IBM Software

Automation Scenarios for a z/VM Cluster and Linux on System z Guests

SHARE Session 11809

Tracy Dean, IBM
tld1@us.ibm.com

August 2012

Agenda

- **Recommended practices**
- **Requirements for these scenarios**
- **Overview of product being used**
  - IBM Operations Manager for z/VM
  - Considerations for z/VM V6.2 SSI
    - Apply to many automation solutions
- **Automation scenarios**
  - Can be product agnostic
  - Live demos
  - Configuration options and sample code
- **Summary and reference information**

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
- **Automation and operational monitoring**
  - 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

Recommended Practices – Operational Management

Generate alerts and/or automatically recover from

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

Schedule automated system maintenance procedures

- Spool cleanup based on policies
- Minidisk cleanup (from logs) – may include archiving
- Orderly shutdown and relocation of critical guests to another member in SSI cluster

IBM Software

Requirements

Implementing these Scenarios

Automation requirements for z/VM system

- **Take an action based on a message on a console**
  - Provide data from the message to the action
- **Send commands to Linux guests**
- **Trigger an action if spool usage reaches a specified percent full**
  - Provide data about spool usage to the action
- **Trigger an action if page space usage reaches a specified percent full**
  - Provide data about page space usage to the action
- **Chain any actions (triggered by messages, schedules, etc.)**
- **Suspend and resume message rules, schedules, spool/page monitors, etc.**
- **Issue commands real-time on a service machine console**
- **Add messages to a console view from local or remote sources**
- **Detect a user ID logging off**
- **Suppress lines when viewing a console**

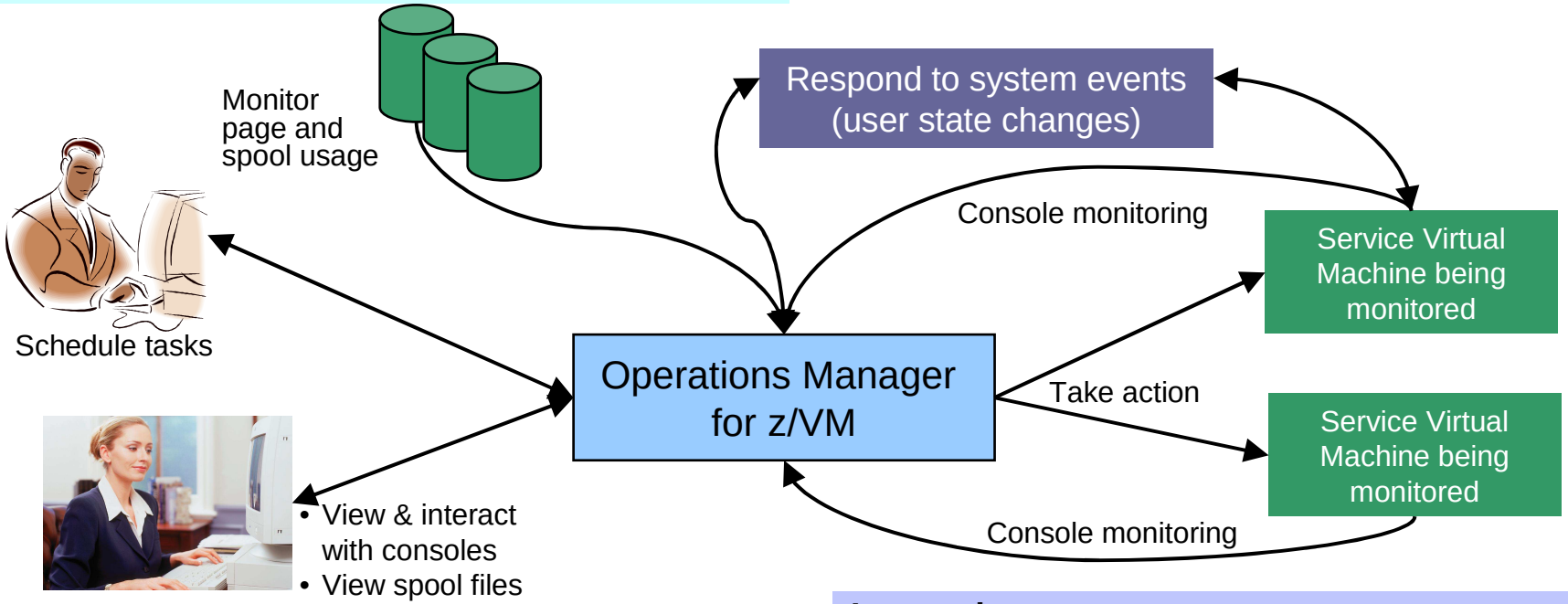
IBM Software

Automating Operations
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



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**

Monitor Service Machine Consoles

Operations Manager

Test Data
TEST Message 1
TEST Message 2
TEST 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
…

syslog data
slog Message 1
slog Message 2
slog Message 3
…

Filtered

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
…

Unfiltered

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

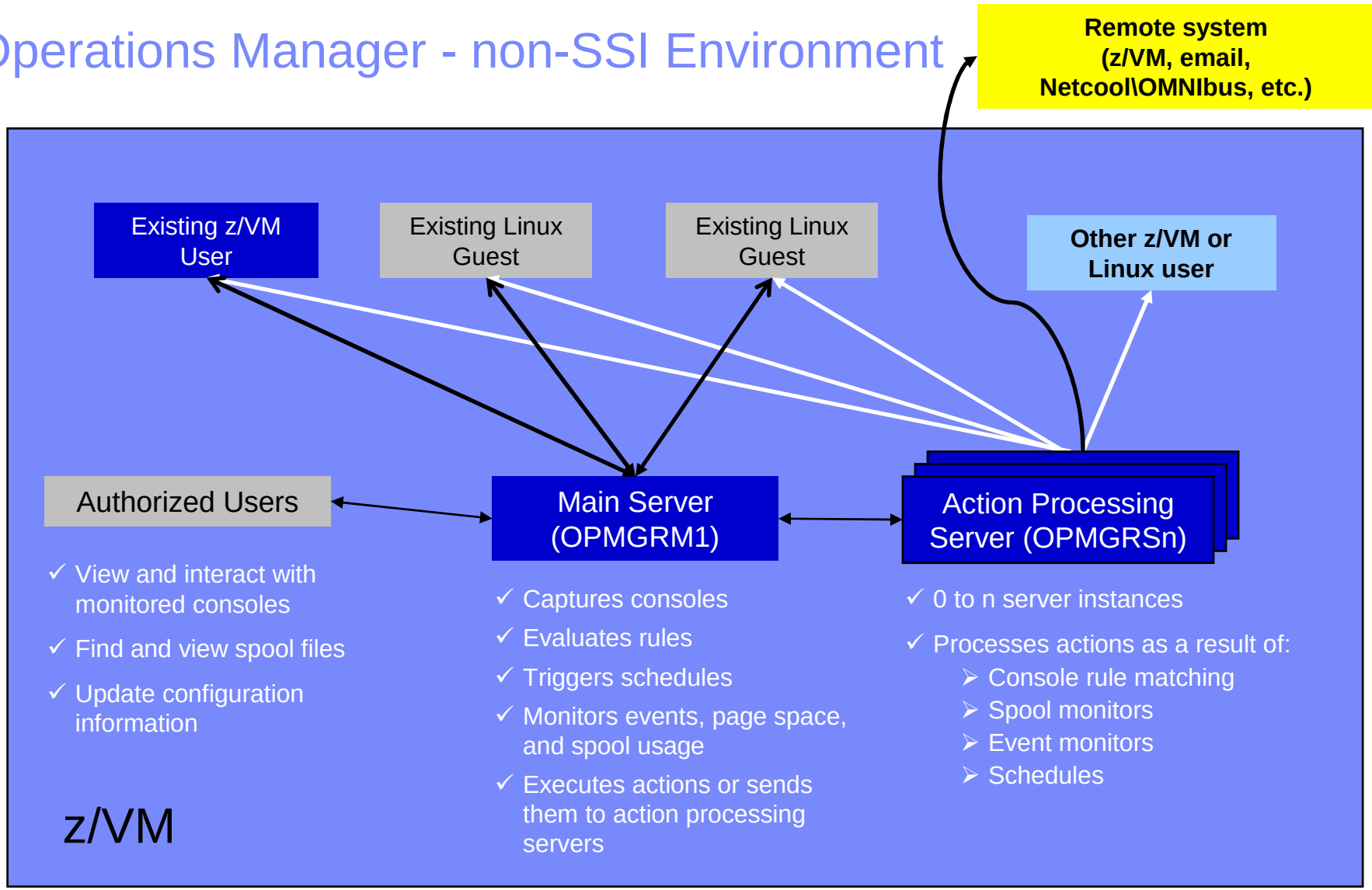
Monitor Service Machines

- **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
    - Change color, highlight, hold, or suppress a console message
    - CP or CMS commands
    - REXX EXECs
- **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
- **Take multiple actions based on one message**
  - Chain actions together
- **Rules apply to consoles received by local Operations Manager server**

View and Interact with Consoles

- **Authorized users can view live consoles of monitored service machines and guests**
  - Multiple users can view the same console simultaneously
  - No need to logon to the service machine to see its console
  - Test data and Linux syslog data treated as a “console”
  - Views can be defined to look at a group of consoles in one view
- **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**
- **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

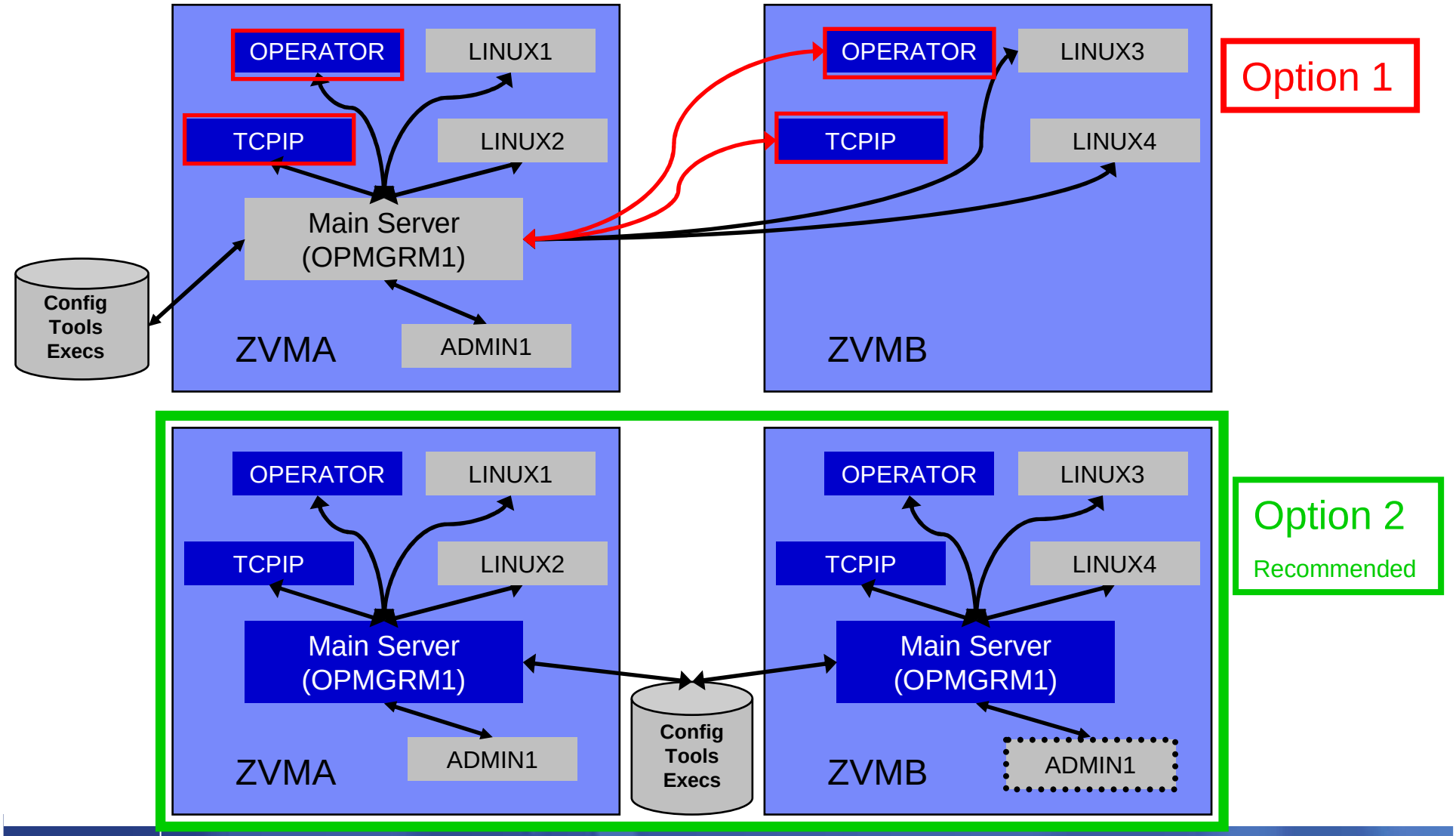
Operations Manager - non-SSI Environment



Single Config User

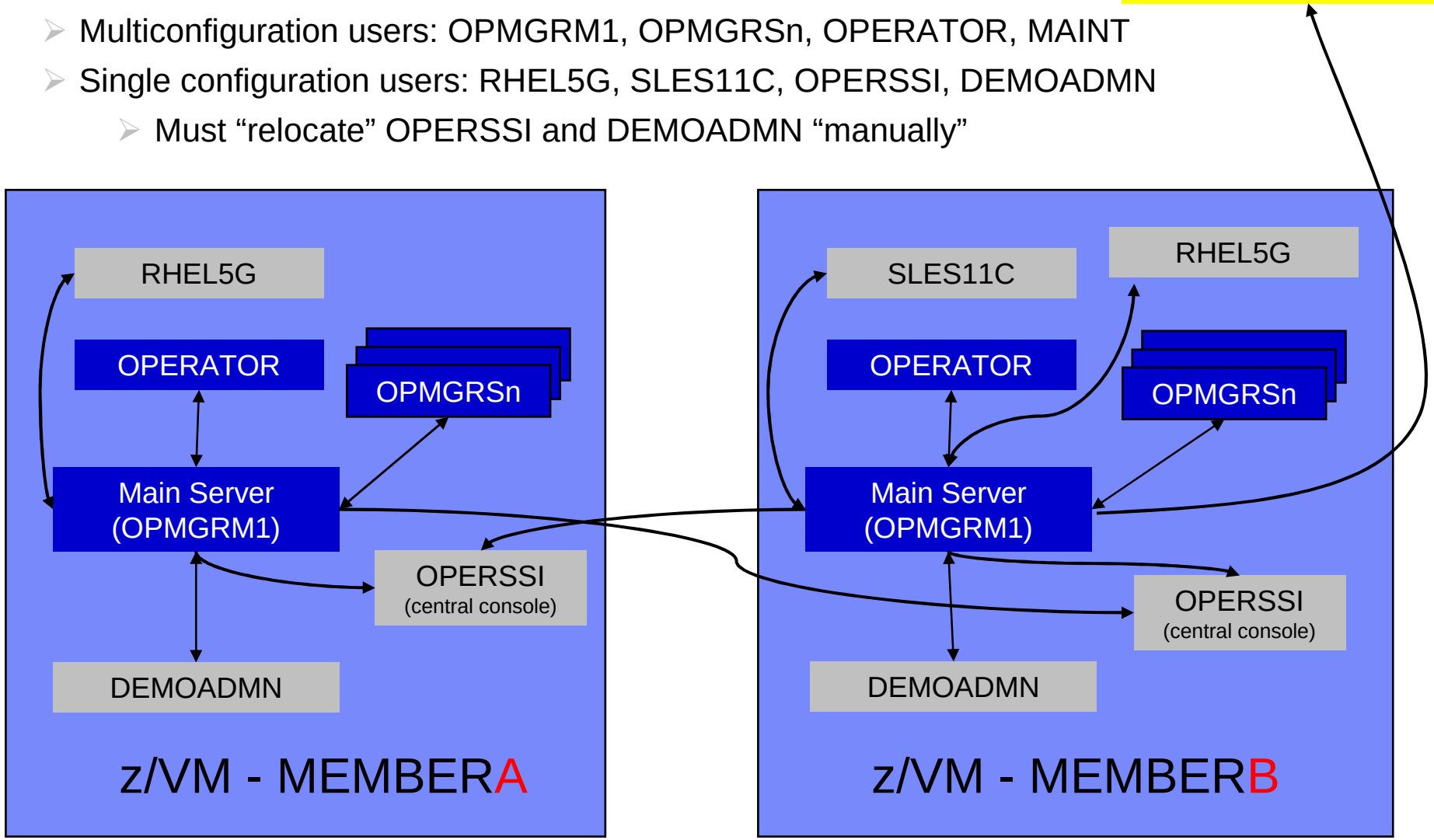
Multiconfig User

SSI Considerations for Console Monitoring



Operations Manager in SSI Cluster - Example

- Multiconfiguration users: OPMGRM1, OPMGRSn, OPERATOR, MAINT
- Single configuration users: RHEL5G, SLES11C, OPERSSI, DEMOADMN
  - Must “relocate” OPERSSI and DEMOADMN “manually”



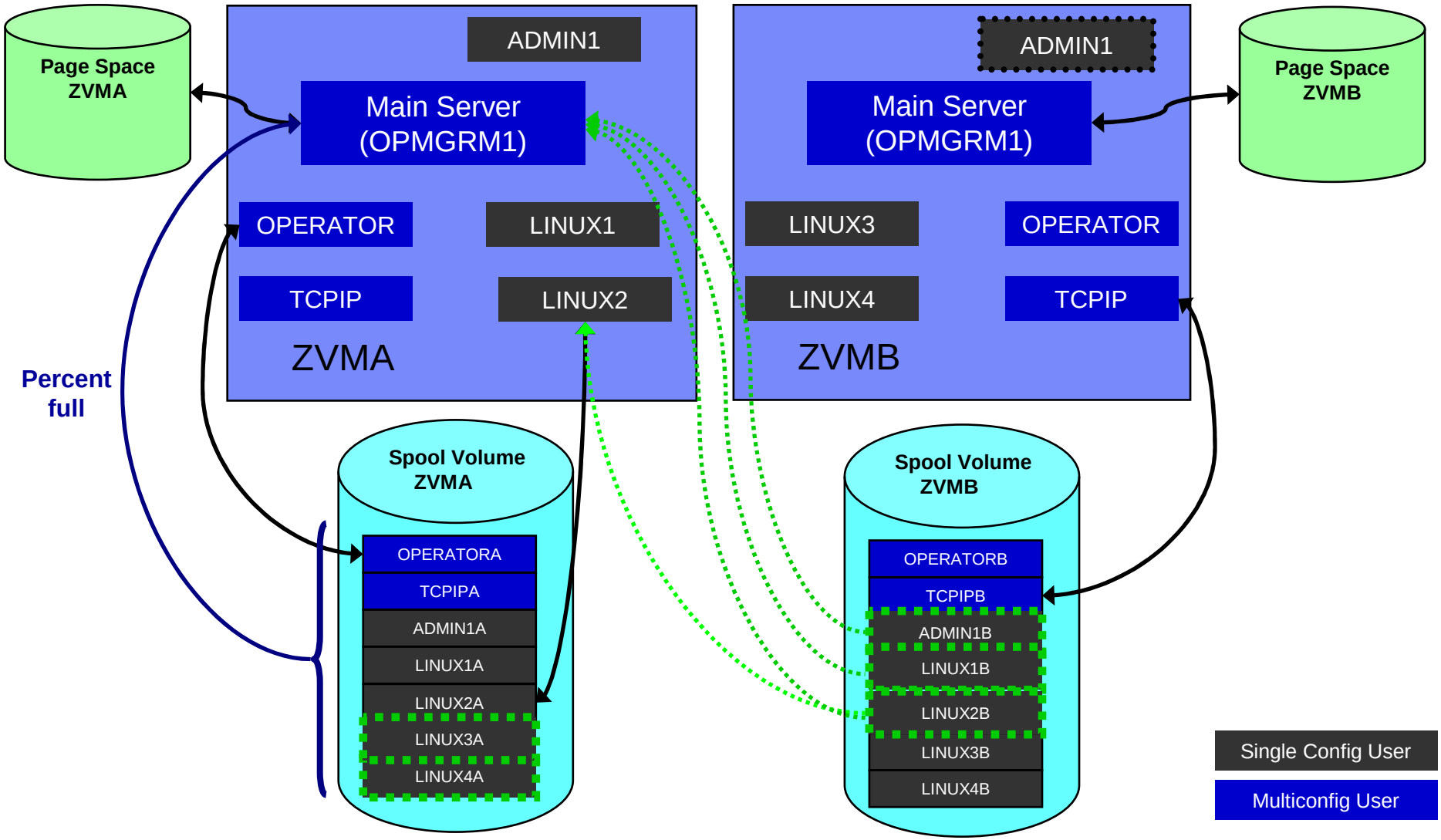
Monitor Service Machines - Considerations

- **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 when 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 Single System Image Environments**
  - Have all consoles monitored by an Operations Manager server on the same member as the monitored guest
    - 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

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
    - View the contents of an individual spool file
    - Purge, transfer, or change a spool file

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
- **Spool data – visibility to authorized users**
  - Spool data for multiconfiguration users
    - Only 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 users
    - Files created while logged onto that member are always visible on that member
    - Files owned by the user but created while logged onto another member are only visible to the local member when the user is logged on (or running disconnected) on the local member
- **Another way of putting it**
  - Spool data created on a member is always visible on that member
    - Whether the owning user is currently logged on or not
    - This includes
      - Files created by single configuration users while logged onto that member
      - Files created by multiconfiguration users with subconfig info for that member
  - Spool data owned by single configuration users is seen by the local member when the user is logged on (or running disconnected on) the local member
    - Even if data was originally created while logged onto another member of the cluster
- **Recommendation**
  - Have an Operations Manager server on each member to monitor spool and page space

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
  - 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 and spool monitoring
- **No impact from SSI**

Respond to System Events

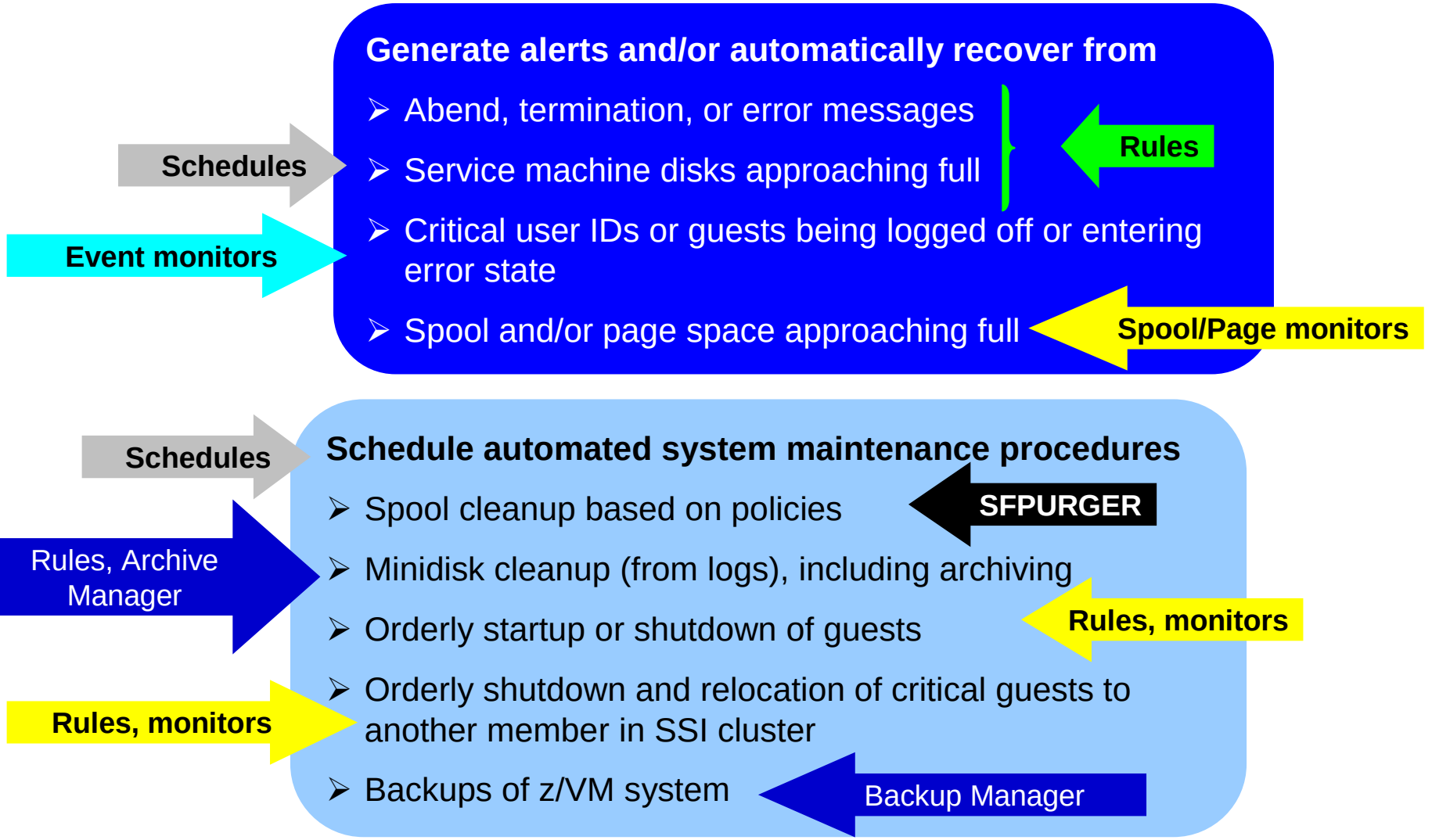
- **Create monitors for z/VM system events (*VMEVENT) related to user IDs**
  - Class 0

    0 - Logon
    1 - Logoff
    2 - Failure condition (typically CP READ)
    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 - Time bomb 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 and console and spool monitors

Dynamic Configuration

- **Initial configuration file loaded at startup**
  - May imbed other configuration files
- **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
  - Load a new or updated configuration file
  - Commands in action routines
  - Request reload from user IDs on other members of a cluster
    - Use SMSG OPMGR1 at <member> CONFIG …

Recommended Practices – Operational Management



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

Reference Information

- **Product Web site**
  - Start at http://www.ibm.com/software/stormgmt/zvm/
  - Product pages include
    - Publications
    - Pre-requisites
    - Announcements
    - Presentations
    - White papers
    - Support
- **e-mail**
  - Mike Sine, sine@us.ibm.com, Technical Marketing
  - Tracy Dean, tld1@us.ibm.com, Product Manager
- **White papers on Operations Manager website (Library page)**
  - 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
- **White paper and presentation on Backup and Restore Manager website (Library page)**
  - Getting Started with Installation, including SFS server creation and installation of Backup Mgr
  - Backing up z/VM and Linux on System z – Tivoli Storage Manager vs Backup Manager

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**
   a. Using POSTZMSG interface to Netcool/OMNIbus
   b. Using SNMP interface to Netcool/OMNIbus
3. **Send an 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 a 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**

IBM Software

Automation Scenarios

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

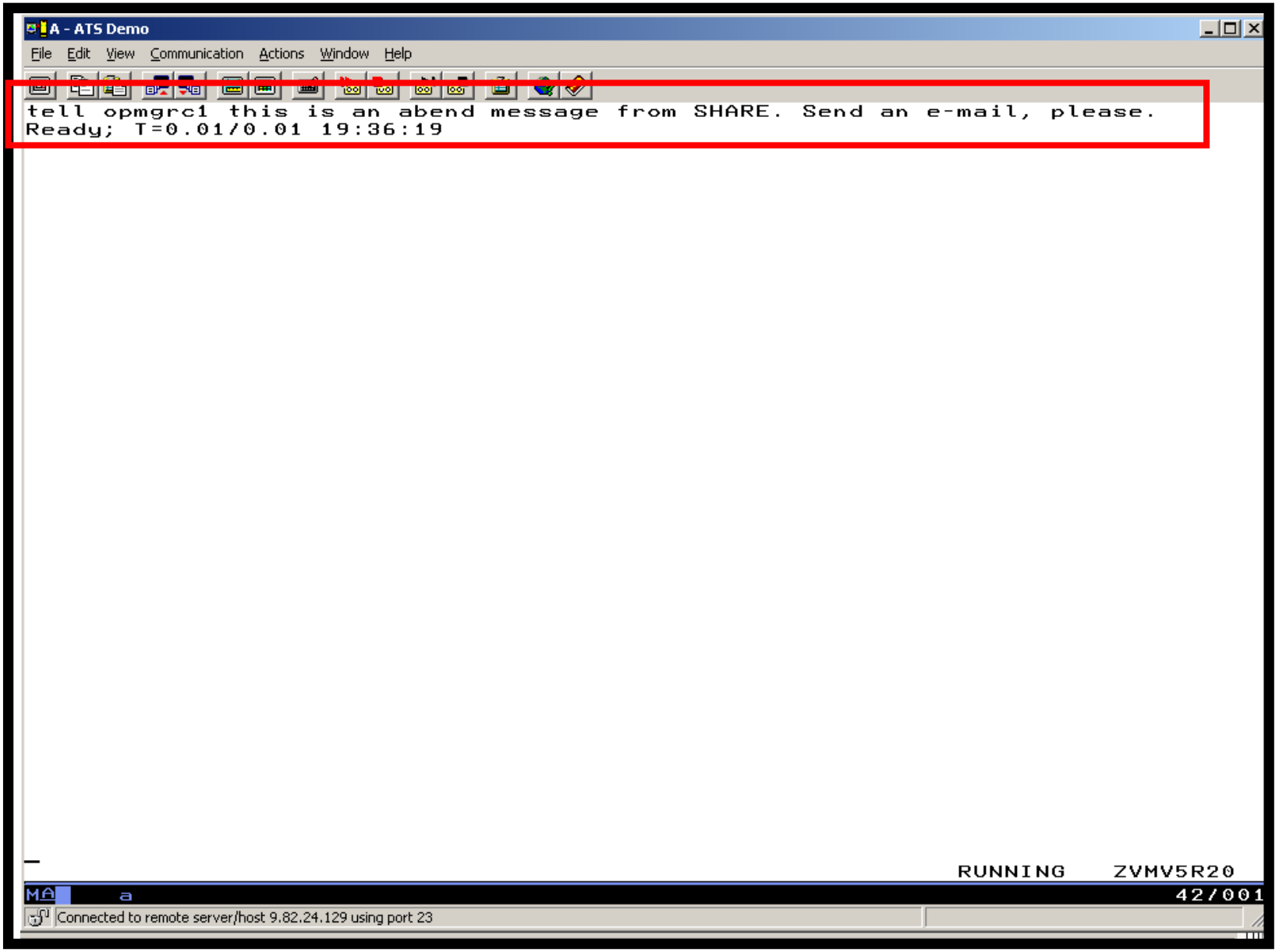
- **From any VM user ID:**

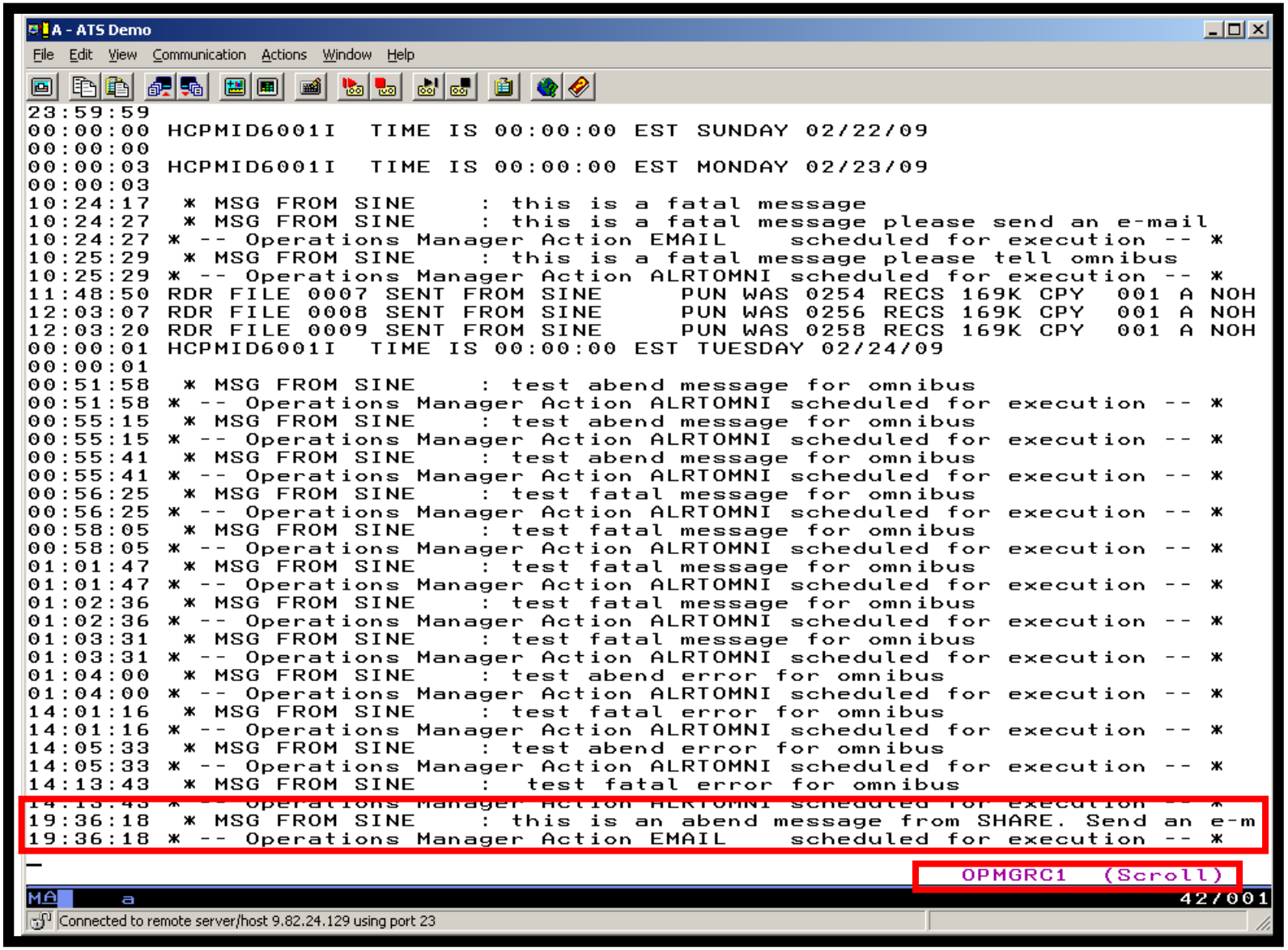
```
tell opmgrc1 this is an abend message from SHARE. Send an e-mail, please.
```

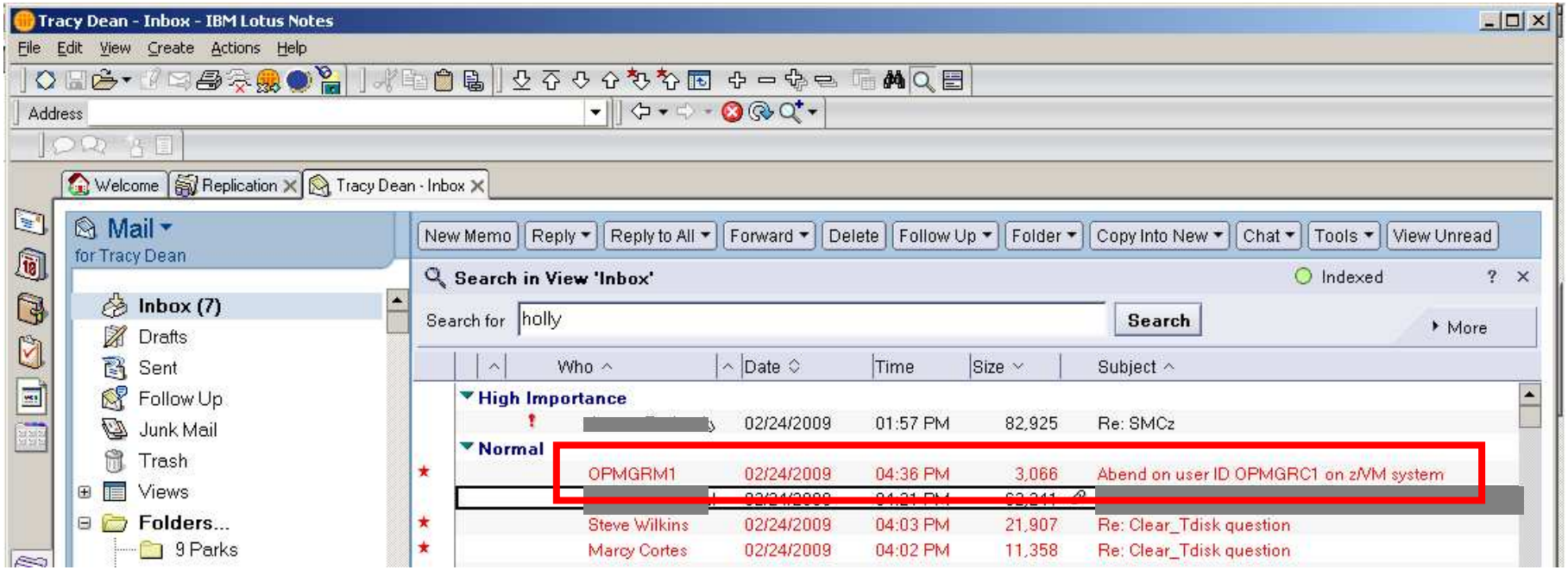
- **From an authorized VM user ID, view the console of OPMGRC1:**

