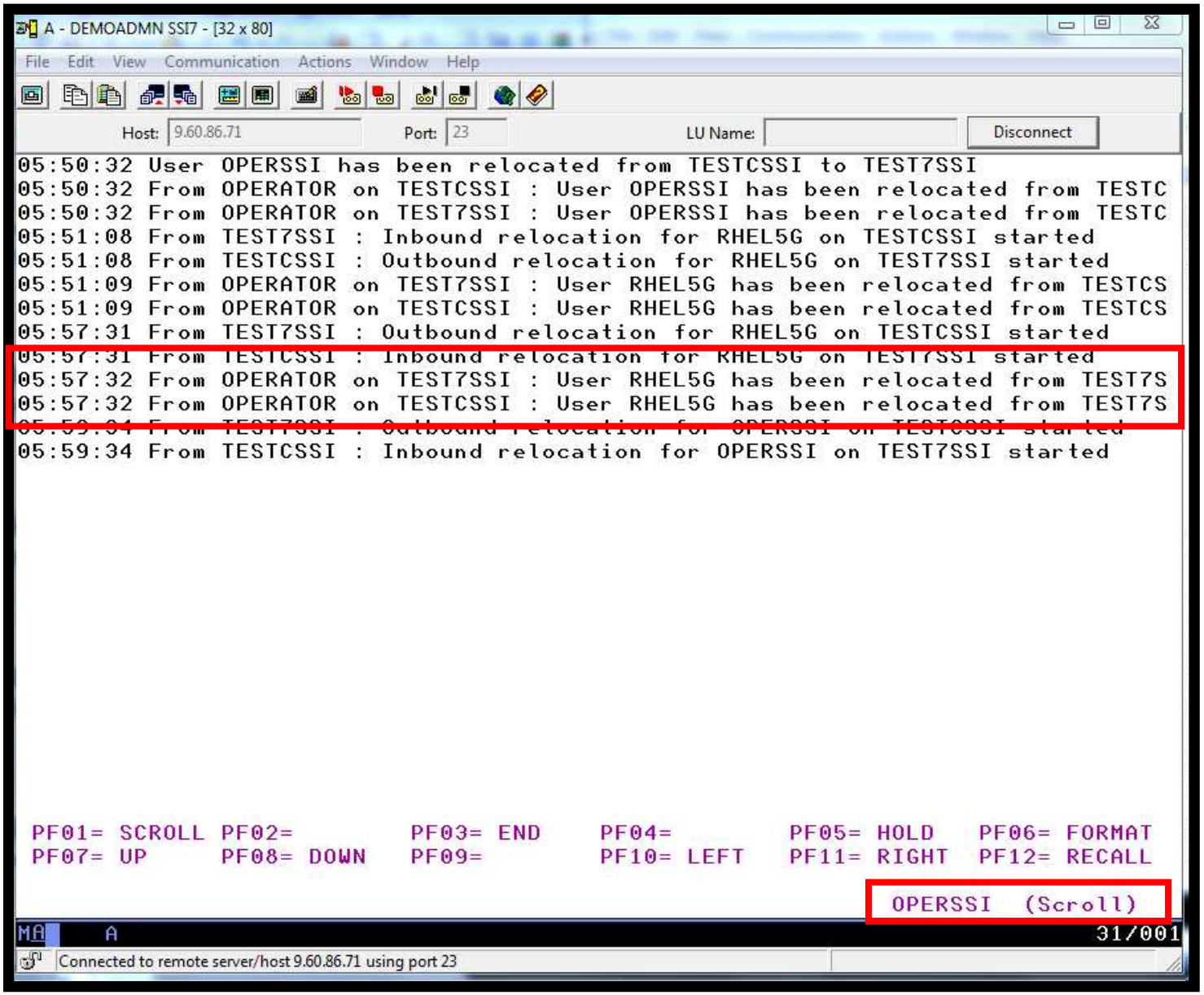
- **Note the messages received on OPERSSI on TEST7SSI from OPERATOR on both TESTCSSI and TEST7SSI indicating RHEL5G was relocated**
- **Note the message received on OPERSSI on TESTCSSI indicating OPERSSI has been relocated**

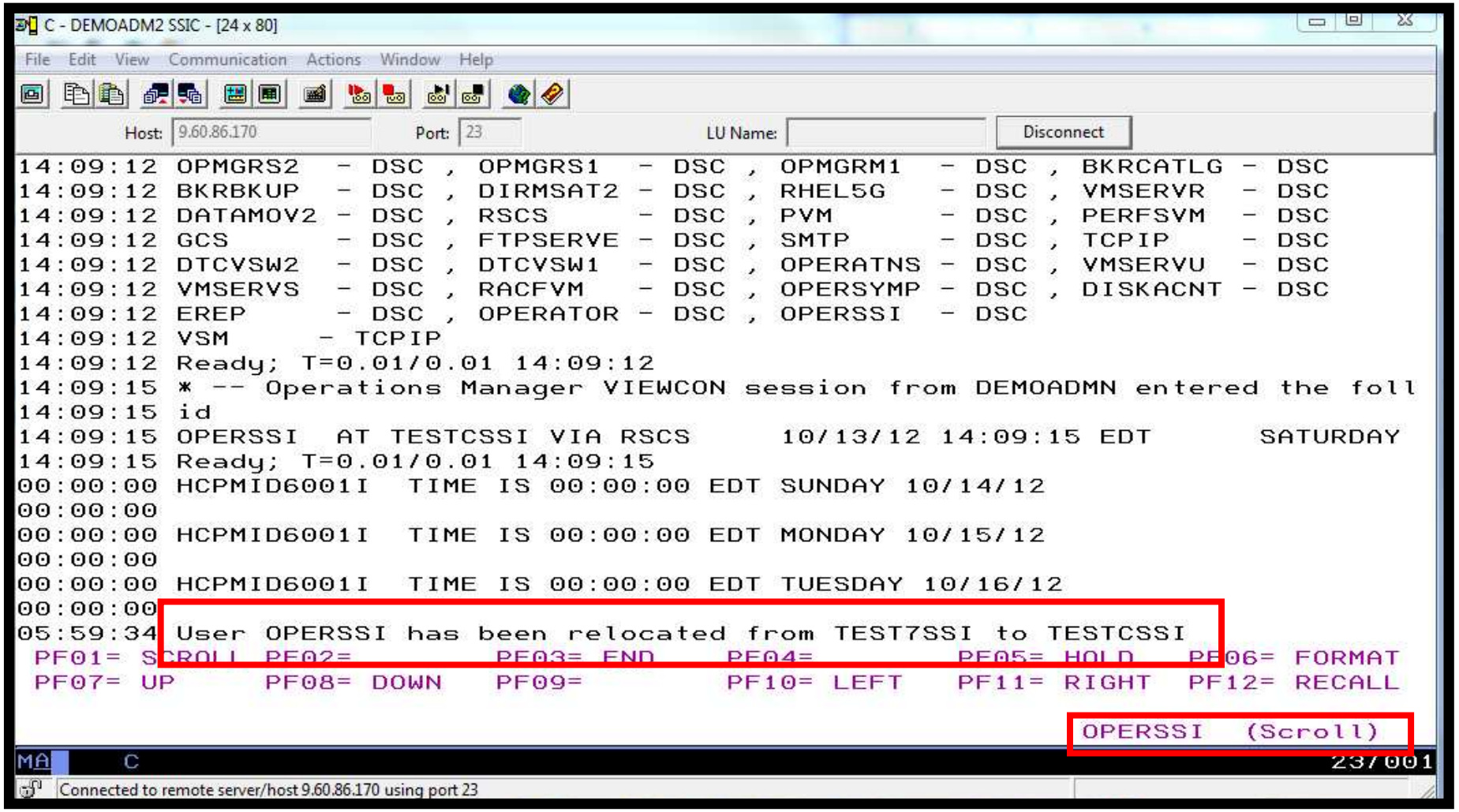












Scenario 10a: How Do You Do That?

Event monitor in Operations Manager:

```
*
*** Notify OPERSSI console when relocations started
DEFEMON NAME(RELOC),+
  TYPE(9,10),+
  ACTION(RELOC)
```

Action in Operations Manager:

```
*
DEFACTN NAME(RELOC),+
  COMMAND(EXEC MSG2OPER &u &3 &4 &5 junk),+
  ENV(LVM)
```

Scenario 10a: How Do You Do That?

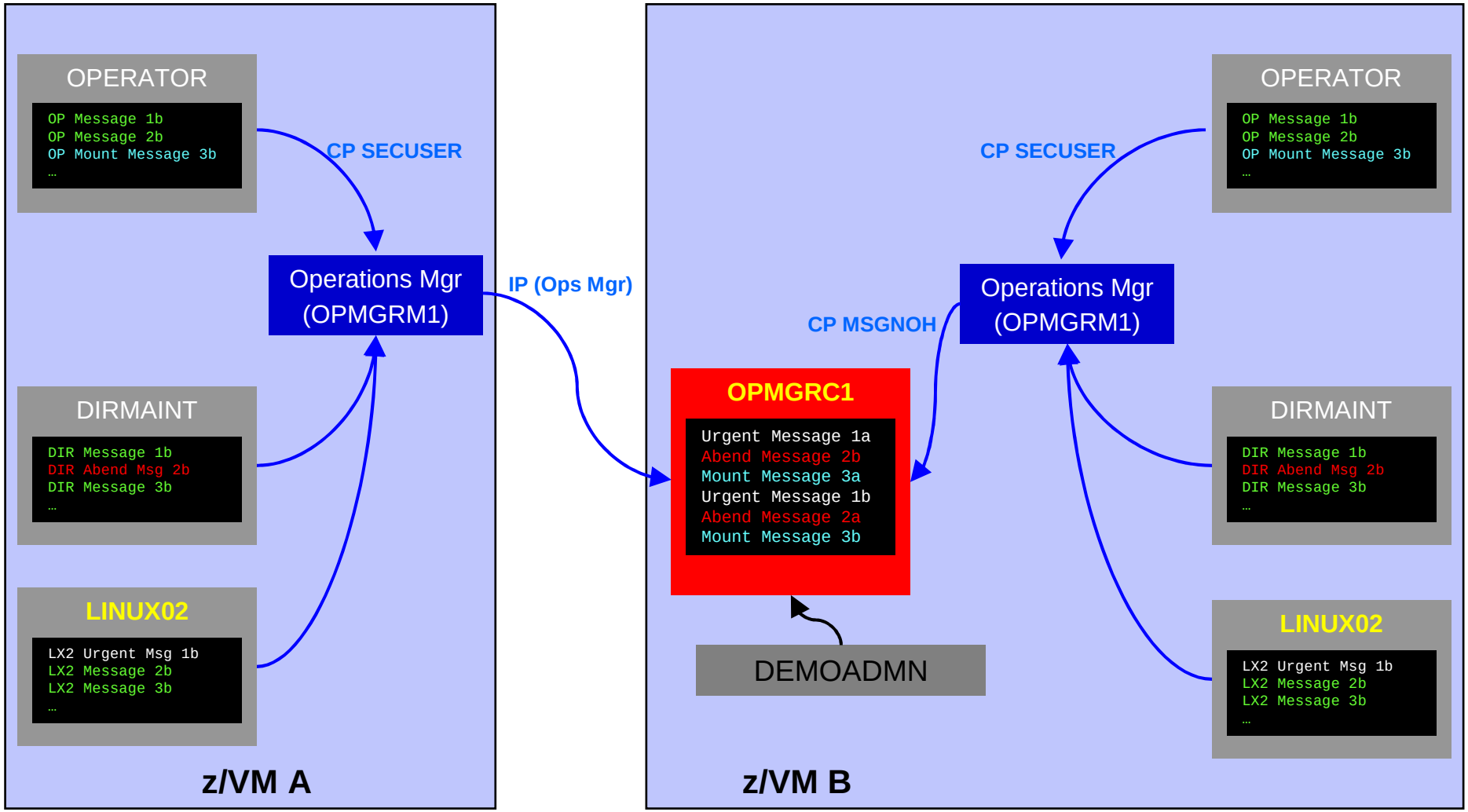
MSG2OPER EXEC (excerpts):

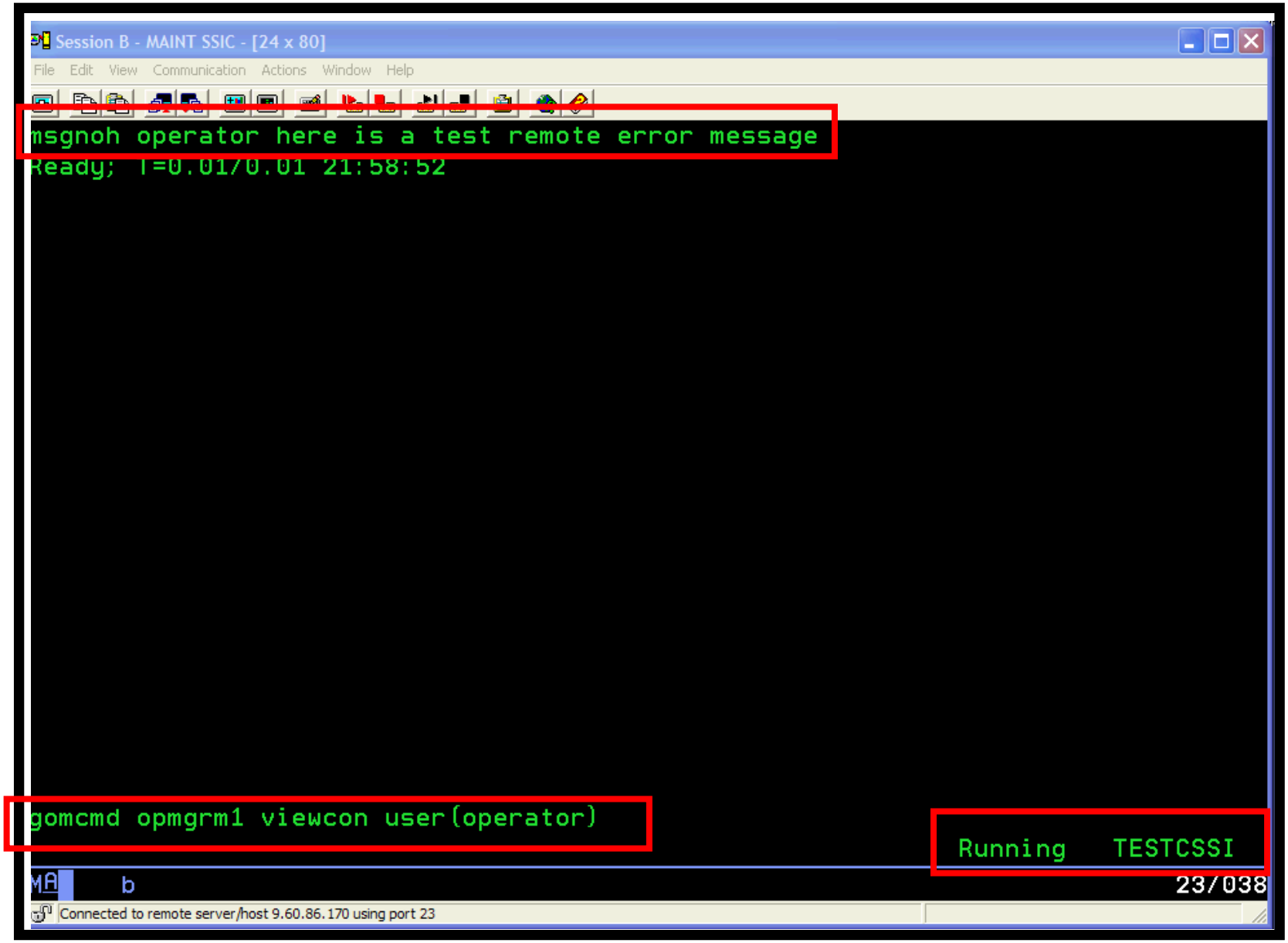
```
/*  Send a message to a central console OPERSSI for SSI cluster     */
/*                                                                  */
trace r
Address Command
Parse arg userid euser event sourcesys msgtext

/* Get local TCP/IP hostname */
parse value Search_TCPIP_Data("hostname") with getrc tcphostname .
if getrc > 4 then tcphostname = "unknown_host_name"

if userid = '_GOMEMON' then
  do
    if event = 9 then
      msgtext = 'Outbound relocation for' euser 'on' sourcesys 'started'
    else
      msgtext = 'Inbound relocation for' euser 'on' sourcesys 'started'
    'CP MSGNOH OPERSSI AT TEST7SSI From' tcphostname ':' msgtext
  end
```

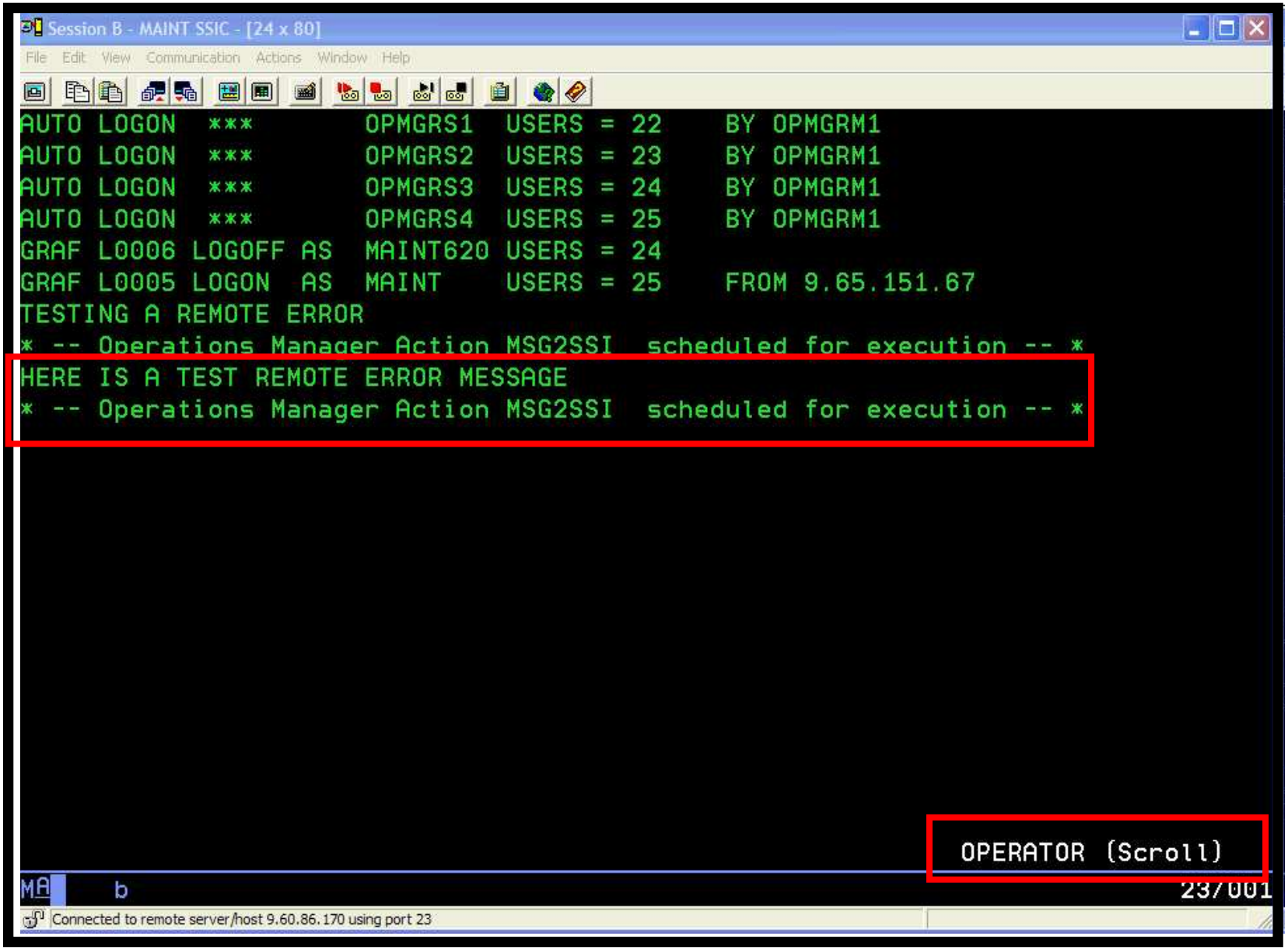
Scenario 10b: Create a Central Operations Console across multiple z/VM systems that are **not in an SSI cluster**

- **Use Operations Manager to watch for error, warning, fatal messages on service machine consoles on one or more systems**
  - OPERATOR, DIRMAINT, TCP/IP, RACF, etc.
  - Linux guests
  - Linux syslog
- **Route these messages to a central operations console on one of the z/VM systems**
- **Operations staff watches one operations console for signs of trouble across multiple z/VM systems**
  - View individual service machine consoles for more details when needed

Creating a Central Console Across Multiple LPARS







Scenario 10b: Detailed Steps

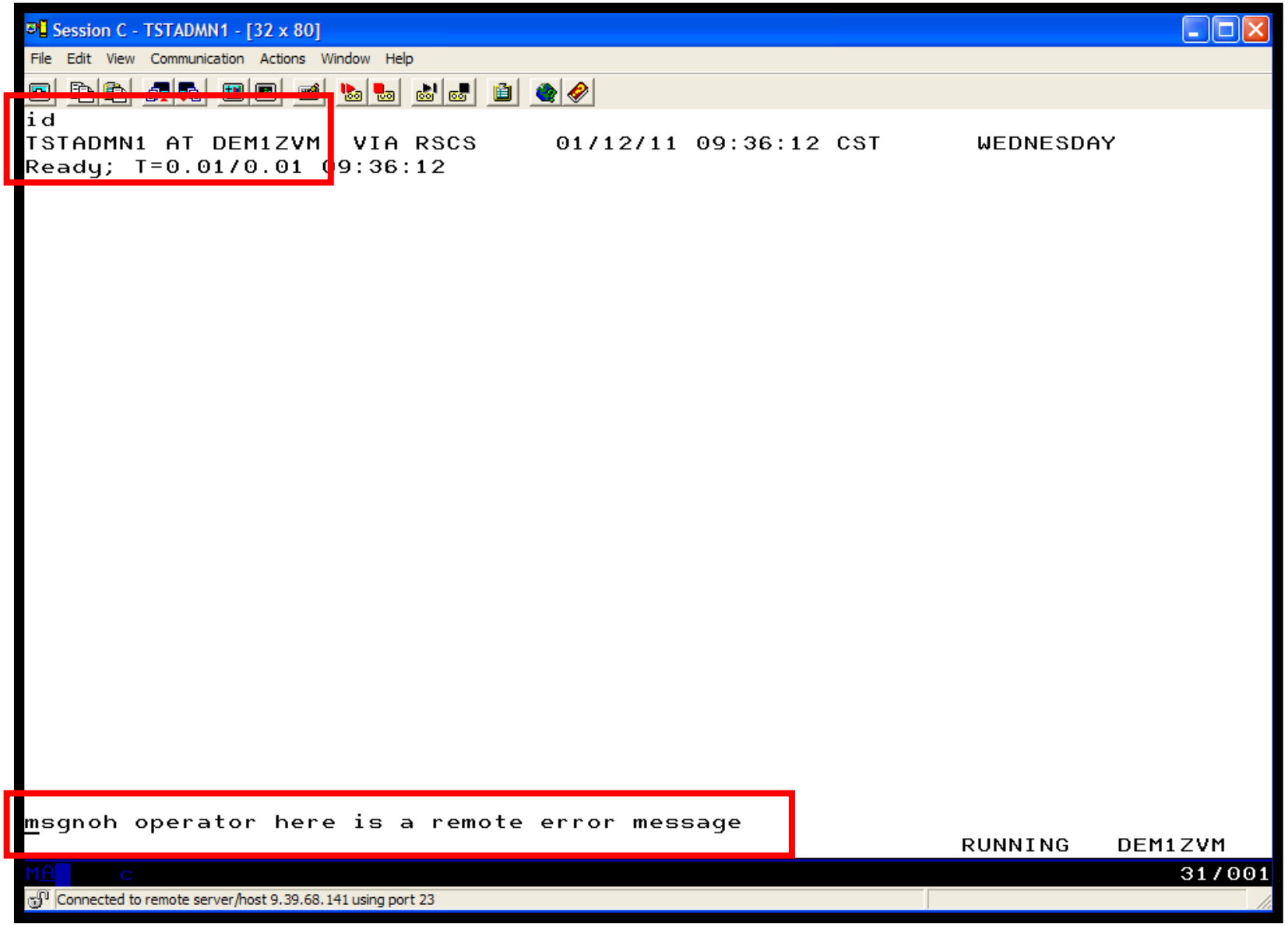
- **On System A (DEM1ZVM) put an “error” message on the OPERATOR console**
  - Must contain the text “remote error”

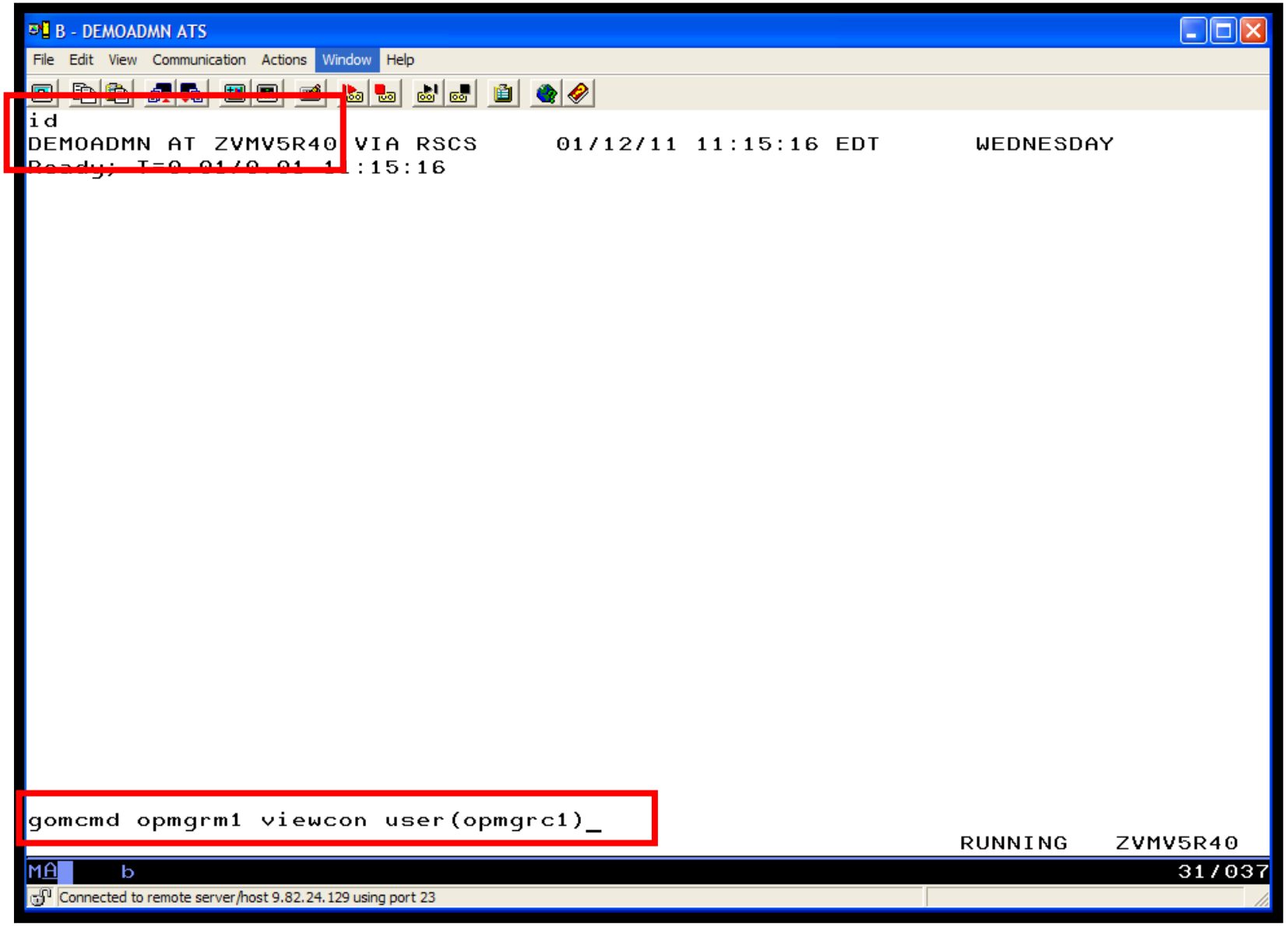
```
msgnoh operator here is a remote error message
```

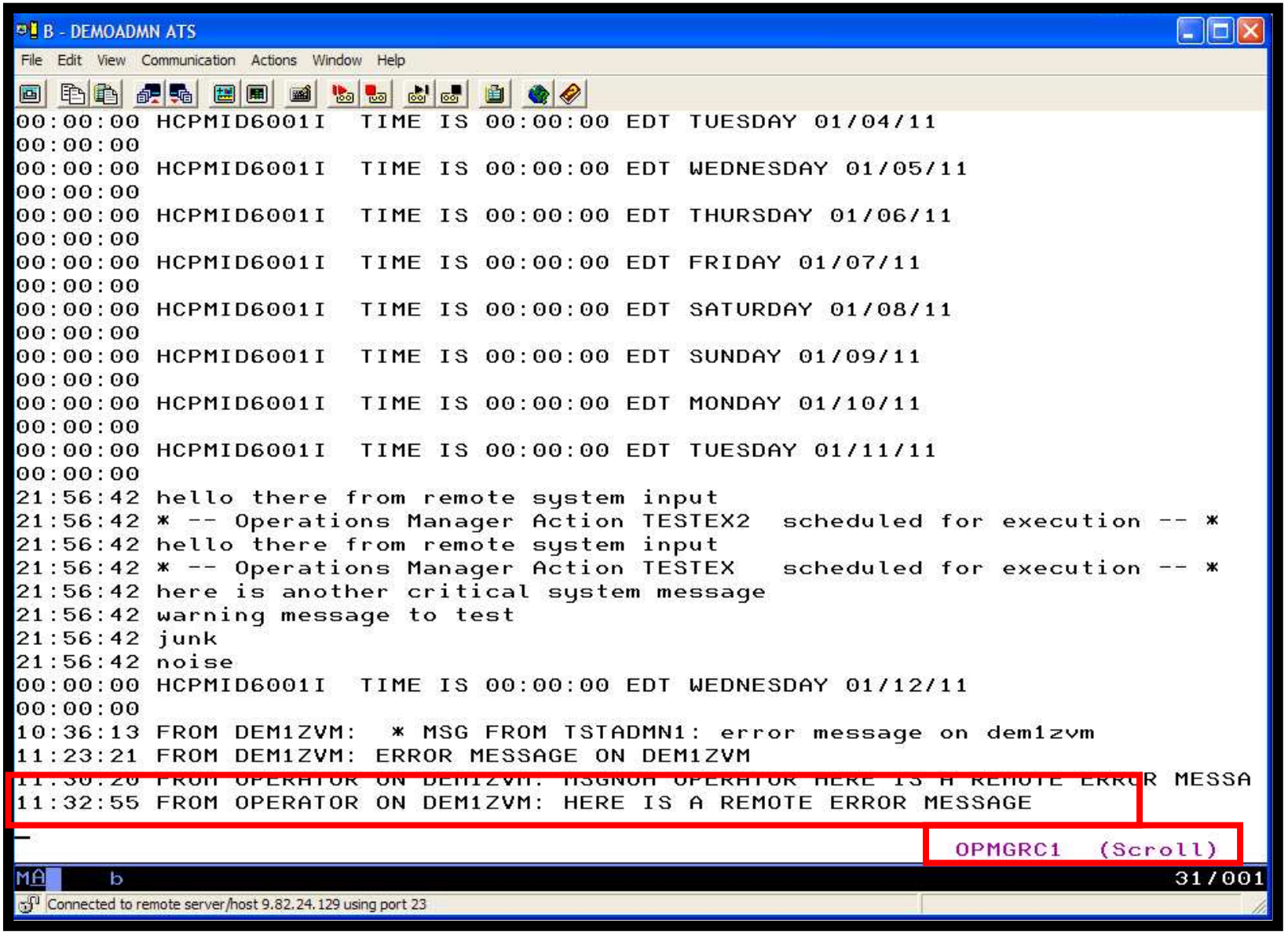
- **View the “Operations Console” (user ID OPMGRC1) on System B (ZVMV5R40) to see the message**

```
gomcmd opmgrm1 viewcon user(opmgrc1)
```

- **Note the message received on OPMGRC1 on ZVMV5R40 from OPERATOR on DEM1ZVM**







Scenario 10b: How Do You Do That?

Console rule in Operations Manager on System A:

```
*
DEFRULE NAME(OPERMSGS),+
  MATCH(*remote error*),+
  USER(OPERATOR),+
  ACTION(MSG2GBRG)
```

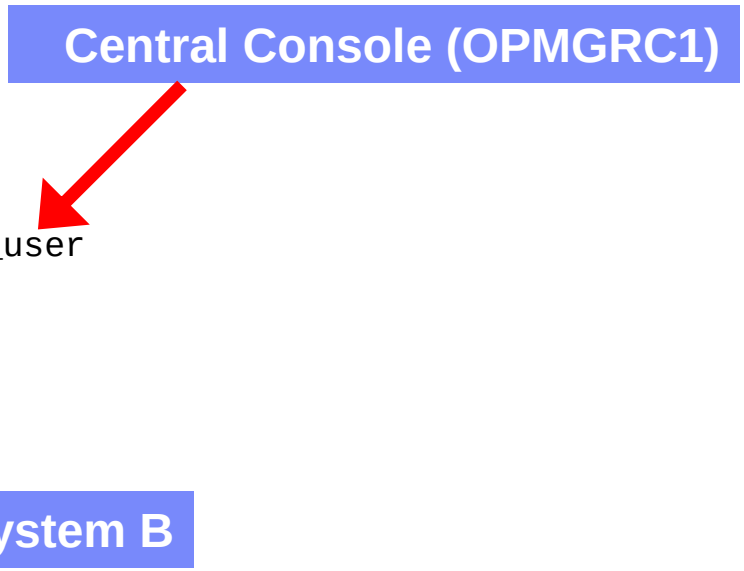
Action in Operations Manager on System A:

```
*
DEFACTN NAME(MSG2GBRG),+
  COMMAND(EXEC MSG2OPS OPMGRC1 From &u on DEM1ZVM: &t),+
  OUTPUT(LOG),+
  ENV(LVM)
```

Scenario 10b: How Do You Do That?

MSG2OPS EXEC on System A:

```
/*  Send a message to a console in Ops Mgr on another system        */
/*                                                                  */
trace r
Address Command
Parse arg cons_user msgtext
'PIPE var msgtext | > TEMP NOTE A'
'EXEC GOMRSIF TEMP NOTE A 9.82.24.129 63000' cons_user
Exit
```



Scenario 10b: How Do You Do That?