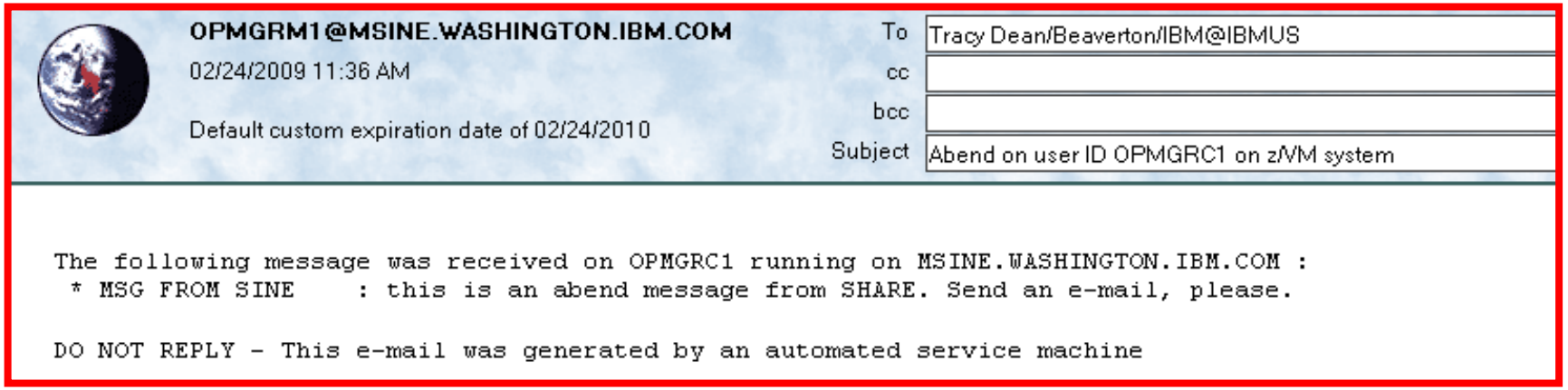
```
gomcmd opmgrm1 viewcon user(opmgrc1)
```

- **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*),+
  EXUSER(ESMTS112),+
  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*),+
  EXUSER(ESMTS112),+
  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.
```

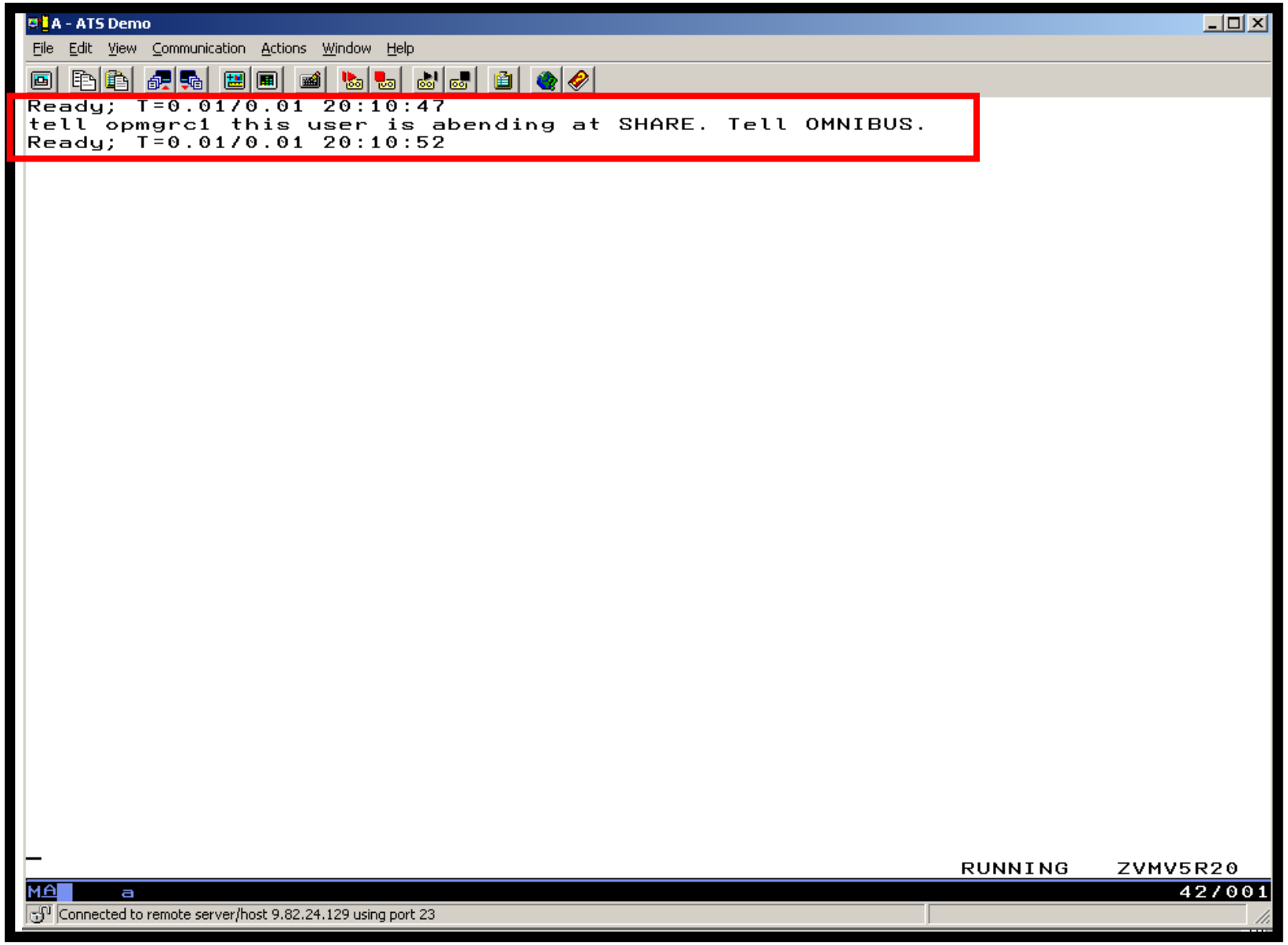
- **From an authorized VM user ID, view the console of OPMGRC1:**

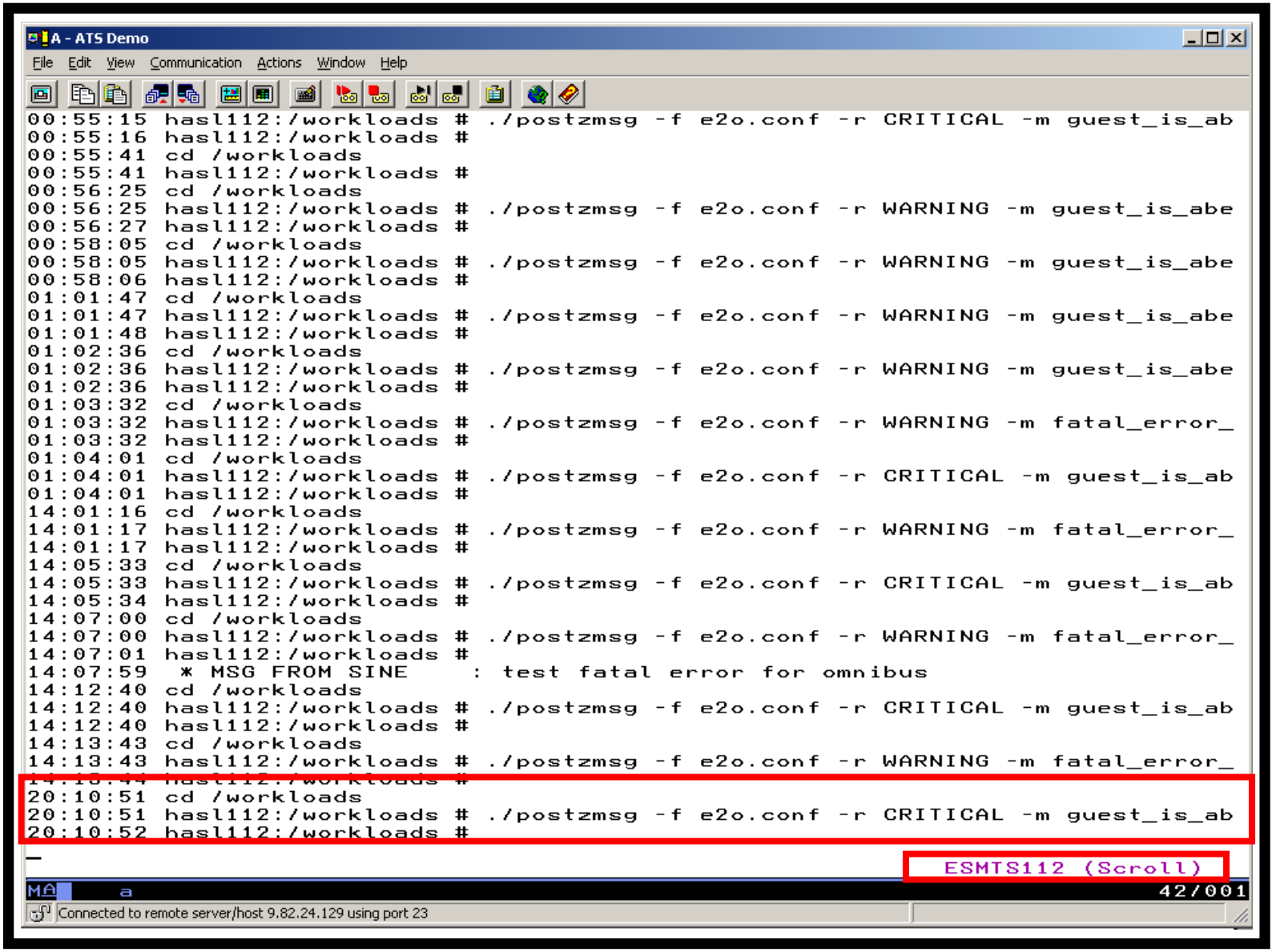
```
gomcmd opmgrm1 viewcon user(opmgrc1)
```

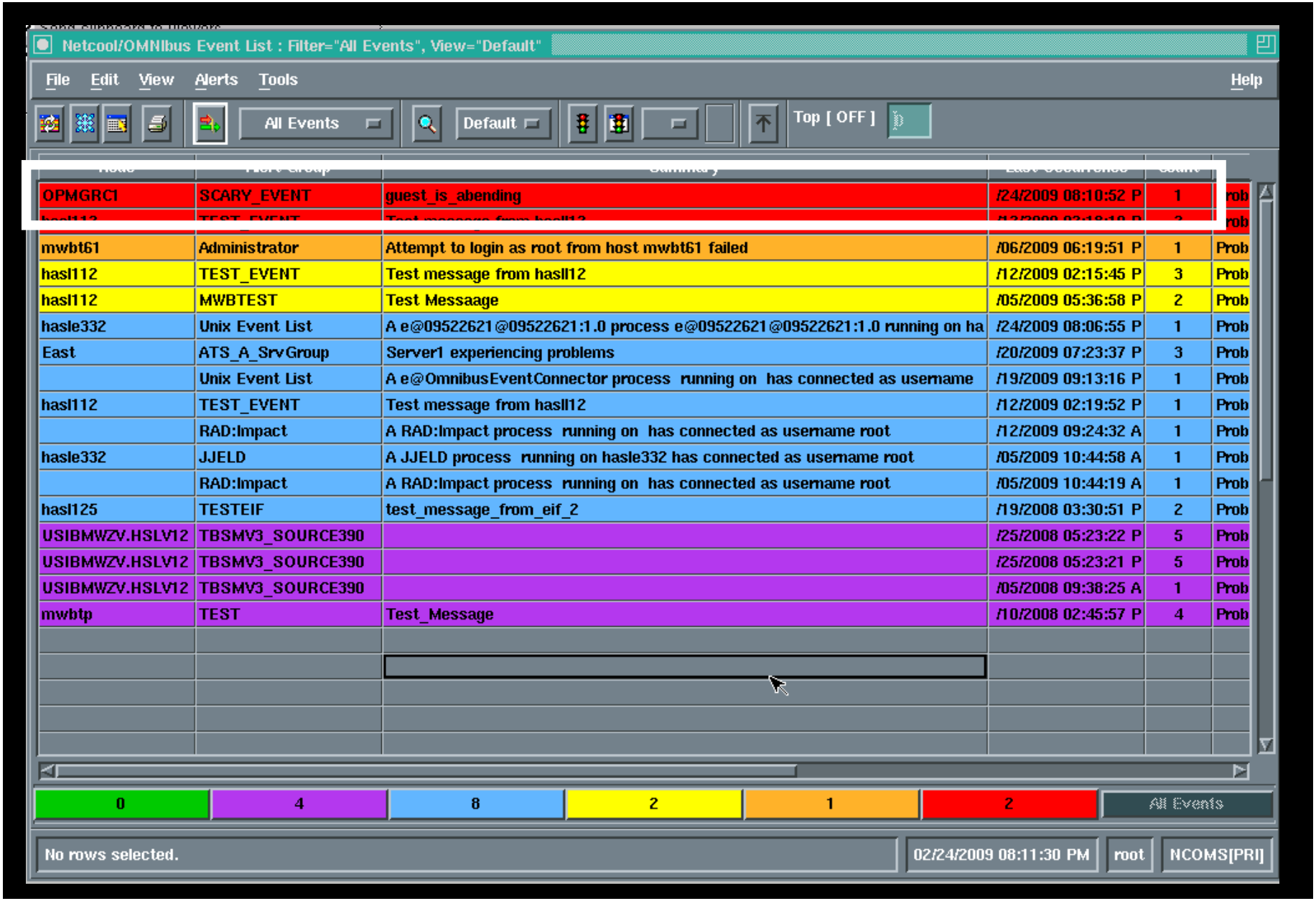
- **From an authorized VM user ID, view the console of the Linux guest that runs POSTZMSG:**

```
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)
- **Send an alert to OMNIbus if this word appears on a console**
  - Use SNMPTRAP command on z/VM
- **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

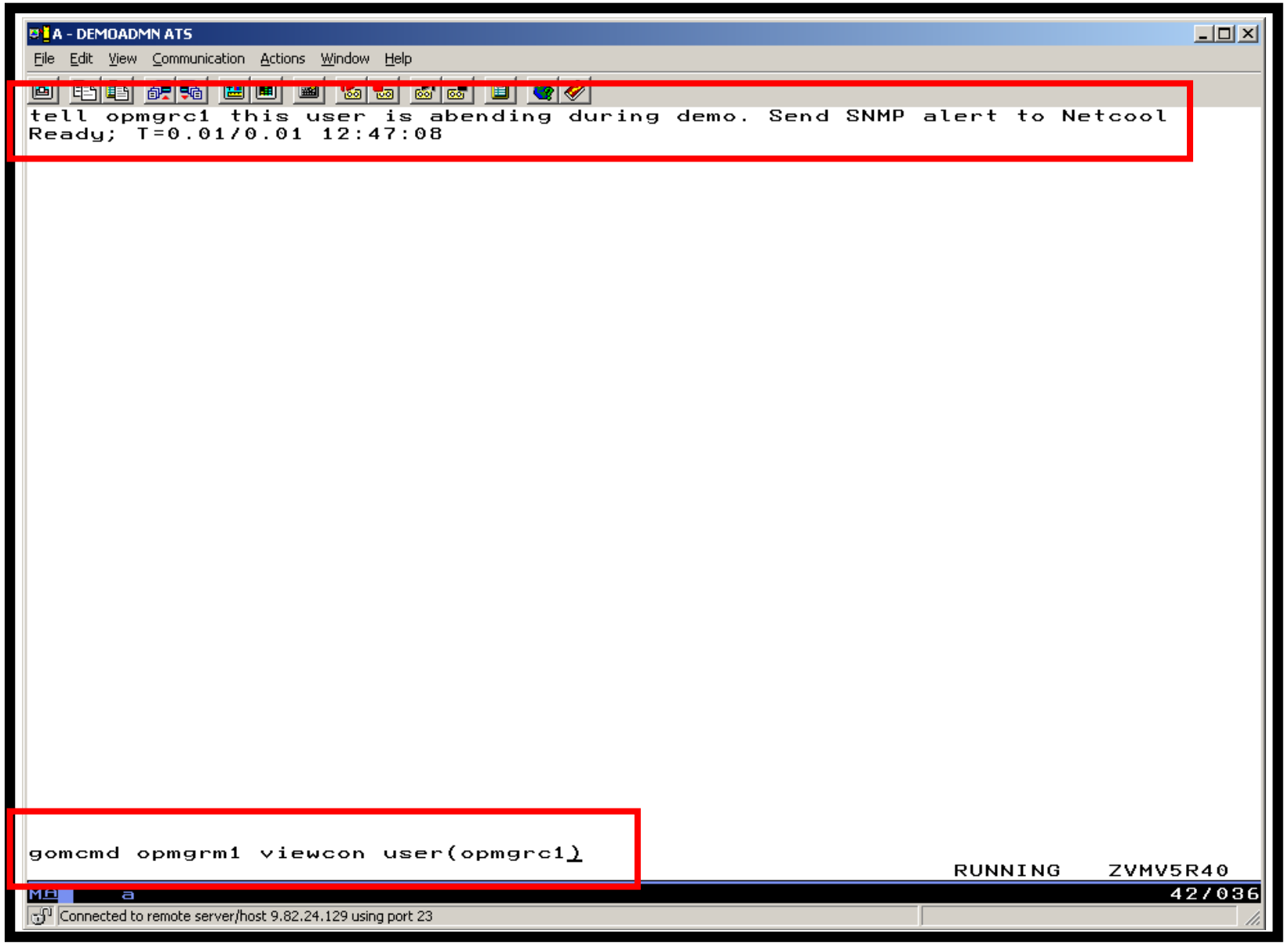
- **View “All Events” in OMNIbus**
- **From any VM user ID:**

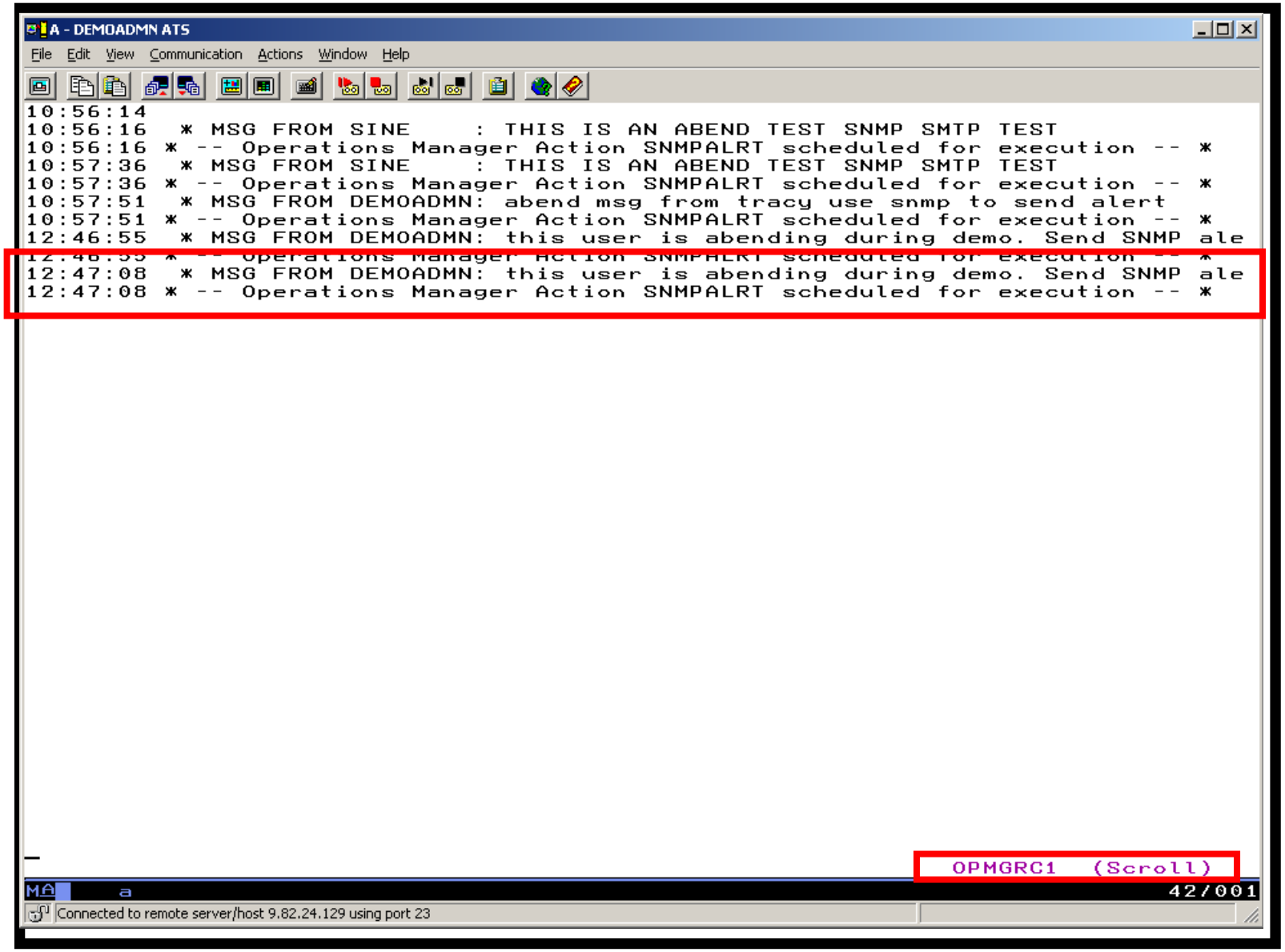
```
tell opmgrc1 this user is abending during demo. Send SNMP alert to Netcool
```

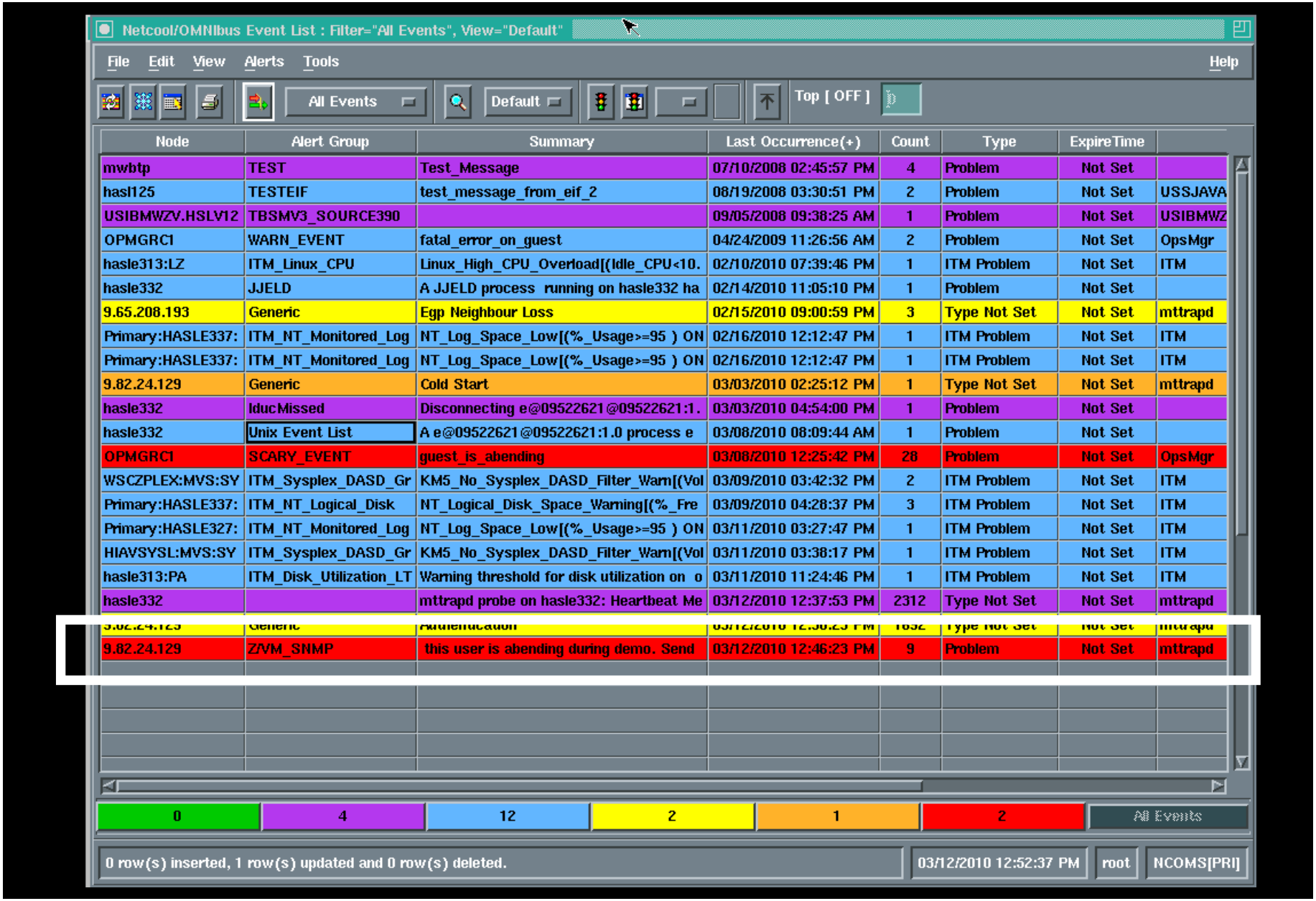
- **From an authorized VM user ID, view the console of OPMGRC1:**

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

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







Scenario 2b: How Do You Do That?

Rule and action in Operations Manager:

```
*
* Send an alert to OMNIbus using SNMP for abend
* msgs on consoles
DEFRULE NAME(ABNDSNMP),+
  MATCH(*abend*snmp*),+
  ACTION(SNMPALRT),+
  PARM(ABEND)
*
DEFACTN NAME(SNMPALRT),+
  COMMAND(EXEC SNMP2OMN &T),+
  ENV(SVM)
```

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**
- **Usage exceeds the specified limit**
  - For demo purposes, we'll dynamically resume (re-activate) an existing spool monitor that requires the spool to only be ??% full
    - Must reactivate on each member of a cluster
- **Automatically send a message to a central console for the entire cluster**
- **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 3b: Send an Email if Spool Usage is Too High

- **Operations Manager monitors the spool usage (percent full)**
- **Usage exceeds the specified limit**
  - For demo purposes, we'll dynamically resume (re-activate) an existing spool monitor that requires the spool to only be 25% full
- **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(splfull2)
```

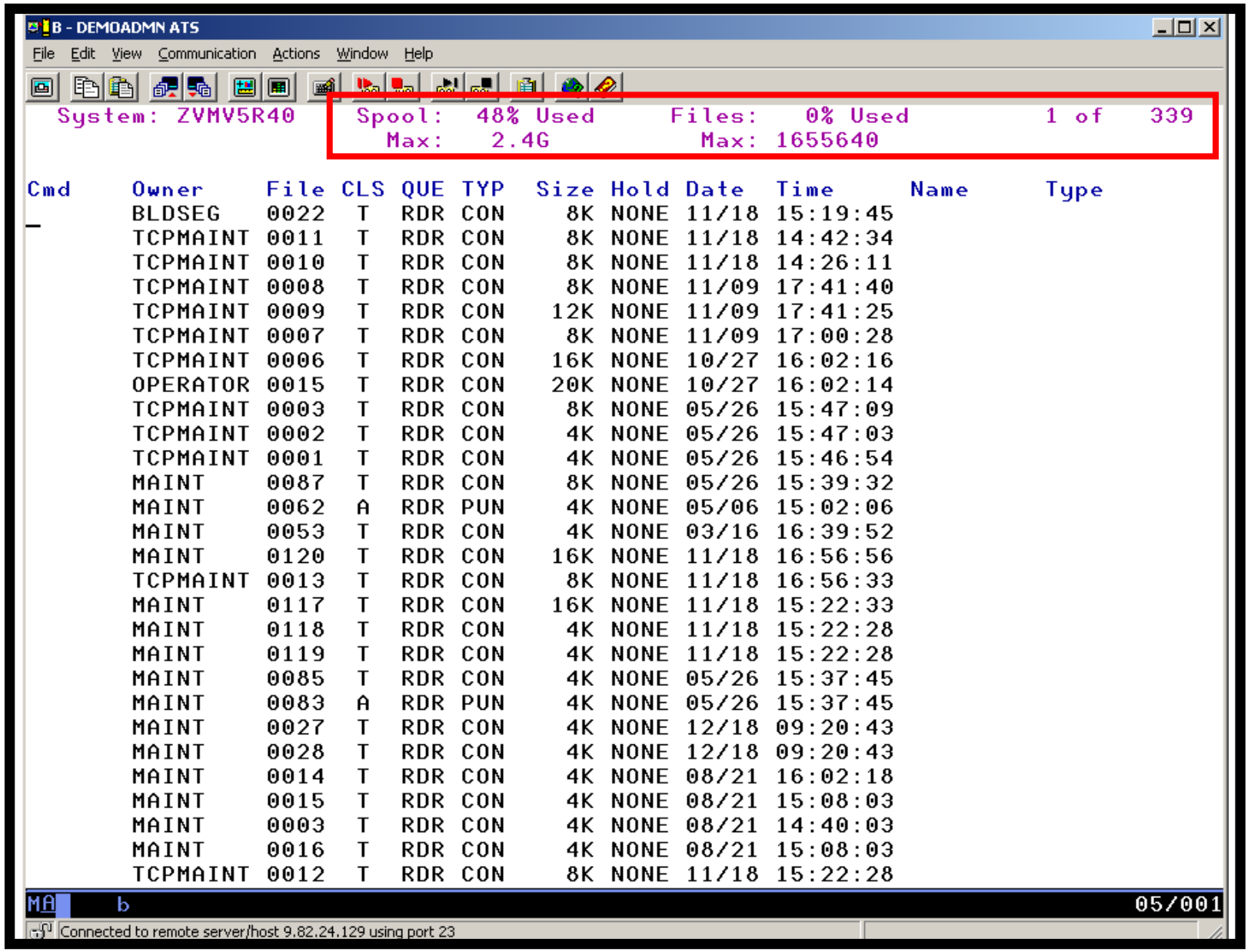
- **Check the Operations Manager log to see the spool monitor triggered:**

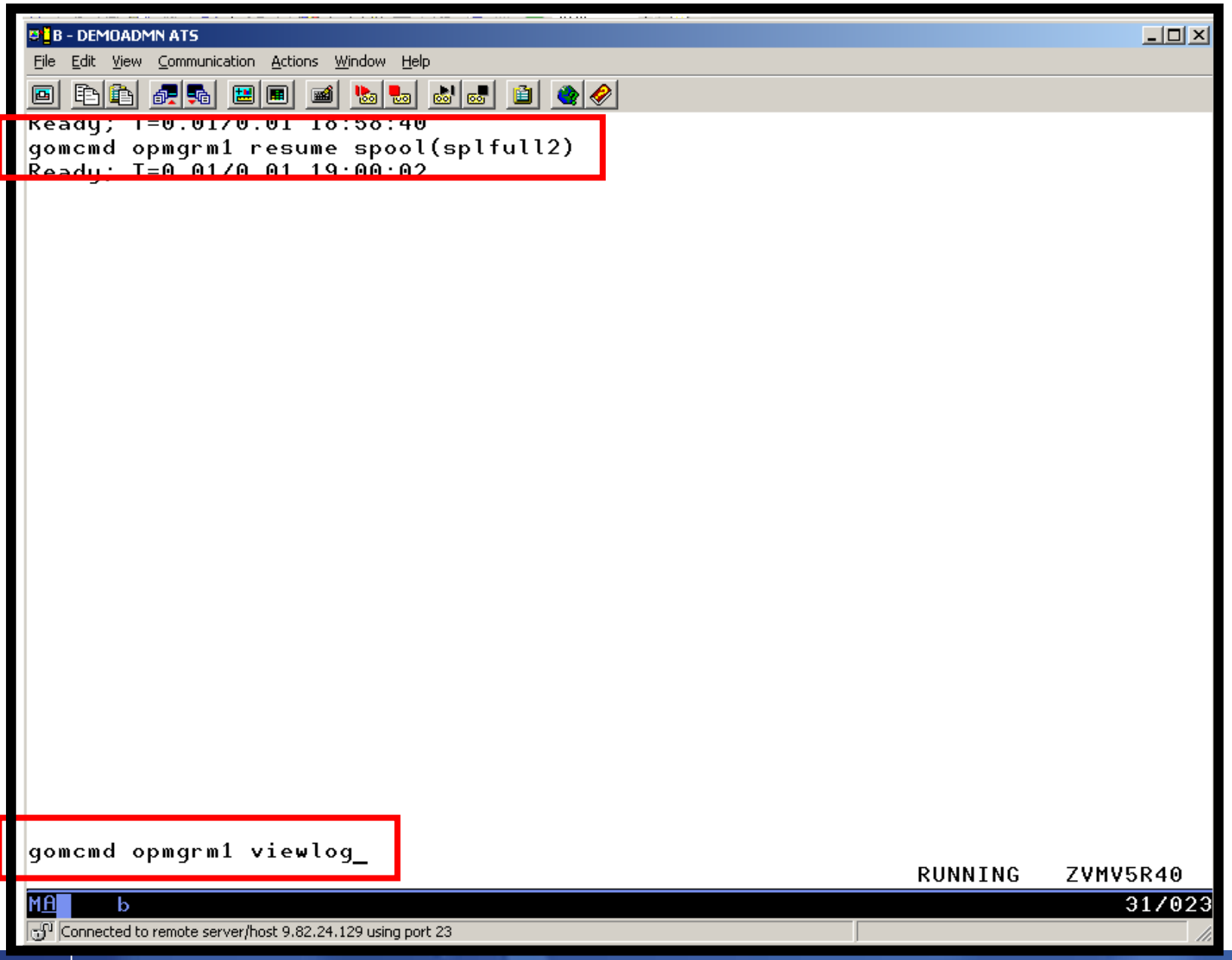
```
gomcmd opmgrm1 viewlog
```

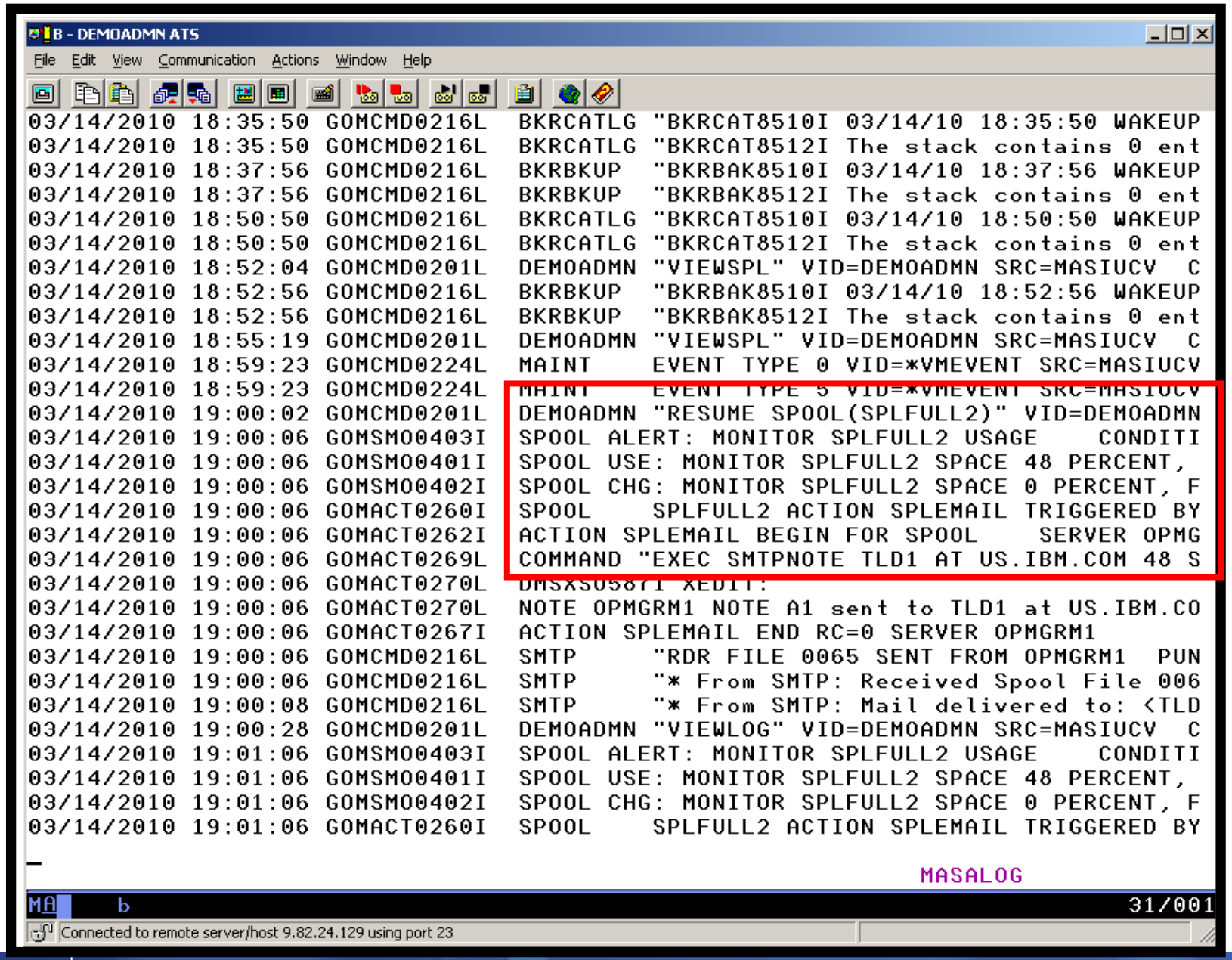
- **Check the inbox of the appropriate person to see the e-mail**

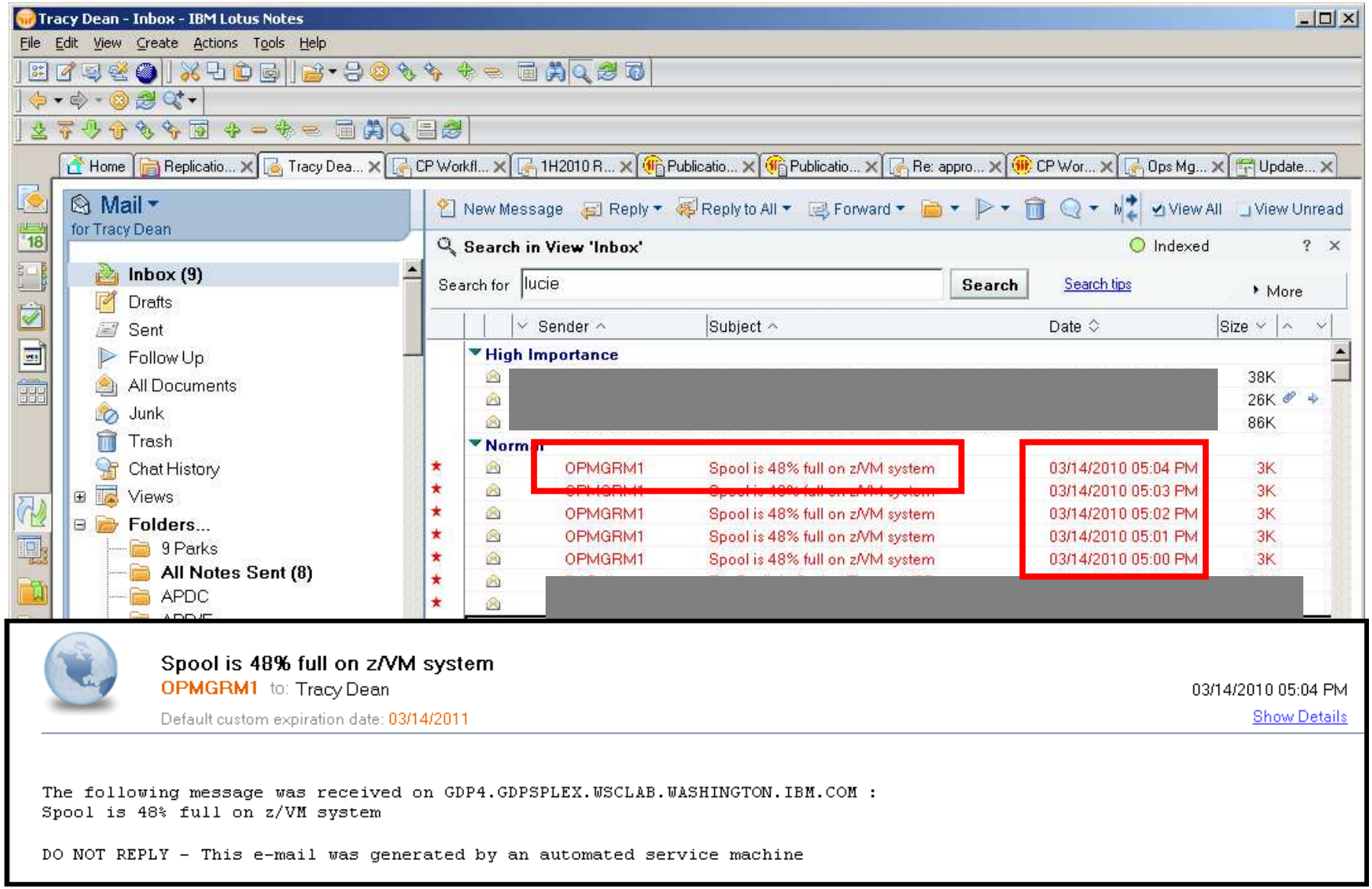
- **From a user ID with Operations Manager privileges:**