TCP/IP listener definition in Operations Manager on System B:

```
*
DEFTCPA NAME(TESTDATA),+
  TCPUSER(TCPIP),+
  TCPAPPL(GOMRSIF),+
  TCPADDR(000.000.000.000),+
  TCPPORT(63000)
```

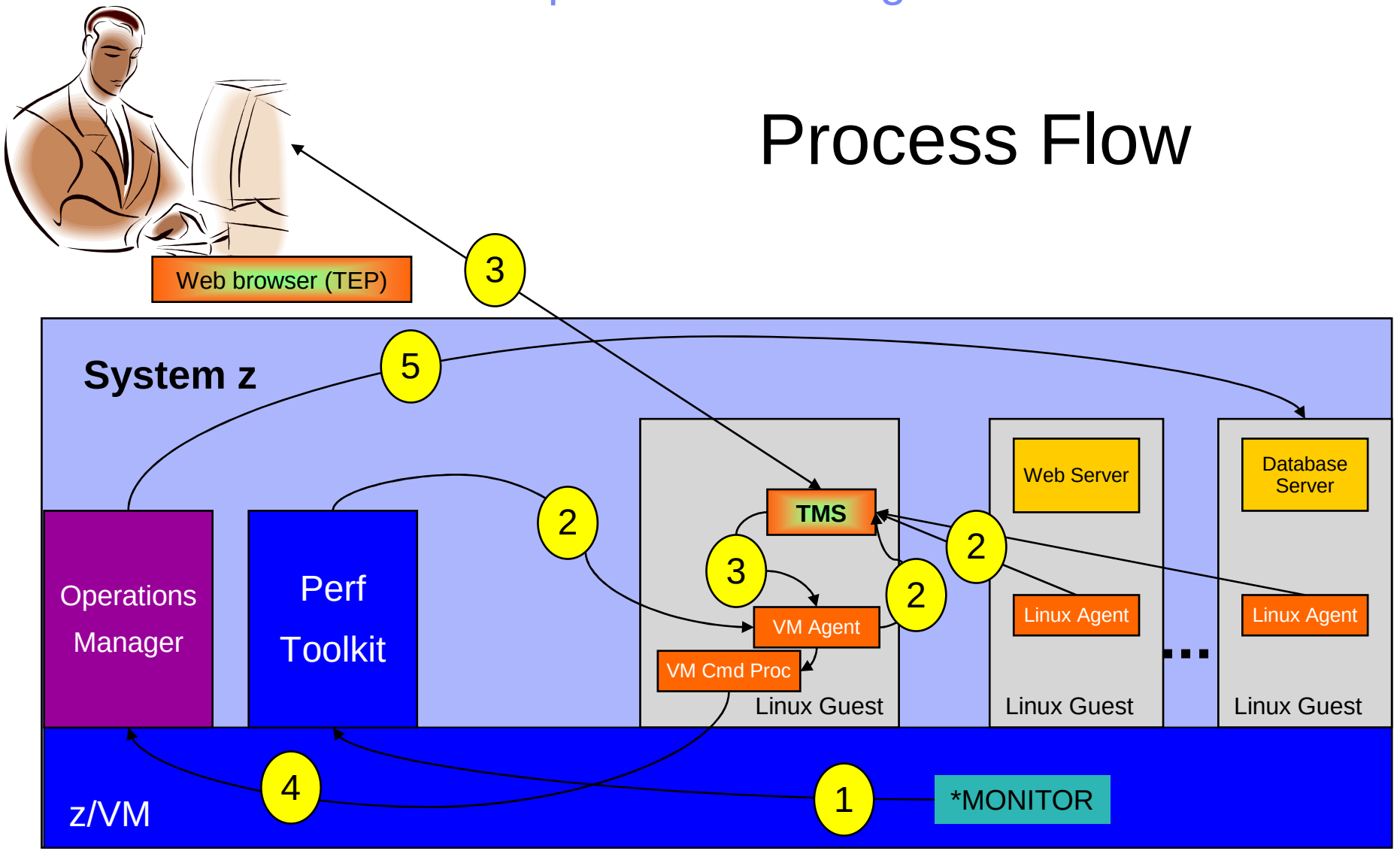
- **May also need to update TCP/IP on System B to allow Operations Manager to listen on port 63000**
- **Can alternatively use TELL (instead of GOMRSIF) to send messages from System A to System B, but requires RSCS**

Scenario 11
Integration with OMEGAMON XE on z/VM and Linux

- **Use Operations Manager to take action based on a triggered situation in OMEGAMON XE on z/VM and Linux**
- **Virtual CPU consumption is high for a Linux guest**
- **OMEGAMON detects the situation, creates an event, and sends message to Operations Manager**
- **Action is triggered by a rule in Operations Manager**
- **Operations Manager checks SHARE status of guest and issues CP commands to tune the guest**
  - SET QUICKDSP
  - SET SHARE
- **Event is resolved in OMEGAMON when virtual CPU consumption of guest is back down**

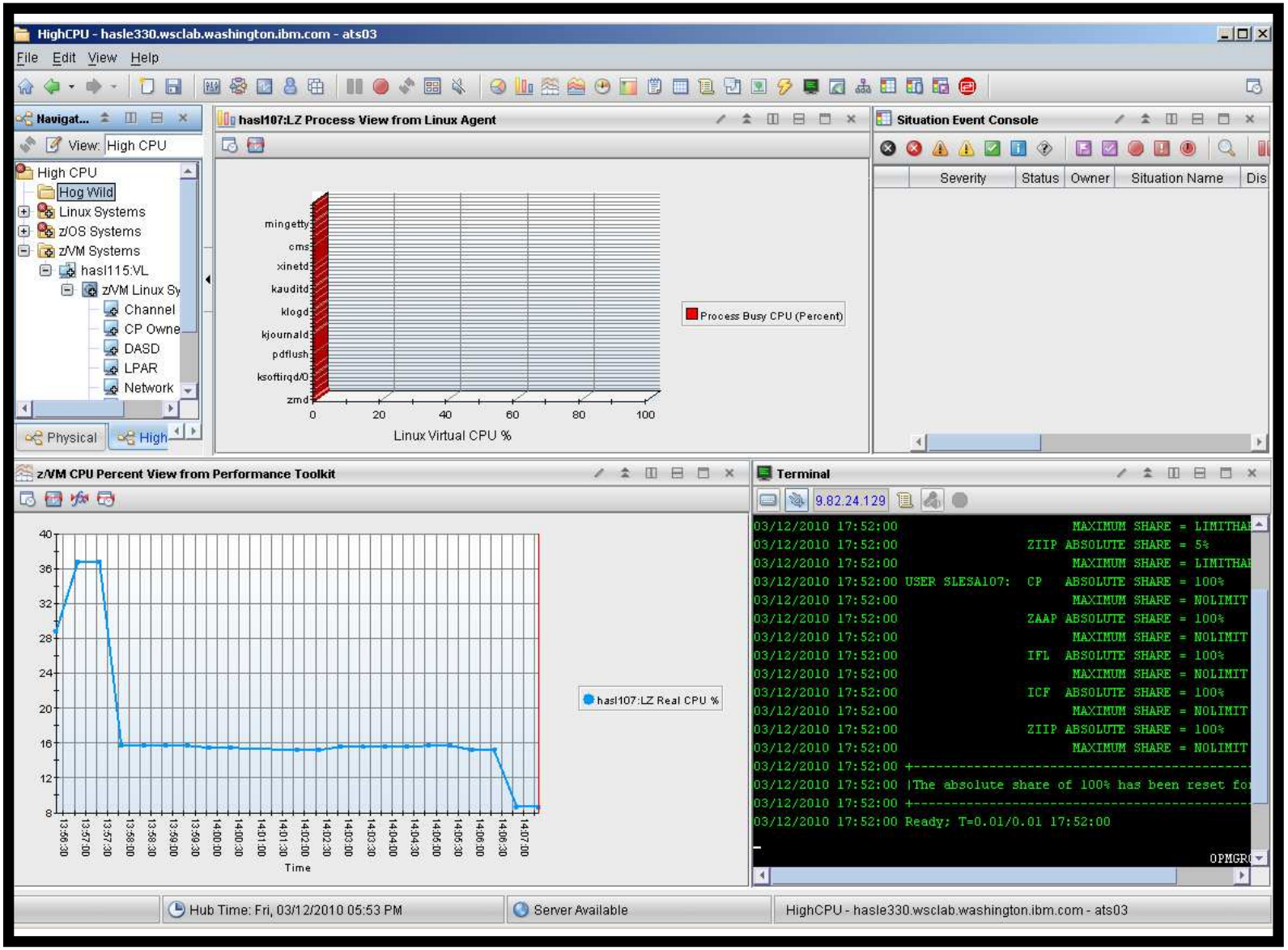
OMEGAMON XE and Operations Manager for z/VM

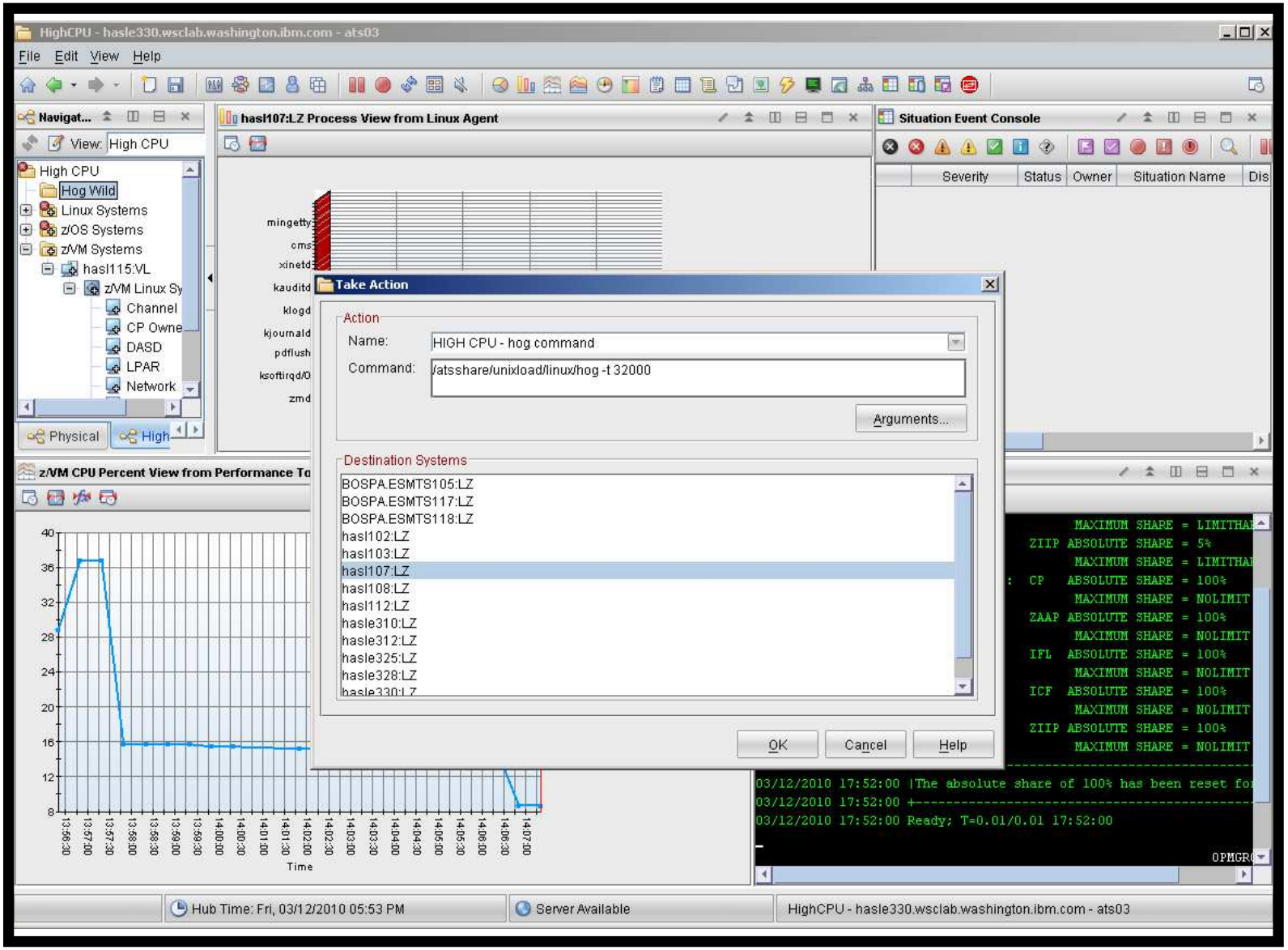
Process Flow

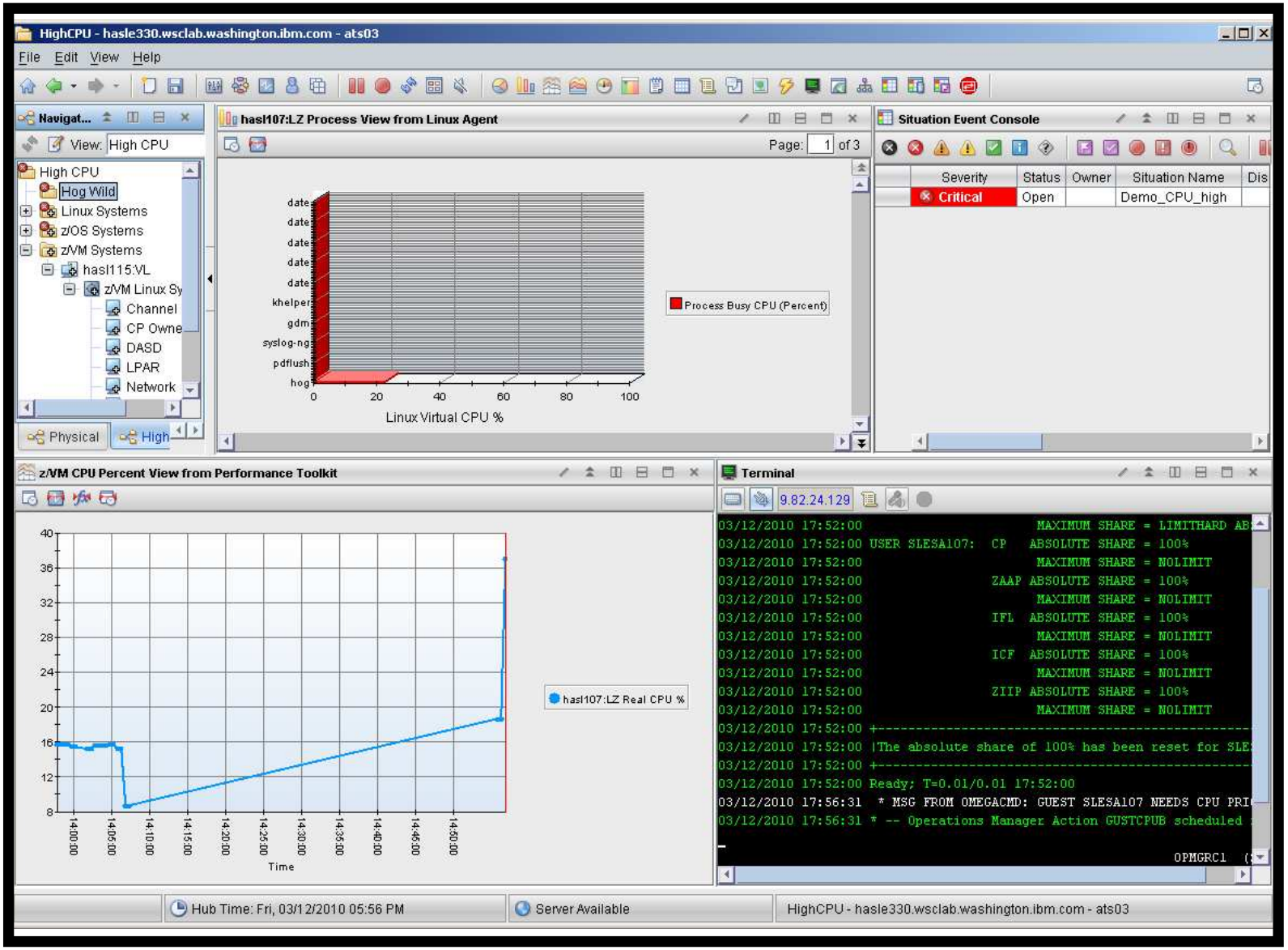


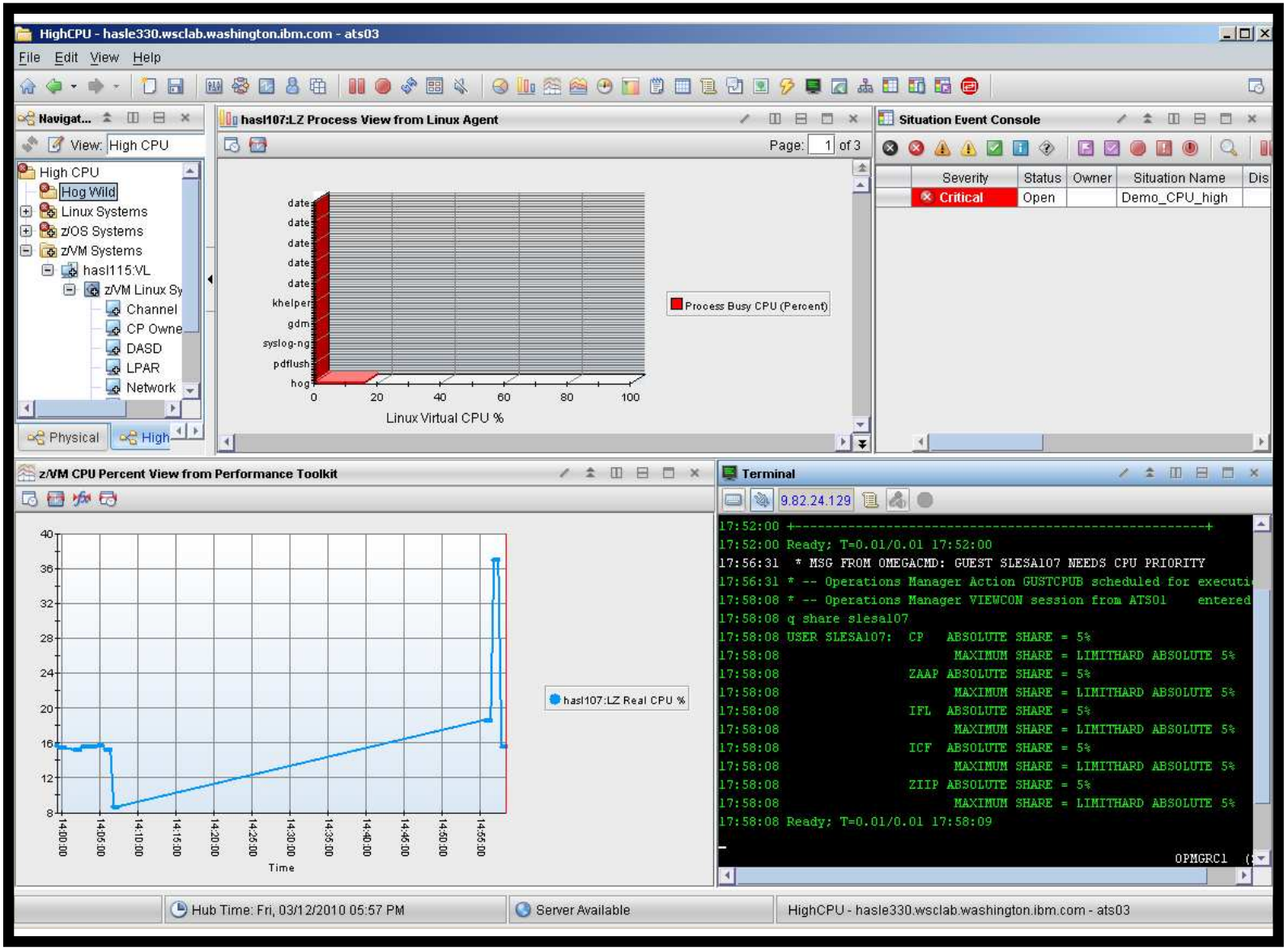
Scenario 11: Detailed Steps

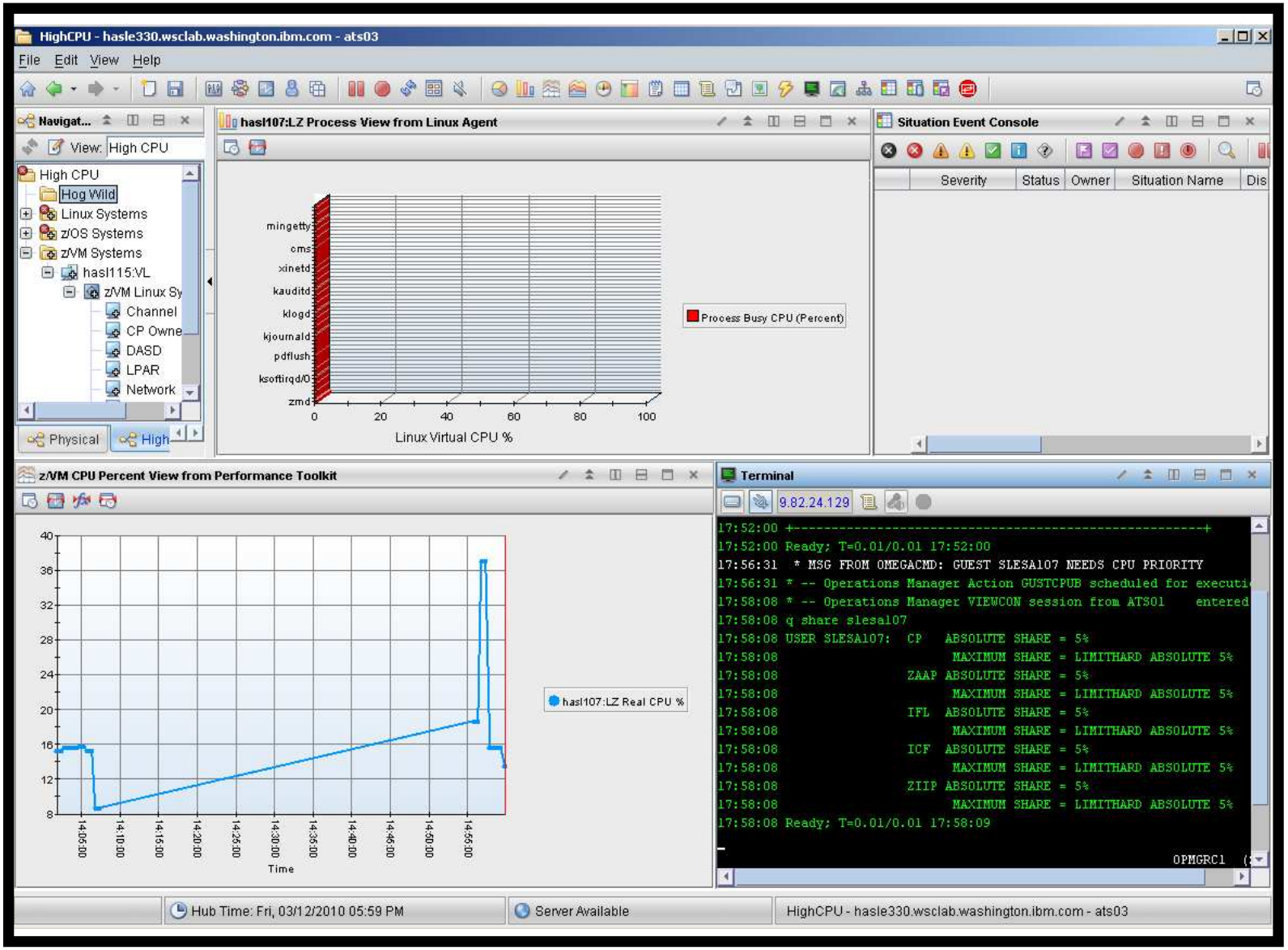
- **Create and start an application on a Linux guest that uses more than 20% of virtual CPU**
  - HOG command on our demo system
- **Updates to Tivoli Enterprise Portal**
  - z/VM CPU graph shows guest CPU % as it runs the application
  - Event pops up on situation event console to say higher than 20%
- **Use Operations Manager to watch z/VM user console used by OMEGAMON**
  - Message receive from OMEGAMON to address high CPU on the guest
  - Message from Operations Manager indicating action is triggered
- **Updates on Tivoli Enterprise Portal**
  - CPU used by that guest decreases below 20%
  - Event closed (removed from the event console)









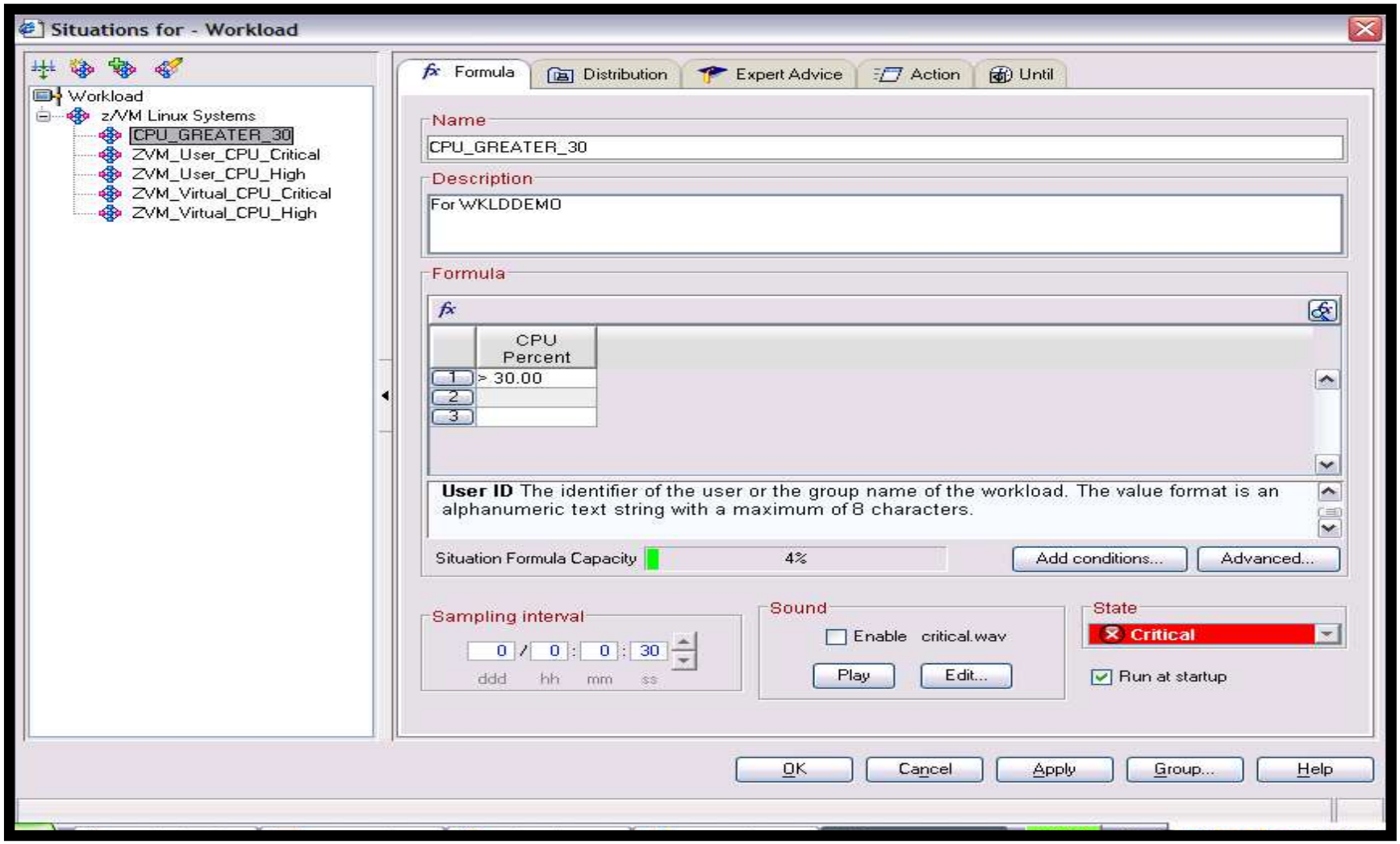


Scenario 11: How Do You Do That?

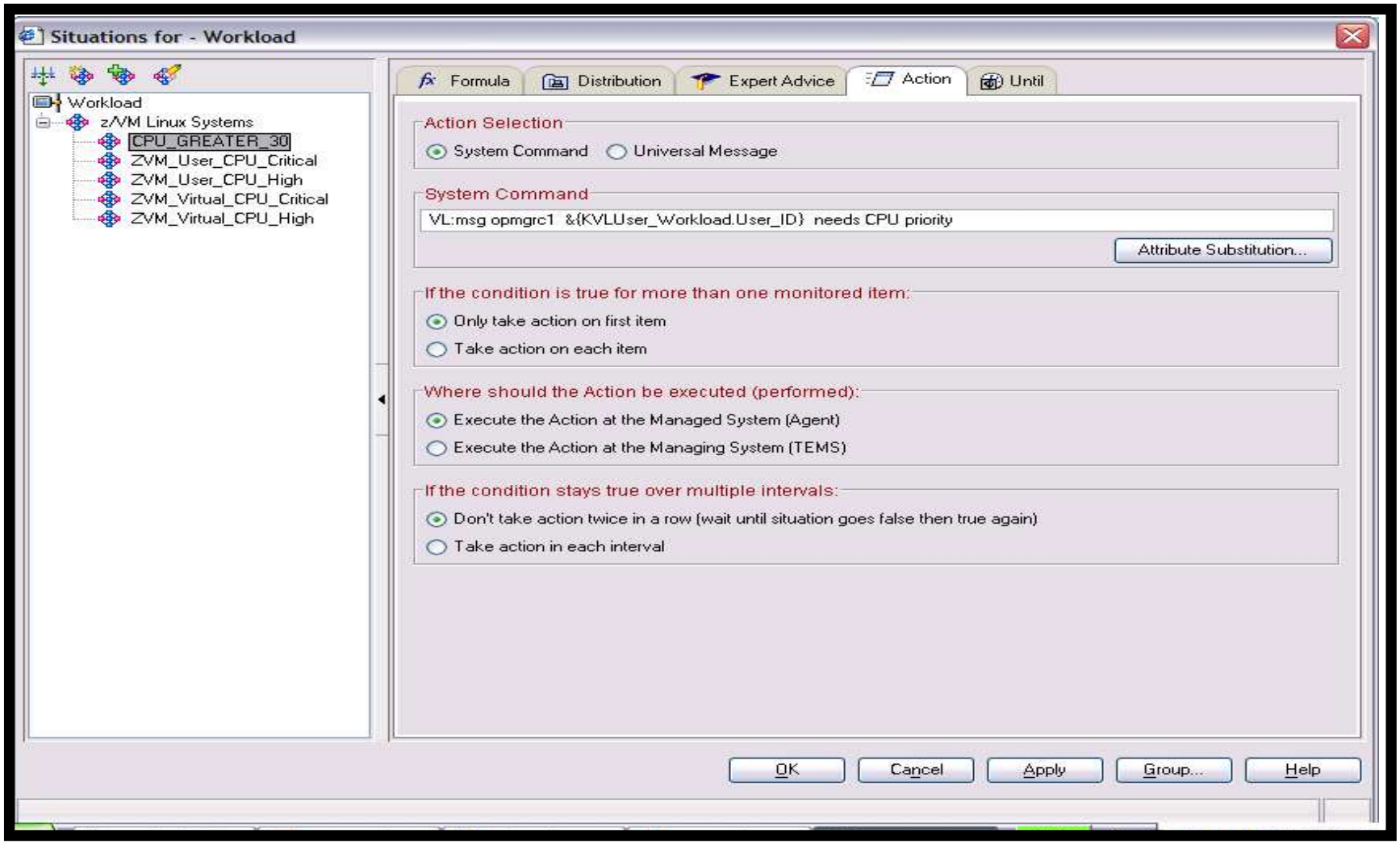
Rules in Operations Manager:

```
*
* Adjust SHARE of Linux guest if CPU usage is too high
* Watch for message from OMEGAMON
DEFRULE NAME(GUSTCPU),+
  MATCH(*NEEDS CPU PRIORITY*),+
  ACTION(GUESTCPU)
*
* Highlight message from OMEGAMON and call EXEC to check and adjust
* SHARE of Linux guest
DEFACTN NAME(GUESTCPU),+
  INPUT(AHI),+
  NEXTACTN(GUSTCPUB)
*
DEFACTN NAME(GUSTCPUB),+
  COMMAND(EXEC VCPU &4),+
  ENV(LVM),+
  OUTPUT(LOG)
```

Scenario 11: Detailed Steps
OMEGAMON Configuration

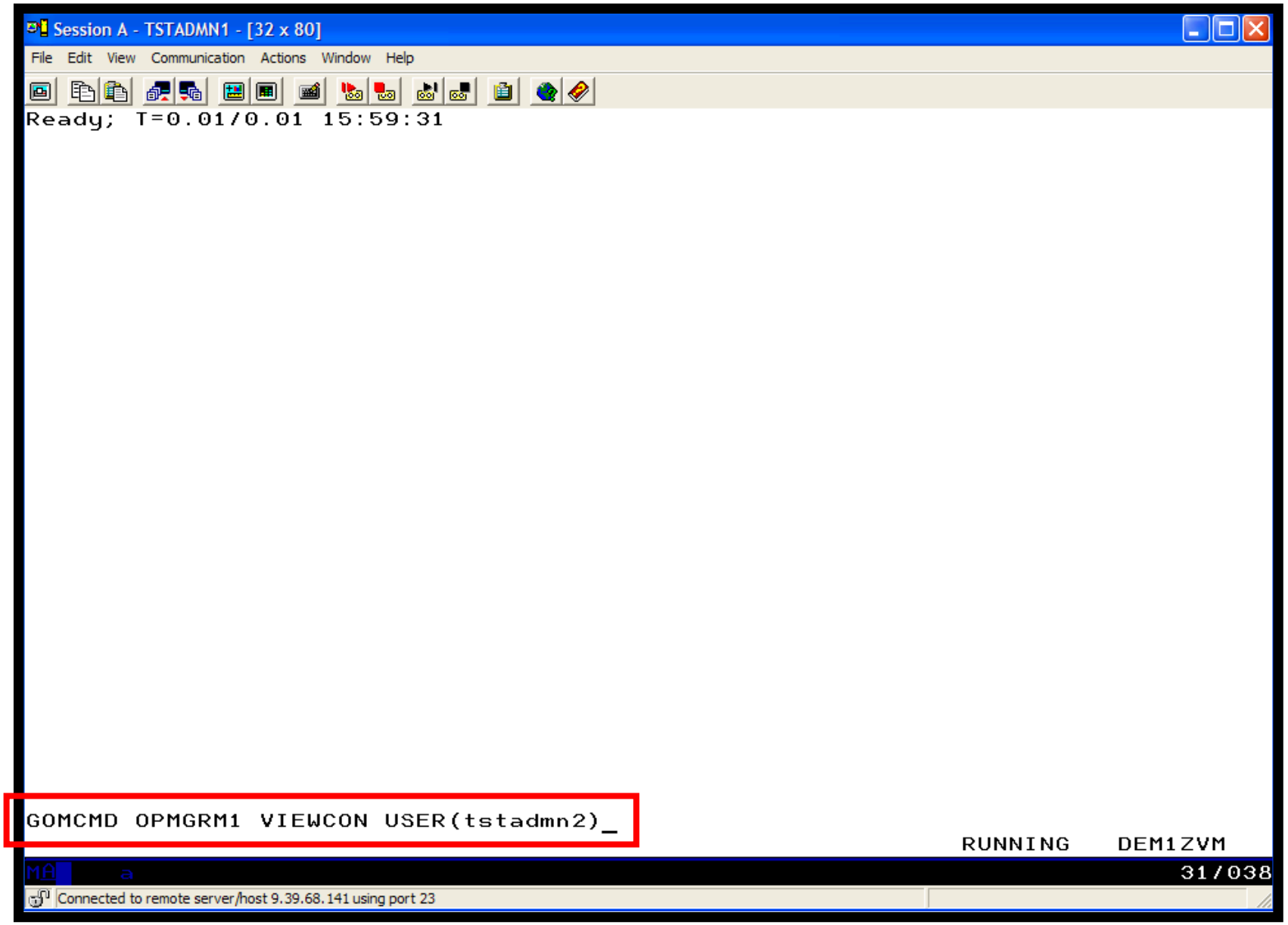


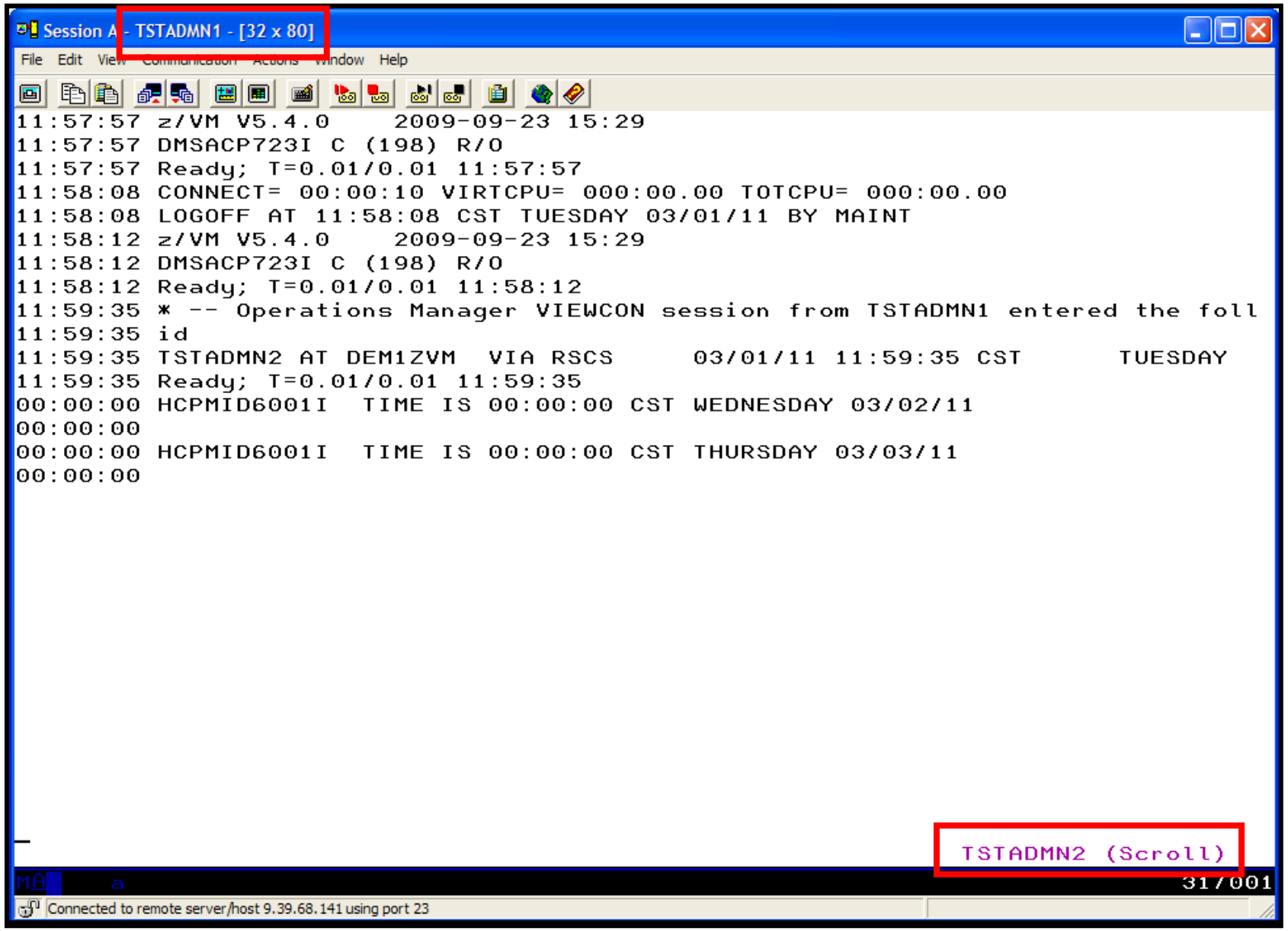
Scenario 11: Detailed Steps
OMEGAMON Configuration