```
gomcmd opmgrm1 suspend spool(splfull2)
```









Scenario 3b: How Do You Do That?

Spool monitor and action in Operations Manager:

```
*
*
DEFSMON NAME(SPLFULL2),+
  USAGE(025-099),+
  ACTION(SPLEMAIL),+
  PARM(SPOOL)
*
DEFACTN NAME(SPLEMAIL),+
  COMMAND(EXEC SMTPNOTE tld1 at us.ibm.com &4 &p),+
  ENV(LVM)
```

Scenario 3b: How Do You Do That?

SMTPNOTE EXEC (excerpts)

```
/*  */
Parse arg mail_user dummyat mail_node baduser errtype msgtext
if errtype = 'ABEND' then
  errtext = 'Abend on user ID' baduser 'on z/VM system'
else
  if errtype = 'SPOOL' then do
    errtext = 'Spool is' baduser'% full on z/VM system'
    msgtext = errtext
  end
  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.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 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

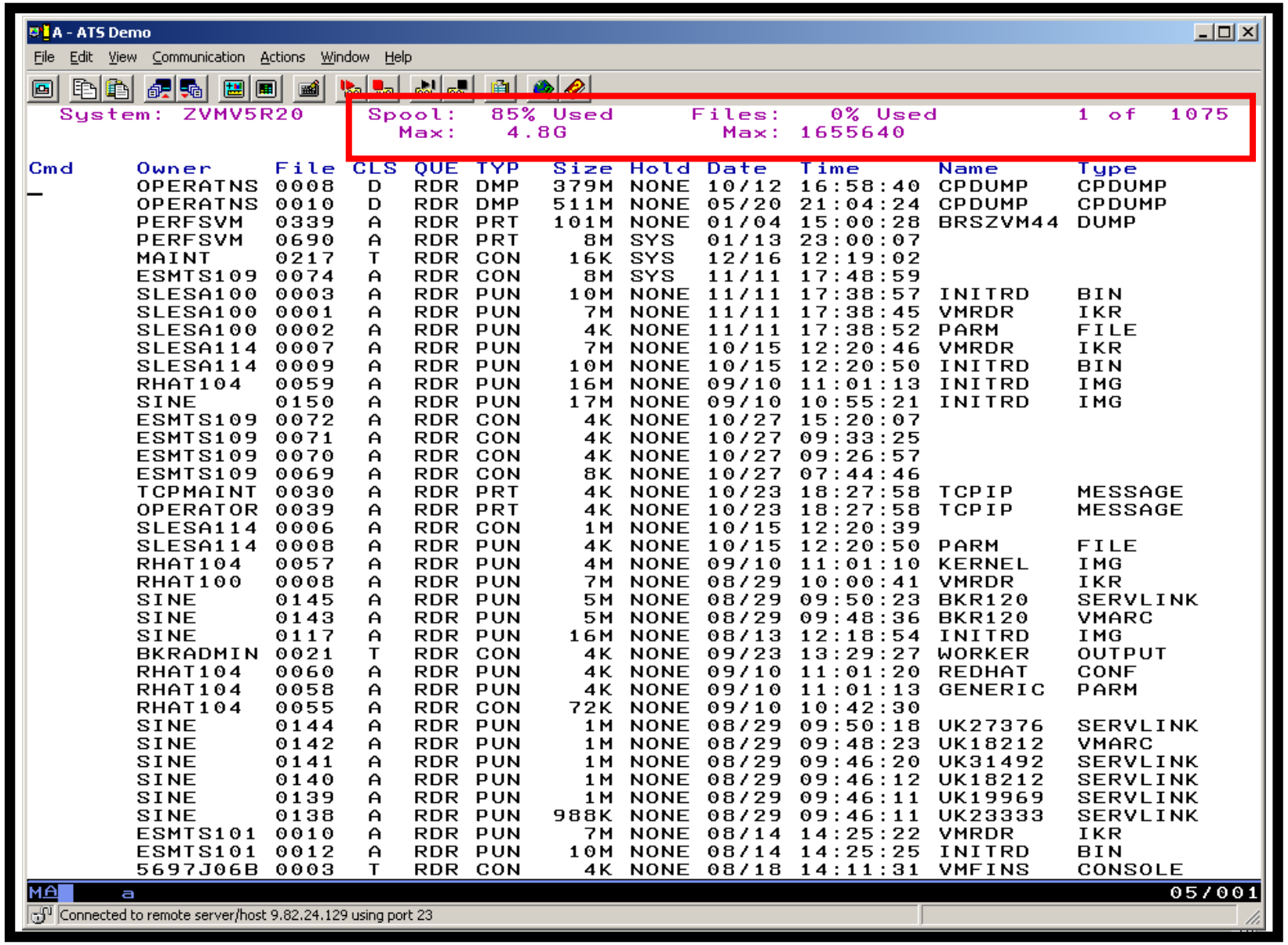
Scenario 4: Detailed Steps

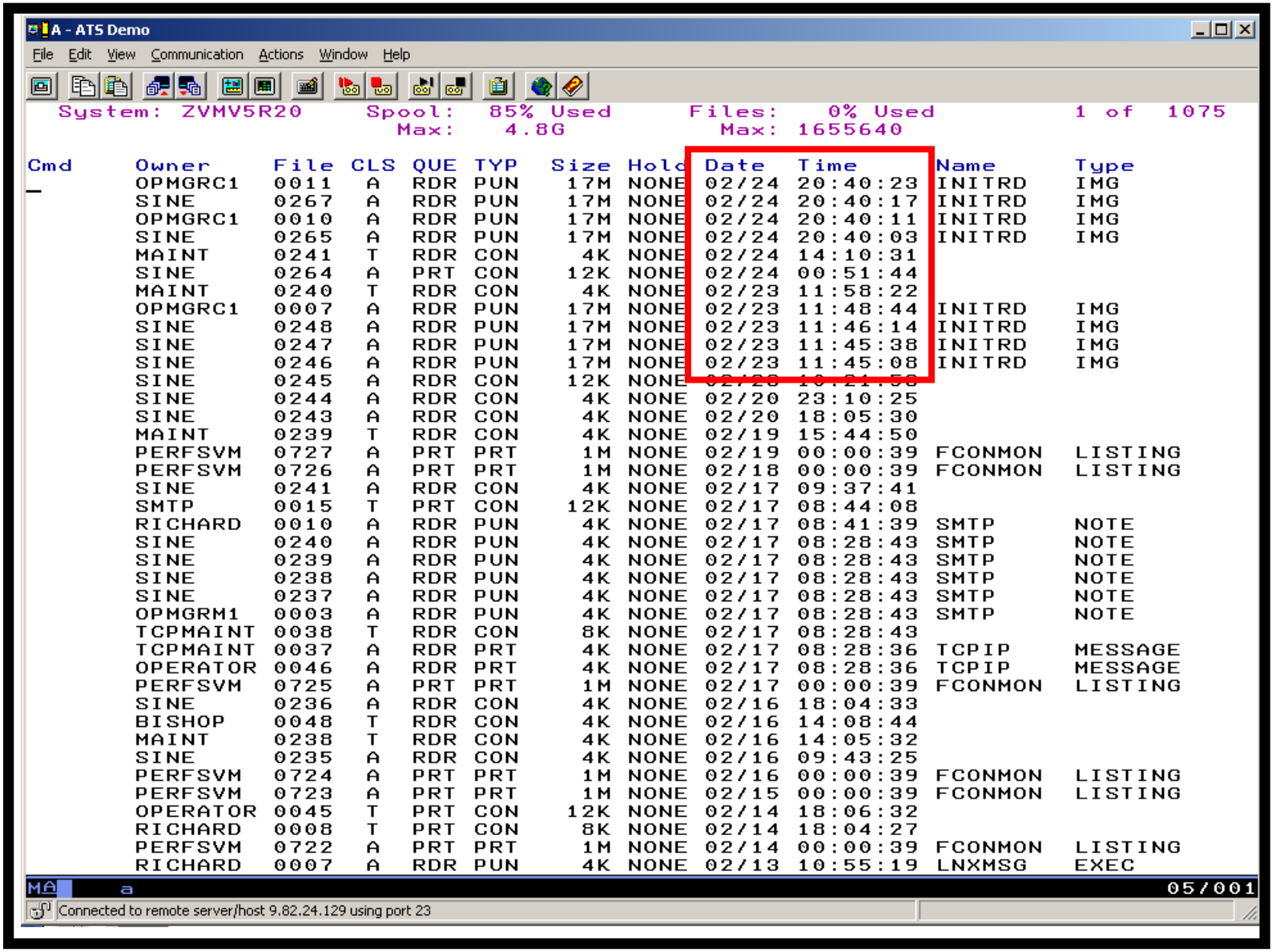
- **From an authorized VM user ID, view the spool files:**

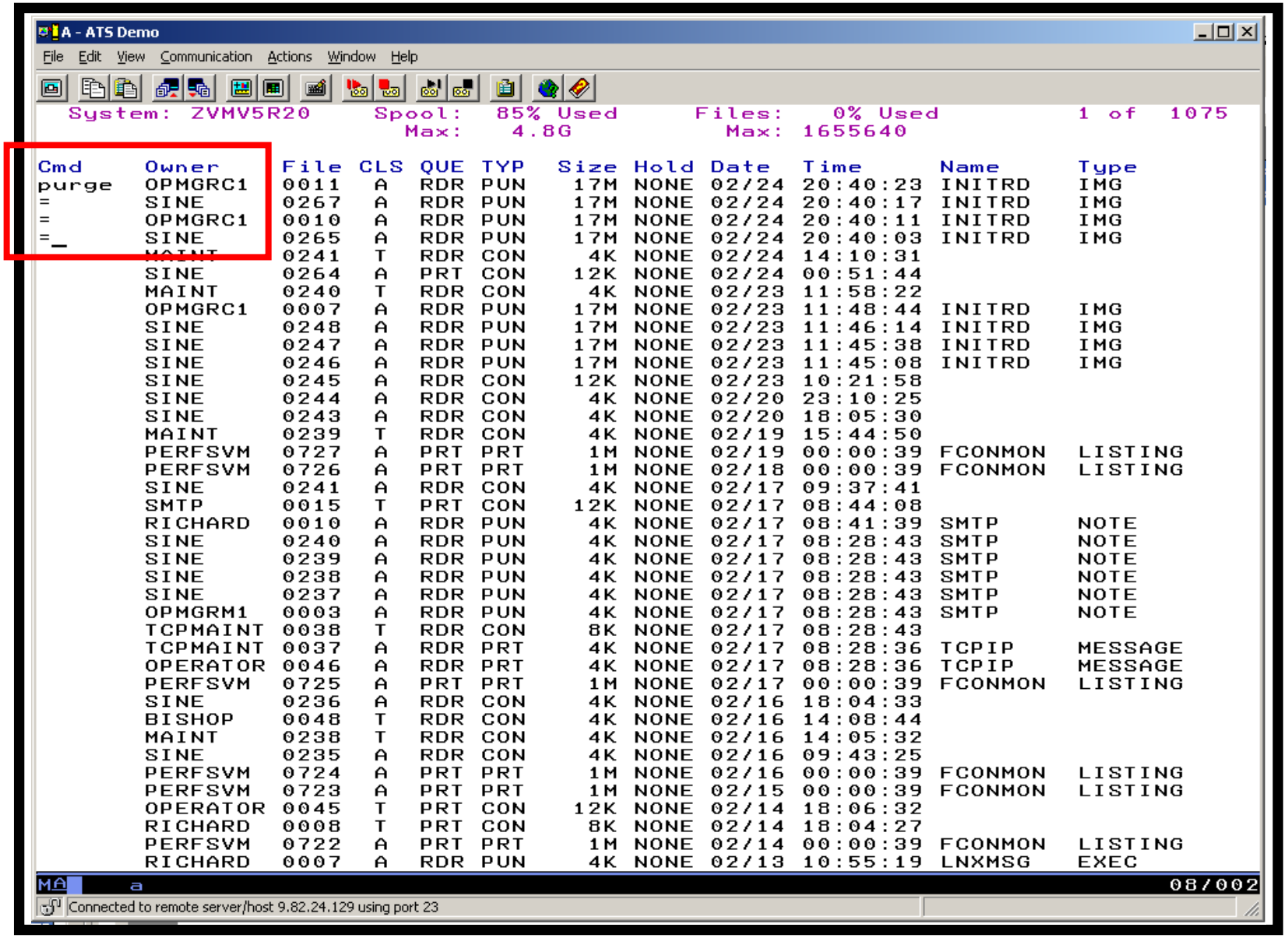
```
gomcmd opmgrm1 viewspl
```

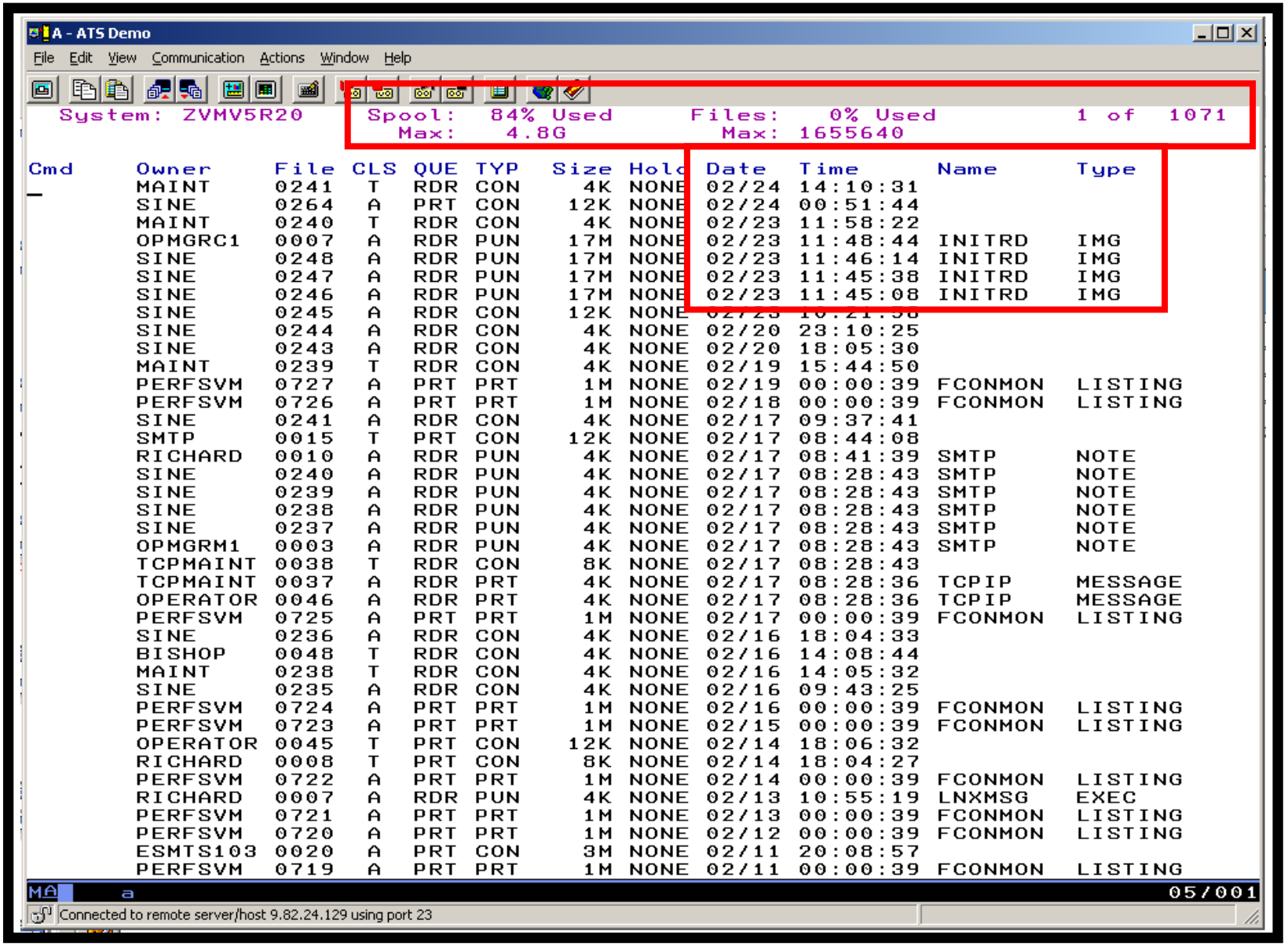
- **Sort by date**
  - Put cursor on date column header and hit F6
- **Find the spool files just sent and type PURGE next to them**
- **From an authorized VM user ID, view the log to see that the spool monitor is no longer triggered:**

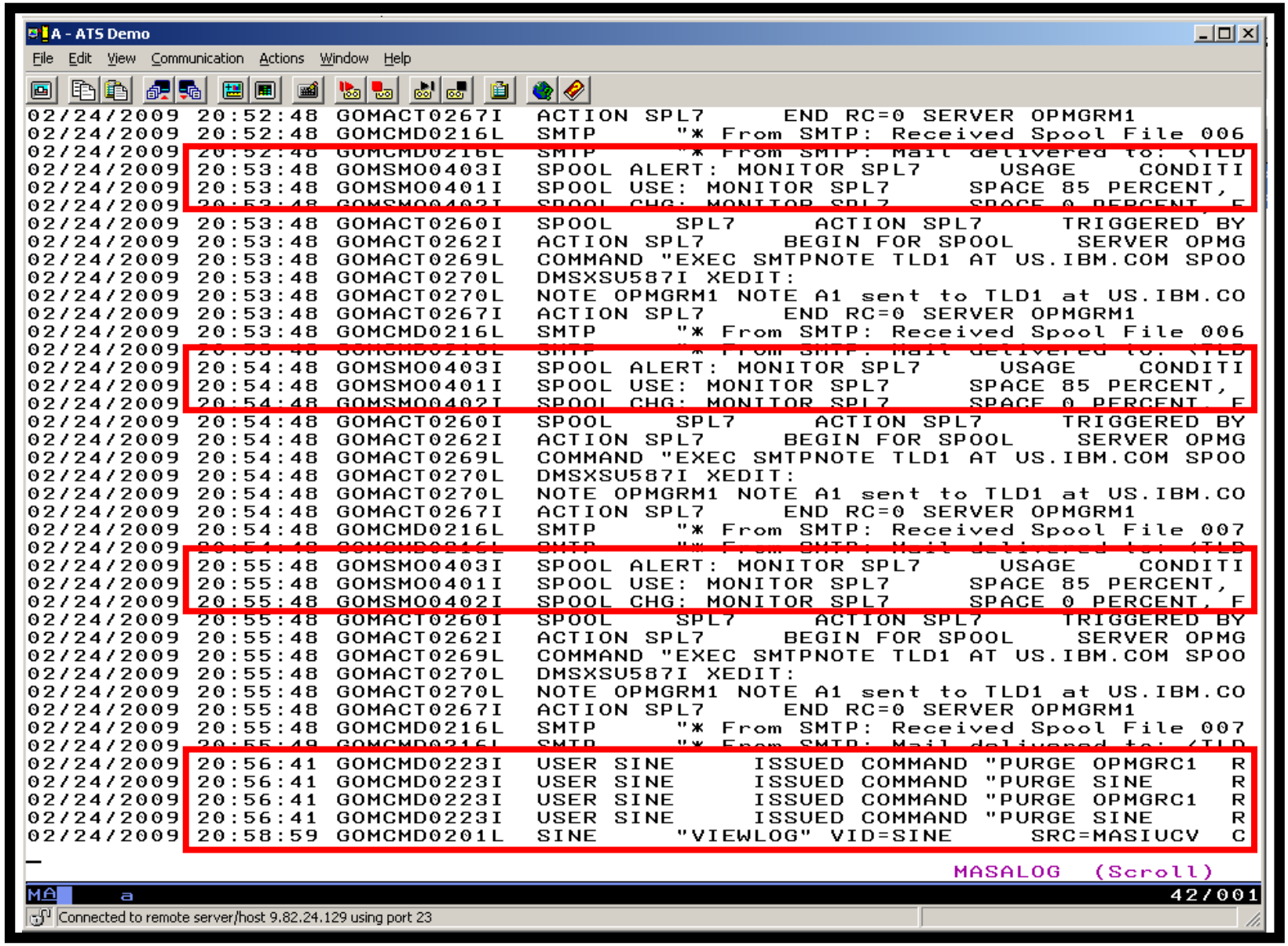
```
gomcmd opmgrm1 viewlog
```











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**

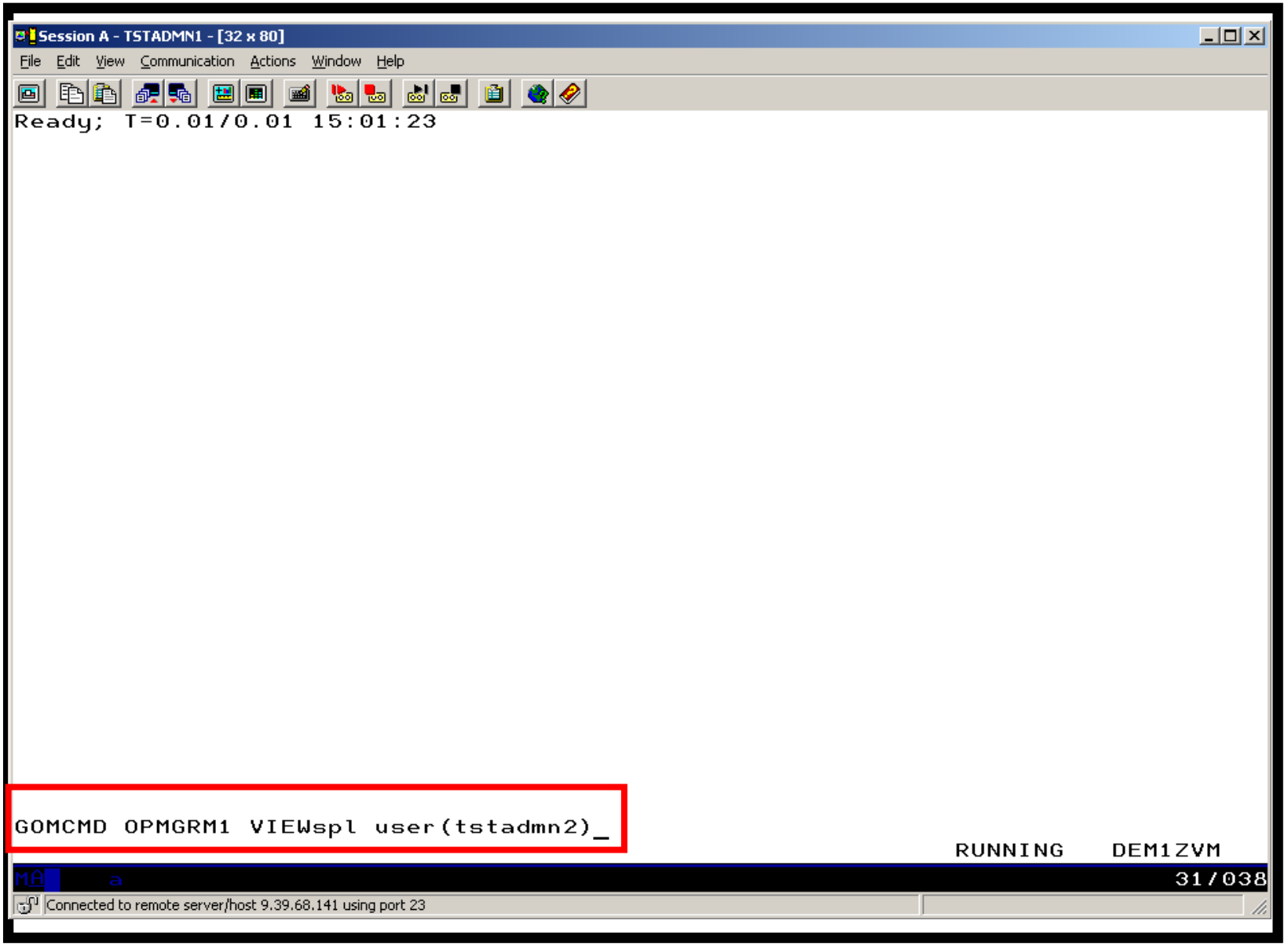
```
gomcmd opmgrm1 delschd name(demo)
```

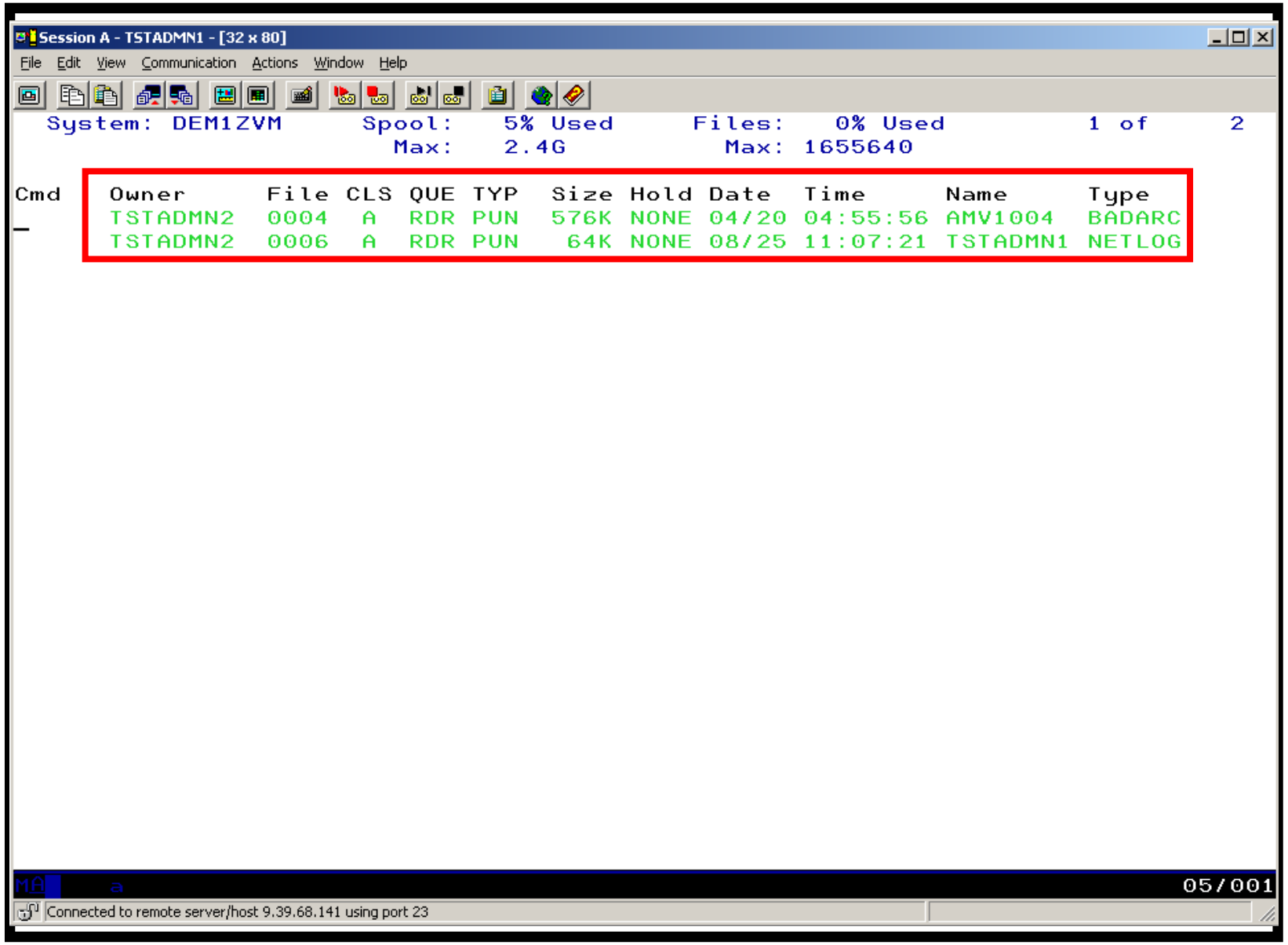
- **Schedule SFPURGER for execution**
  - It will purge any files of class Z

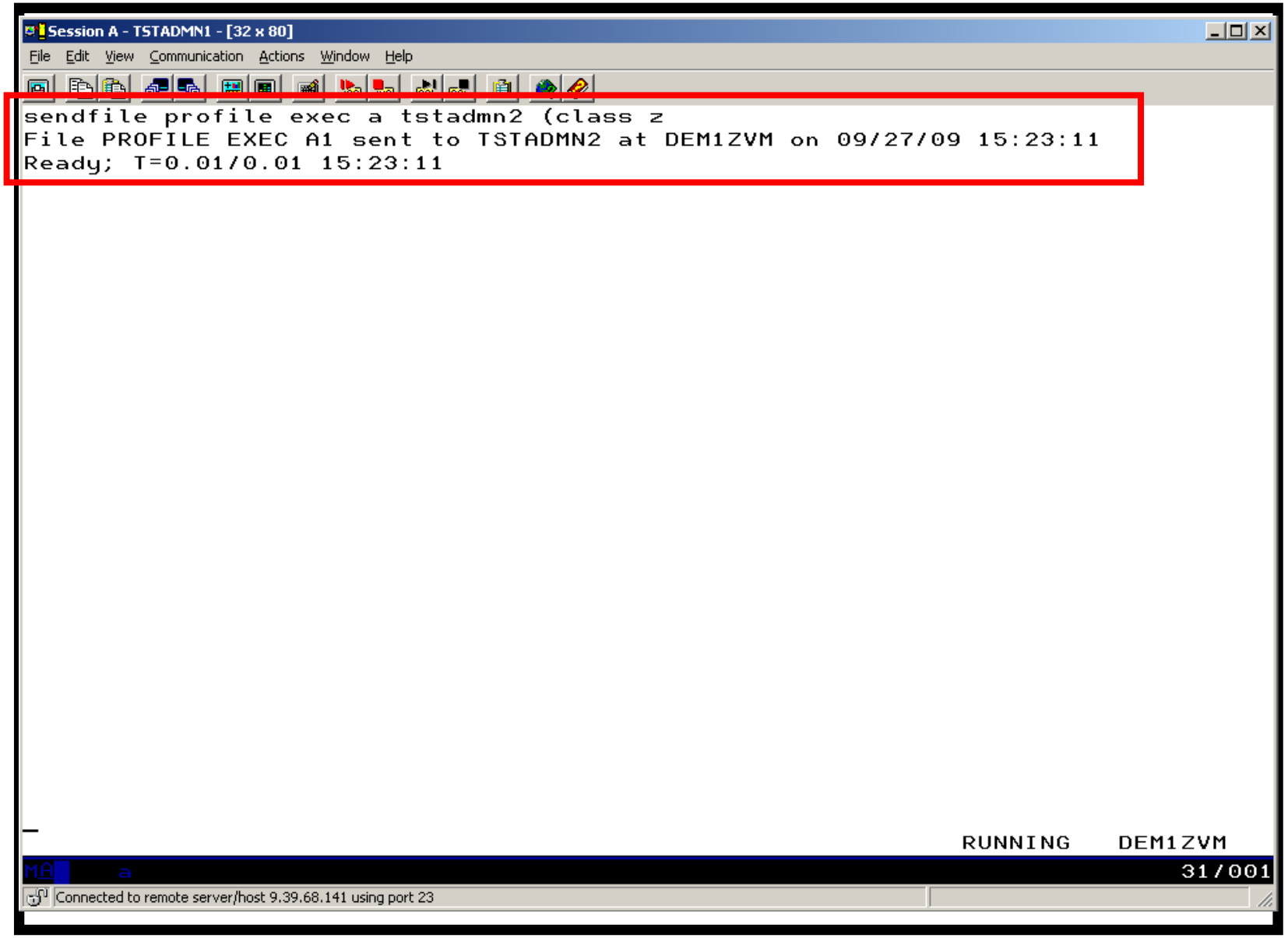
```
gomcmd opmgrm1 defschd name(demo),action(sfpurger),WHEN(now)
```