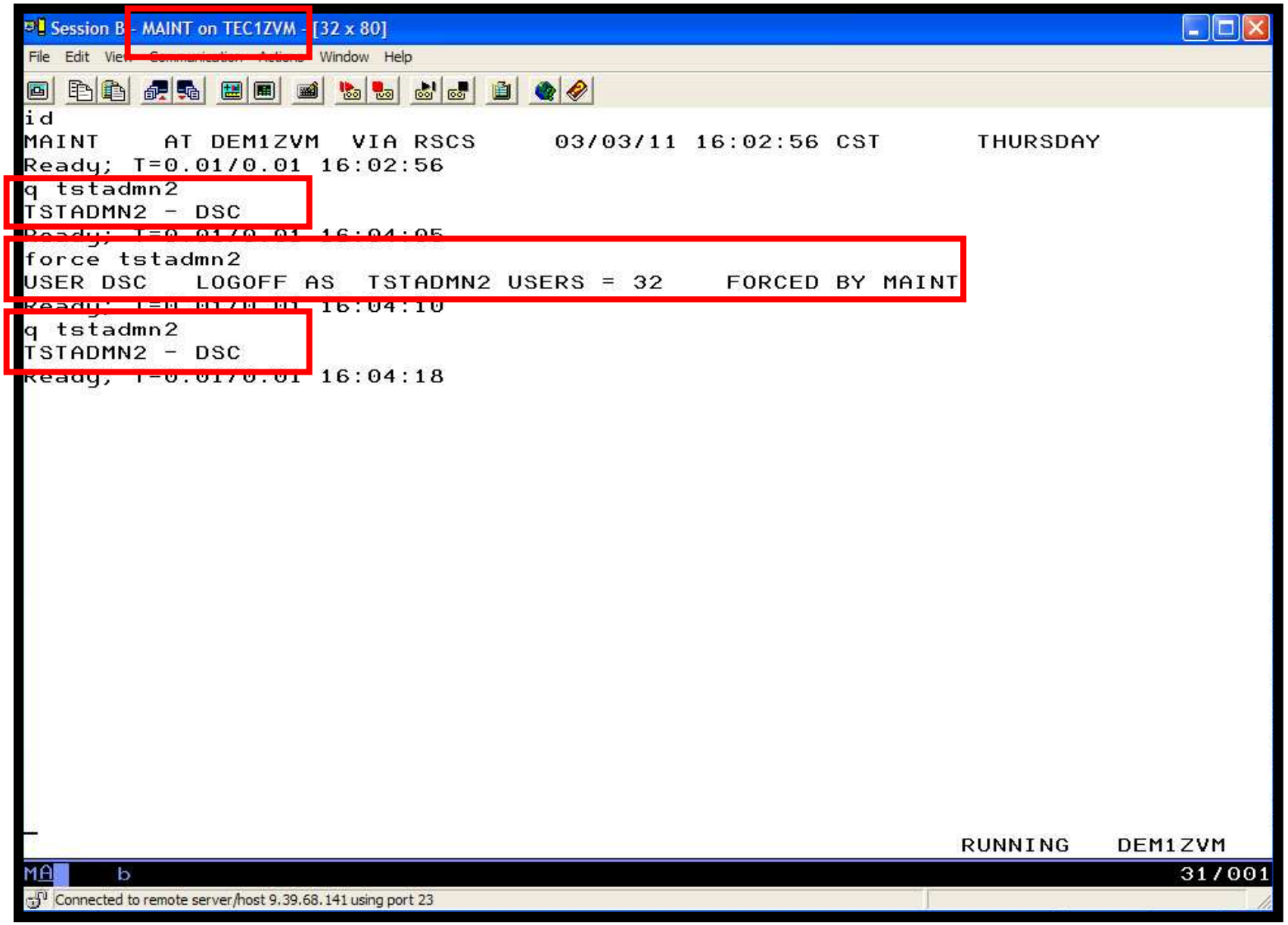


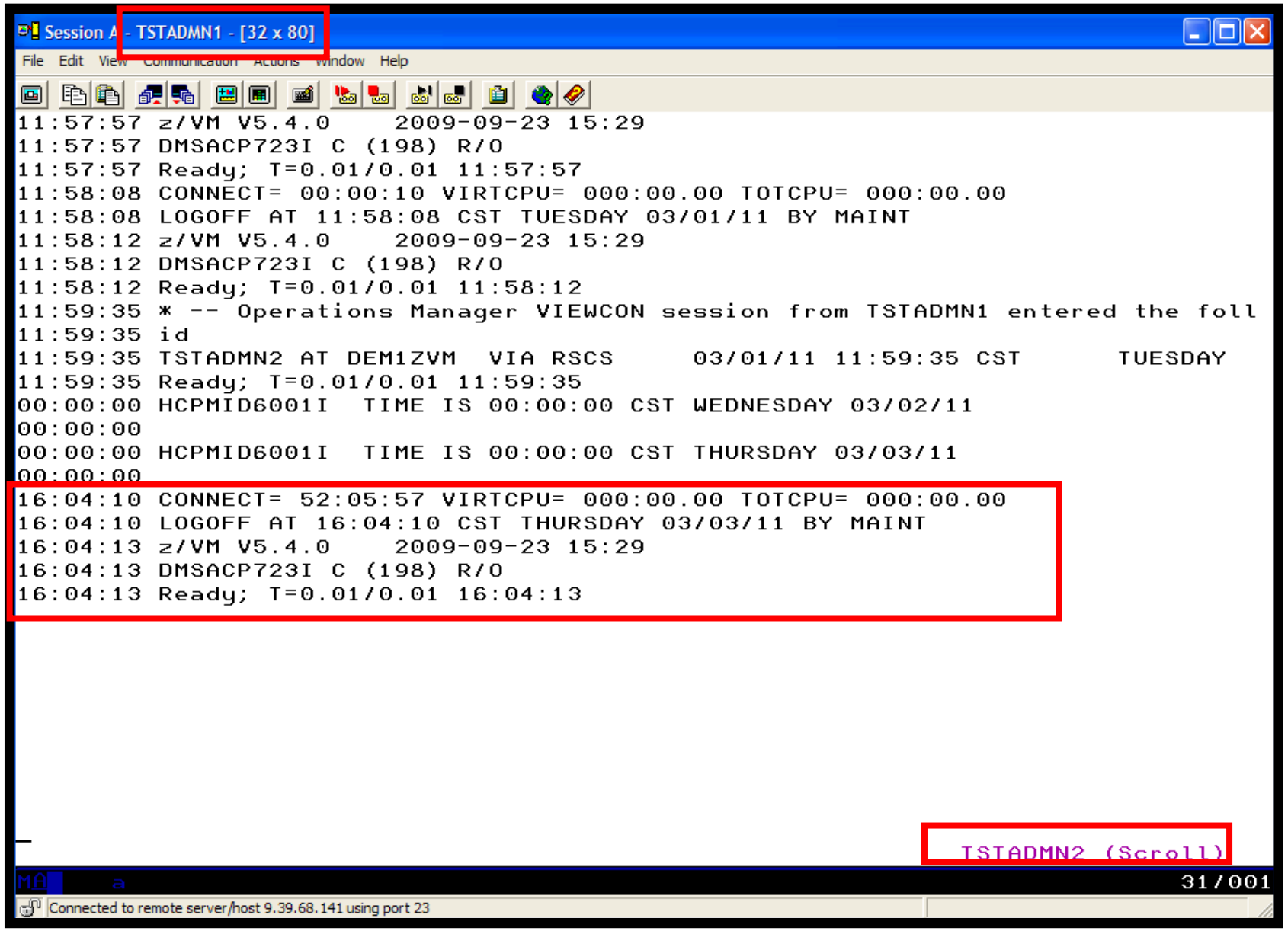
Scenario 12:
Monitor Service Machines for LOGOFF Status – and AUTOLOG them

- **Monitor specific service machines to make sure they stay logged on**
  - Demo will monitor TSTADMN2 user ID
    - Could monitor a group of user IDs
  - If it changes from logged on to logged off status, then restart it
- **Dynamically pass the user ID to the action**
  - Re-use action for multiple monitors or user IDs









Scenario 12: How Do You Do That?

Console rule and action in Operations Manager:

```
*
DEFEMON NAME(ADMIN2),+
  TYPE(1),+
  USER(TSTADMN2),+
  ACTION(AUTOLOG1)
*
DEFACTN NAME(AUTOLOG1),+
  COMMAND(CP SLEEP 3 SEC),+
  NEXTACTN(AUTOLOG2),+
  OUTPUT(LOG),+
  ENV(OPMGRS1)
*
DEFACTN NAME(AUTOLOG2),+
  COMMAND(CP XAUTOLOG &3),+
  OUTPUT(LOG),+
  ENV(OPMGRS1)
```

Scenario 13: Monitor Page Space – Send Email if Full

- **Operations Manager monitors the page space usage (percent full)**
  - For demo purposes, page space monitor is currently defined but suspended (not active)
  - We'll dynamically resume (re-activate) the page space monitor
  - Demo monitor requires the page space be only 0% full
- **Usage exceeds the specified limit**
- **Automatically send an e-mail to someone who can evaluate and take action**
- **For demo purposes, suspend (de-activate) the page space monitor when complete**

Scenario 13: Detailed Steps

- **From an authorized VM user ID, see the page space usage:**

```
q alloc page
```

- **From a user ID with Operations Manager privileges:**

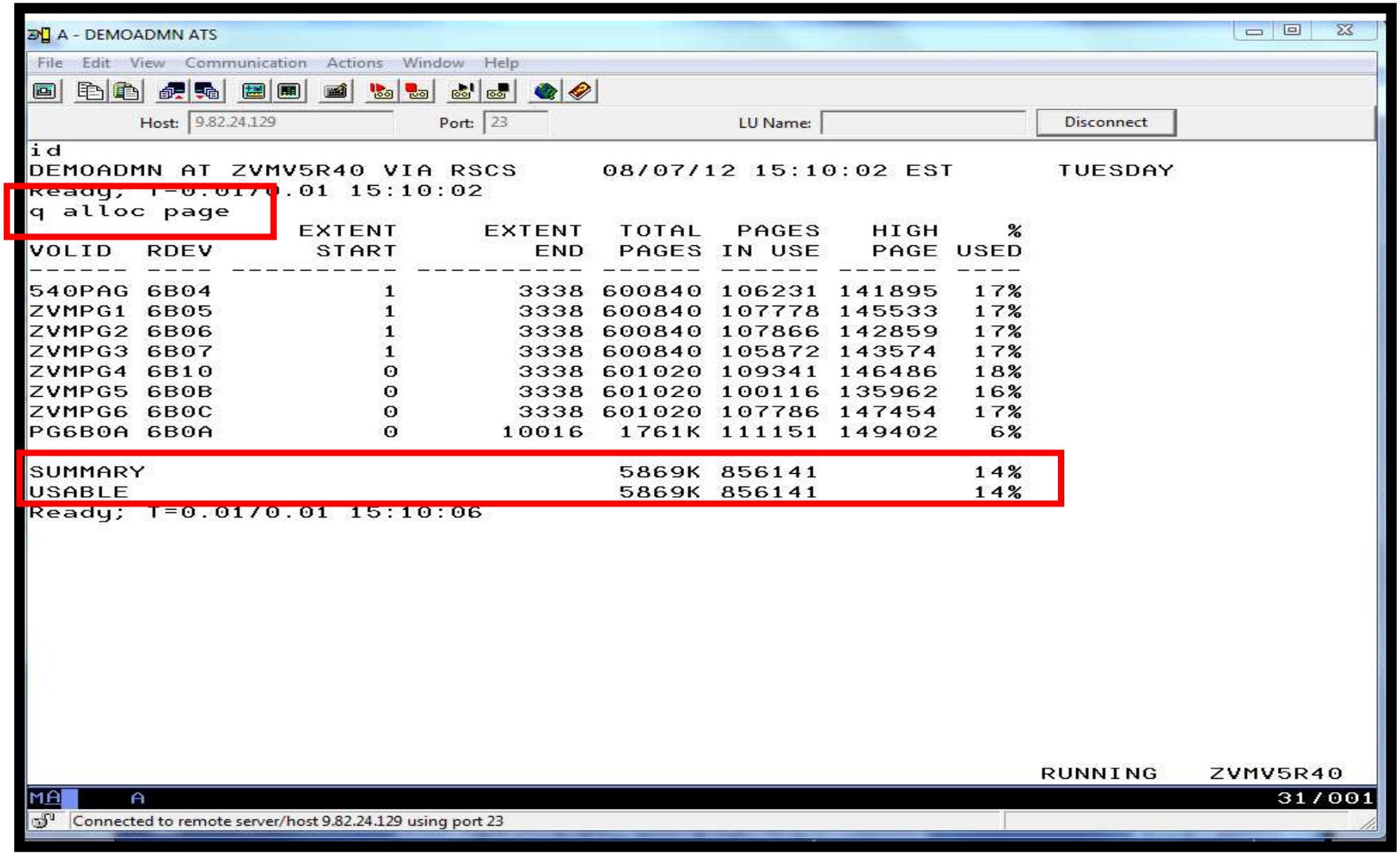
```
gomcmd opmgrm1 resume page(pgfull)
```

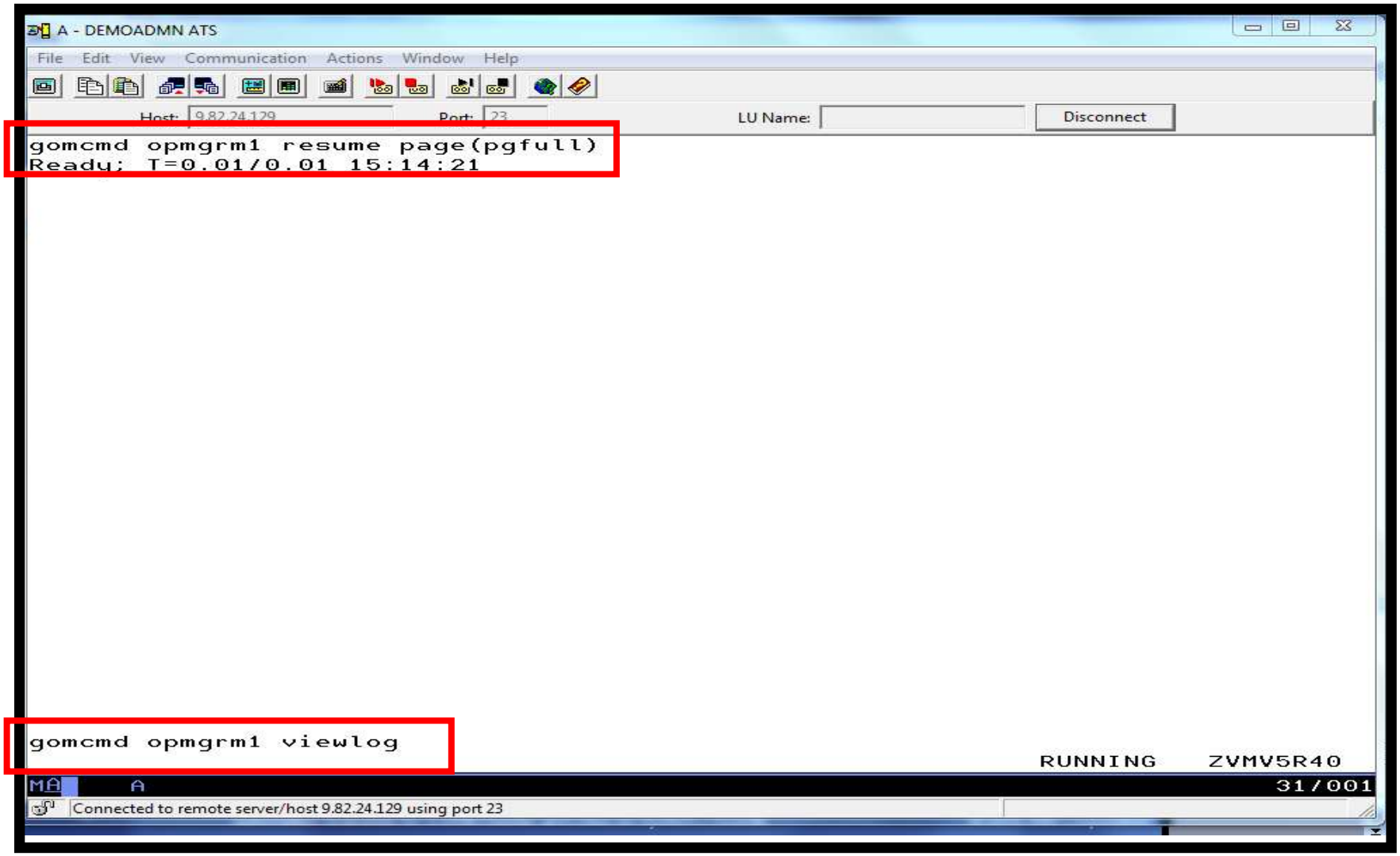
- **Check the Operations Manager log to see the spool monitor triggered:**

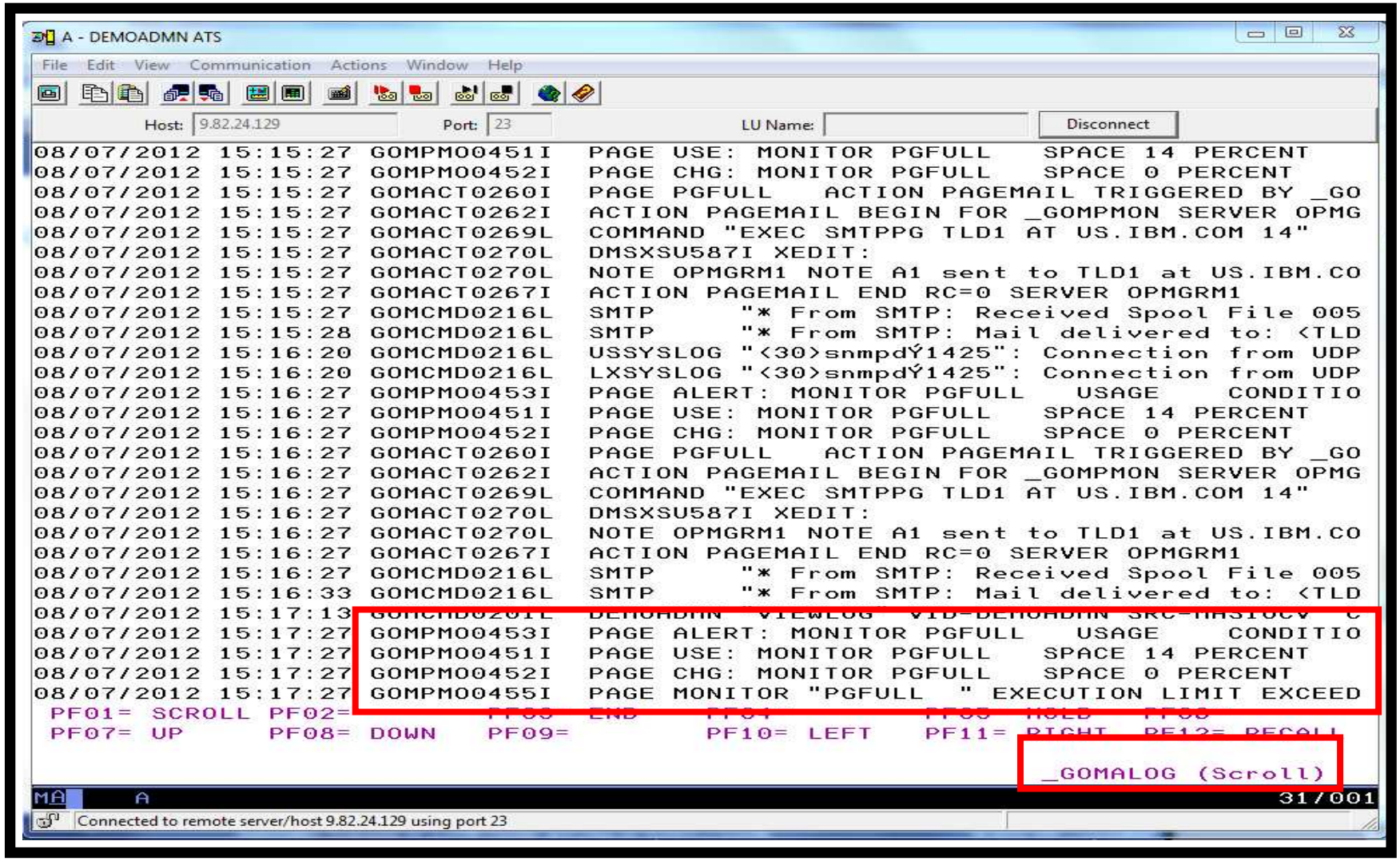
```
gomcmd opmgrm1 viewlog
```

- **Check the inbox of the appropriate person to see the email**
- **From a user ID with Operations Manager privileges:**

```
gomcmd opmgrm1 suspend page(pgfull)
```







Scenario 13: How Do You Do That?

- **Console rule and action in Operations Manager:**

```
*
DEFPMON NAME(PGFULL),+
  USAGE(010-100),+
  INTERVAL(1),+
  LIMIT(3,3600),+
  ACTION(PAGEMAIL)
*
SUSPEND PAGE(PGFULL)
*
DEFACTN NAME(PAGEMAIL),+
  COMMAND(EXEC SMTPPG tld1 at us.ibm.com &4),+
  OUTPUT(LOG),+
  ENV(LVM)
```

Scenario 13: How Do You Do That?

SMTPPG EXEC (excerpts)

```
/*  */
Parse arg mail_user dummyat mail_node pgpct

errtext = ‘Page space is' pgpct‘% full on z/VM system‘

/* Get local TCP/IP hostname */
parse value Search_TCPIP_Data("hostname") with getrc tcphostname .
if getrc > 4 then tcphostname = "unknown_host_name"
parse value Search_TCPIP_Data("domainorigin") with getrc tcpdomain .
if getrc > 4 then tcpdomain = "unknown_domain_name"

fqdomain_name = tcphostname'.'tcpdomain

/* Construct the e-mail */
line.1 = 'OPTIONS: NOACK    LOG    SHORT   NONOTEBOOK ALL CLASS A'
line.2 = 'Date: ' Date() ',' Time()
line.3 = 'From: Operations Manager for z/VM'
line.4 = 'To:  ' mail_user 'at' mail_node
line.5 = 'Subject: ' errtext 'on' fqdomain_name
line.6 = ‘DO NOT REPLY - This e-mail was generated by an automated service machine’
Line.7 = ‘ ‘
line.8 = msgtext
line.0 = 8

'PIPE stem line. | > TEMP NOTE A'

'EXEC SENDFILE TEMP NOTE A (NOTE SMTP'
```

Scenario 14:
Monitor SSI Connectivity between Two Members of a Cluster

- **Create a schedule to query ISLINKs between two members of a cluster**
- **If less than 4 links up, send message to consolidated SSI console (OPERSSI)**
  - For demo purposes, we'll dynamically deactivate a link then reactivate it when done

Scenario 14: Detailed Steps

- **From an authorized VM user ID, see the currently available ISLINKs:**

```
q islink node testcssi
```

- **Deactivate one of the links:**

```
deactivate islink 0d01
```

- **Using Operations Manager, view the central operations console to see the alert:**

```
gomcmd opmgrm1 viewcon user(operssi)
```

- **Schedule is triggered every 2 minutes, so wait 2 minutes and see the messages again**

- **Reactivate the link:**

```
activate islink 0d01
```

Scenario 14: How Do You Do That?

- **Schedule and action in Operations Manager:**

```
*** Check every 10 minutes for any IS links being down
DEFSCHD NAME(ISLINK1),+
  EVERY(00:02),+
  ACTION(QISLINK),+
  PARM(TESTCSSI)
*
DEFACTN NAME(QISLINK),+
  COMMAND(EXEC QISLINK TEST7SSI &p),+
  ENV(SVM)
```

Scenario 14: How Do You Do That?

- **QISLINK EXEC:**

```
/*  Find the number of IS Links available to another node  */
/*  If less than 4, then send message to OPERSSI           */
trace o
Address command
Parse Arg thisnode othernode
'PIPE CP QUERY ISLINK NODE' othernode '| find _____State:____Up| COUNT LINES | VAR numlinks'
If numlinks < 4
 Then 'CP MSGNOH OPERSSI AT TEST7SSI From' thisnode': Number of ISLINKs to' othernode 'is' numlinks
Exit 0
```

Scenario 15:
Suppress Passwords on Linux Consoles

- **TN3270 login to Linux guest displays password**
  - Password on separate line from password prompt
  - Password captured in console and viewable in Operations Manager VIEWCON
- **Use a rule in Operations Manager to suppress the password**
  - I.e. the line following the “password:” prompt
- **Can be expanded to suppress multiple lines following matching text**

Scenario 15: Detailed Steps

- **Use Operations Manager to view the console of a Linux guest:**

```
gomcmd opmgrm1 viewcon user(omeglnx1)
```

- **Enter the login command:**

```
login root
```

- **Enter the password**
  - Note that it's not displayed

Scenario 15: How Do You Do That?

- **Rule and action in Operations Manager:**

```
*
* Change password prompt to red.
* Suppress the password when logging onto OMEGLNX1.
* Have to suppress next 2 lines to include the line Ops Mgr adds
* indicating the user entered a "command"
*
DEFRULE NAME(OMEGPW),+
  MATCH(Password:*),+
  USER(OMEGLNX1),+
  ACTION(SUPPW),+
  SUPNEXT(2)
*
DEFACTN NAME(SUPPW),+
  INPUT(CRE)
```

Scenario 16:
Autolog a Linux Guest and Send Message if Doesn't Start Successfully

- **Define a schedule and action to start a Linux guest**
- **Define a rule looking for the application specific message indicating up and ready for work**
- **Define an idle monitor for the above rule**
  - If "up and ready" message is not found within 1 minute, then send message to central console
- **Idle monitor is suspended until schedule is triggered**
  - Before autologging the Linux guest, automatically resume idle monitor
- **Idle monitor is automatically suspended again once it is triggered**

Scenario 16: Detailed Steps

- **View the configuration file to see the action that will be “scheduled”, plus the rules and monitors**

```
x tracy config
```

- **Run the action that starts the guest (and monitors)**

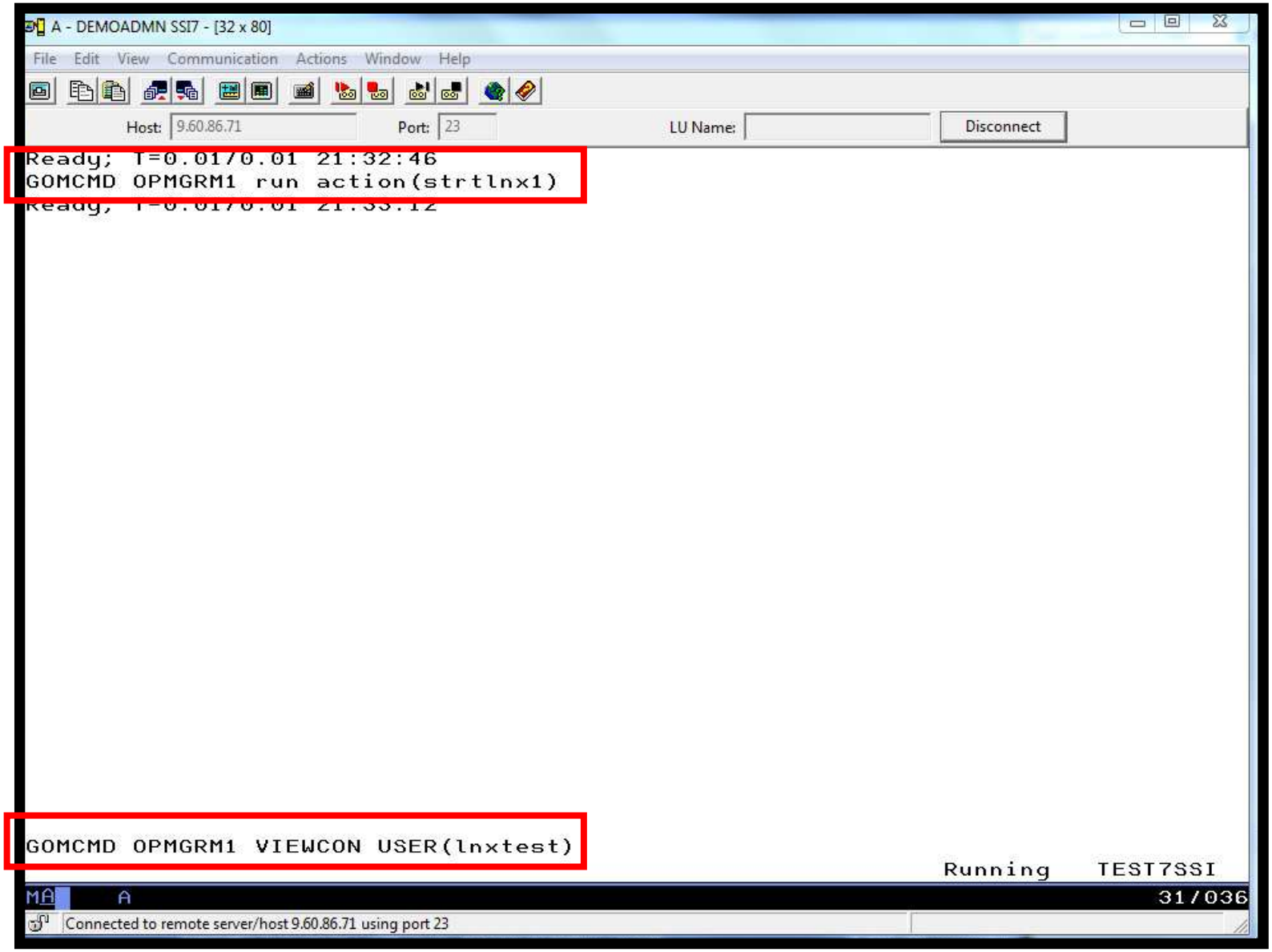
```
gomcmd opmgrm1 run action(strtlnx1)
```

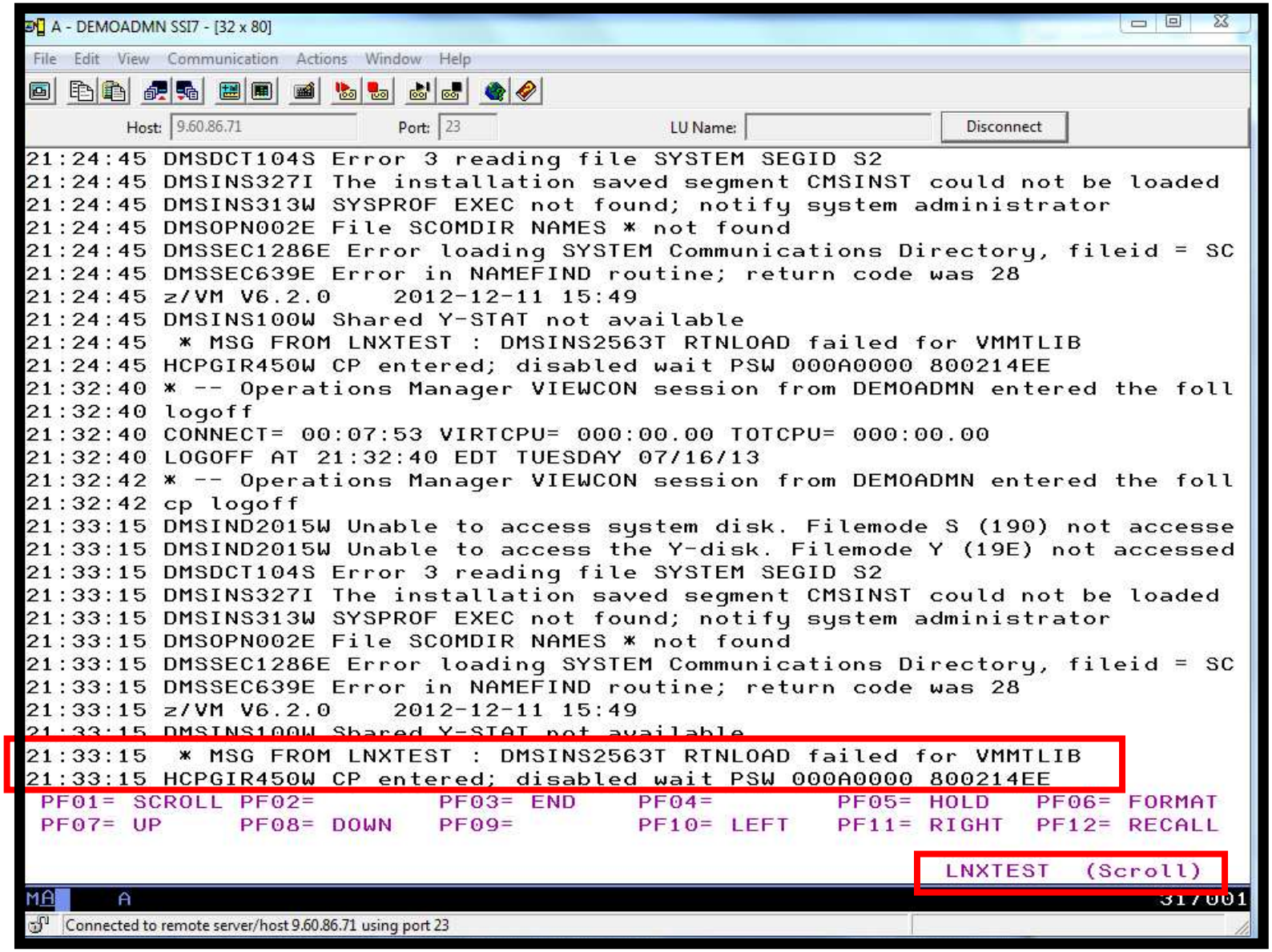
- **View the console of LNXTEST to see that it gets autologged**