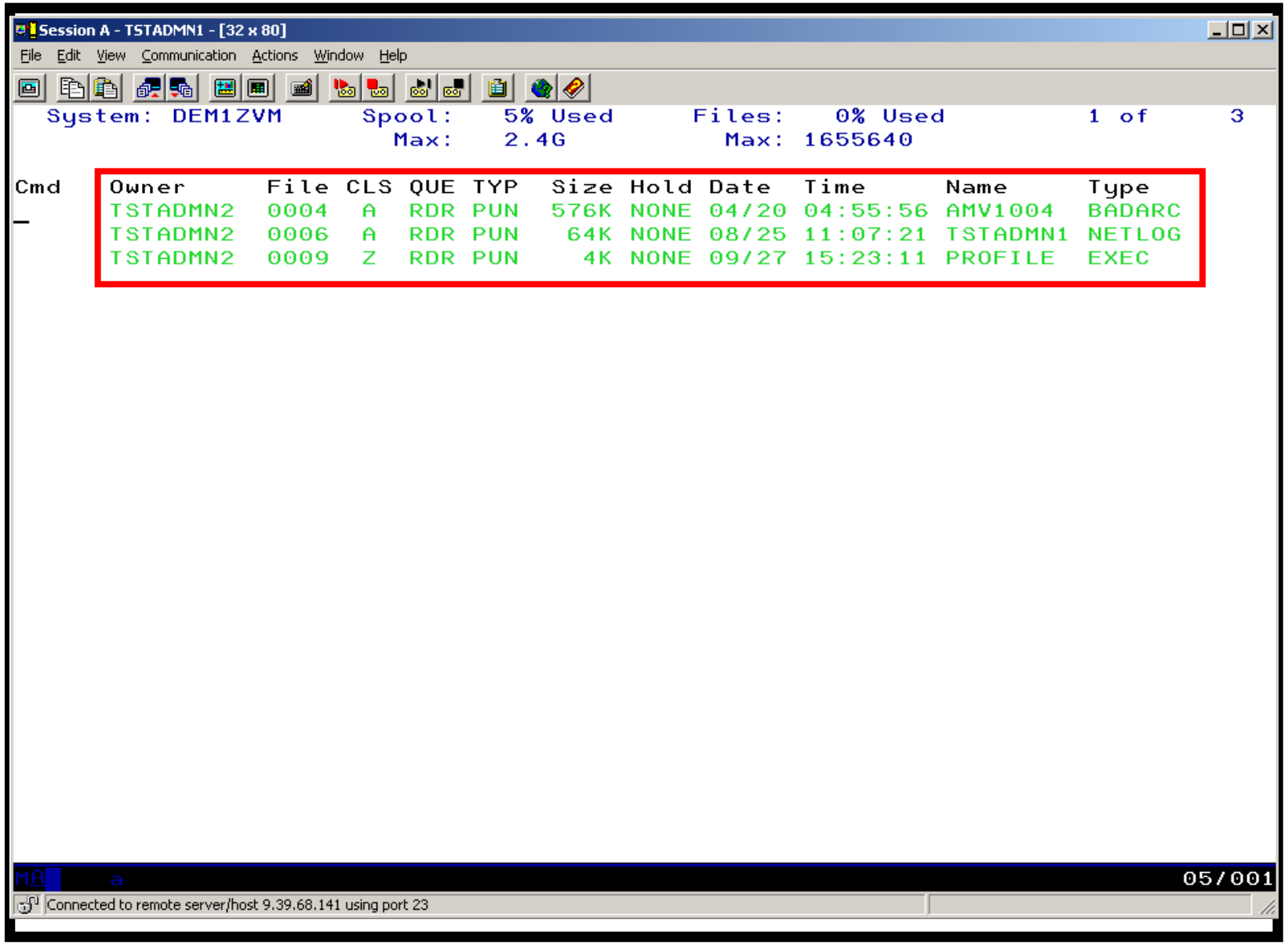
- **View spool files for this user again to see the new file is gone**

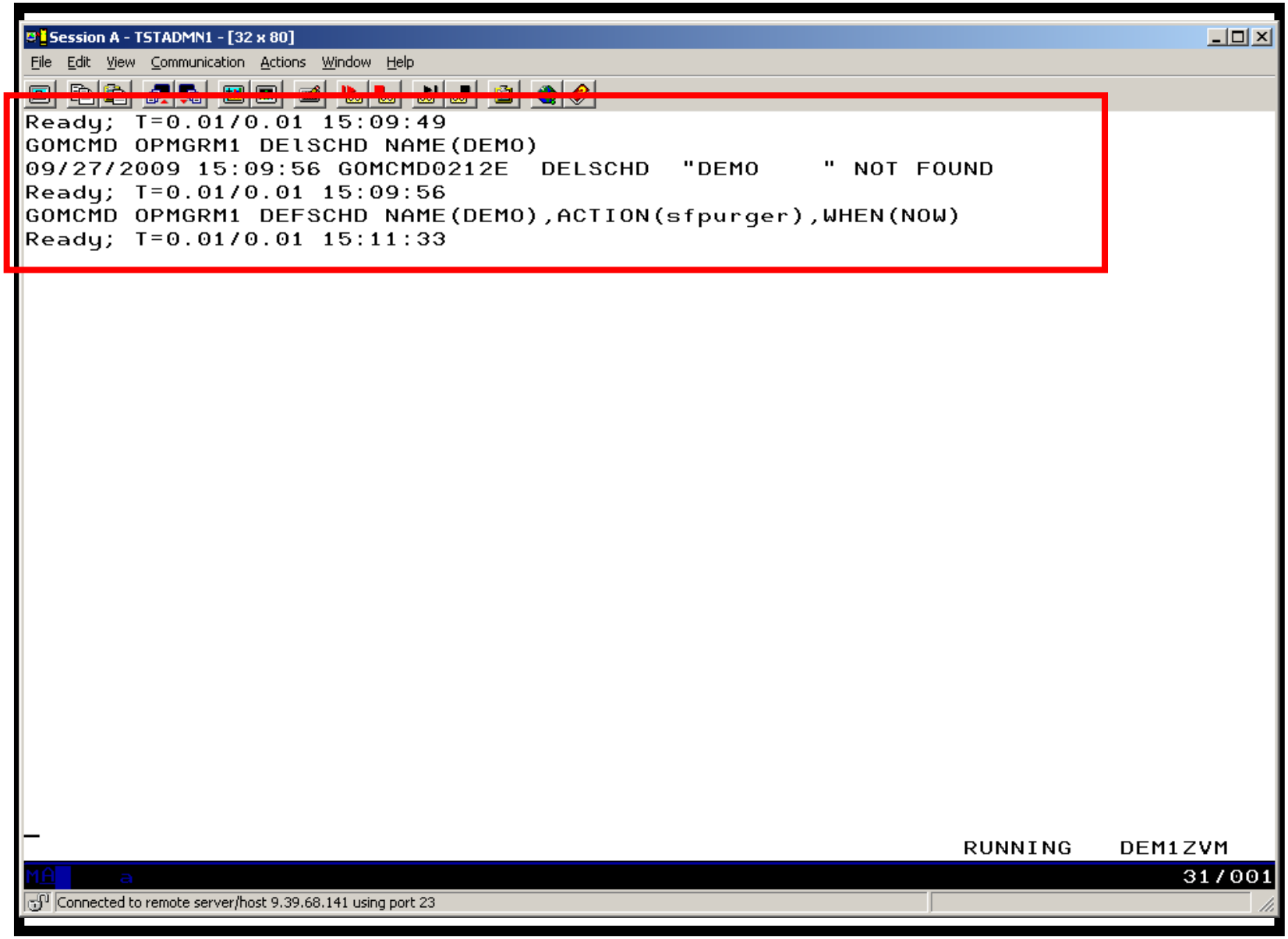
```
gomcmd opmgrm1 viewspl user(tstadmn2)
```

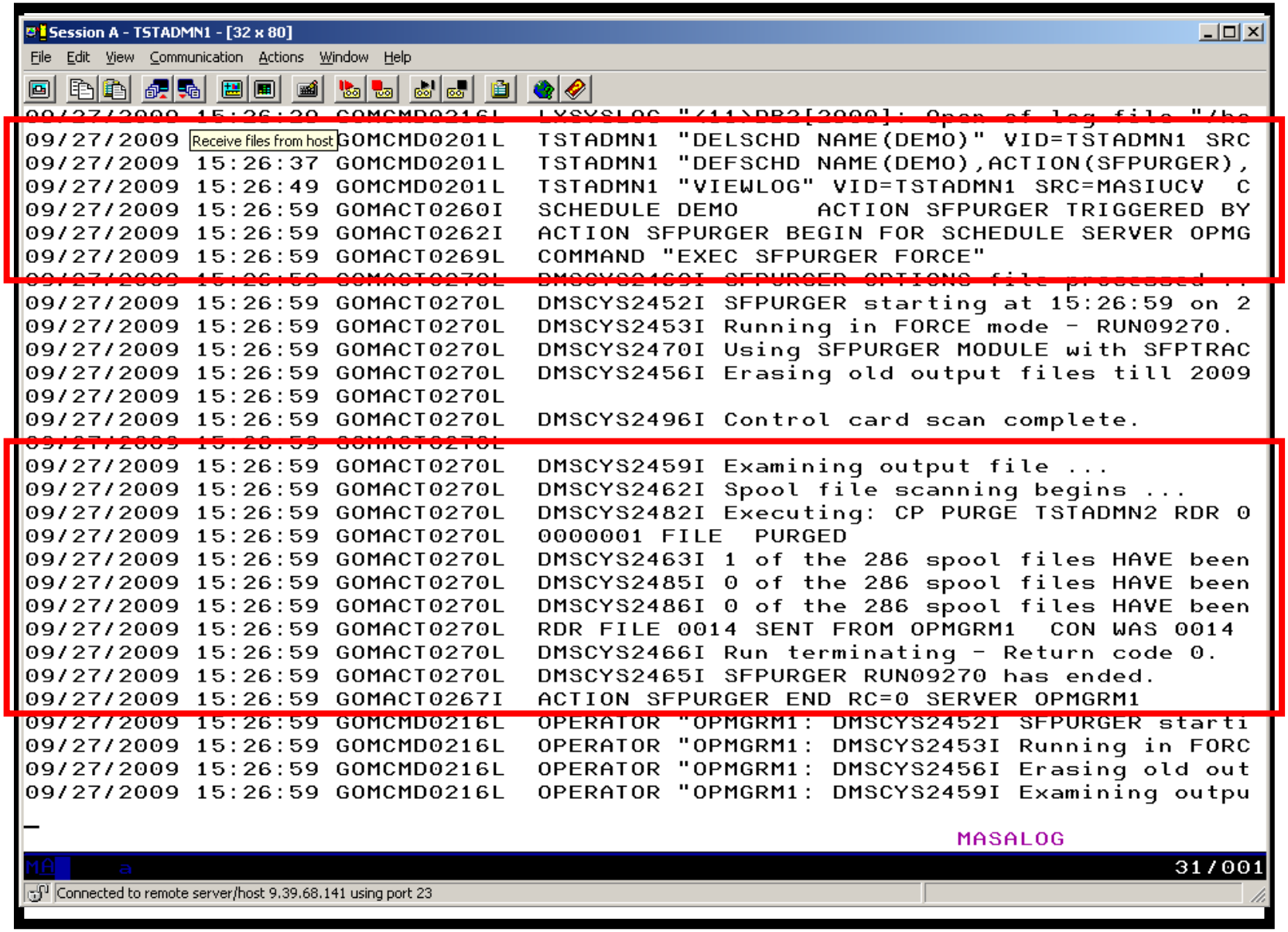


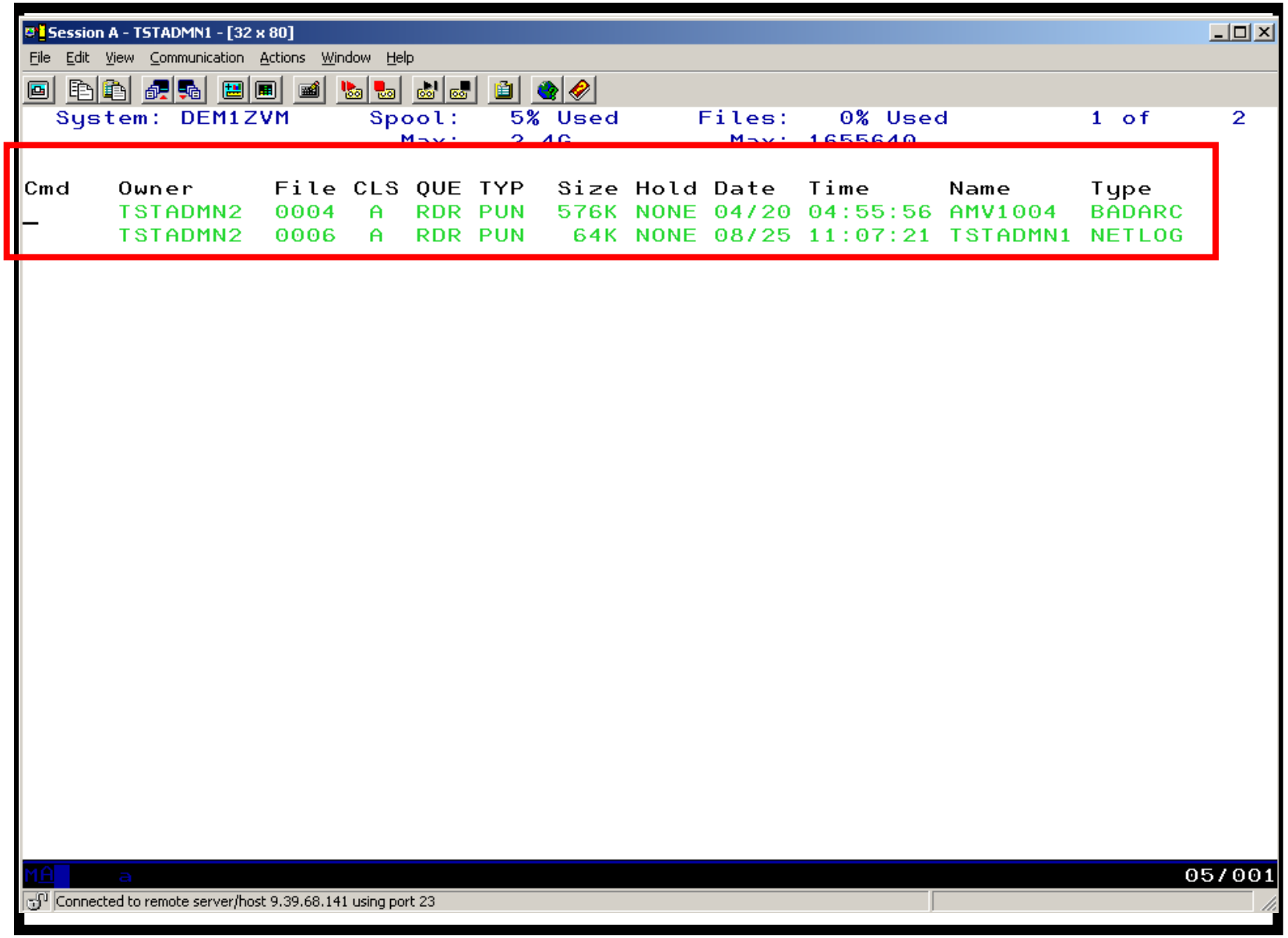












Scenario 5: How Do You Do That?

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
```

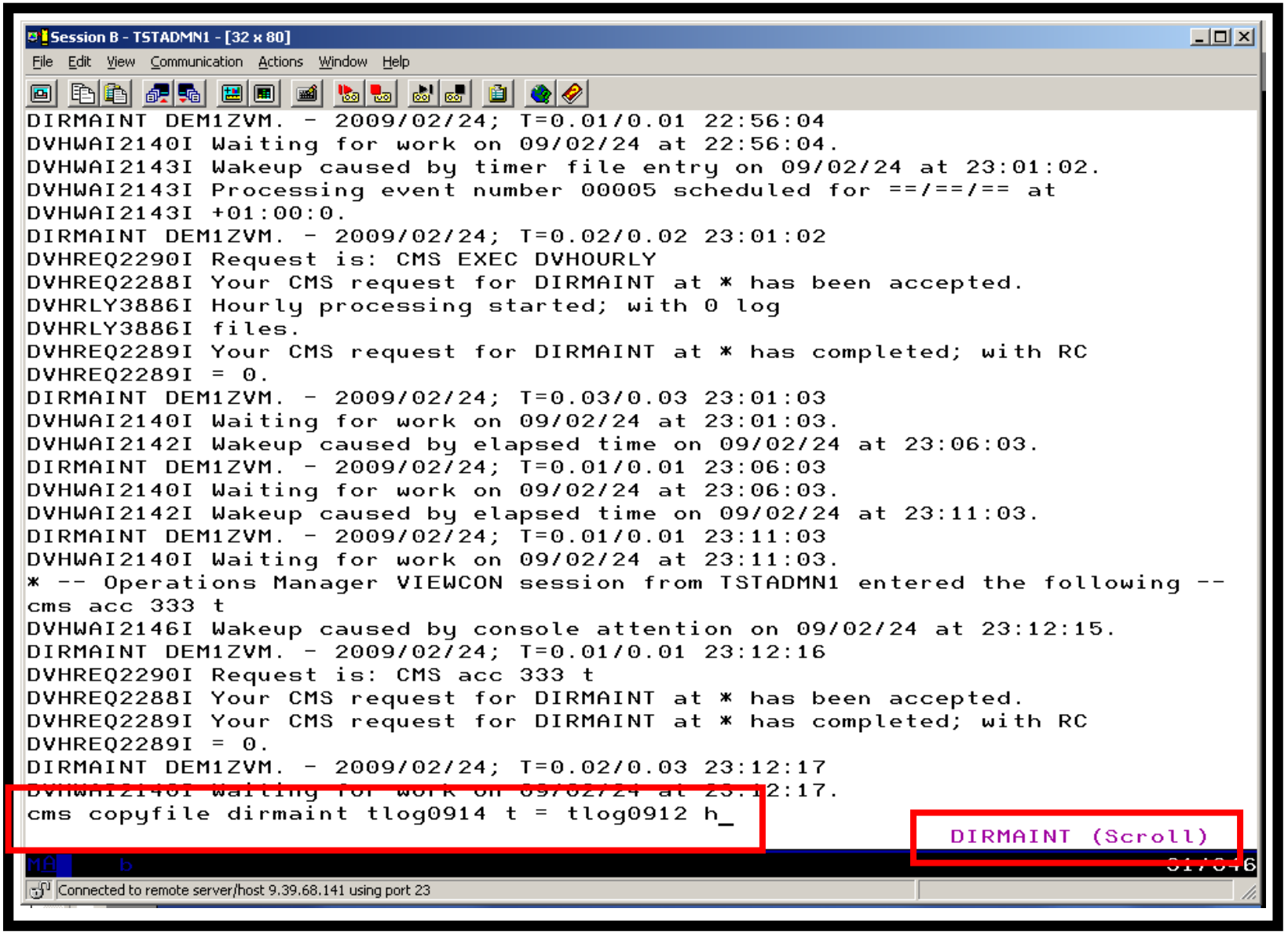
- **Exit the console view and find the files in the archive**

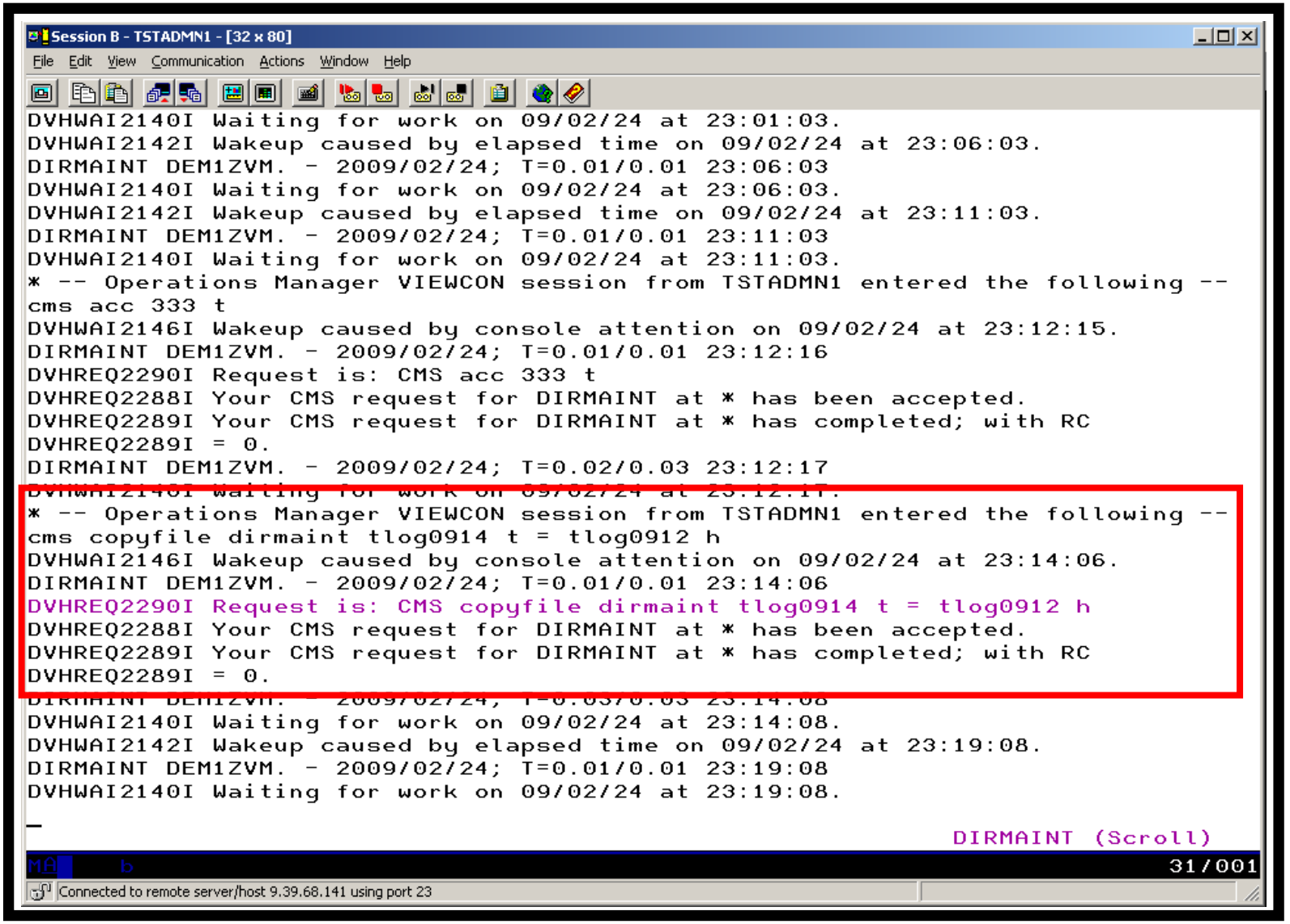
```
amvlist
```

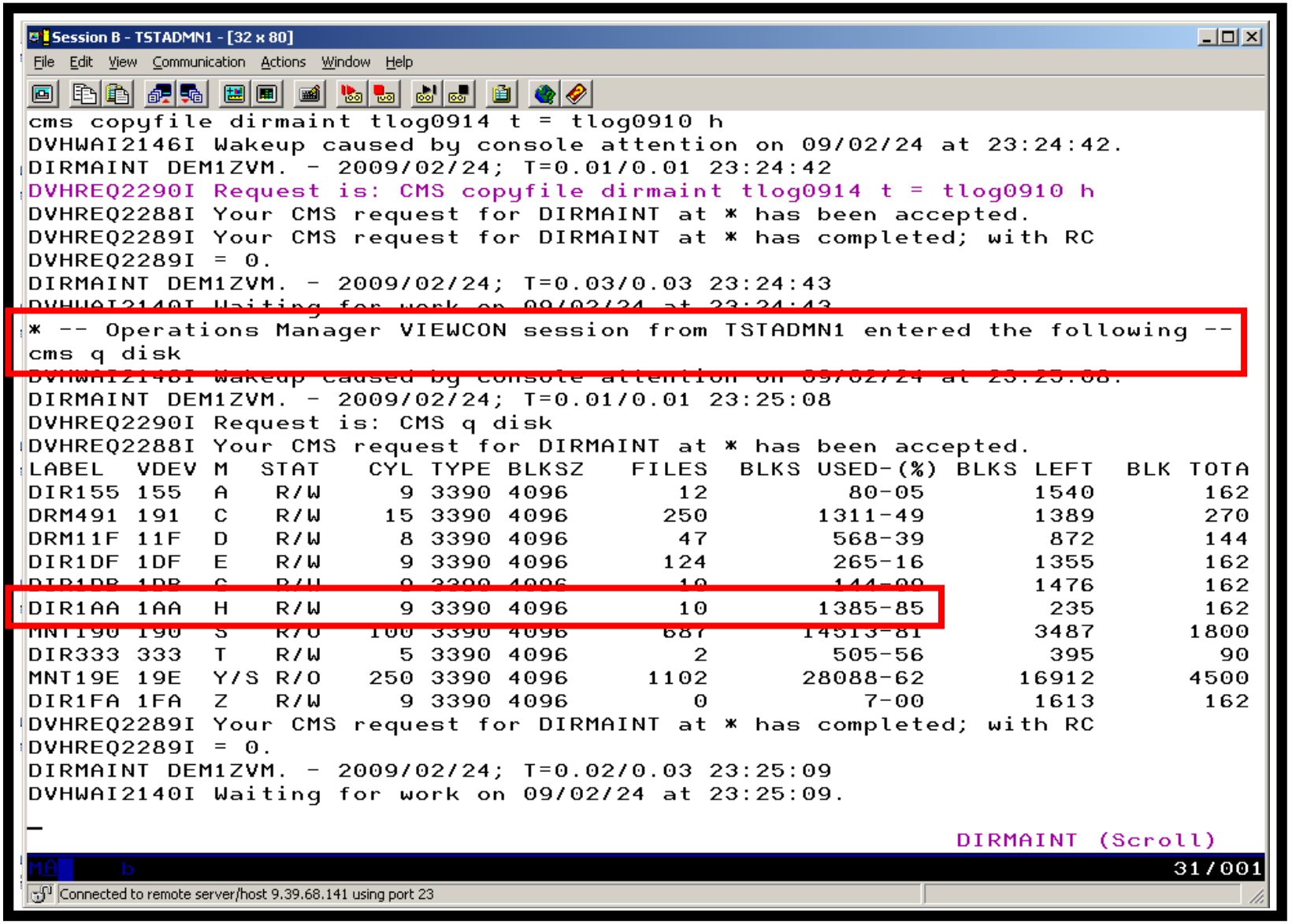
  - Type "archlogs" in the owner field and press ENTER

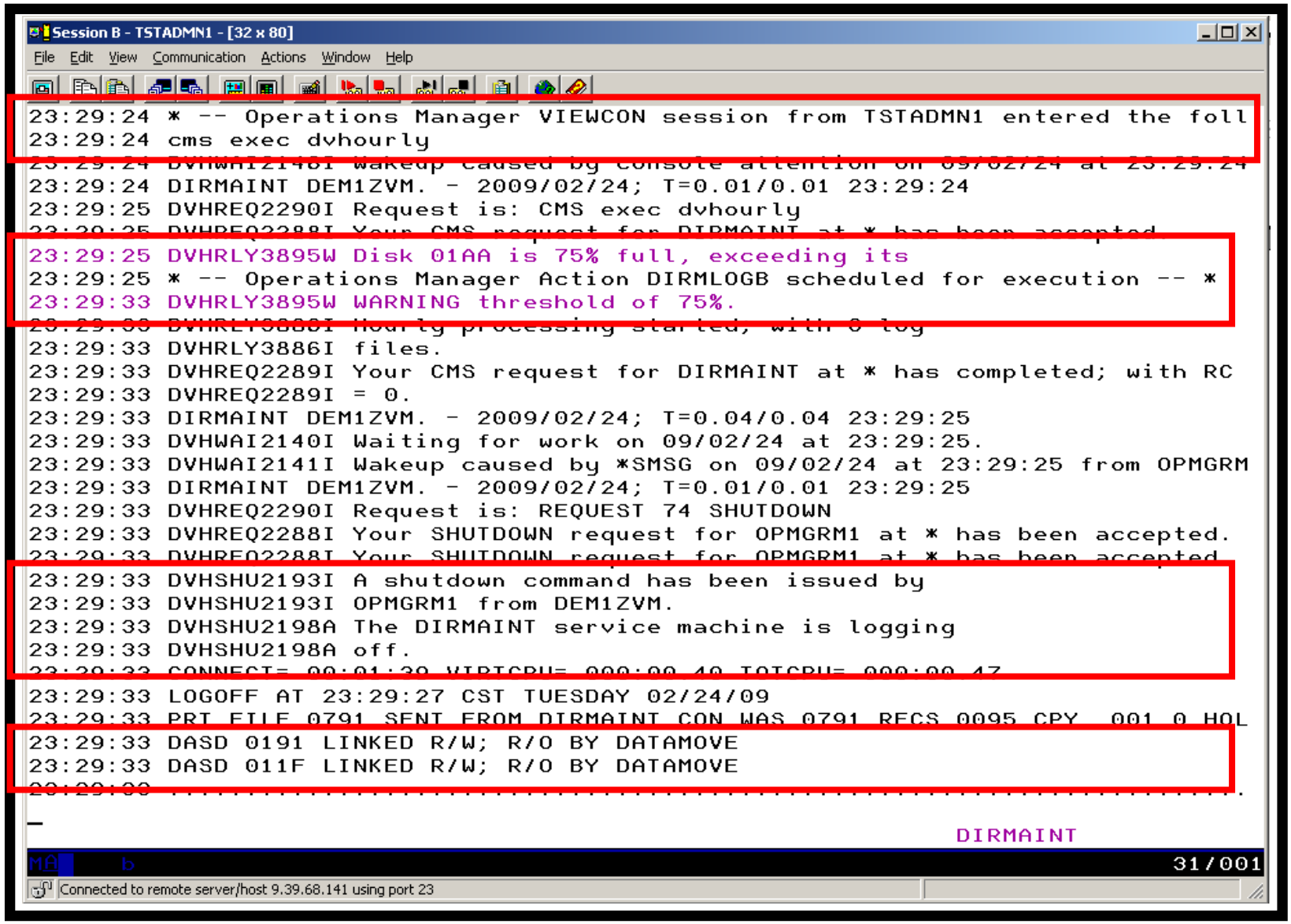
- **Request a copy of the console for further review/debugging**

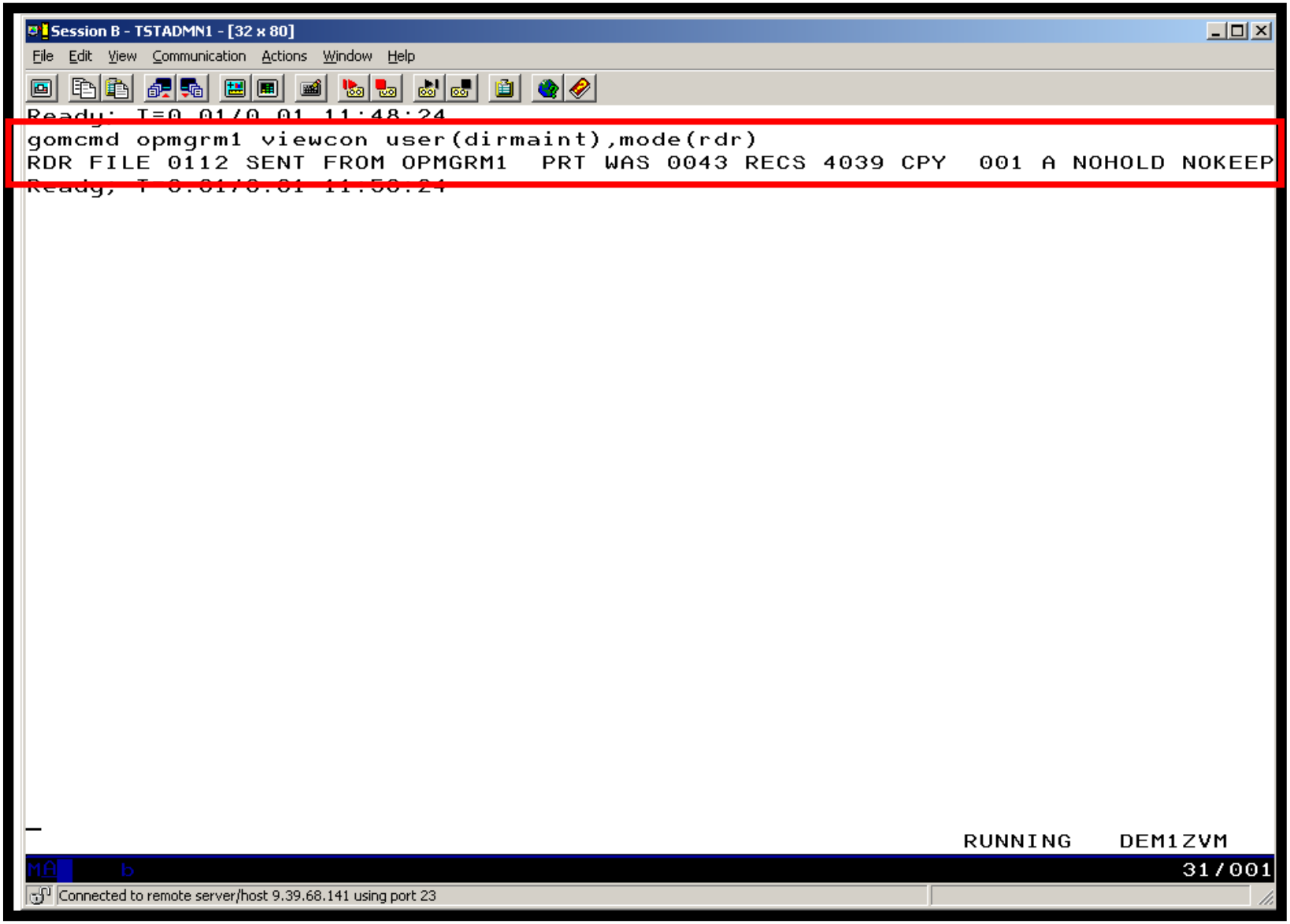
```
gomcmd opmgrm1 viewcon user(dirmaint),mode(rdr)
```

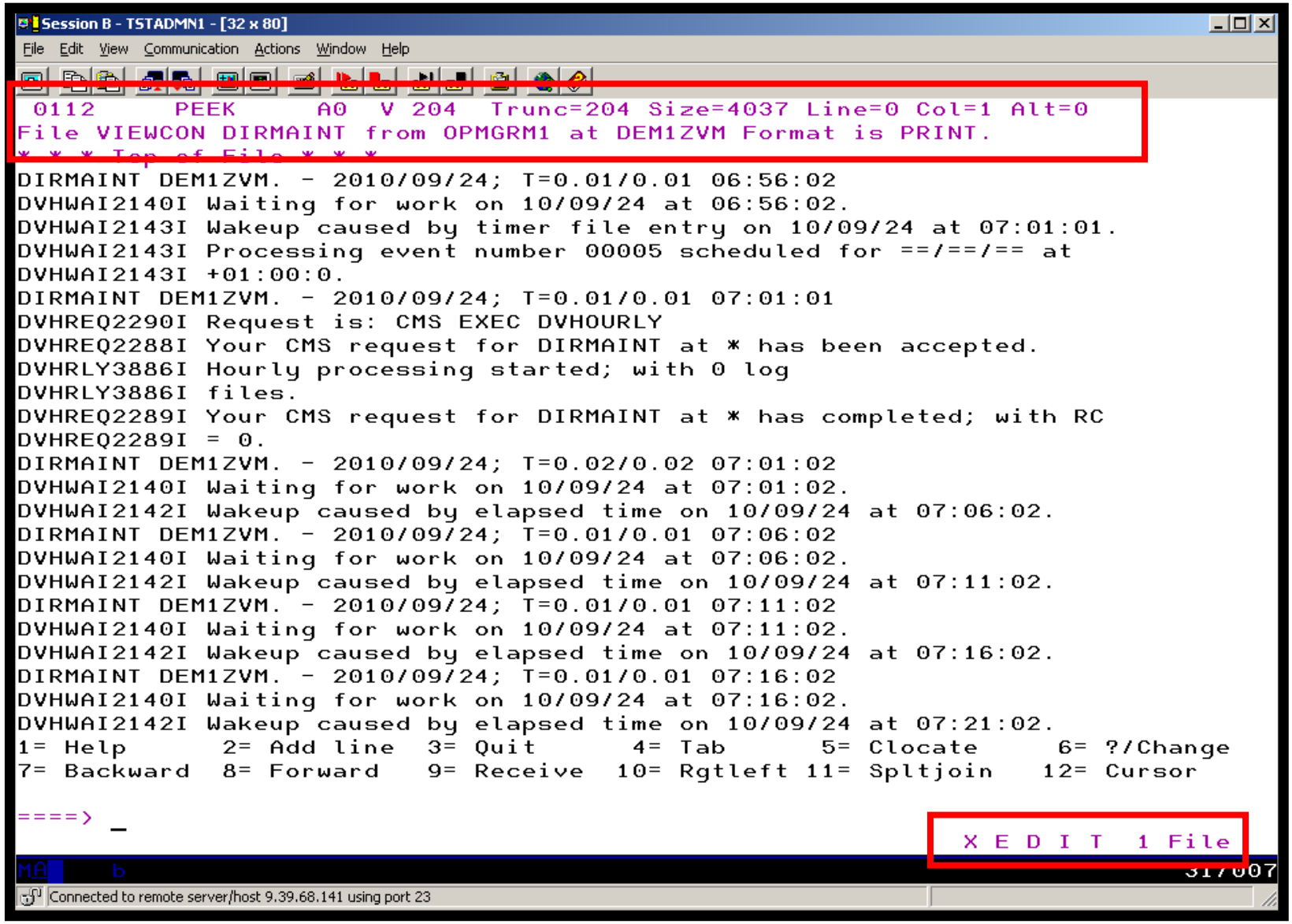












Scenario 6: How Do You Do That?

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
```