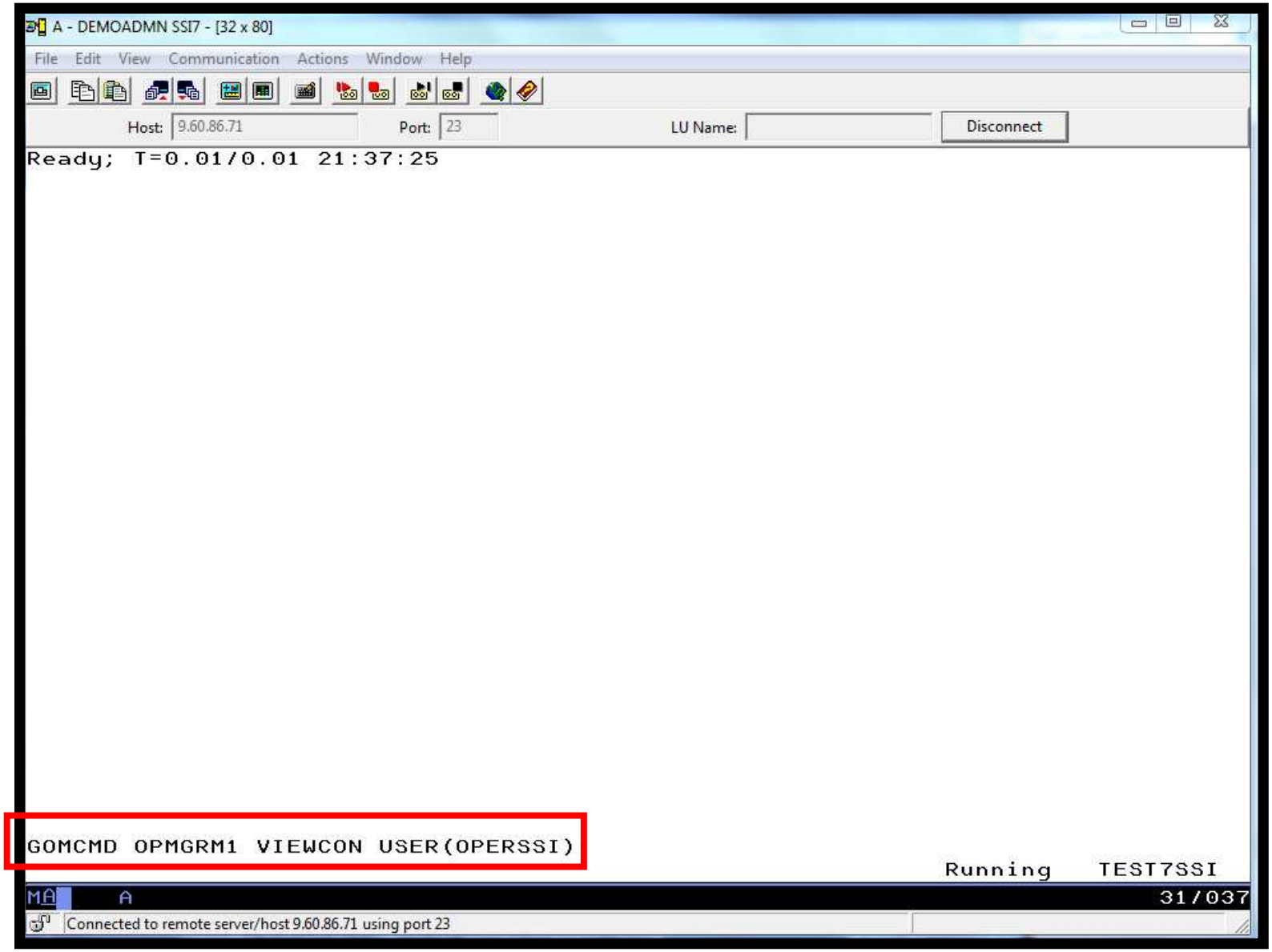
```
gomcmd opmgrm1 viewcon user(lnxtest)
```

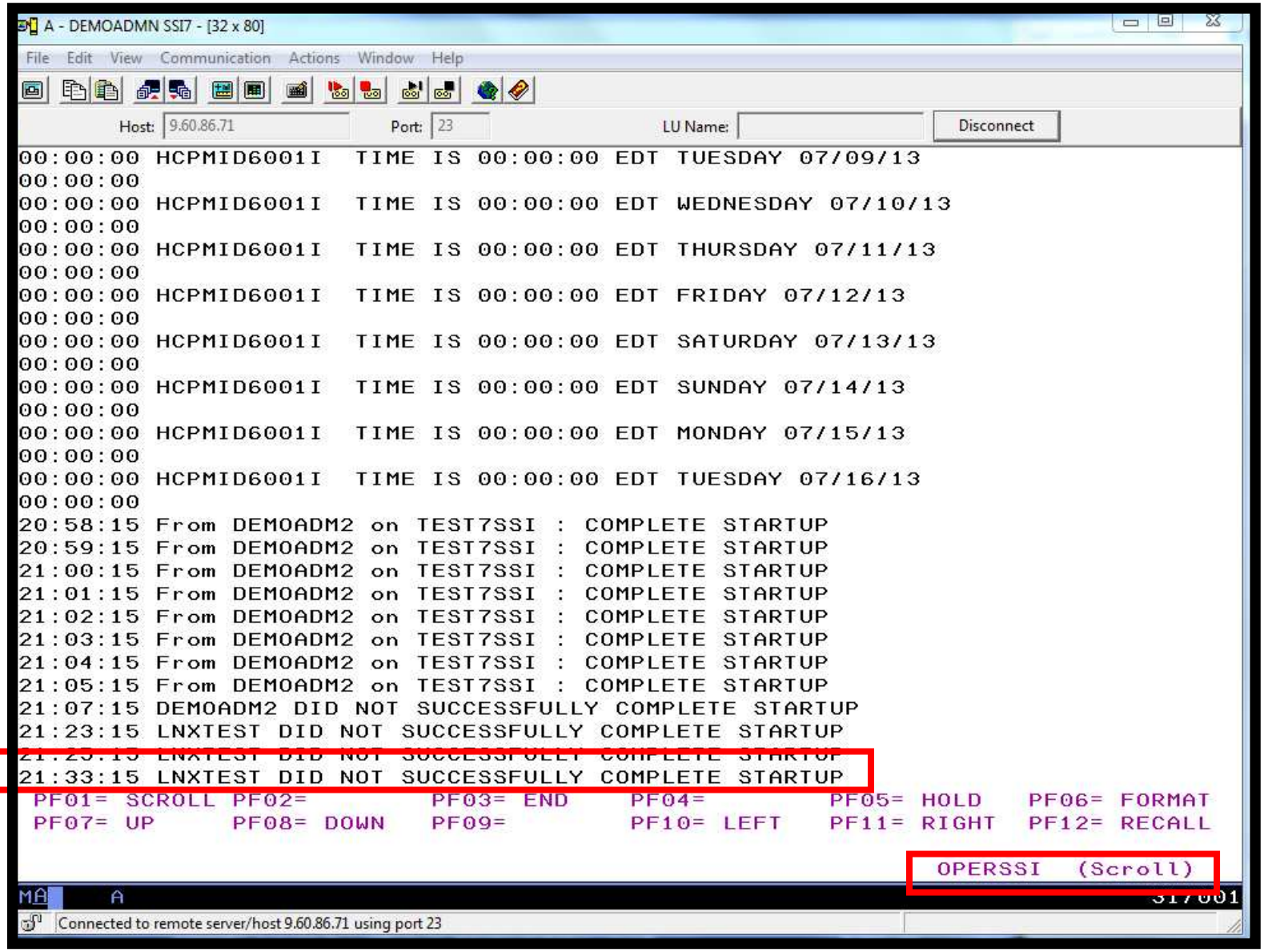
- **View the central console of OPERSSI to see the message that the guest did not start successfully**

```
gomcmd opmgrm1 viewcon user(operssi)
```









Scenario 16: How Do You Do That?

Schedule and action in Operations Manager:

```
*DEFSCHD NAME(STARTLNX),+
*  WHEN(00:01),+
*  ACTION(STRTLNX1)
*
DEFACTN NAME(STRTLNX1),+
  COMMAND('RESUME IDLE(NOLOGON)'),+
  NEXTACTN(STRTLNX2),+
  ENV(GOM)
*
DEFACTN NAME(STRTLNX2),+
  COMMAND(CP XAUTOLOG LNXTEST),+
  ENV(SVM)
```

Scenario 16: How Do You Do That?

- **Watch for successful startup of Linux guest**
- **If successful take no action**

```
DEFRULE NAME(LNXLOGON),+
  MATCH(*LNXTEST successfully started*),+
  USER(LNXTEST),+
  ACTION(NOACT)
*
DEFACTN NAME(NOACT)
```

Scenario 16: How Do You Do That?

- **If Linux doesn't start successfully send message to central console and suspend monitor:**

```
DEFIMON NAME(NOLOGON),+
  RULE(LNXLOGON),+
  OCCUR(1,1),+
  ACTION(MSG2SSI2),+
  PARM(LNXTEST)
*
SUSPEND IDLE(NOLOGON)
DEFACTN NAME(MSG2SSI2),+
  COMMAND(CP MSGNOH OPERSSI &p did not successfully complete startup),+
  NEXTACTN(SUSPIDLE),+
  ENV(SVM)
*
DEFACTN NAME(SUSPIDLE),+
  COMMAND('SUSPEND IDLE(NOLOGON)'),+
  ENV(GOM)
```

Scenario 17: View Live Consoles of Linux Guests, Linux Syslog Data, CMS Service Machines

- **Configure user IDs / guests to be monitored by Operations Manager**
- **Route syslog data from a Linux guest to Operations Manager**
- **From authorized user, view the live console data of**
  - OPERATOR
    - Issue VM commands
  - A Linux guest
    - Issue Linux commands
  - Linux syslog data

Scenario 17: Detailed Steps

- **From an authorized VM user ID, verify OPERATOR and Linux guest are being monitored by Operations Manager**

```
q secuser operator
q observer sles11c
```

- **View the console of OPERATOR**

```
gomcmd opmgrm1 viewcon user(operator)
```

- **Issue VM commands allowed by OPERATOR**

```
id
cp send bkrbkup cms listfile
```

- **View the console of the backup server**

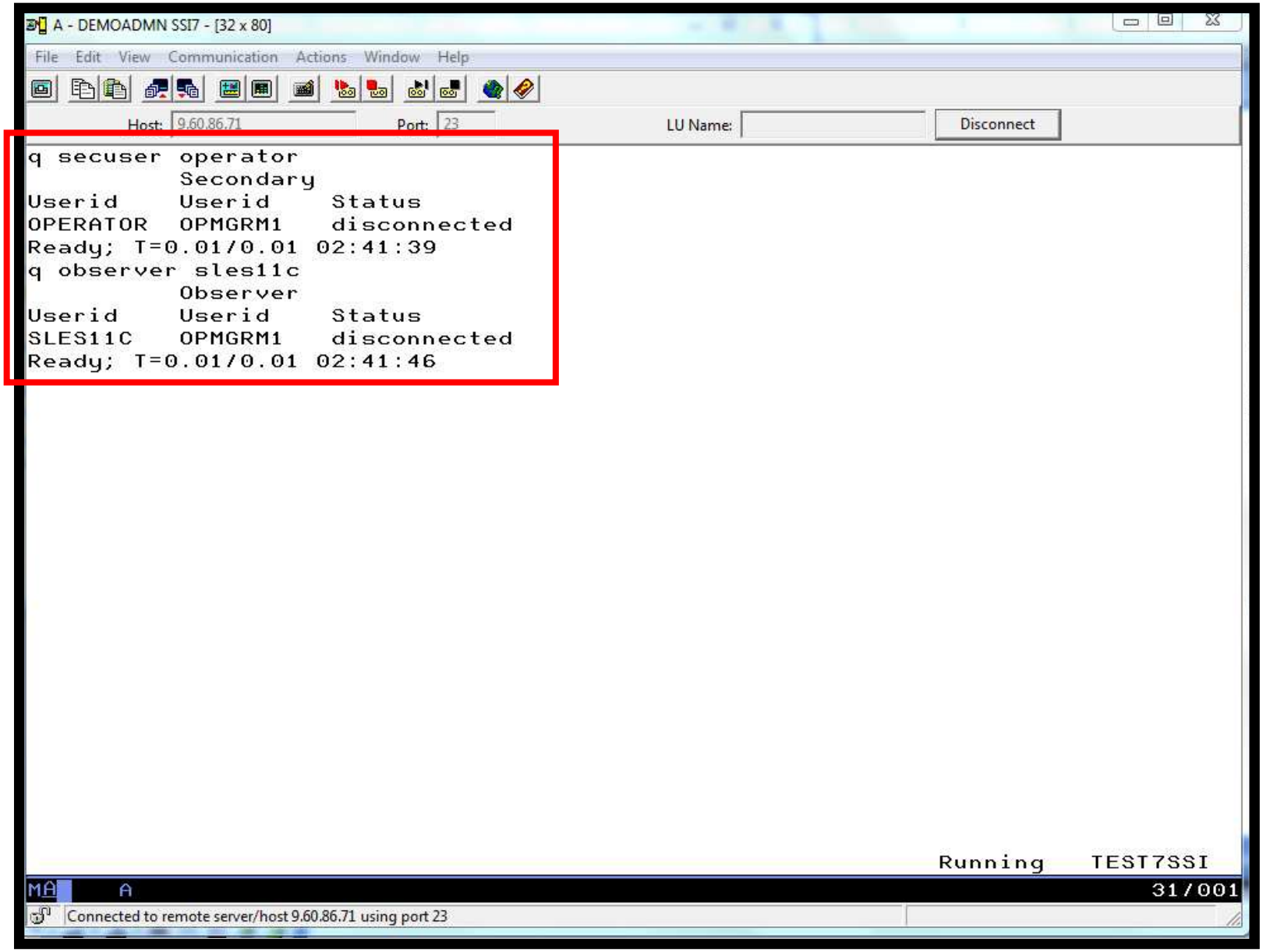
```
gomcmd opmgrm1 viewcon user(bkrbkup)
```

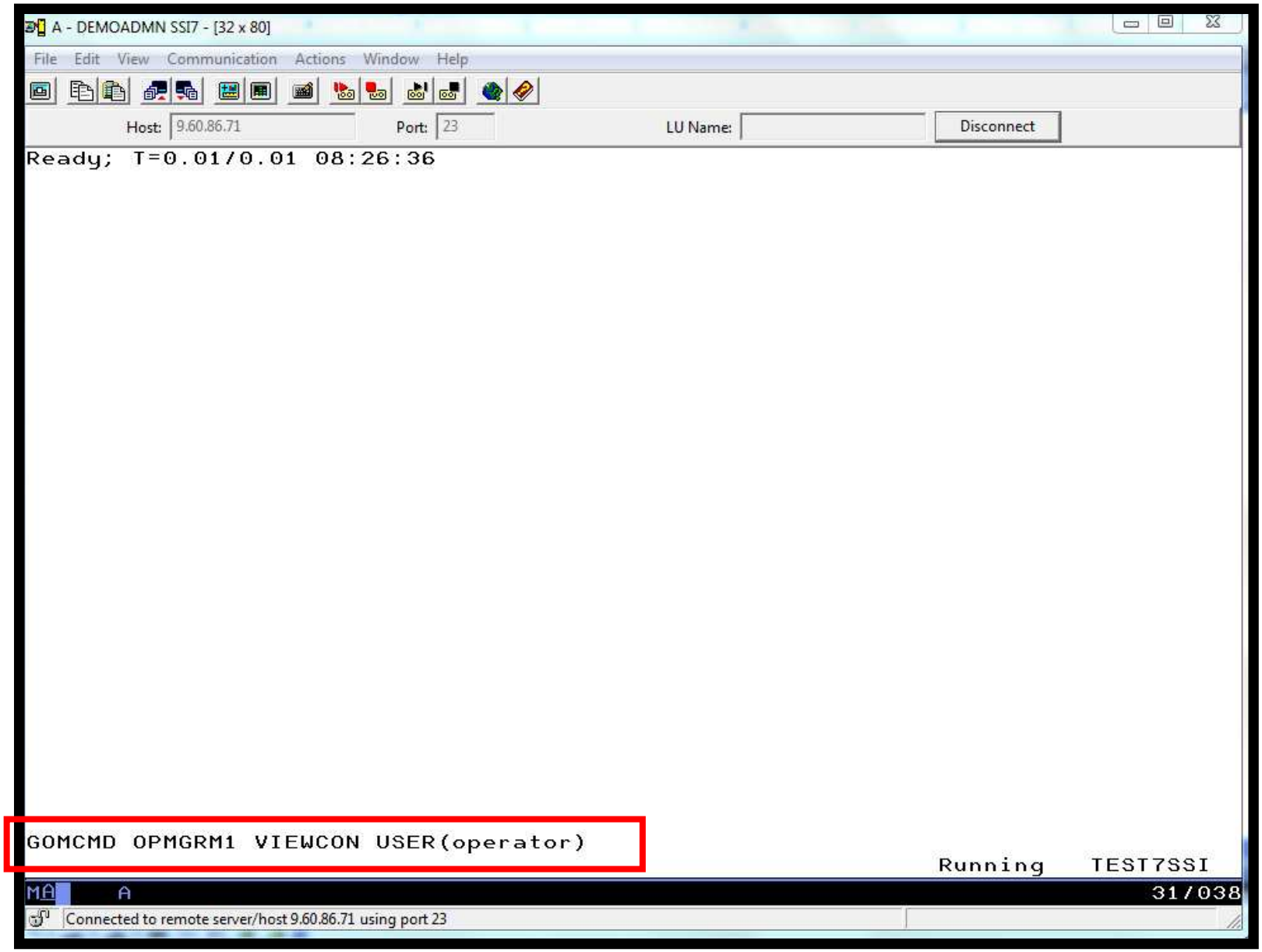
- **View the console of a Linux guest and issue Linux commands**