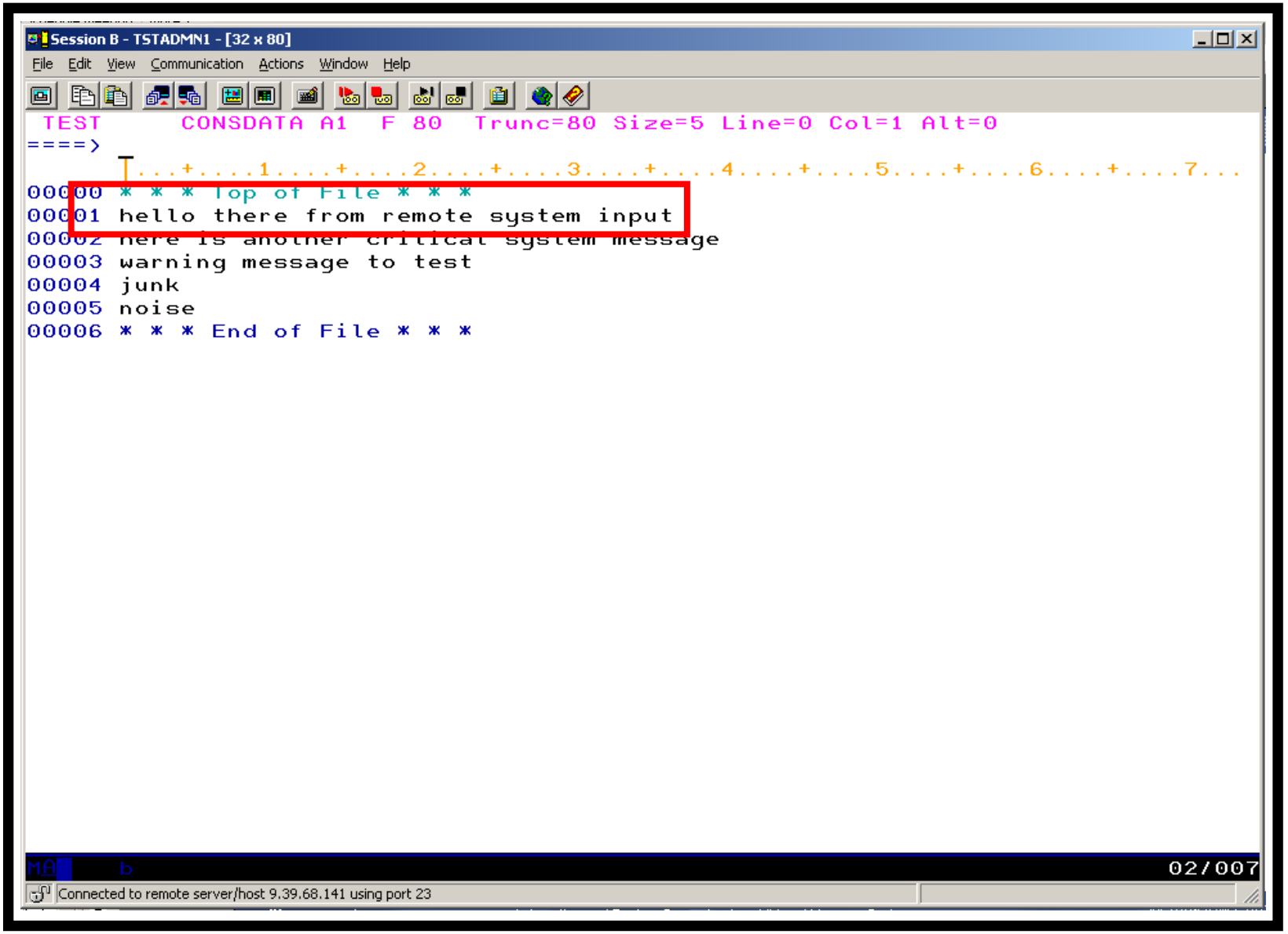
  - Notice the “hello” message in the file

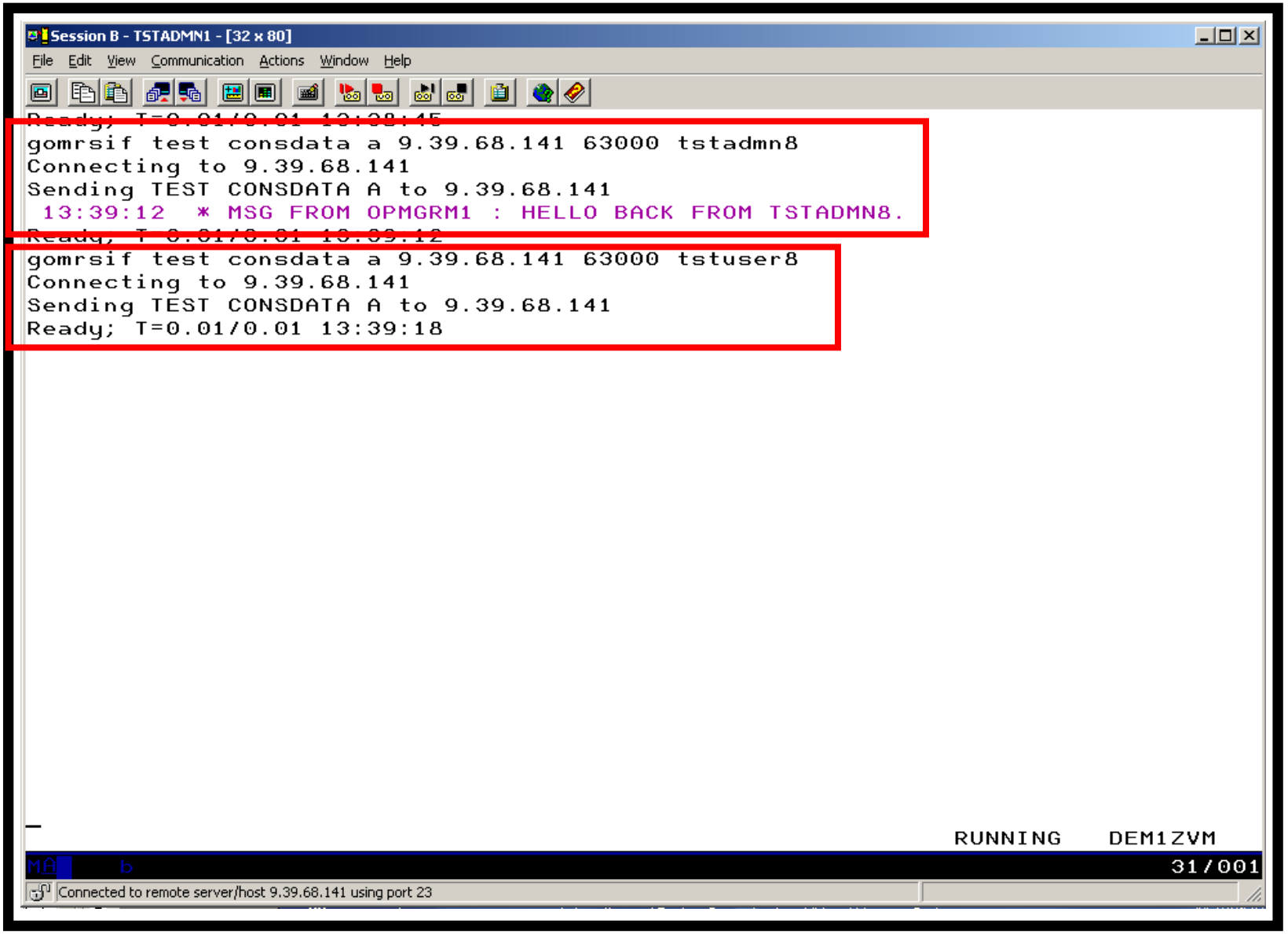
- **From a z/VM user ID, send the test file to Operations Manager**
  - Send it twice, specifying two different “owning” user IDs. One generates a message and one doesn’t:

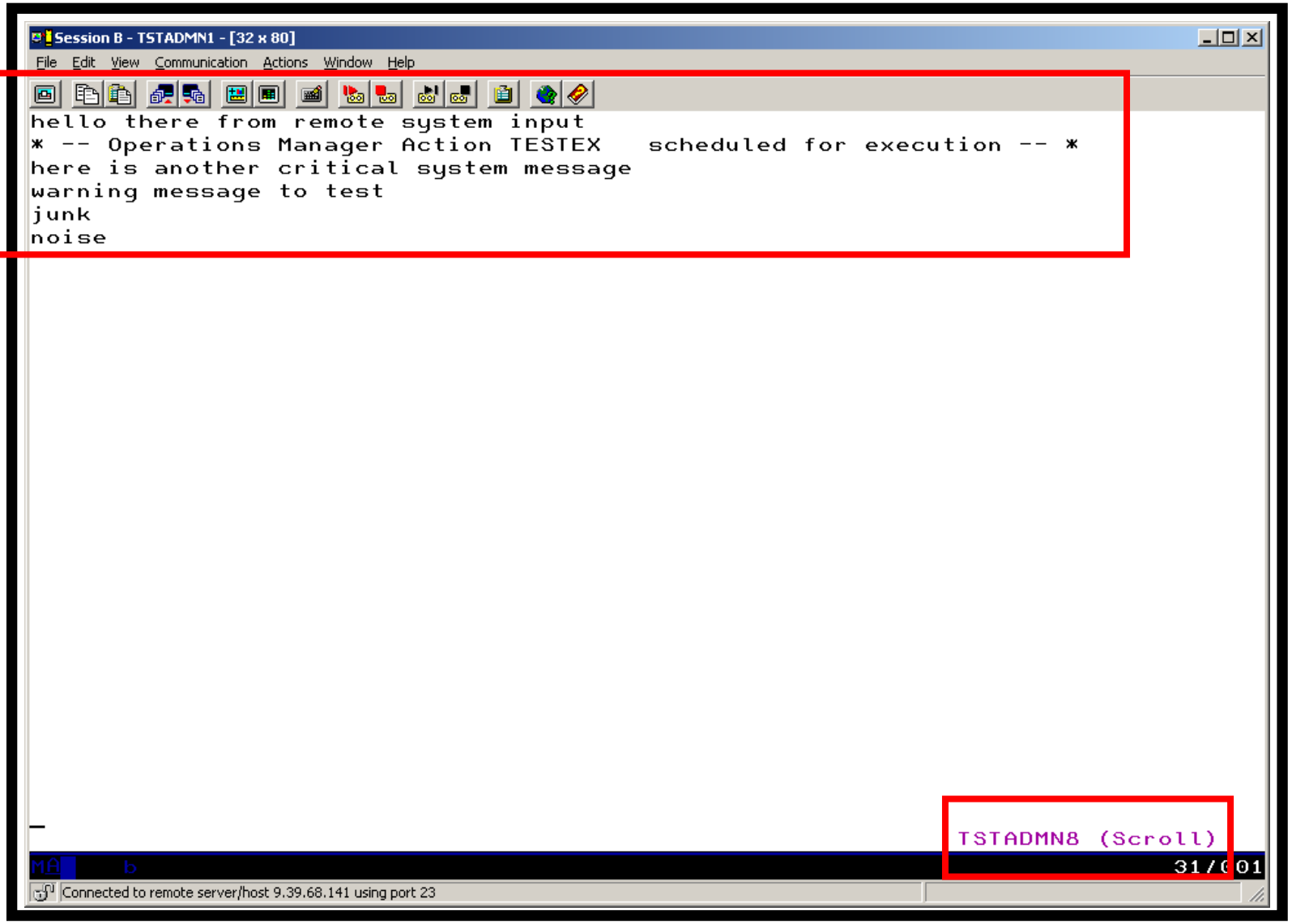
```
gomrsif test consdata a 9.39.64.72 63000 tstadmn8
gomrsif test consdata a 9.39.64.72 63000 tstuser8
```

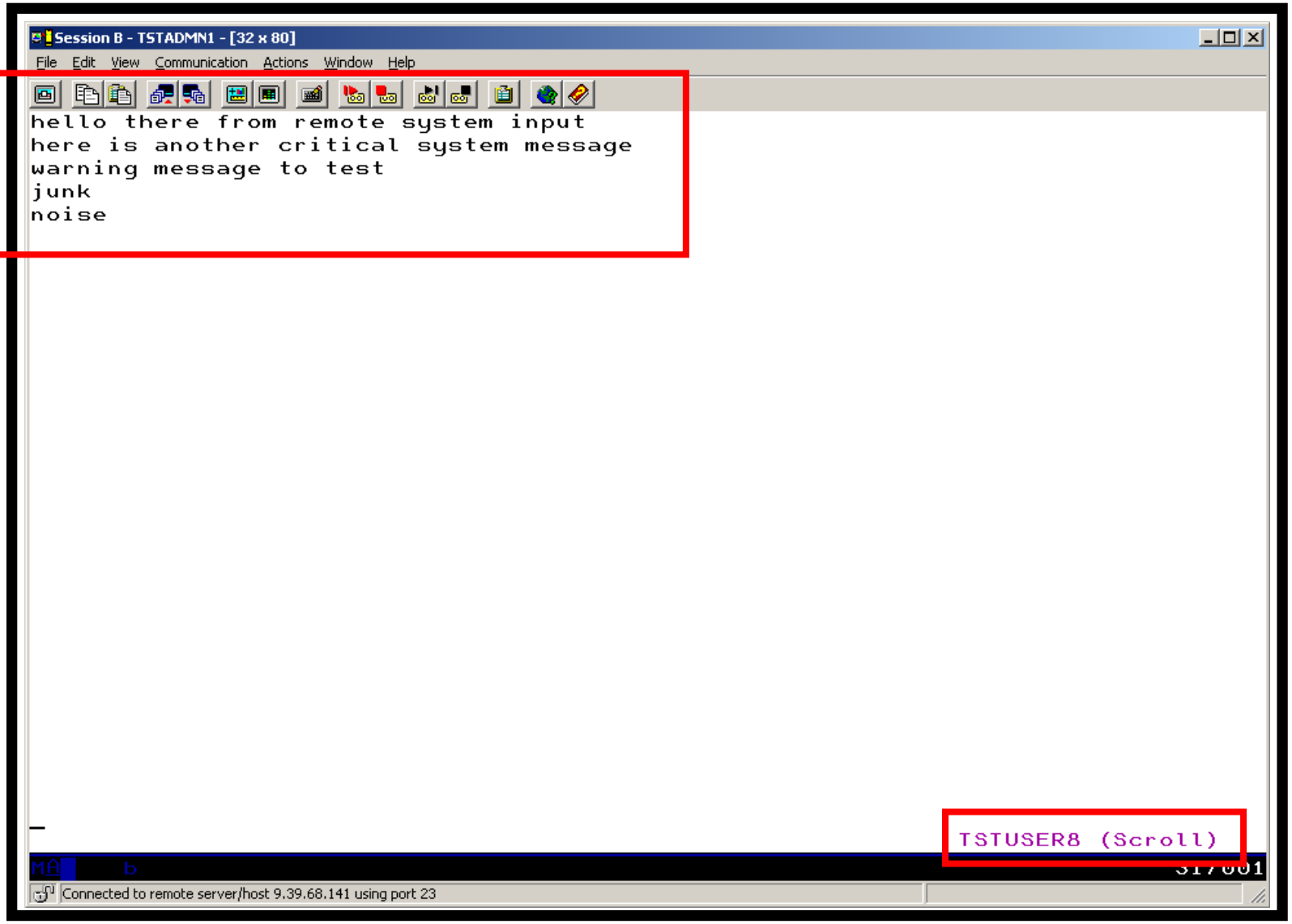
- **From an authorized z/VM user ID, view the consoles of the owning user IDs:**

```
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

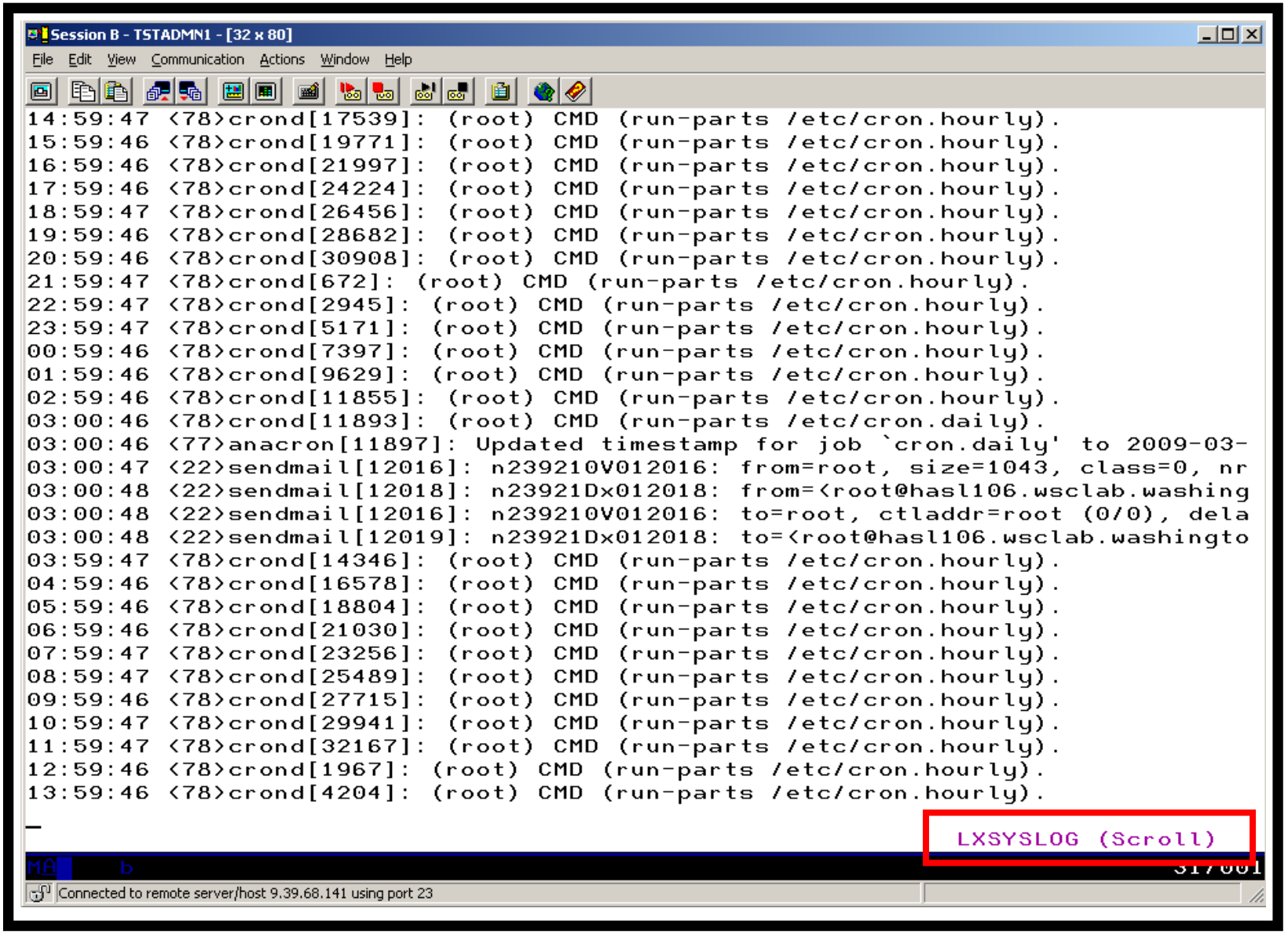
- **From an authorized z/VM user ID, view any syslog data already received**

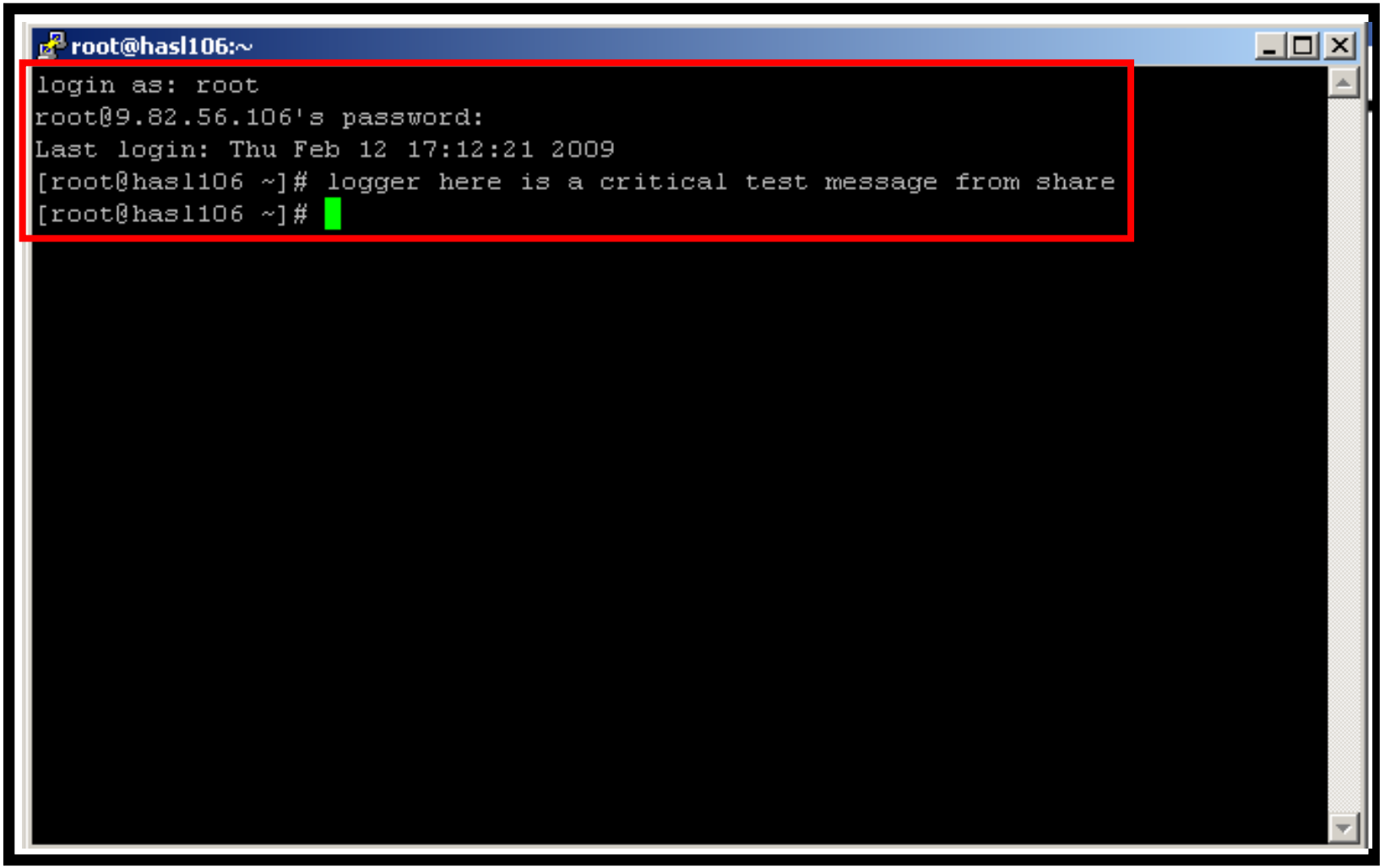
```
gomcmd opmgrm1 viewcon user(lxsyslog)
```

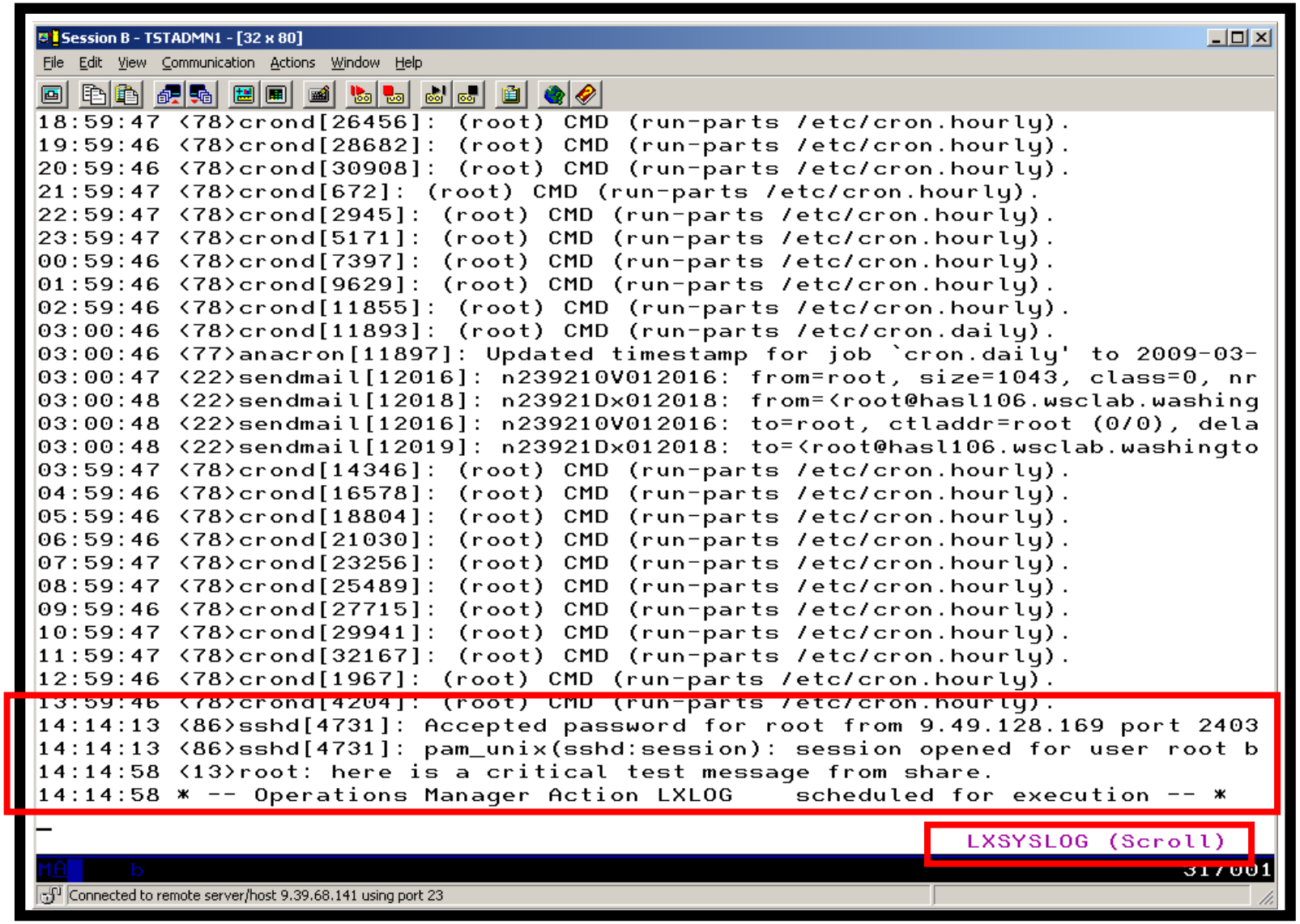
- **Use PUTTY to connect to a Linux guest**
- **Login as root and issue the command**

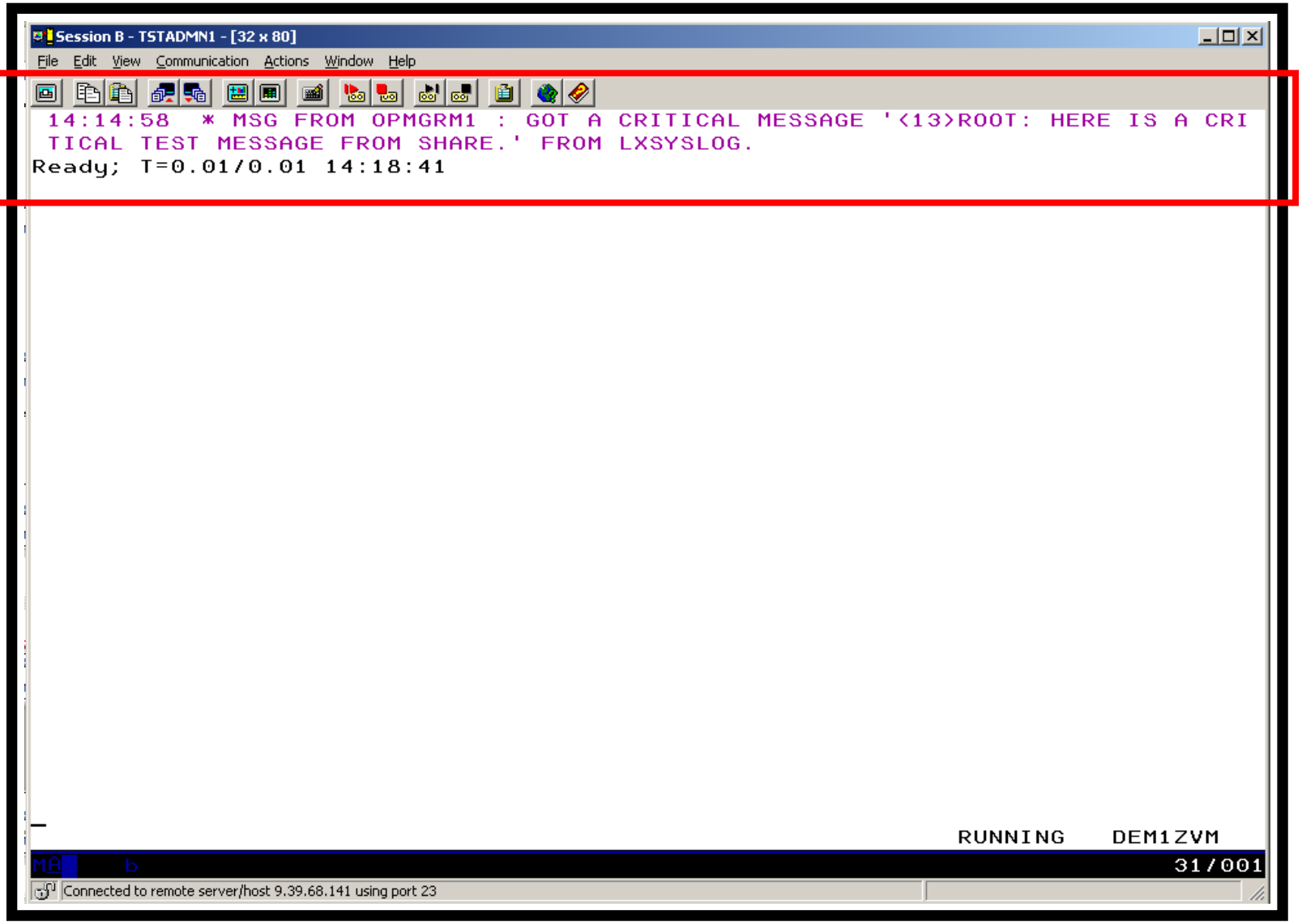
```
logger here is a critical test message from SHARE
```

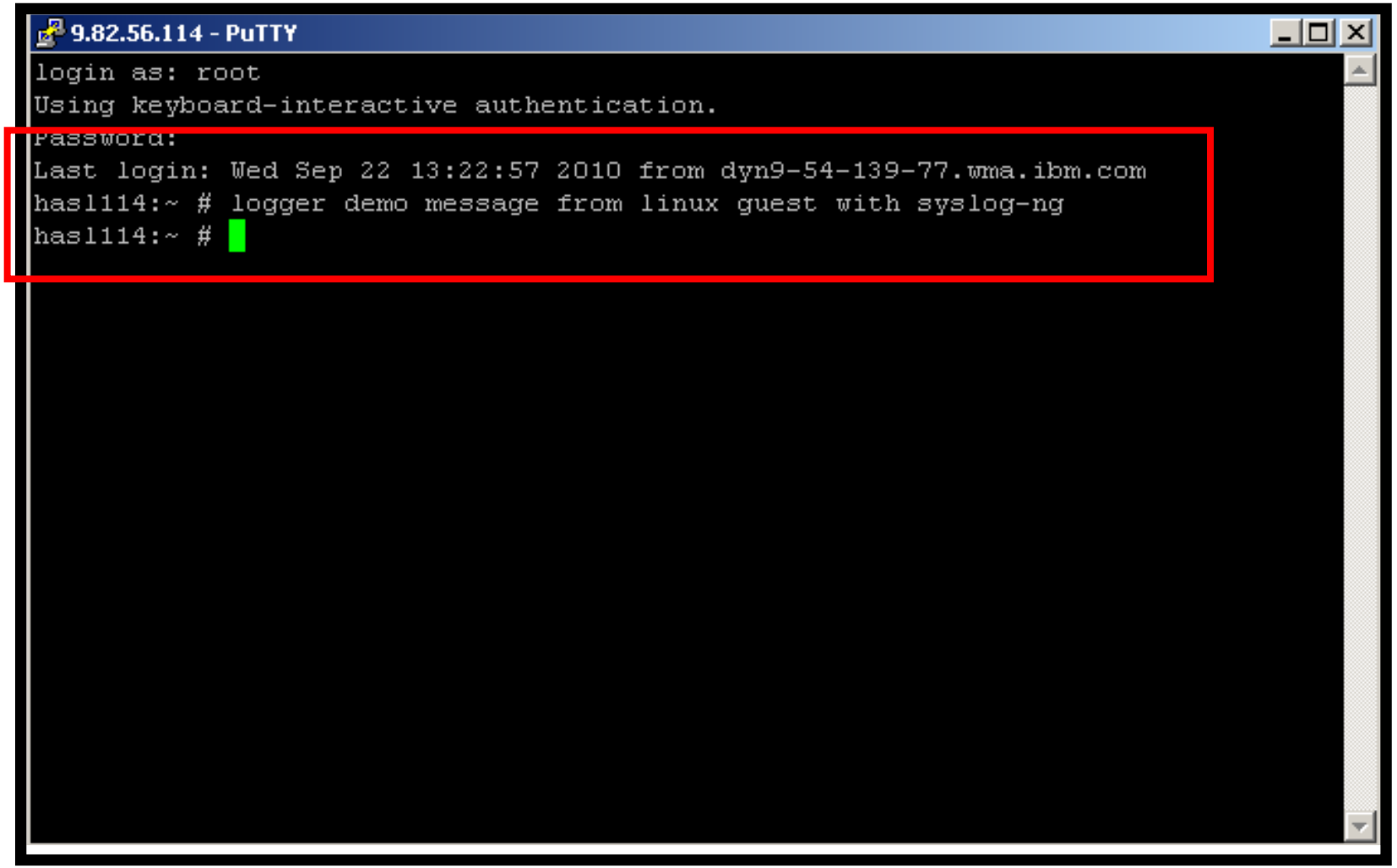
- **Return to the VIEWCON session**
  - See the message in the syslog “console”
  - Using syslog, so no hostname or IP address
- **Repeat from a different Linux guest that uses syslog-ng**

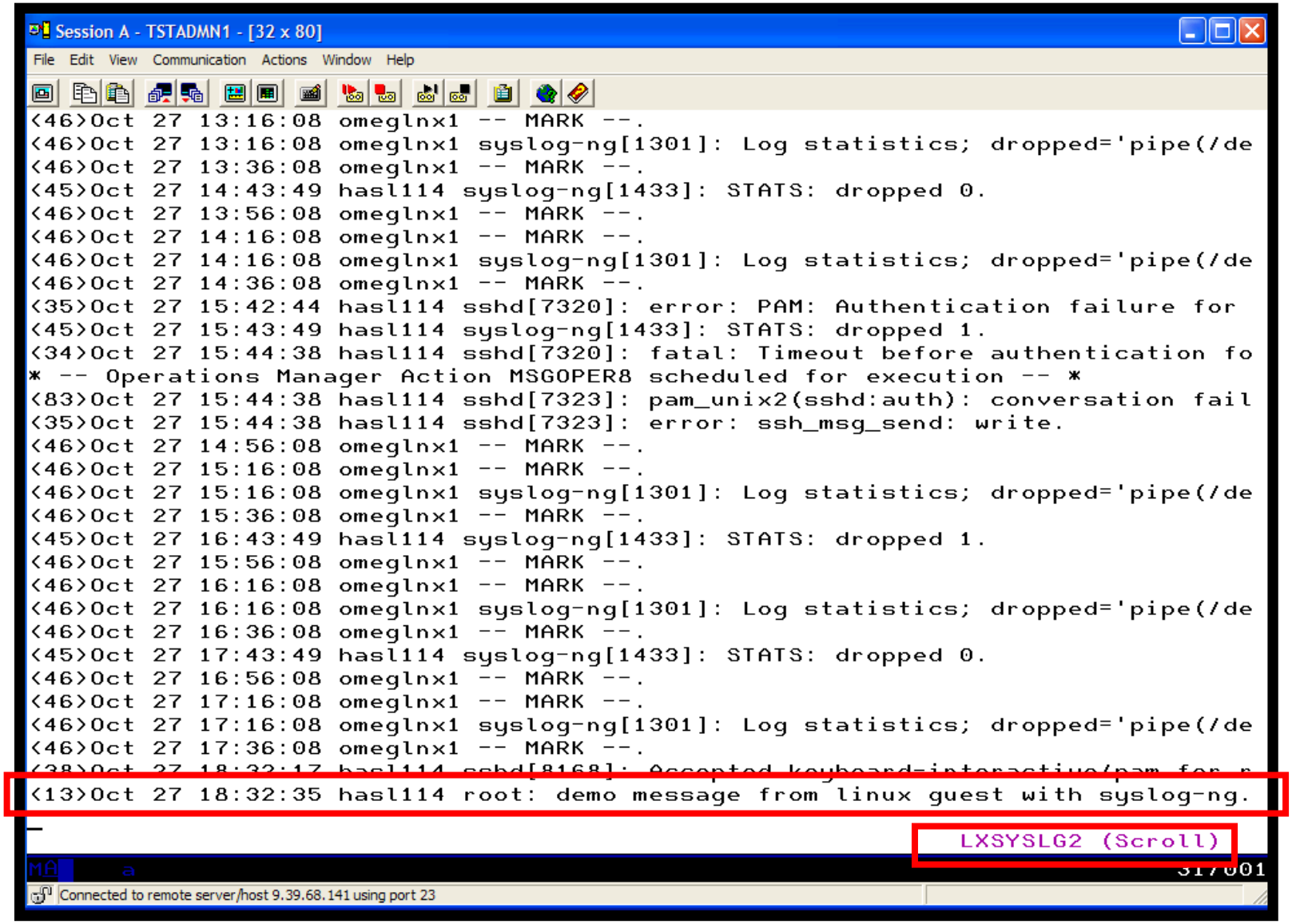


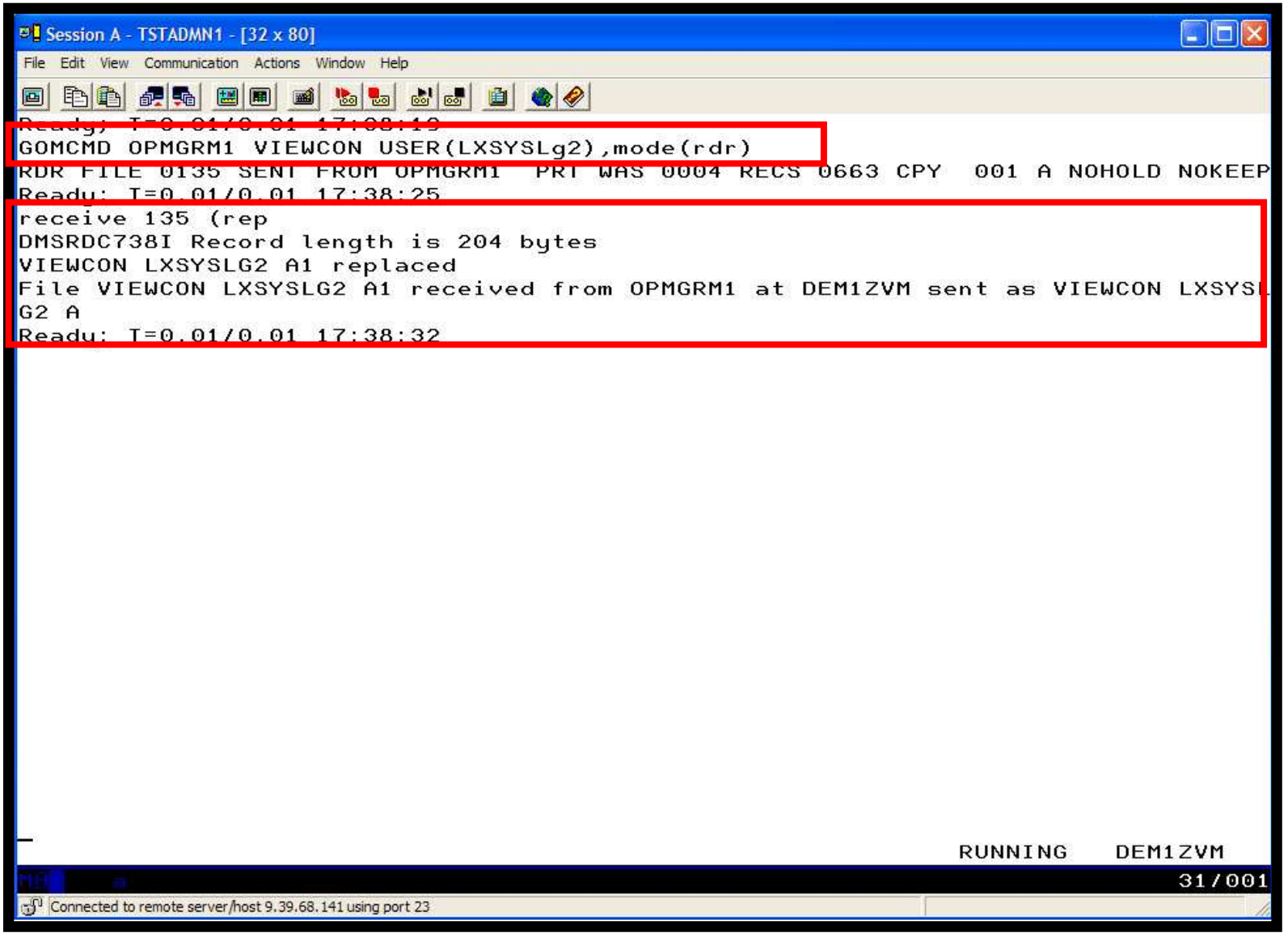


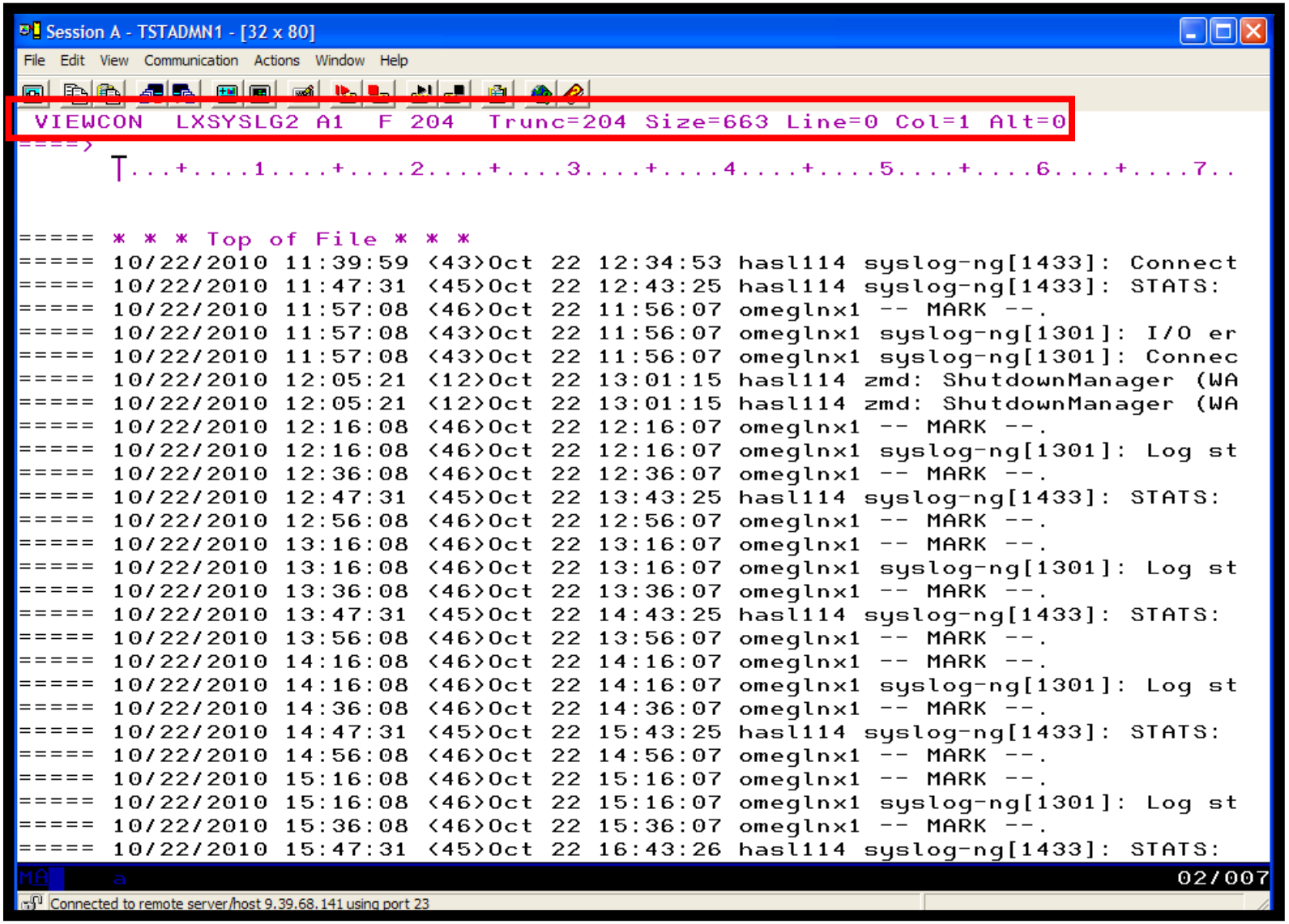












Scenario 8: How Do You Do That?

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**

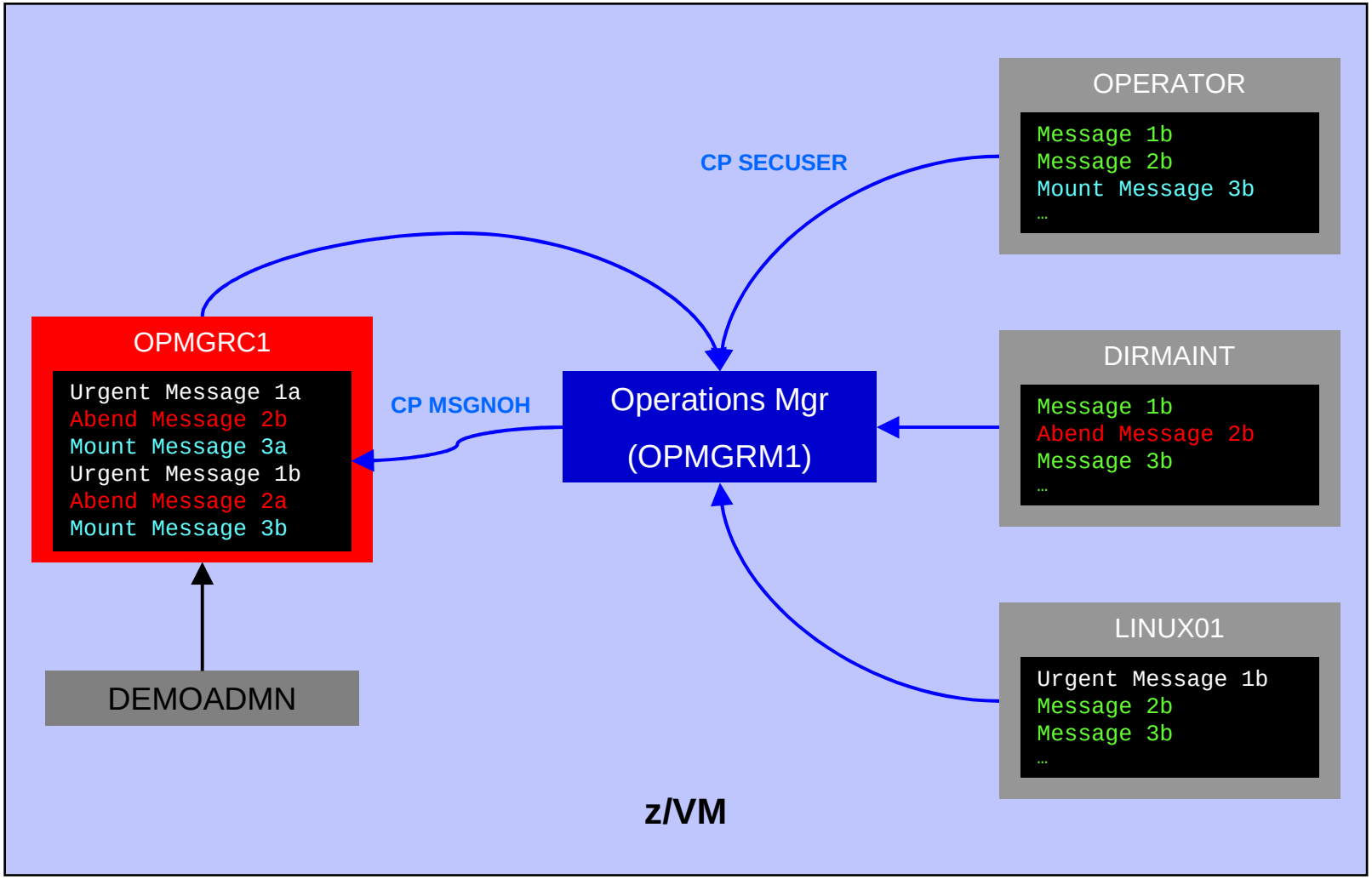
```
*
DEFTCPA NAME(LNXSYSLG),+
  TCPUSER(TCPIP),+
  TCPAPPL(GOMRSYL),+
  TCPADDR(000.000.000.000),+
  TCPPORT(00514),+
  PARM(LXSYSLOG03330417UTF8)
*
DEFTCPA NAME(LNXSYSL2),+
  TCPUSER(TCPIP),+
  TCPAPPL(GOMRSYL),+
  TCPADDR(000.000.000.000),+
  TCPPORT(00515),+
  PARM(LXSYSLG203330417UTF8)
```

- **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 DMSxxxxxxE 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

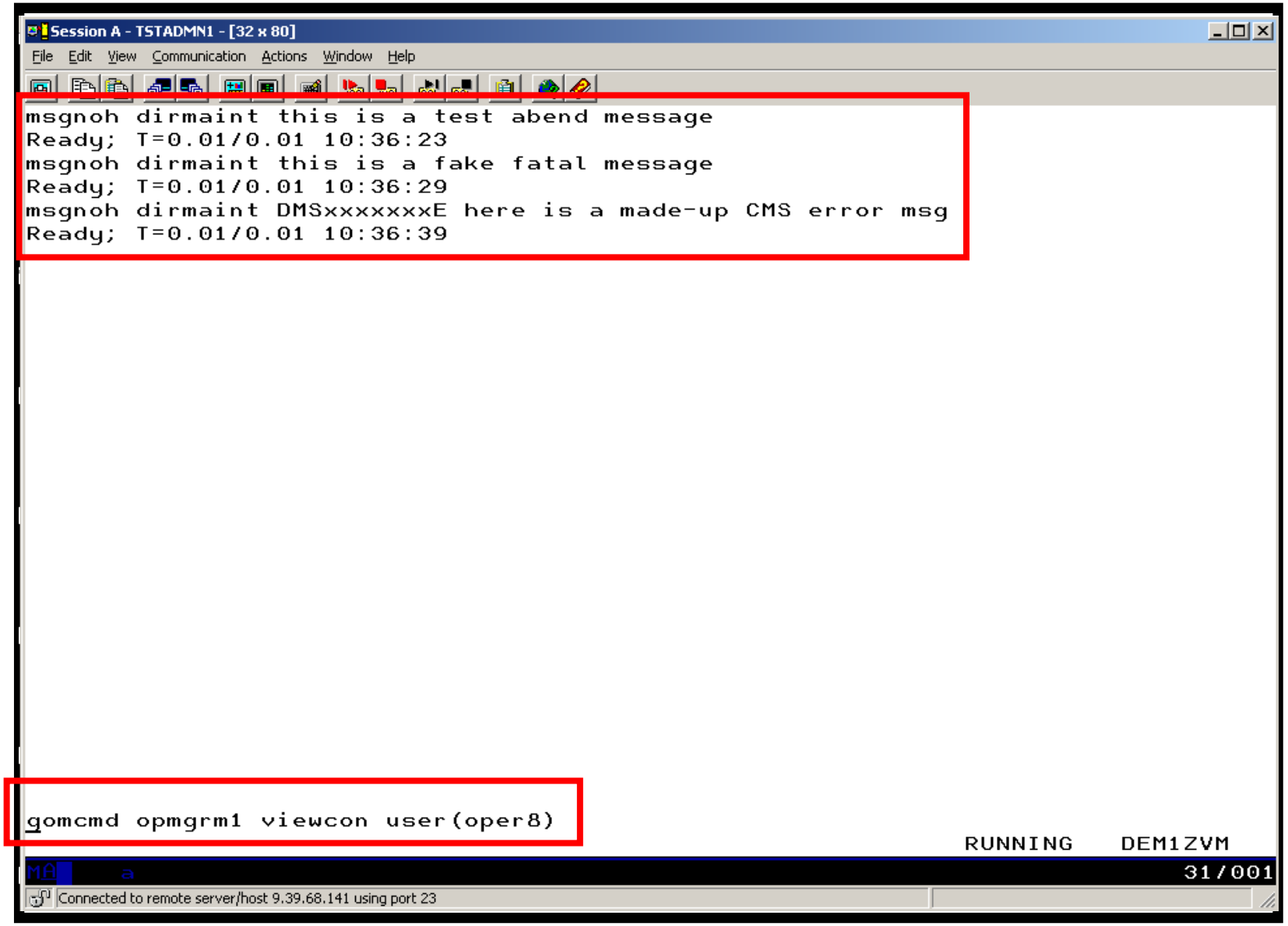
- **From another user ID, run an EXEC to send multiple messages to the Operations Console**

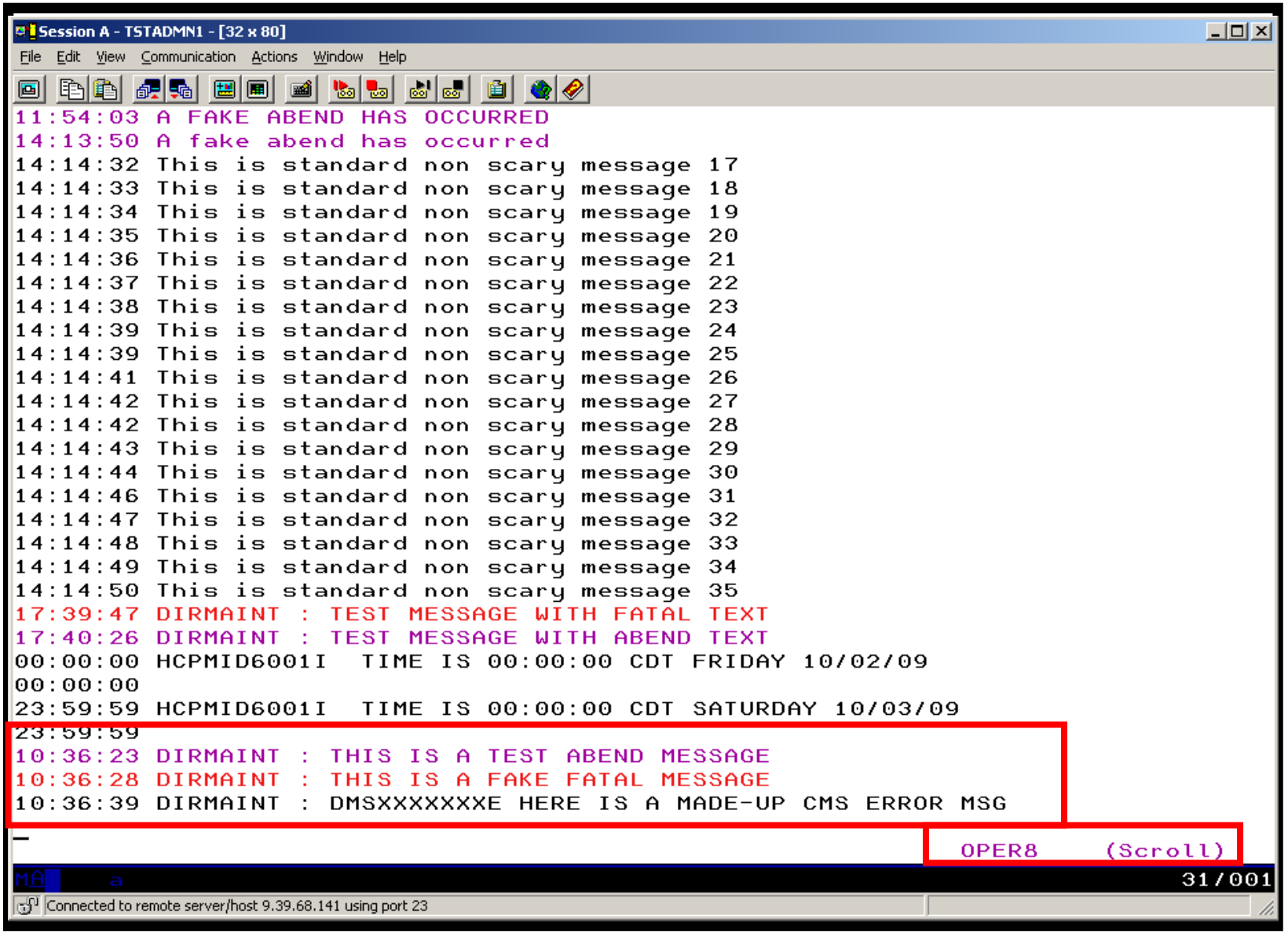
```
lotsmsgs
```

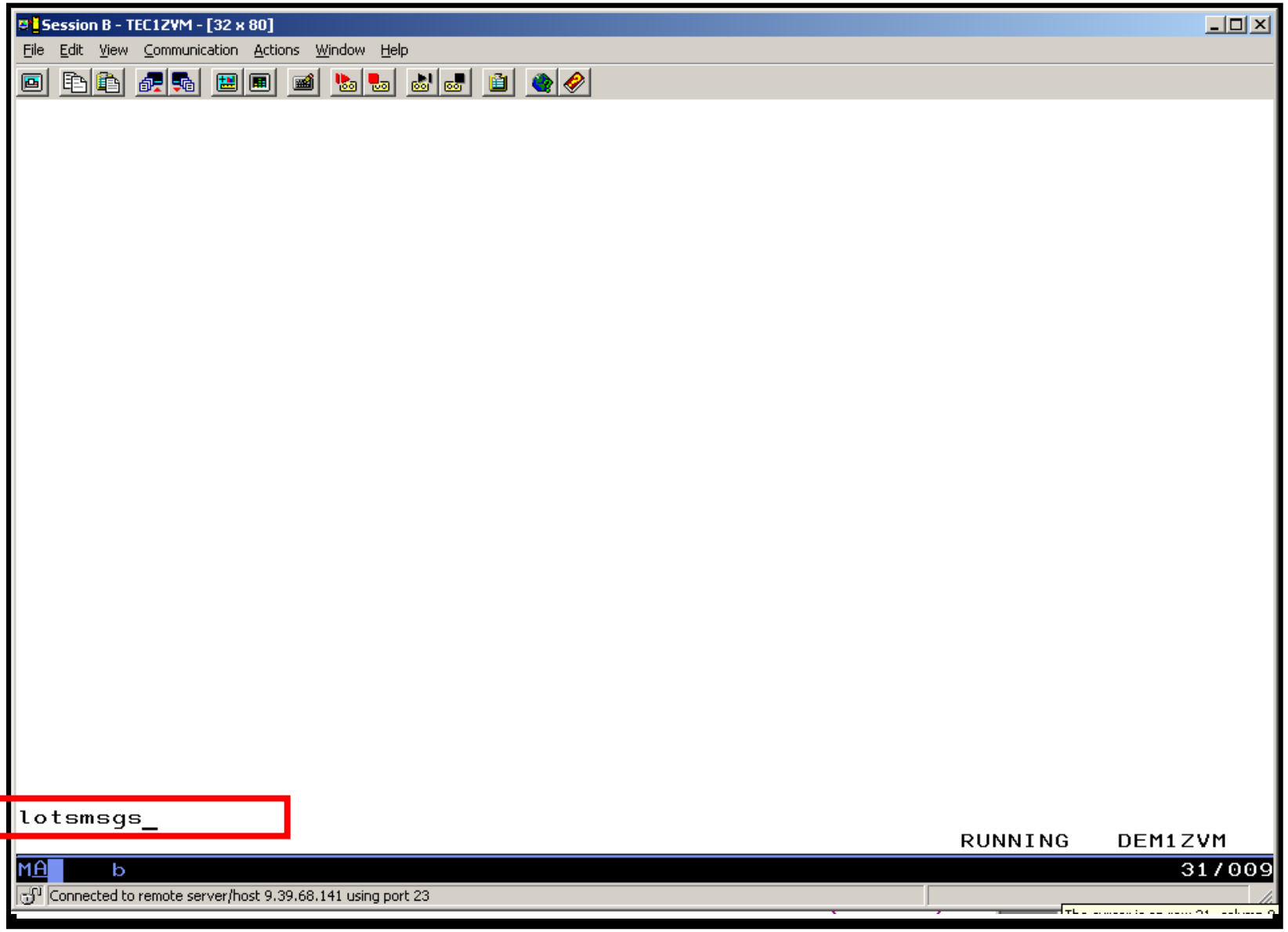
- **View the “Operations Console” to see the messages**

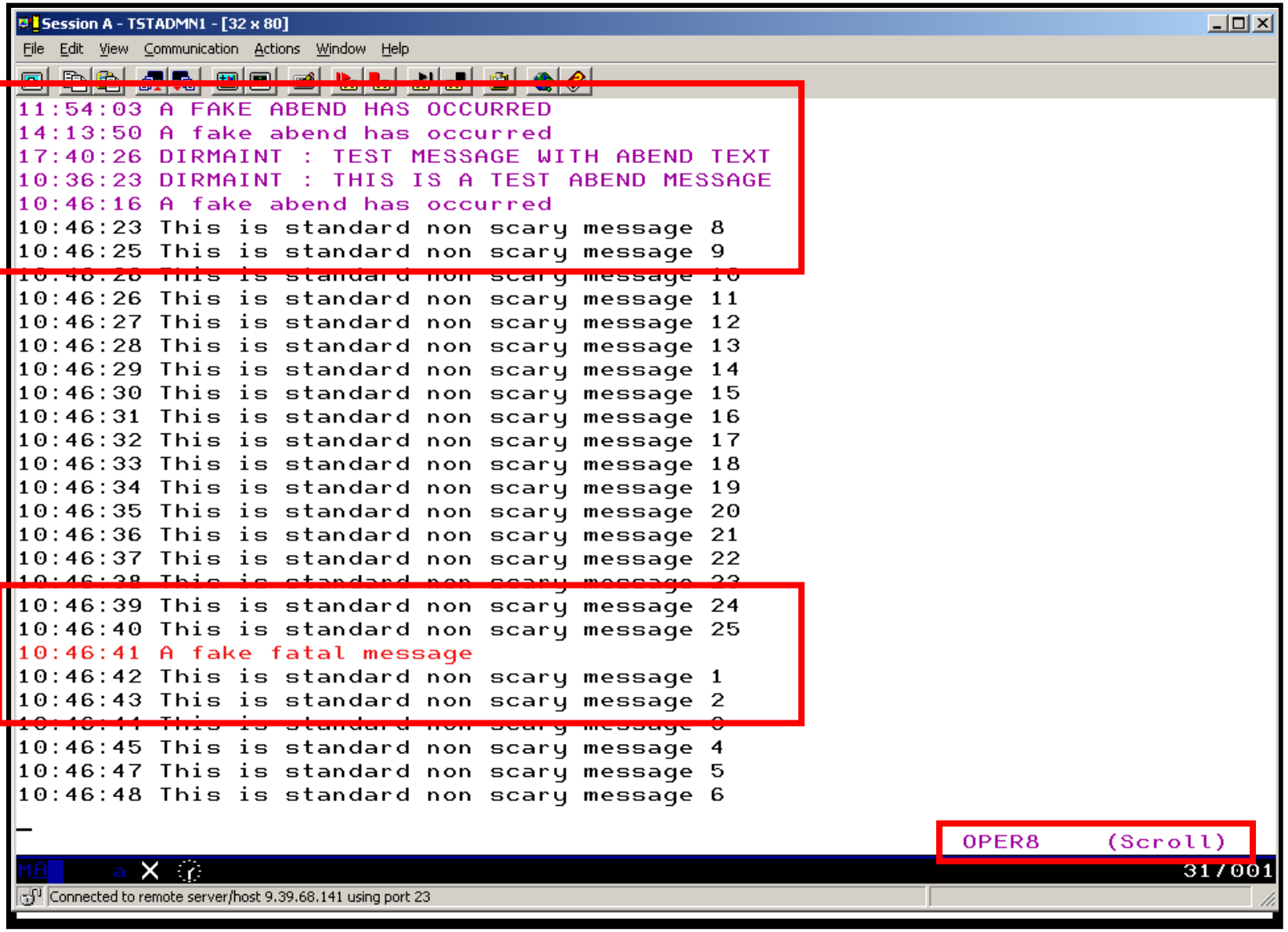
```
gomcmd opmgrm1 viewcon user(oper8)
```

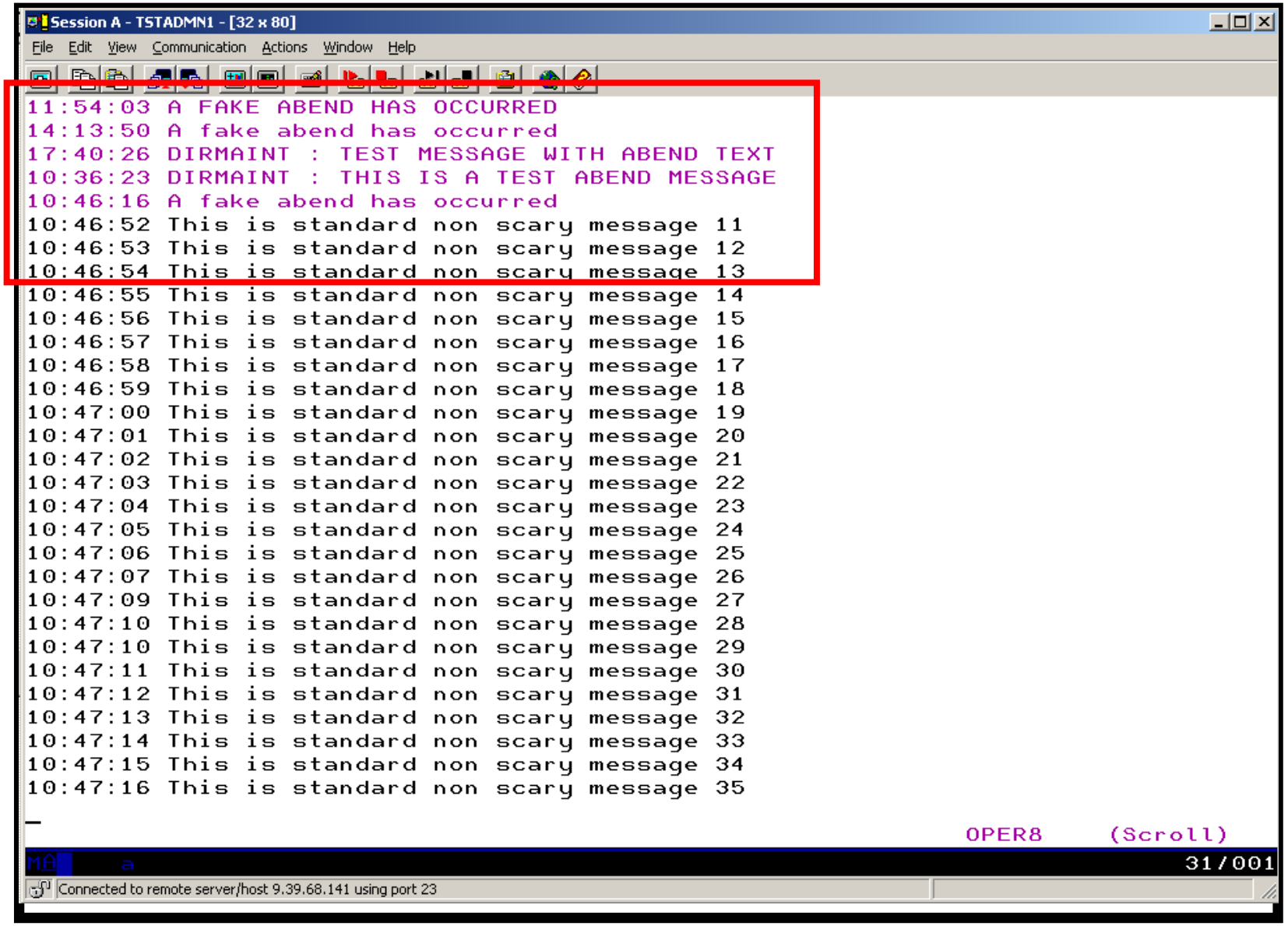
- **Watch the scrolling, held messages, etc.**











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)
```