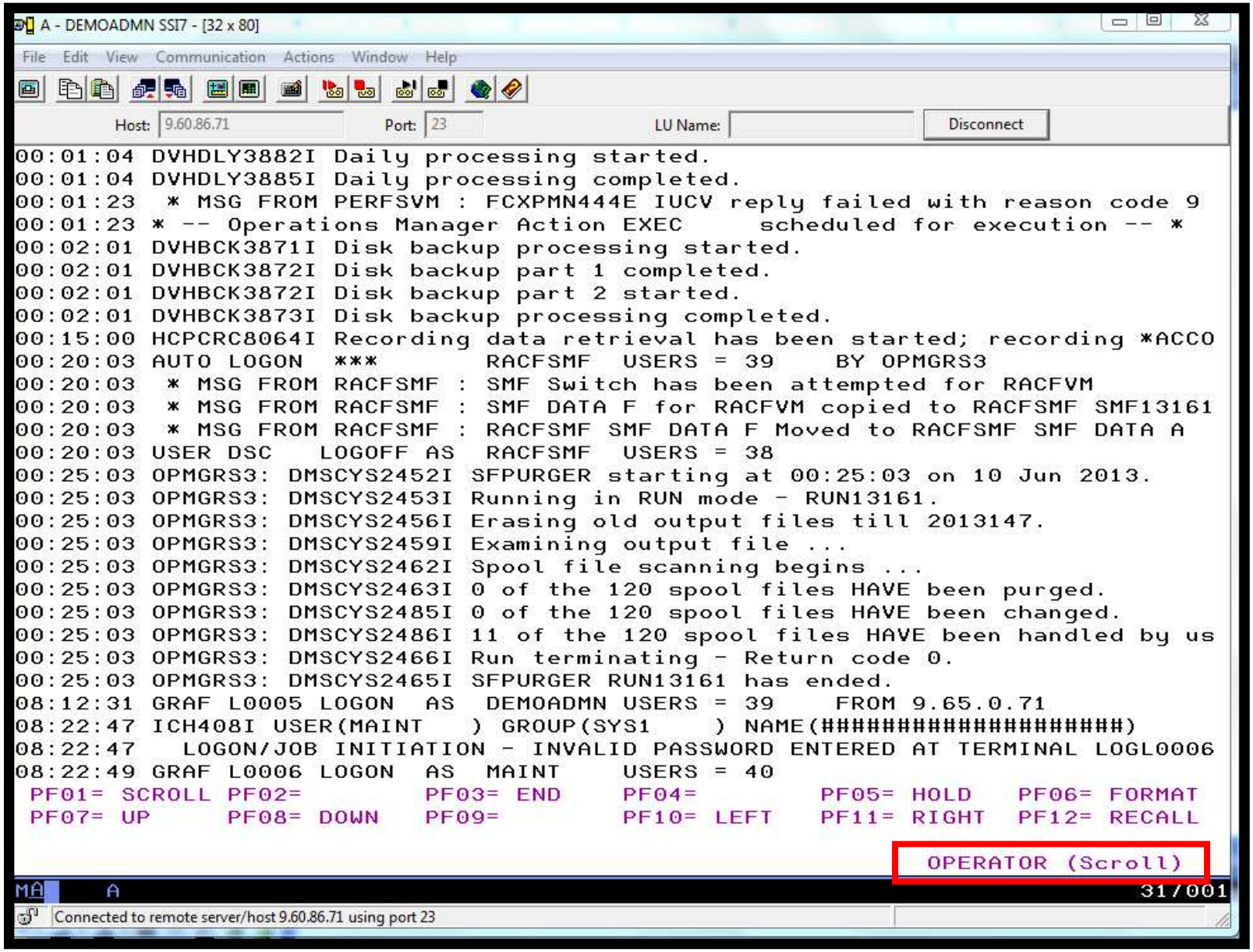
```
gomcmd opmgrm1 viewcon user(sles11c)
echo hello world
```

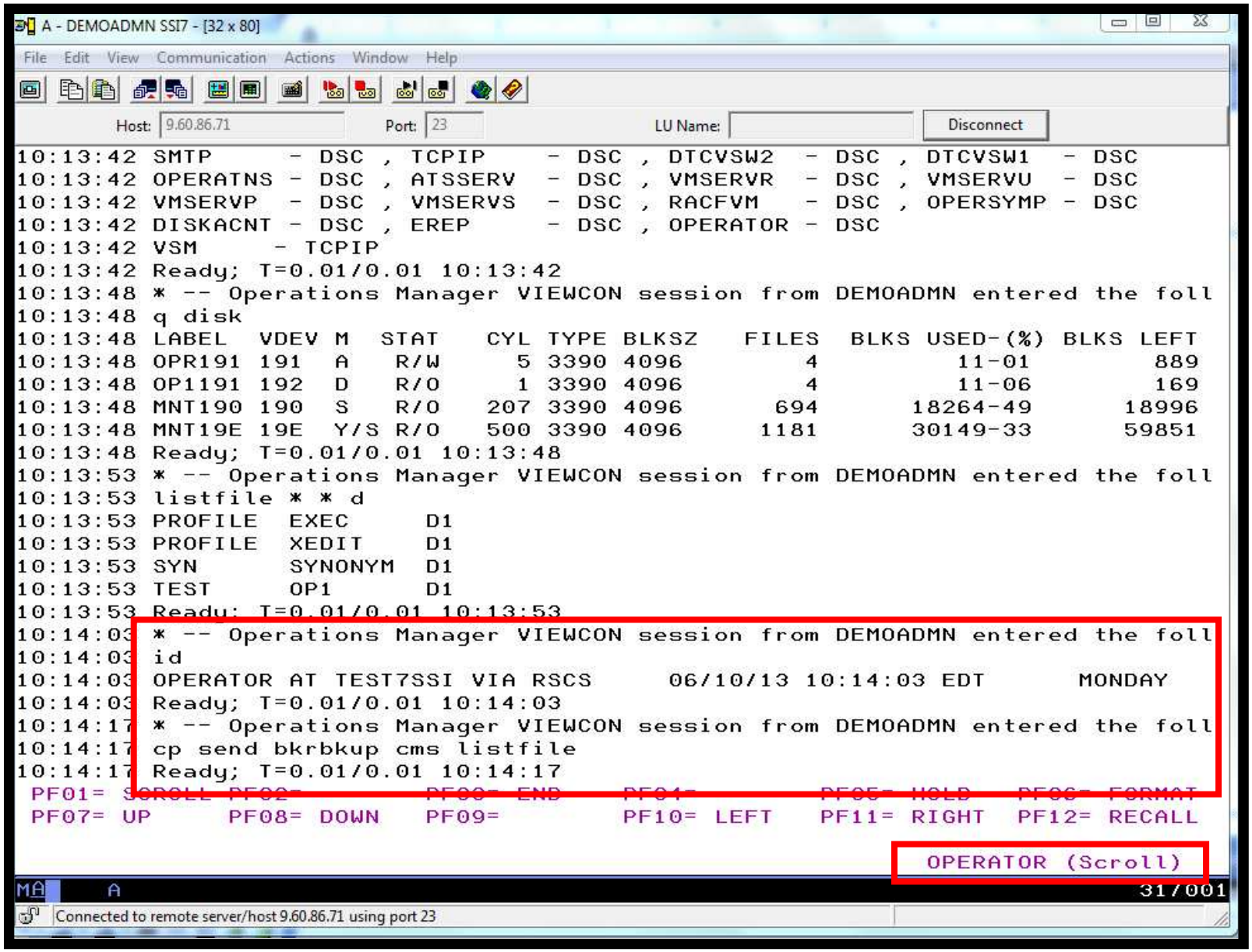
- **View the syslog data from a Linux guest**

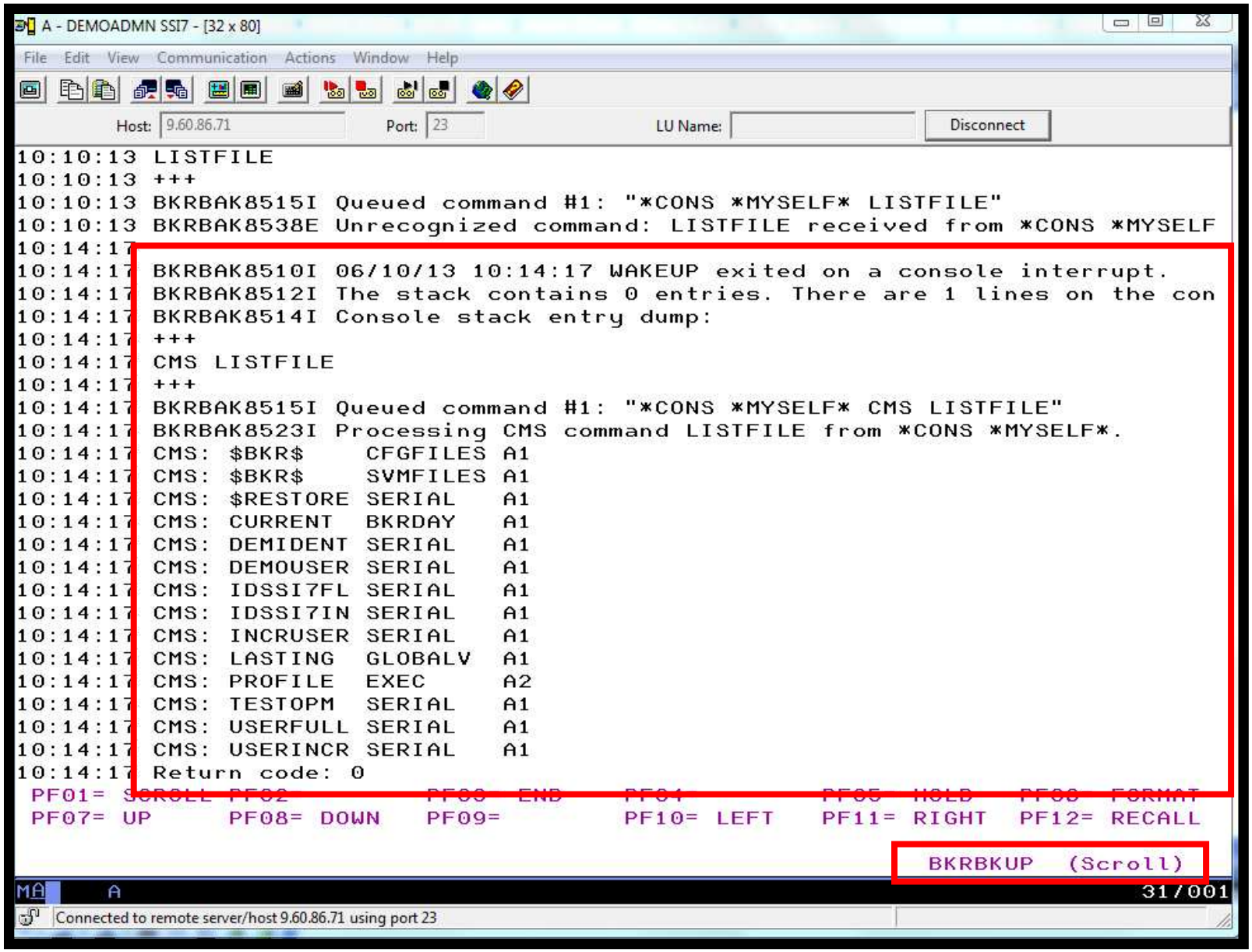
```
gomcmd opmgrm1 viewcon user(lxsyslog)
```

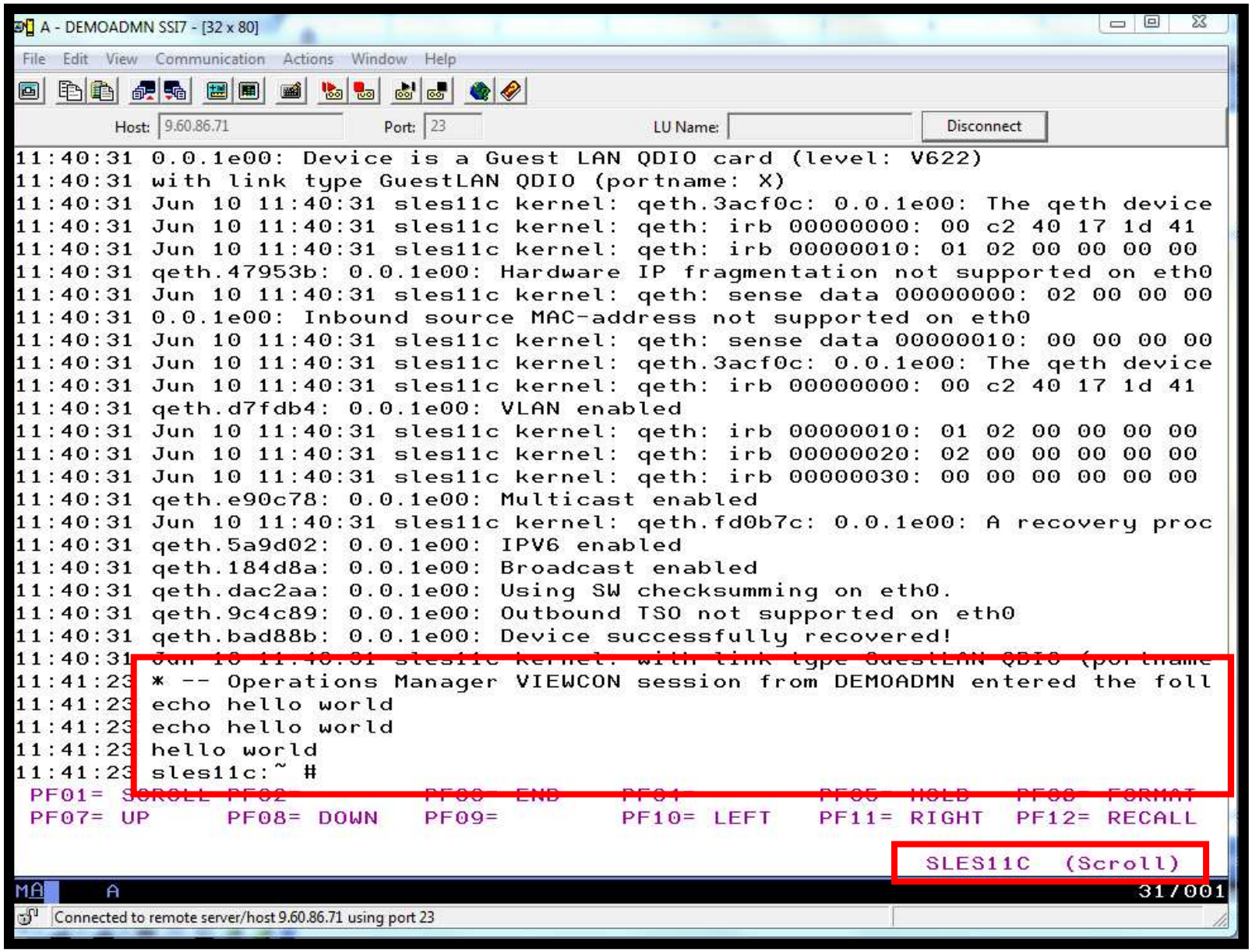


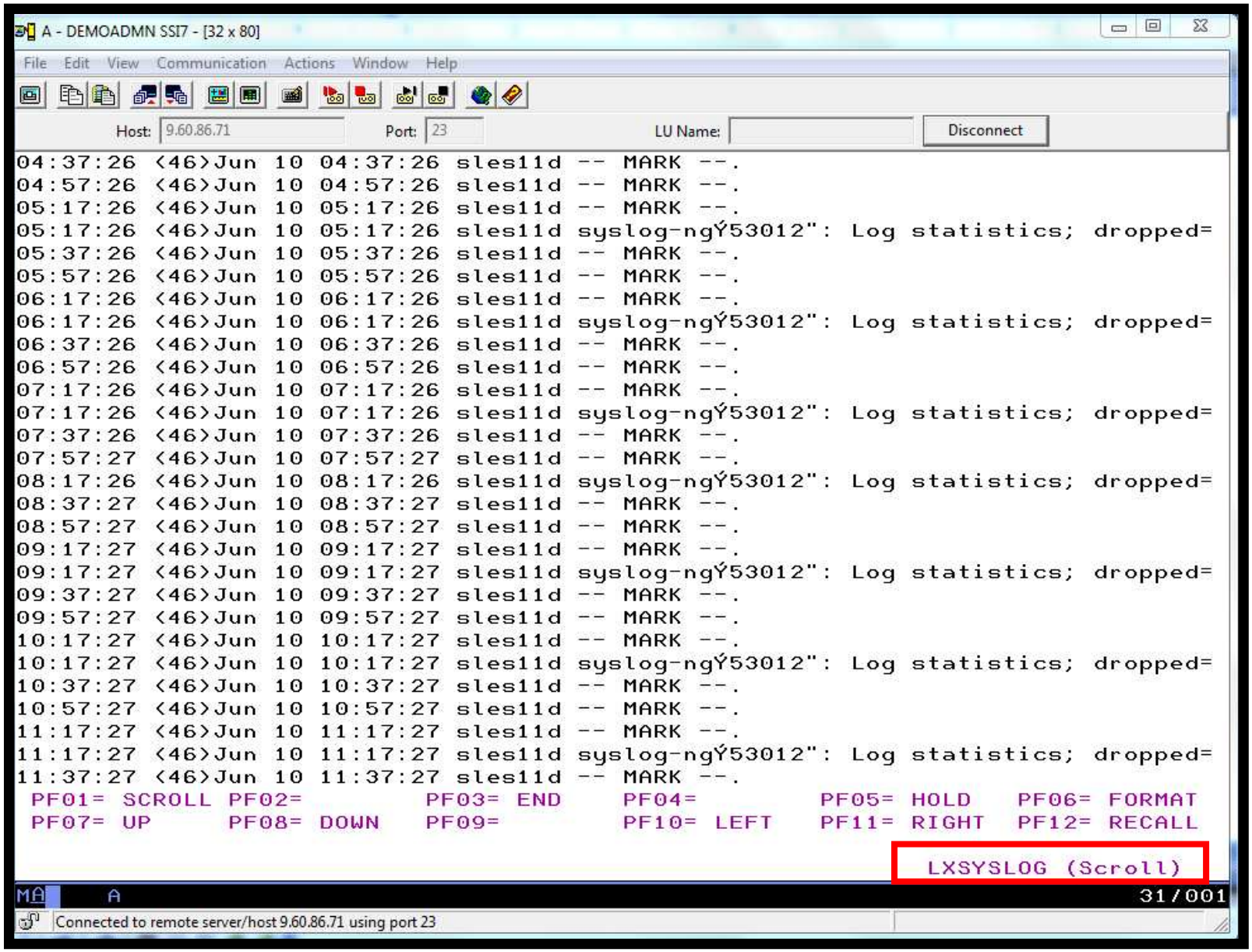












Scenario 17: How Do You Do That?

- **For console data**
  - Make OPMGRM1 the secondary user of OPERATOR and SLES11D
    - Via CONSOLE statement in CP directory entry (recommended)
    - Via SET SECUSER command
- **For Linux syslog data**
  - Set up TCP/IP listener for syslog data

```
*
DEFTCPA NAME(LNXSYSLG),+
  TCPUSER(TCPIP),+
  TCPAPPL(GOMRSYL),+
  TCPADDR(000.000.000.000),+
  TCPPORT(00514),+
  PARM(LXSYSLOG03330417UTF8)
```

  - Update TCP/IP configuration to allow Operations Manager to listen for UDP traffic on the specified port(s)
    - Port 514 used here
  - Update the Linux guest to send its syslog data to the IP address and port of your z/VM system
  - Refer to white paper on Operations Manager web site for details

धन्यवाद
Hindi

多謝
Traditional Chinese

감사합니다
Korean

Спасибо
Russian

Gracias
Spanish

Thank You
English

شكراً
Arabic

Obrigado
Brazilian Portuguese

Danke
German

多谢
Simplified Chinese

Grazie
Italian

Merci
French

நன்றி
Tamil

ありがとうございました
Japanese

ขอบคุณ
Thai