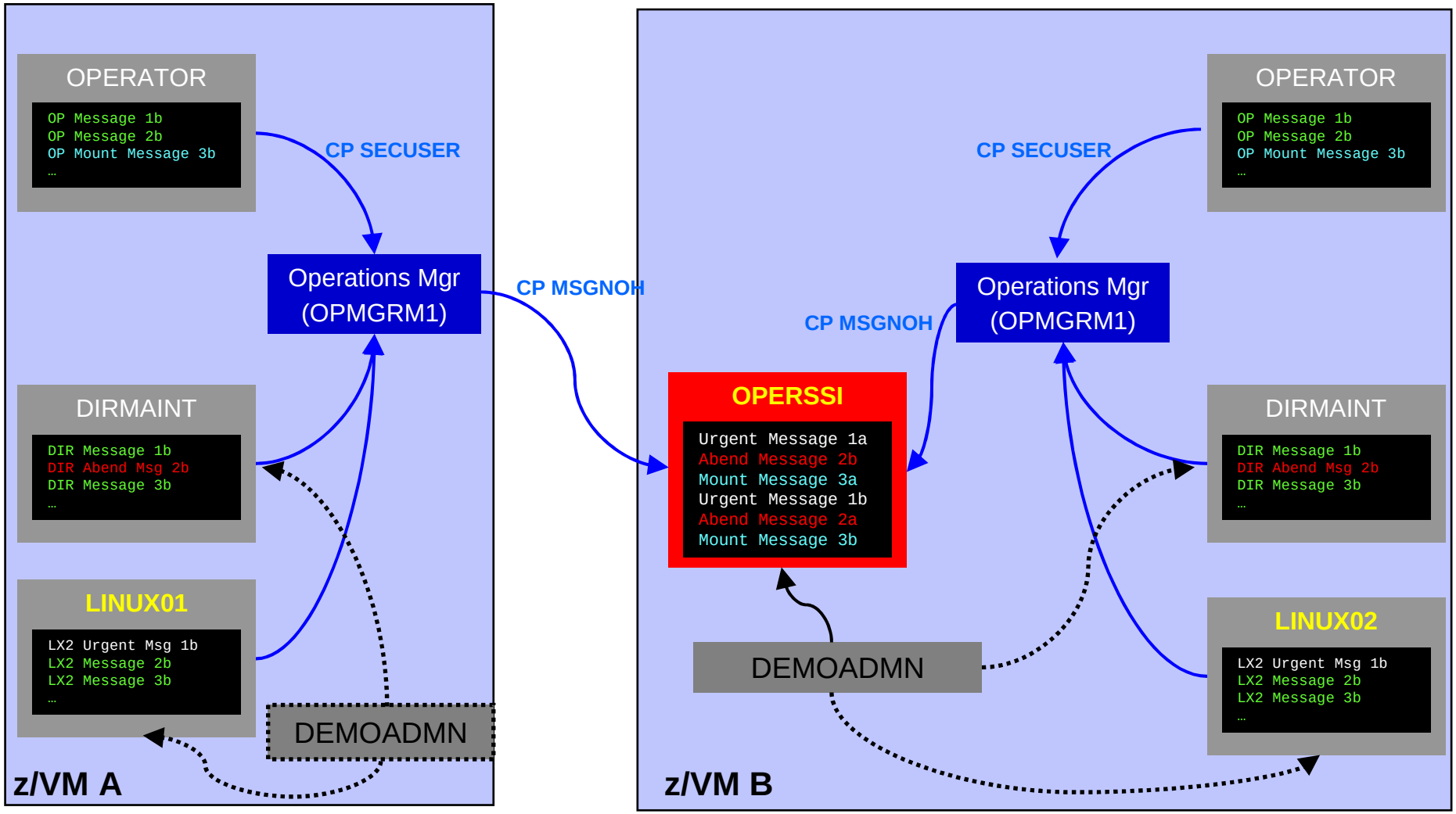
Actions in Operations Manager:

```
*
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 that are **in an SSI cluster**

- **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)**

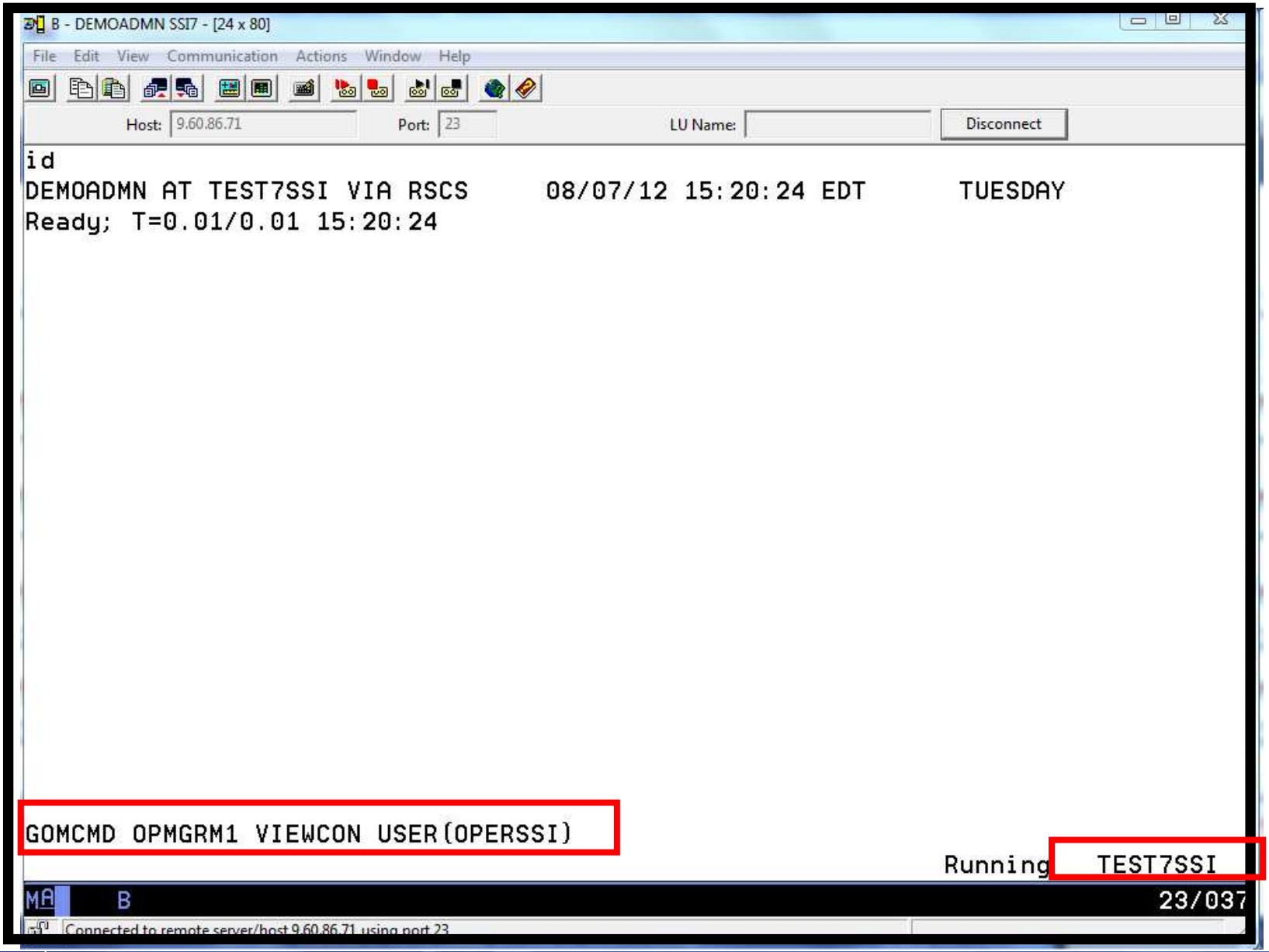
```
gomcmd opmgrm1 viewcon user(operssi)
```

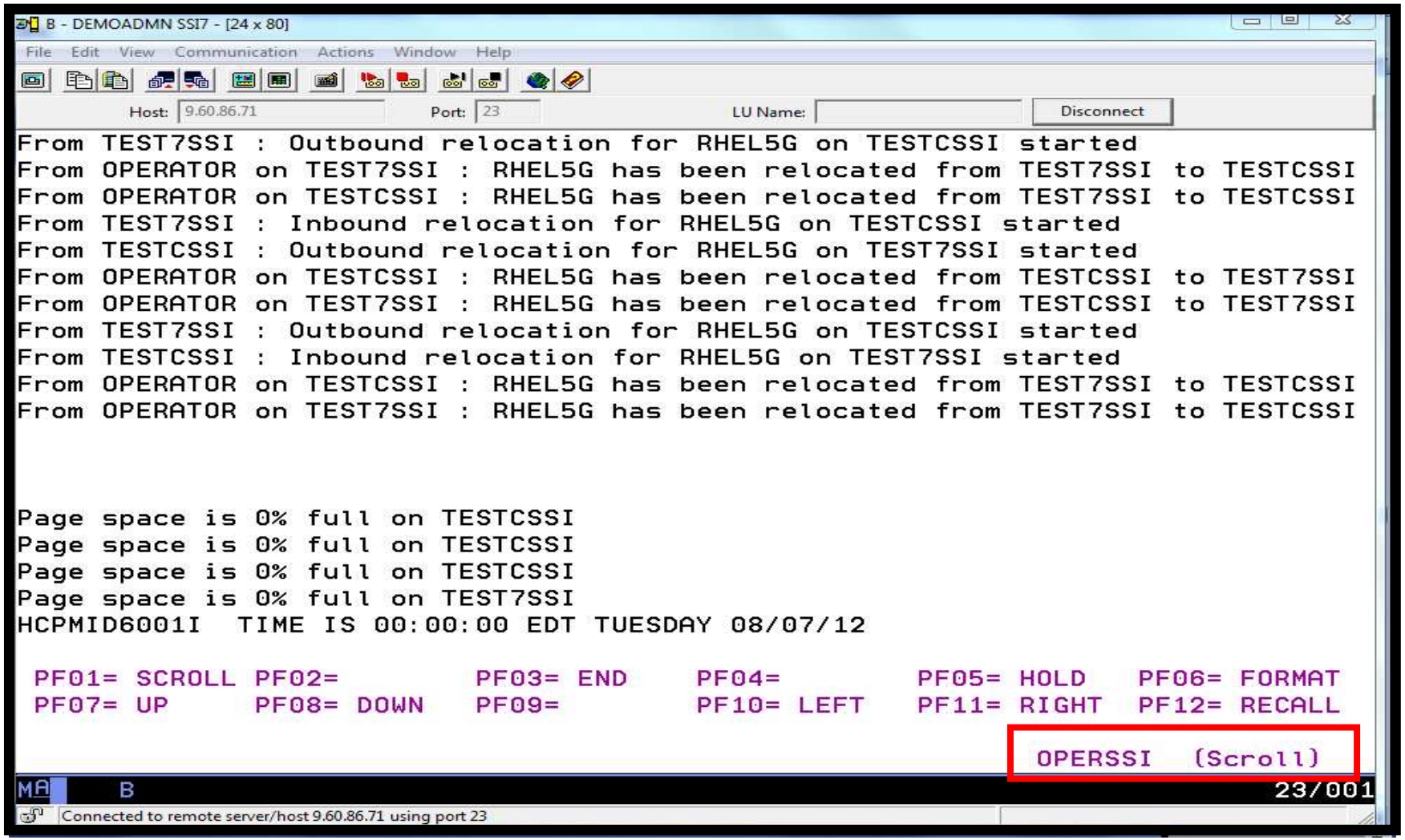
- **On System A (TESTCSSI), find a Linux guest running disconnected locally and relocate it**

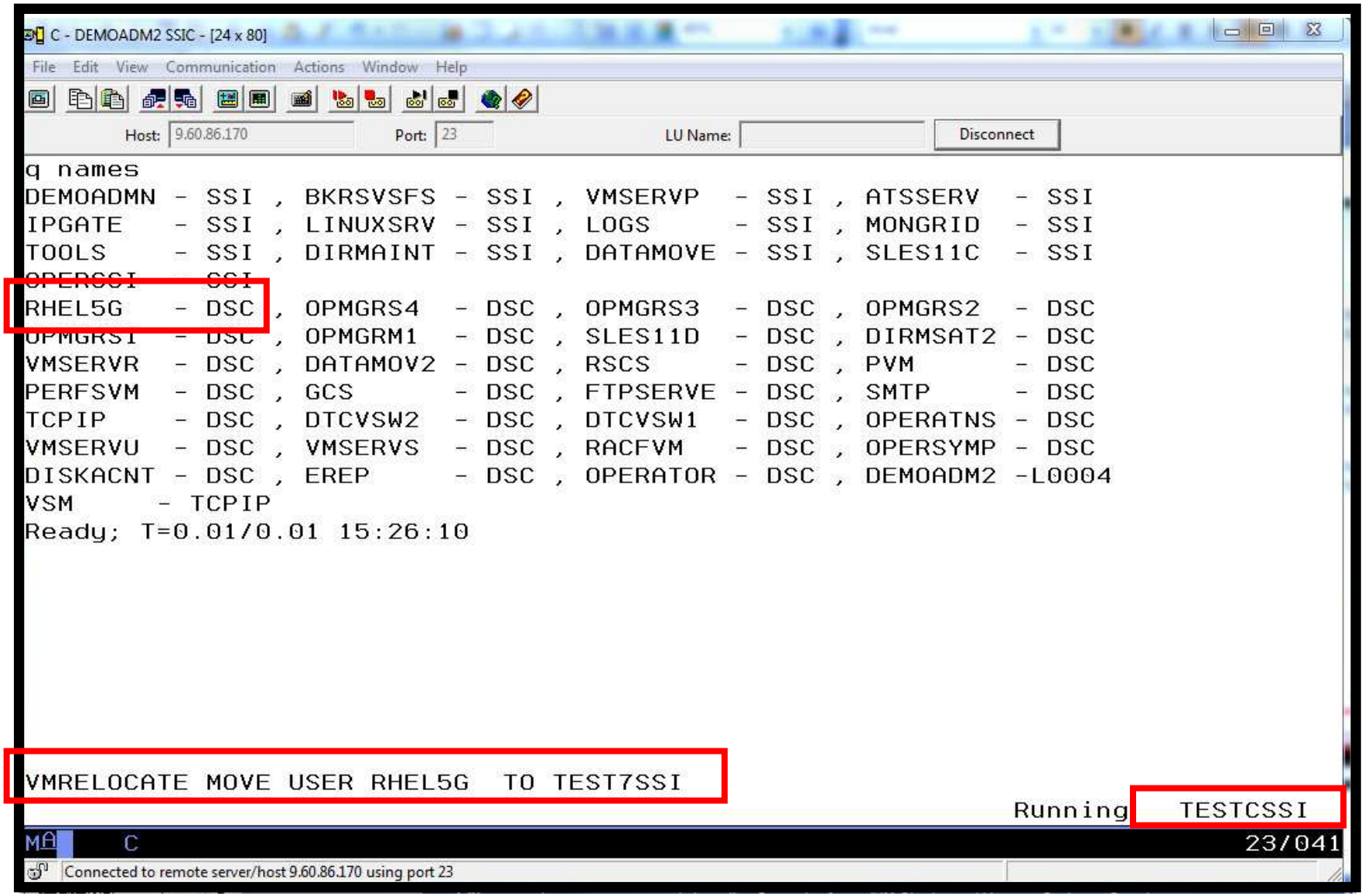
```
q names
```

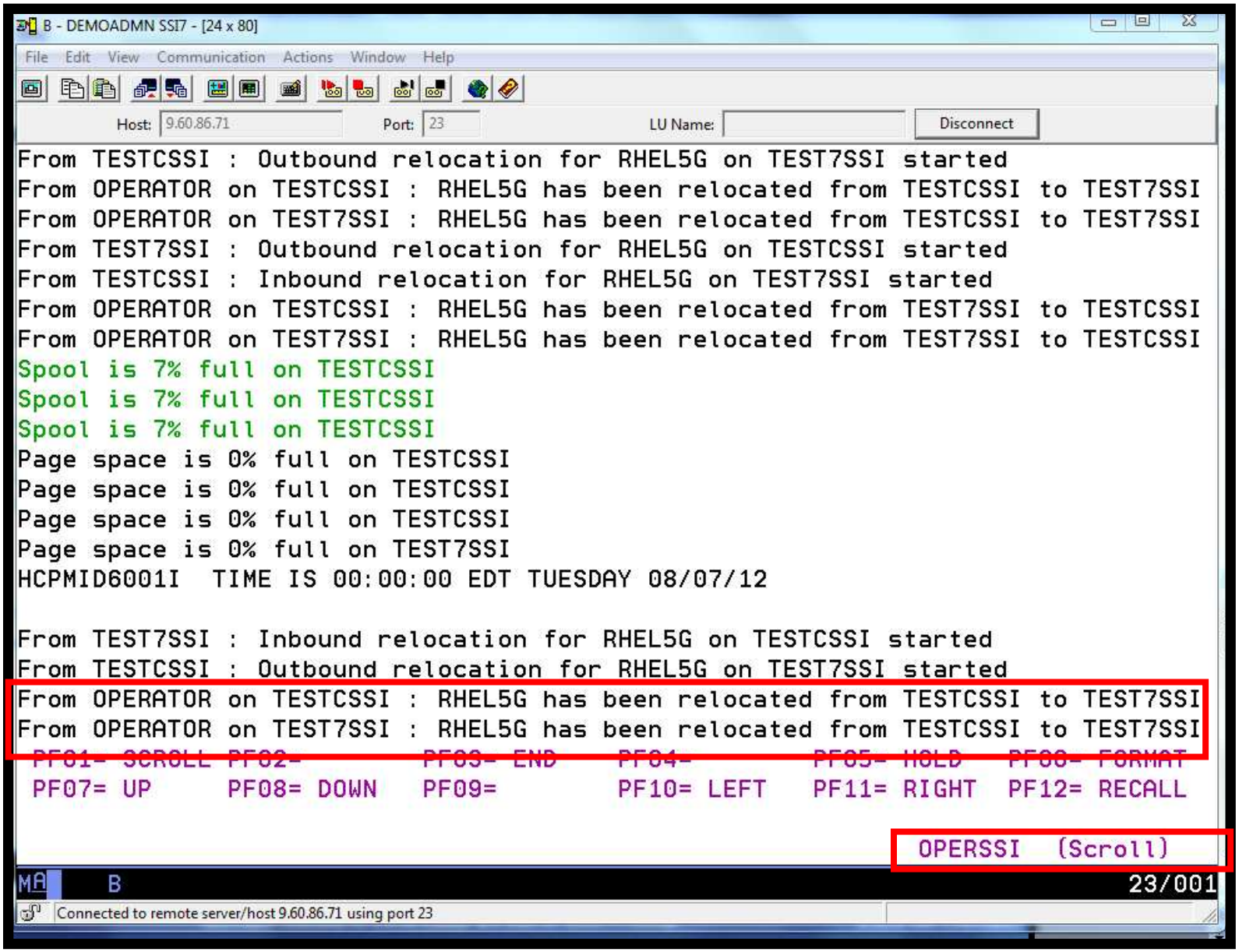
```
VMRELOCATE MOVE USER RHEL5G  TO TEST7SSI
```

- **Note the messages received on OPERSSI on TEST7SSI from OPERATOR on both TESTCSSI and TEST7SSI**









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?

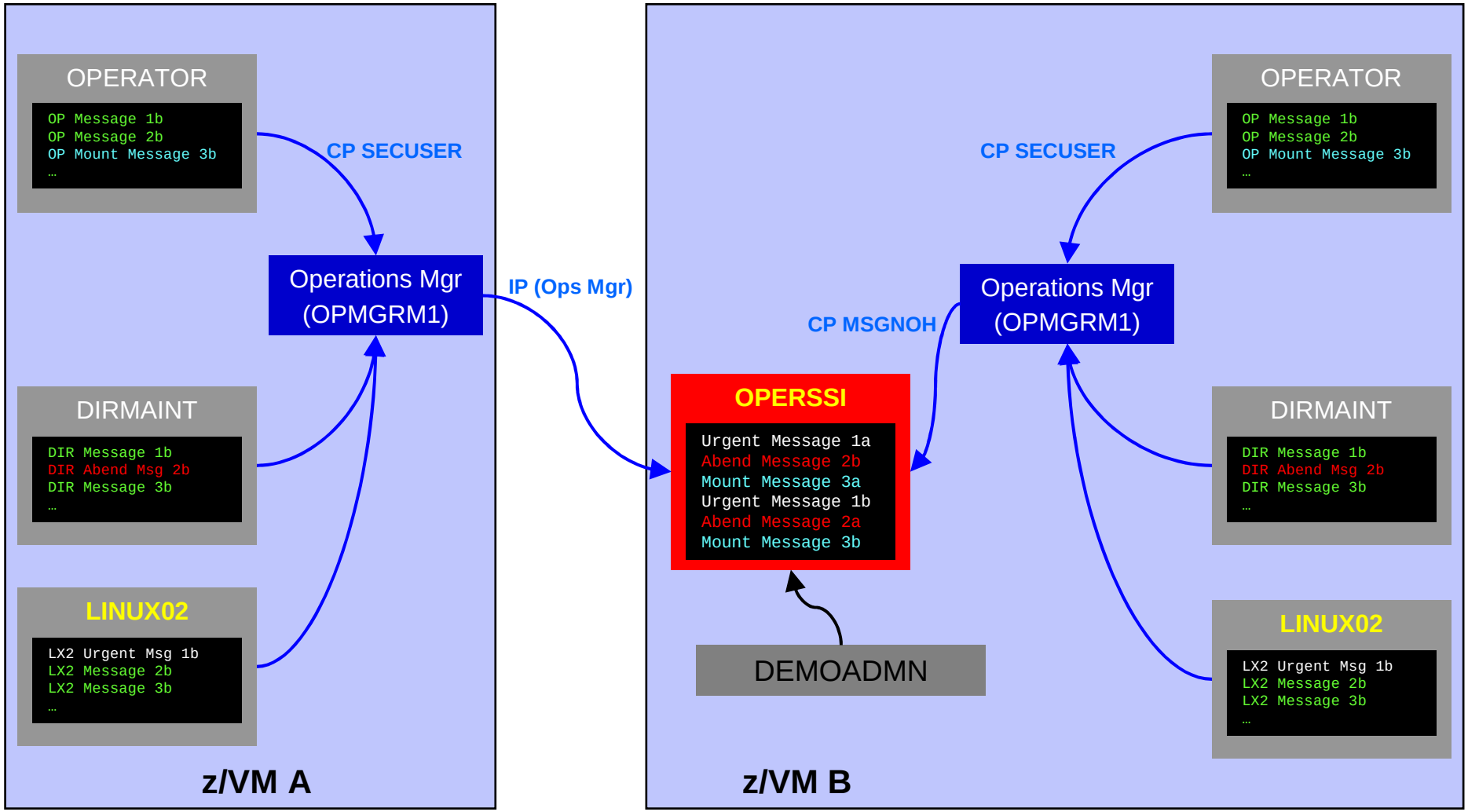
MSG2OPER EXEC (excerpts):

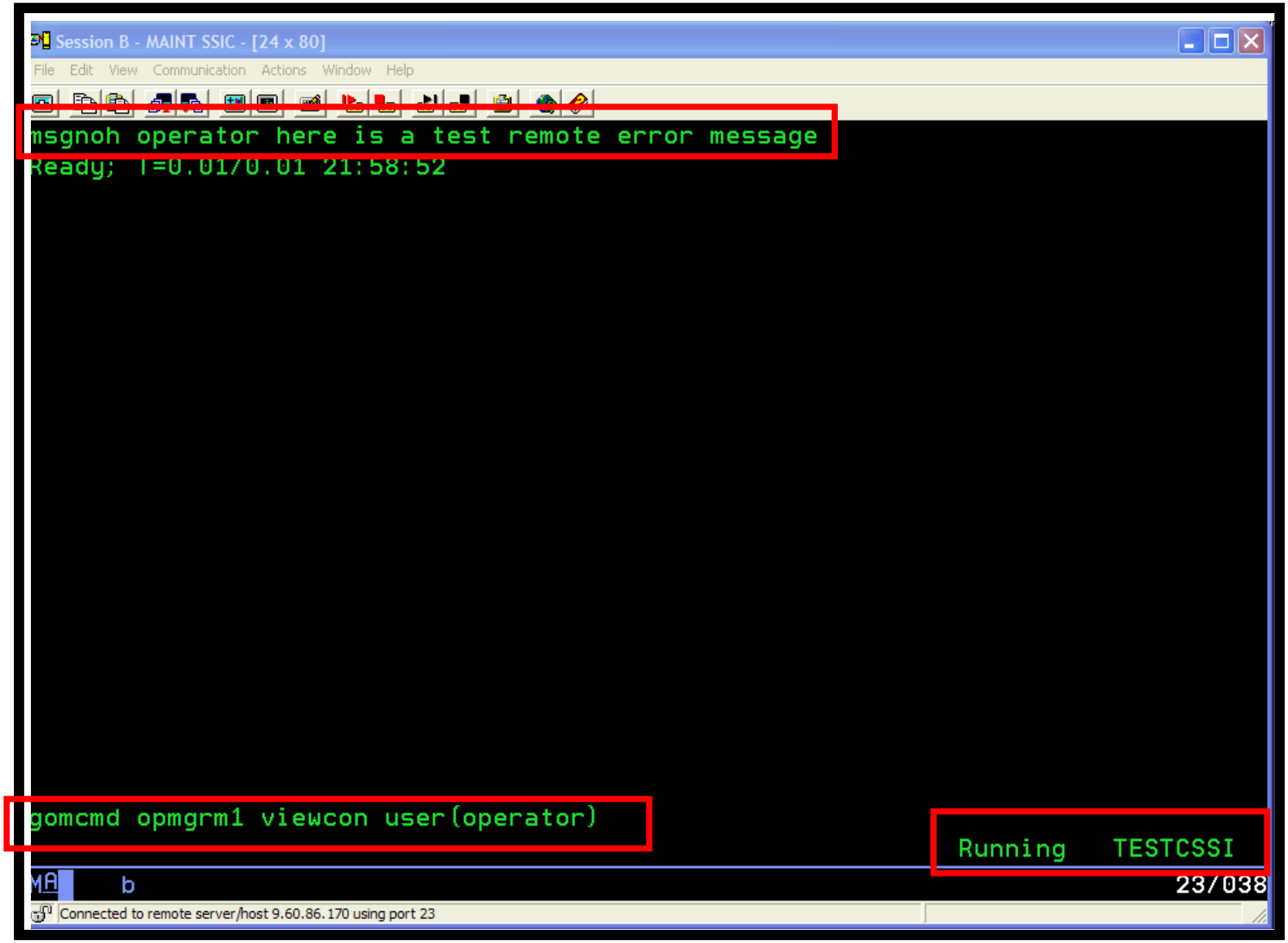
```
/*  Send a message to a central console OPERSSI for SSI cluster     */
/*                                                                  */
trace r
Address Command
Parse arg userid euser event sourcesys msgtext

/* 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
```

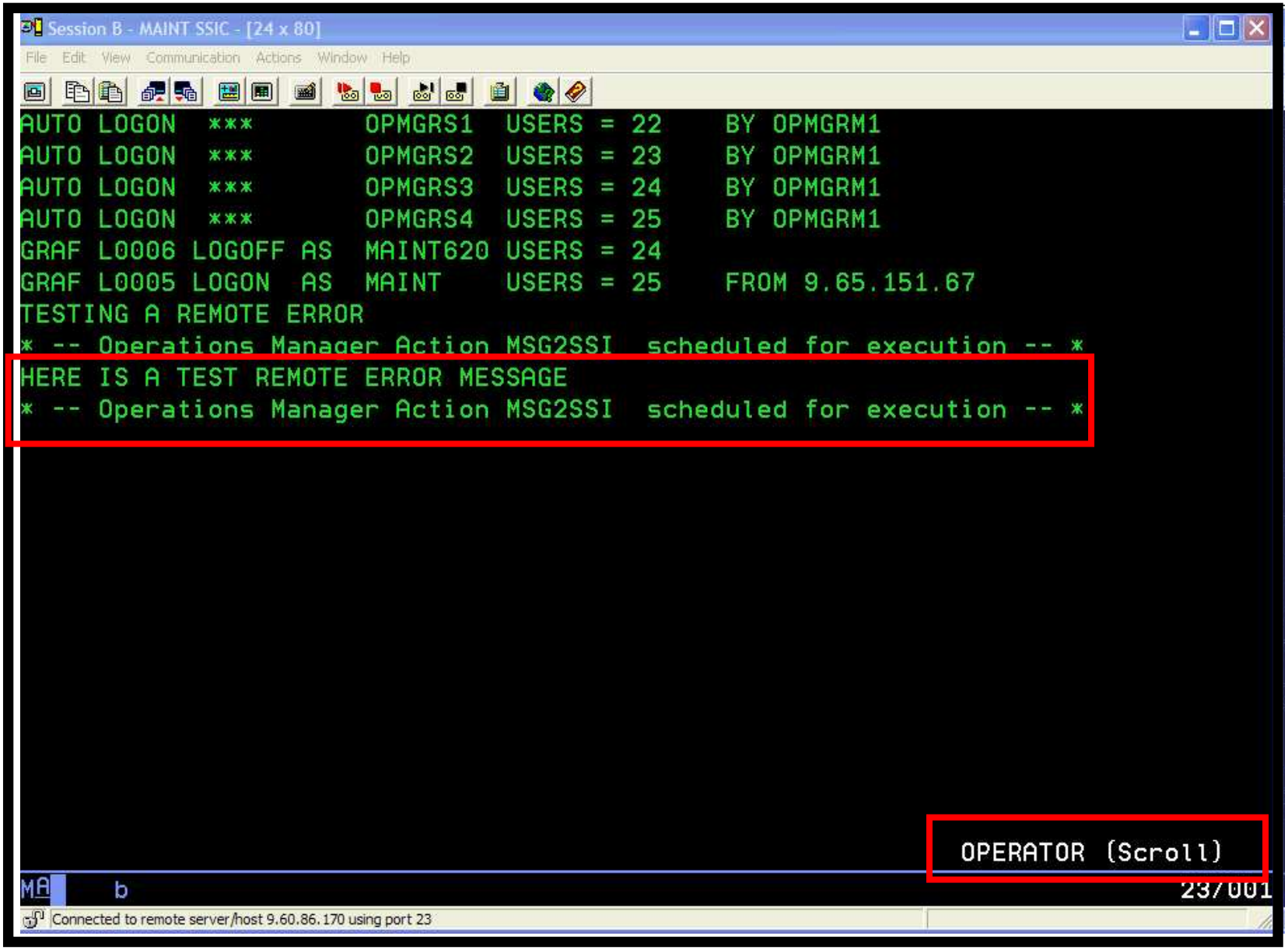
Scenario 10b: Create a Central Operations Console across multiple z/VM systems that are **not** in an SSI cluster

- **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

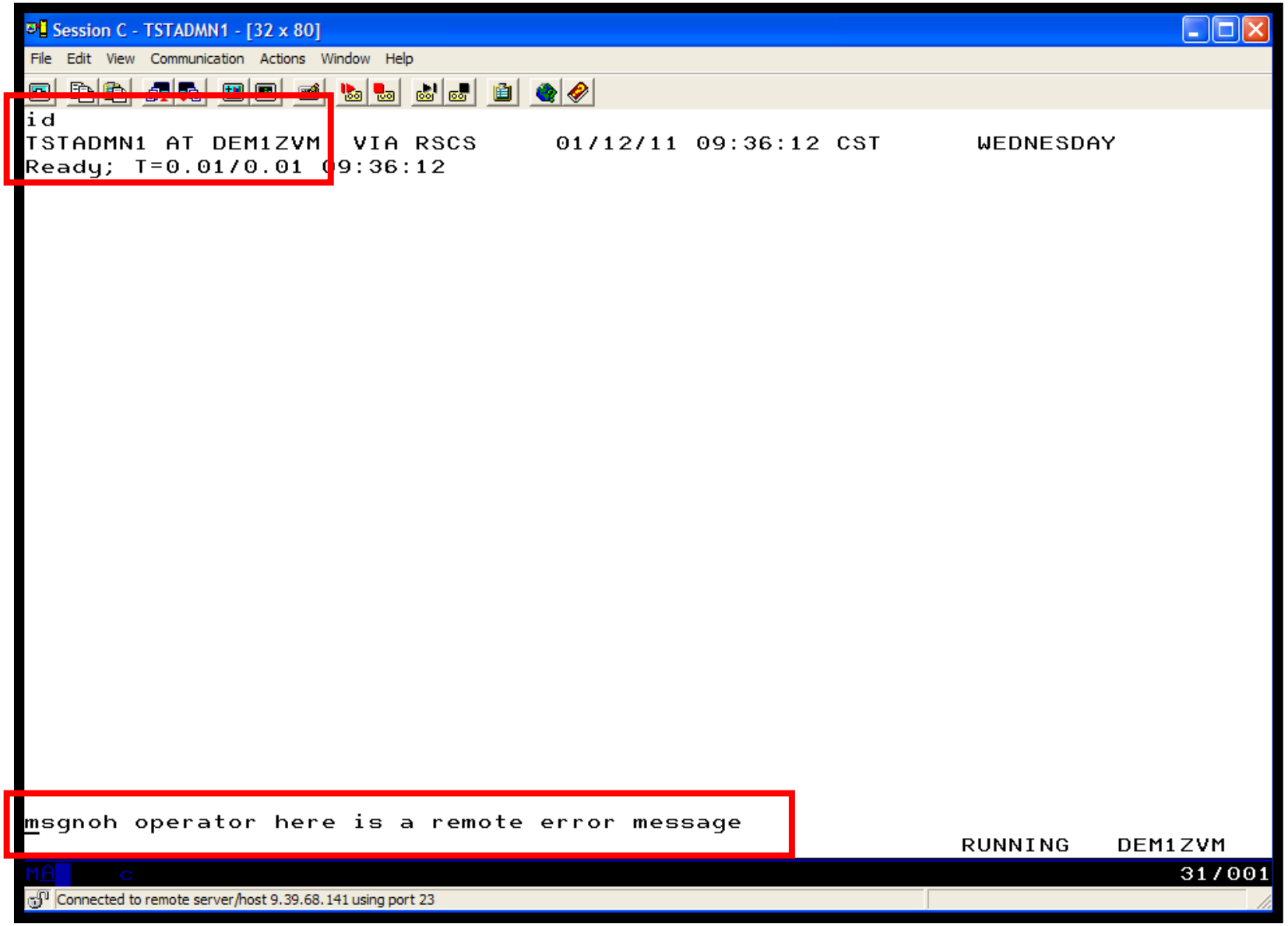
- **On System A (DEM1ZVM) put an “error” message on the OPERATOR console**
  - Must contain the text “remote error”

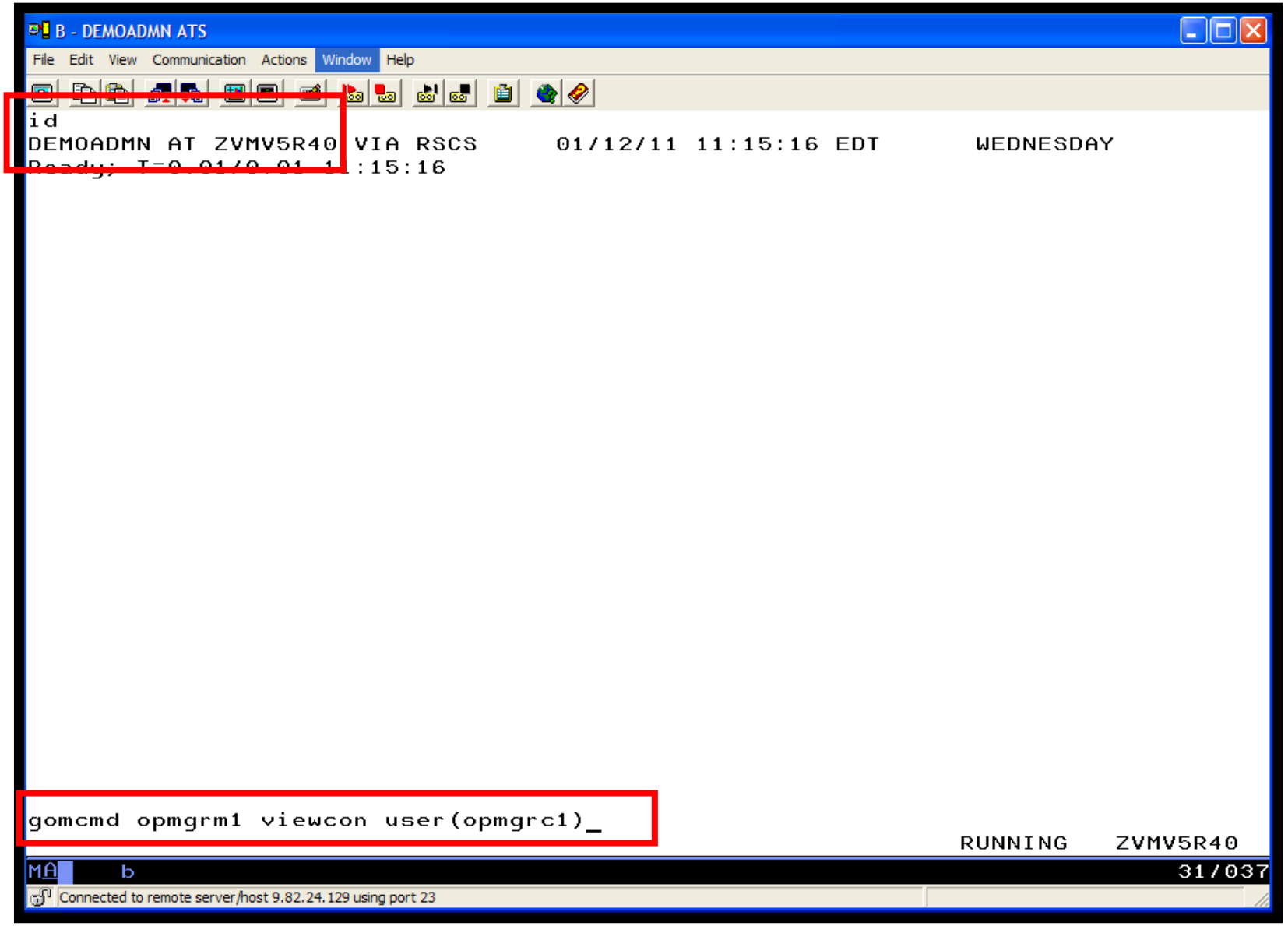
```
msgnoh operator here is a remote error message
```

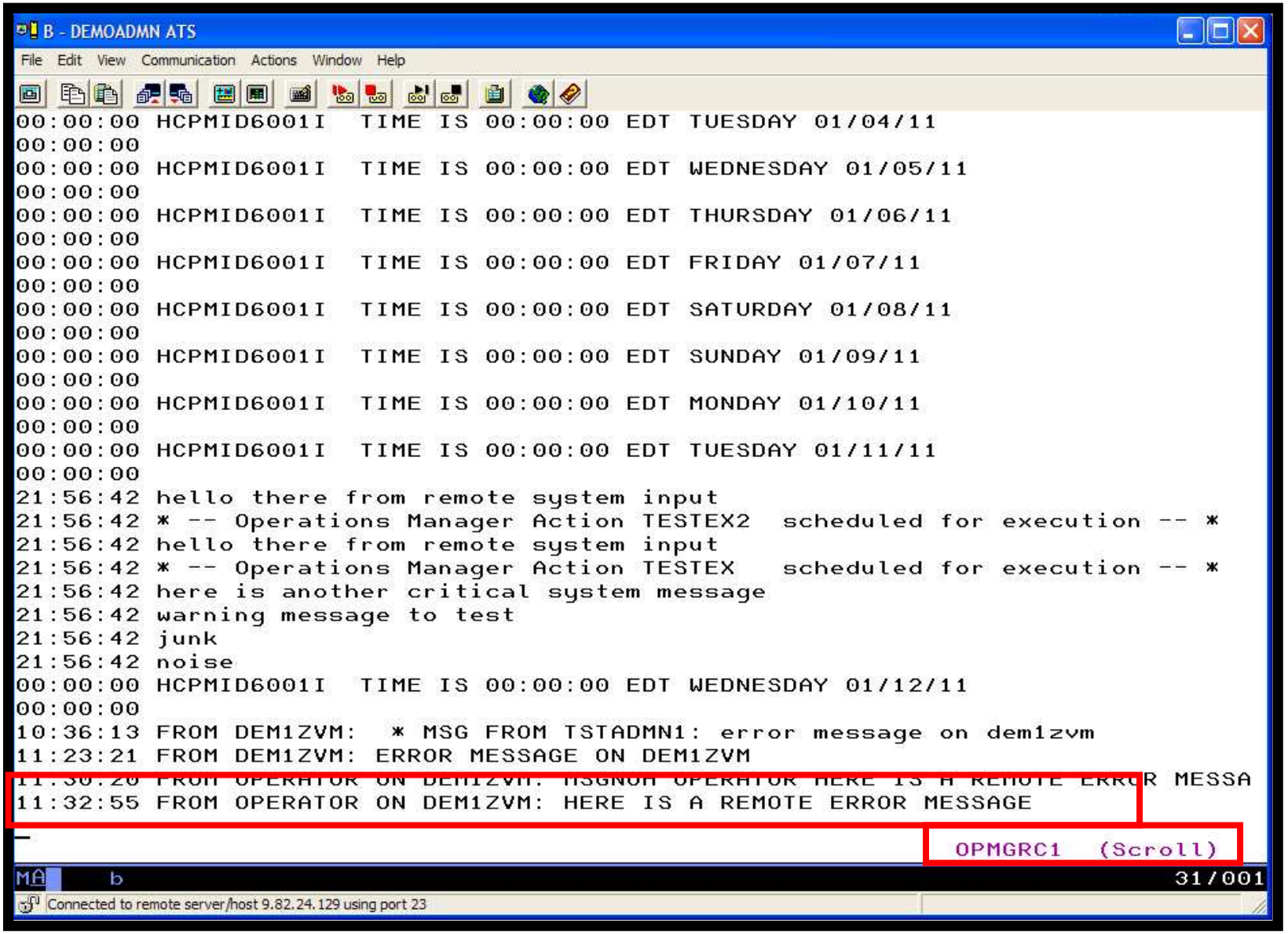
- **View the “Operations Console” (user ID OPMGRC1) on System B (ZVMV5R40) to see the message**

```
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)
```

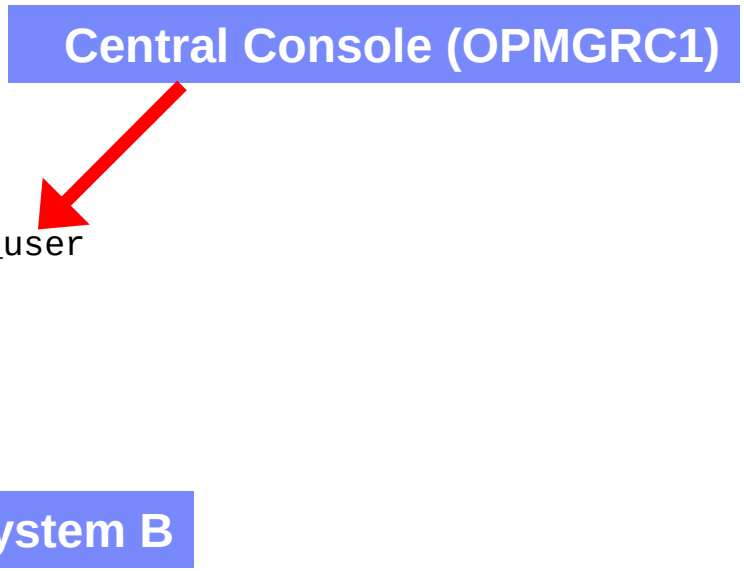
Action in Operations Manager on System A:

```
*
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)
```

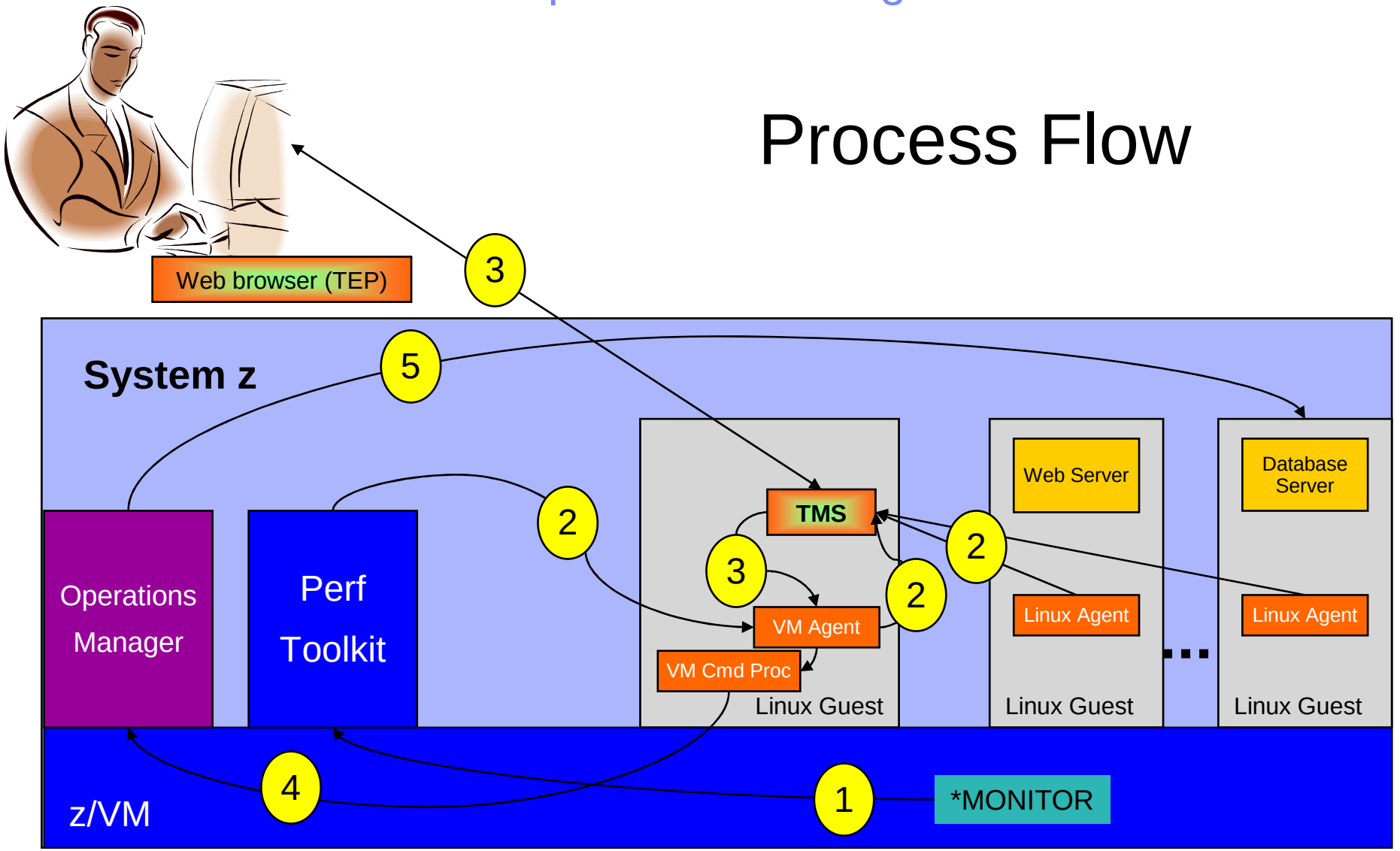
- **May also need to update TCP/IP on System B to allow Operations Manager to listen on port 63000**
- **Can alternatively use TELL (instead of GOMRSIF) to send messages from System A to System B, but requires RSCS**

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**

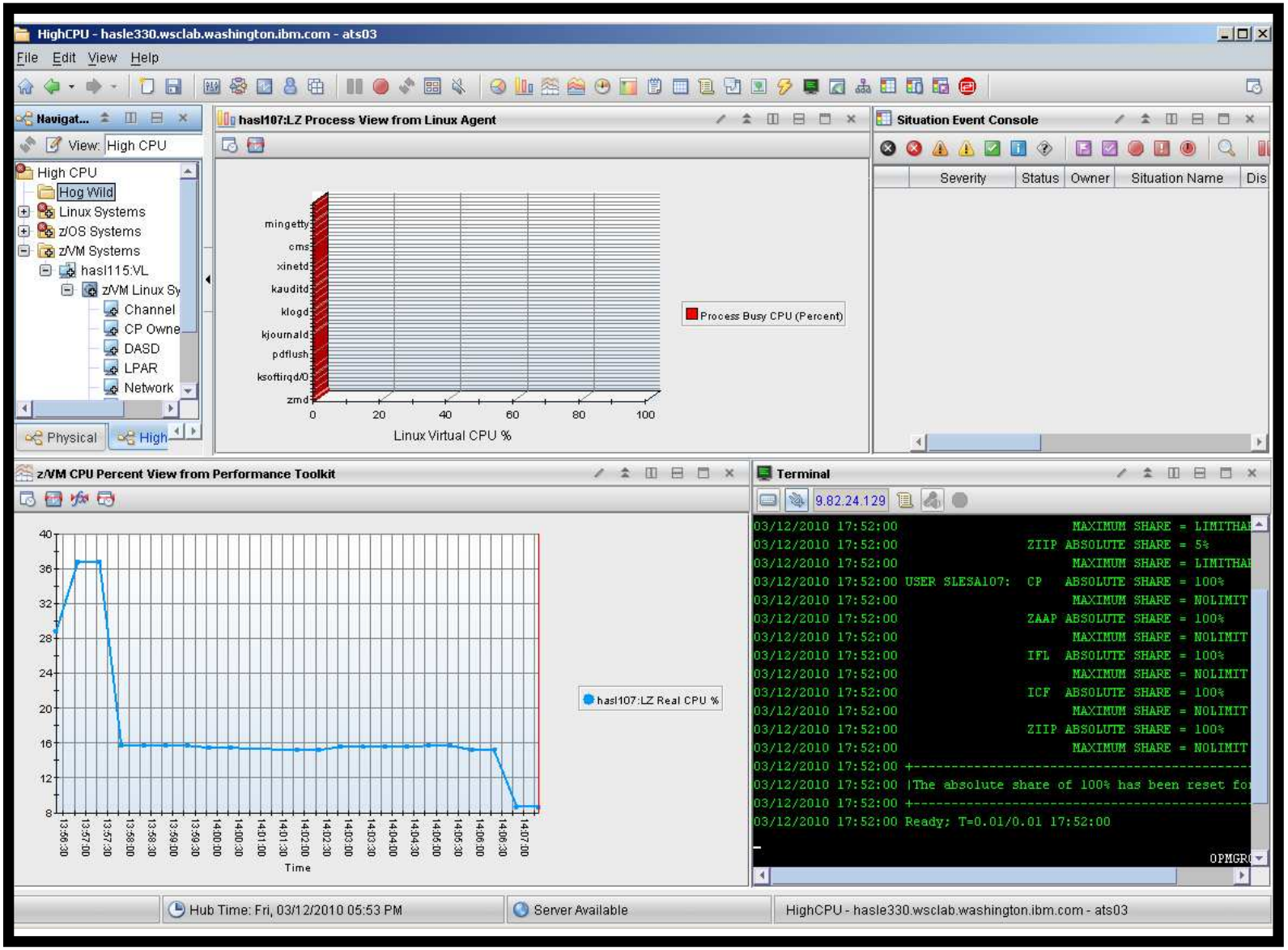
OMEGAMON XE and Operations Manager for z/VM

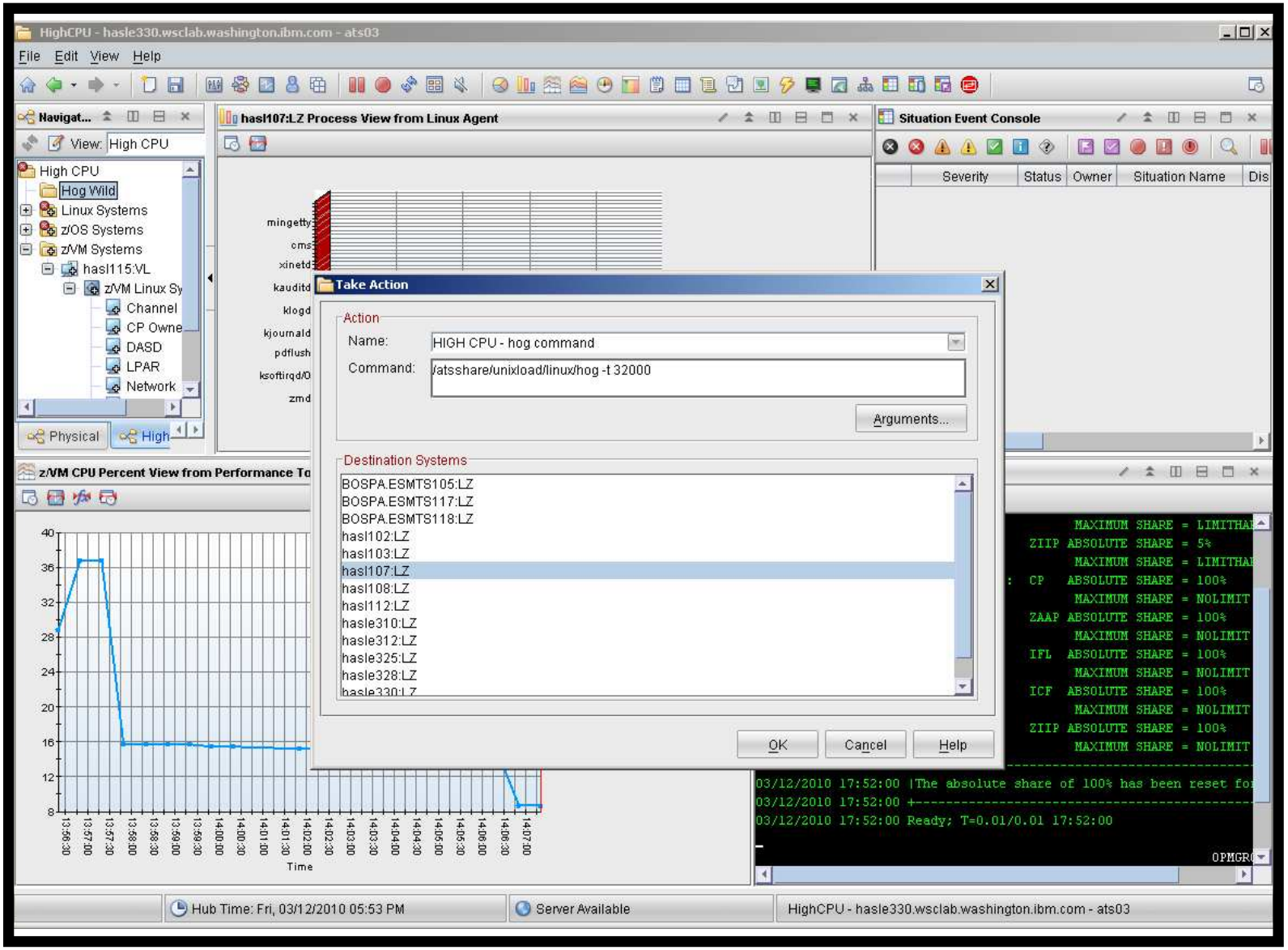
Process Flow

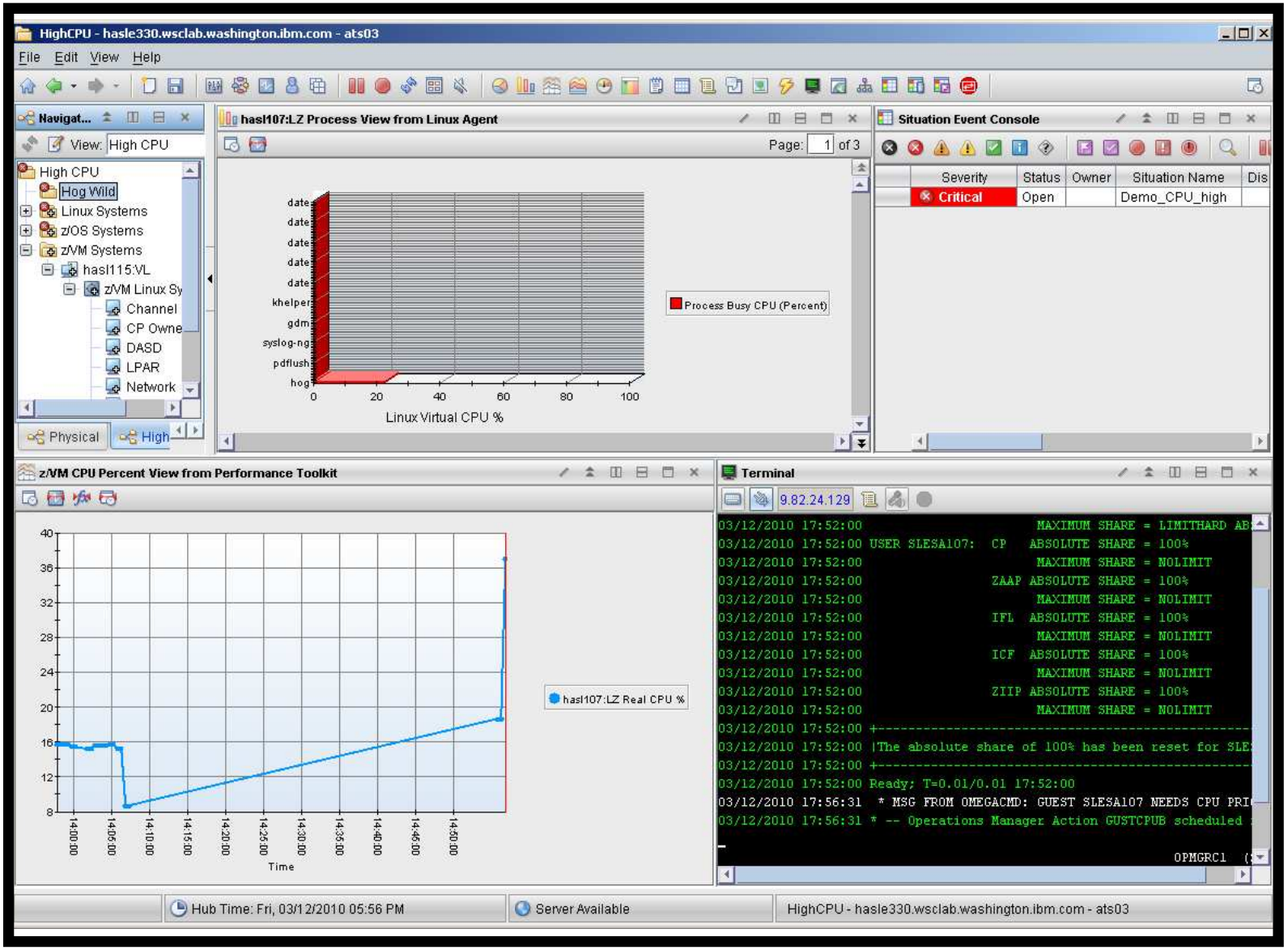


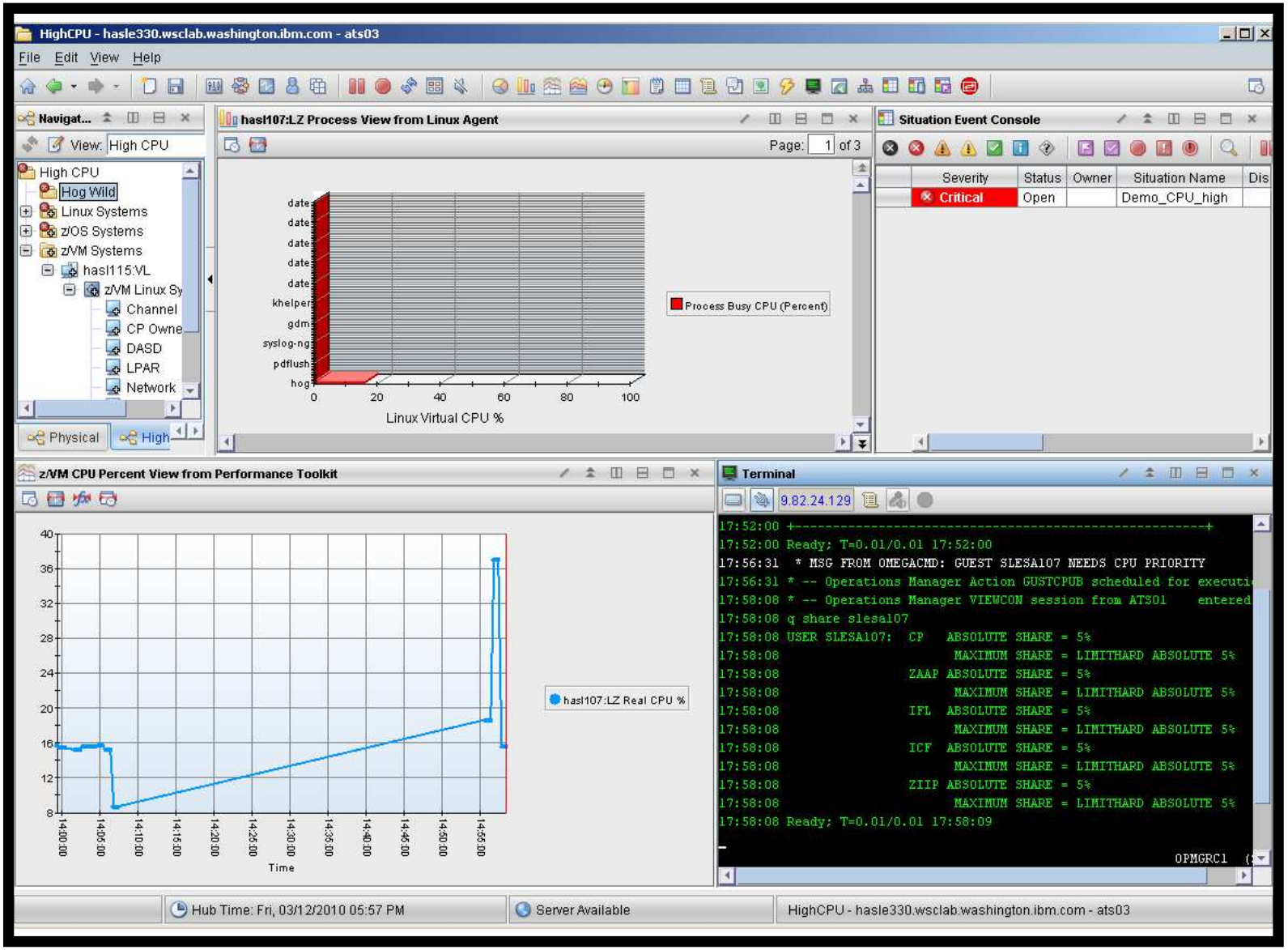
Scenario 11: Detailed Steps

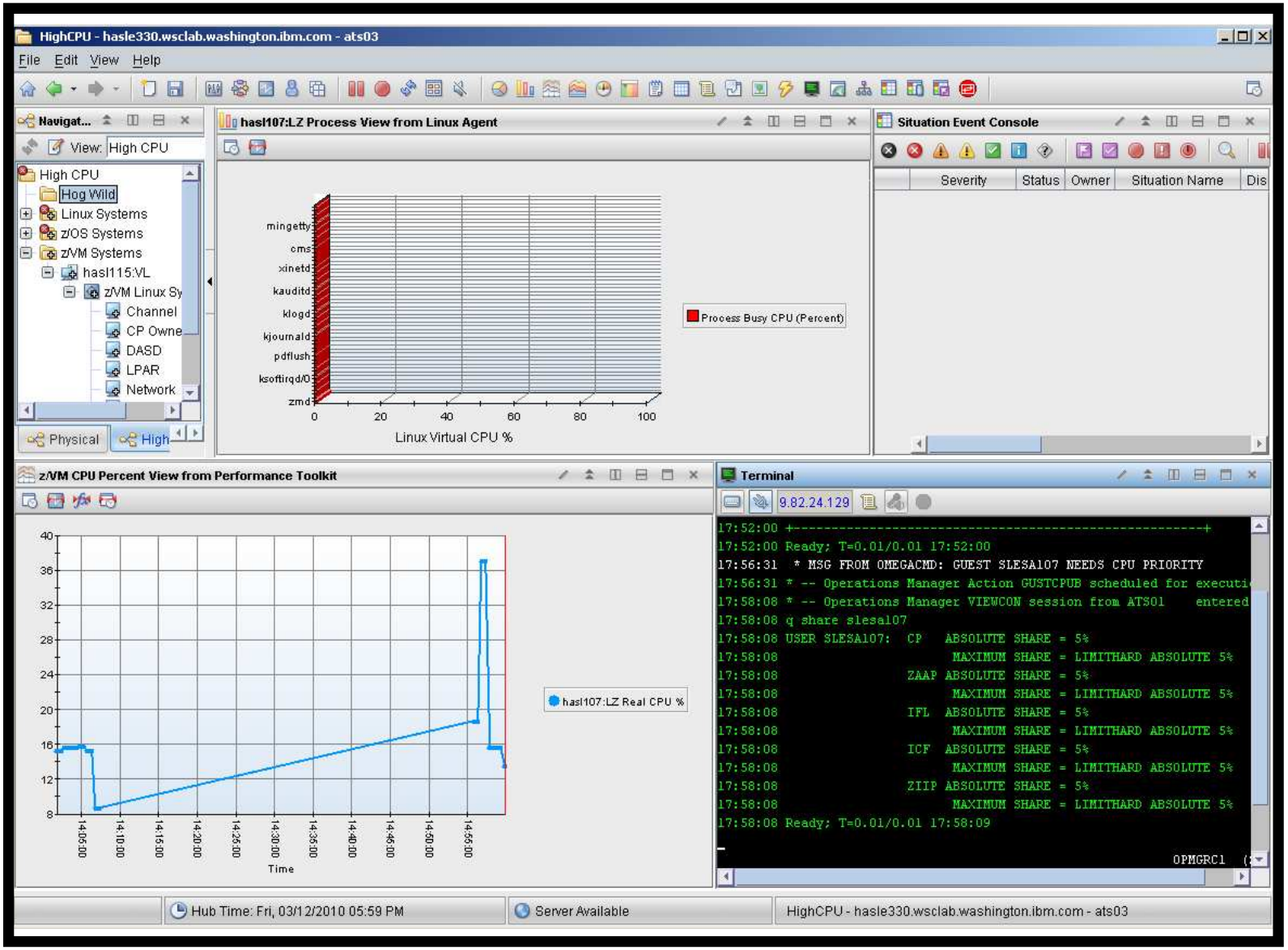
- **Create and start an application on a Linux guest that uses more than 20% of virtual CPU**
  - HOG command on our demo system
- **Updates to Tivoli Enterprise Portal**
  - z/VM CPU graph shows guest CPU % as it runs the application
  - Event pops up on situation event console to say higher than 20%
- **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)









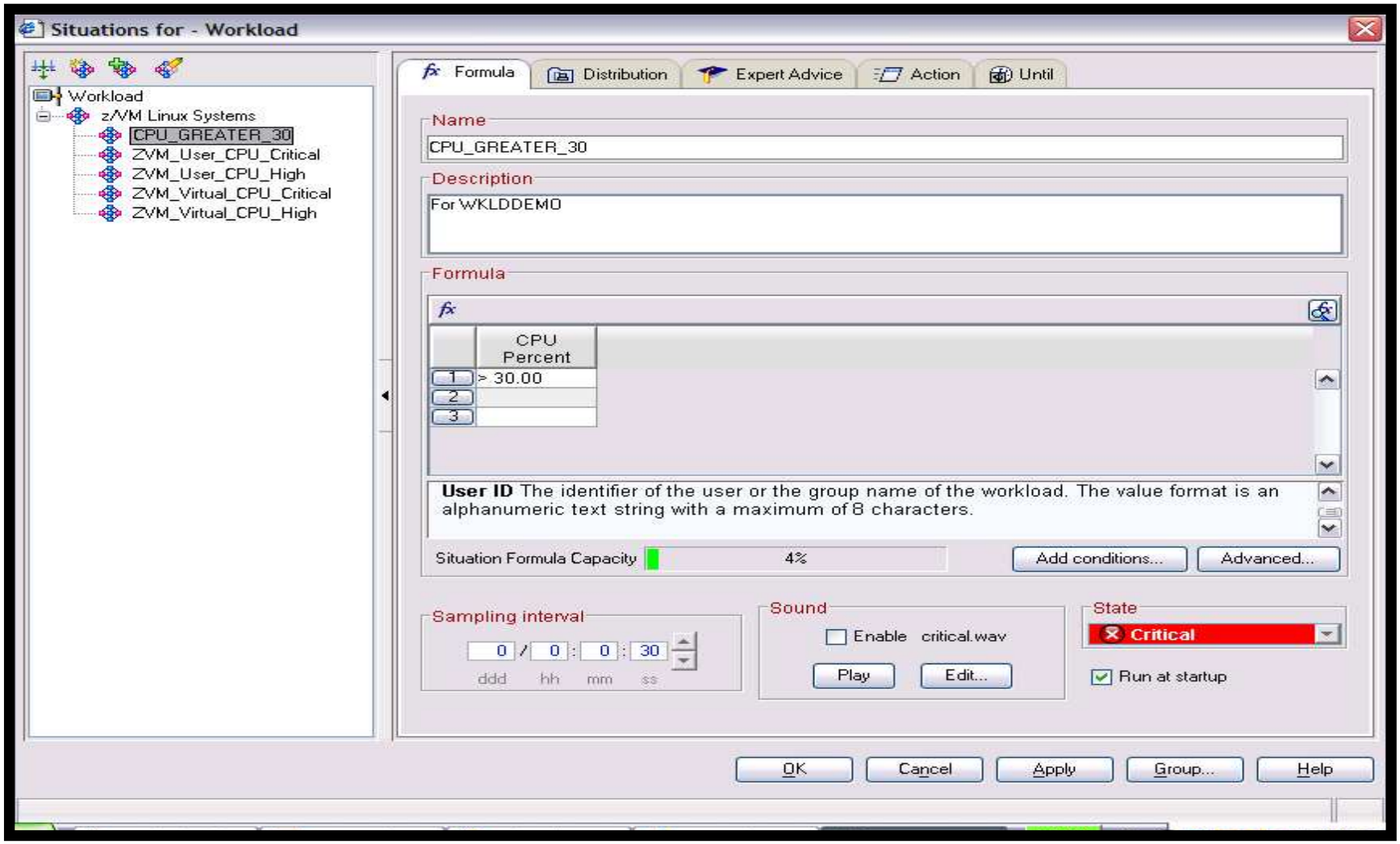


Scenario 11: How Do You Do That?

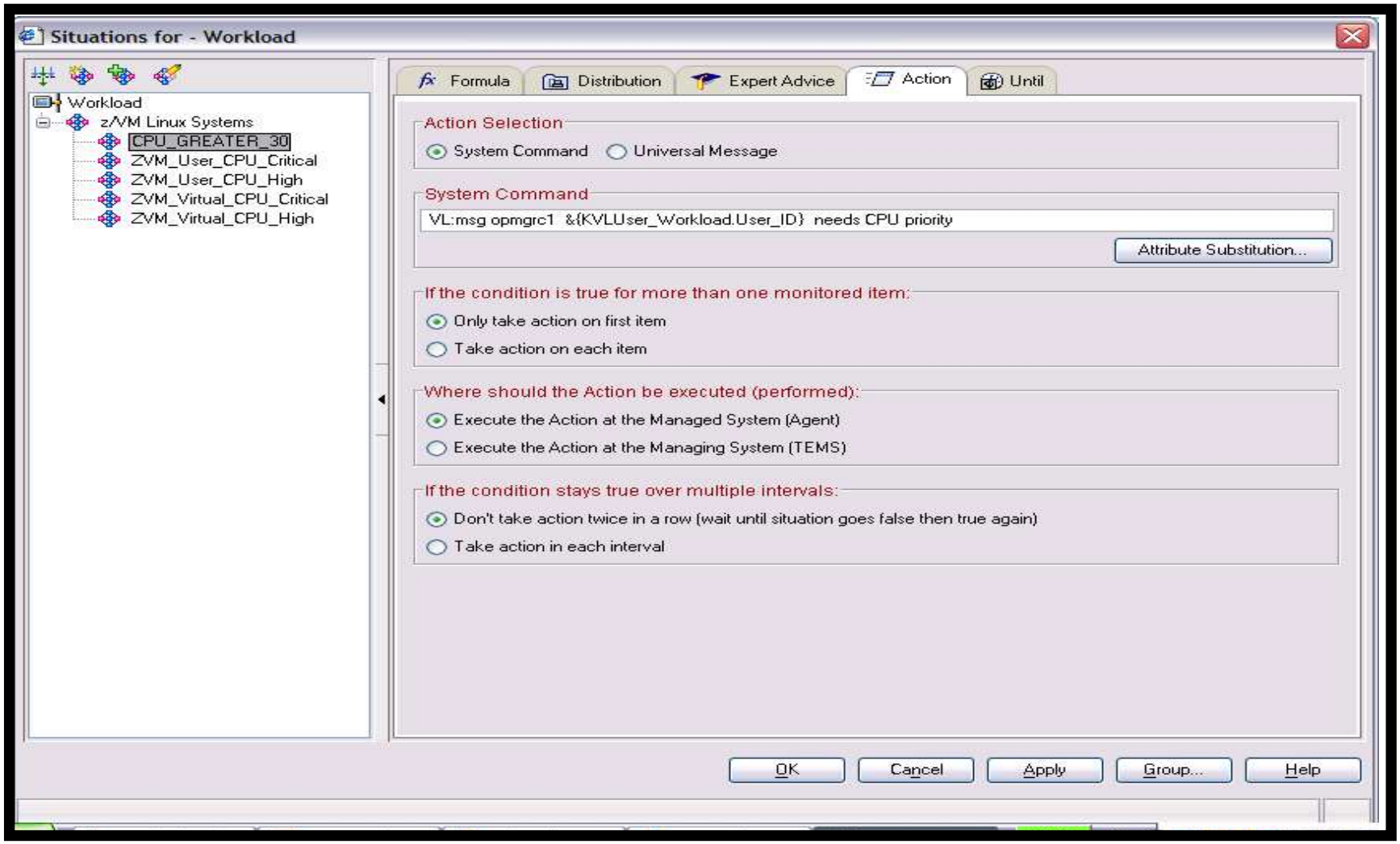
Rules in Operations Manager:

```
*
* 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

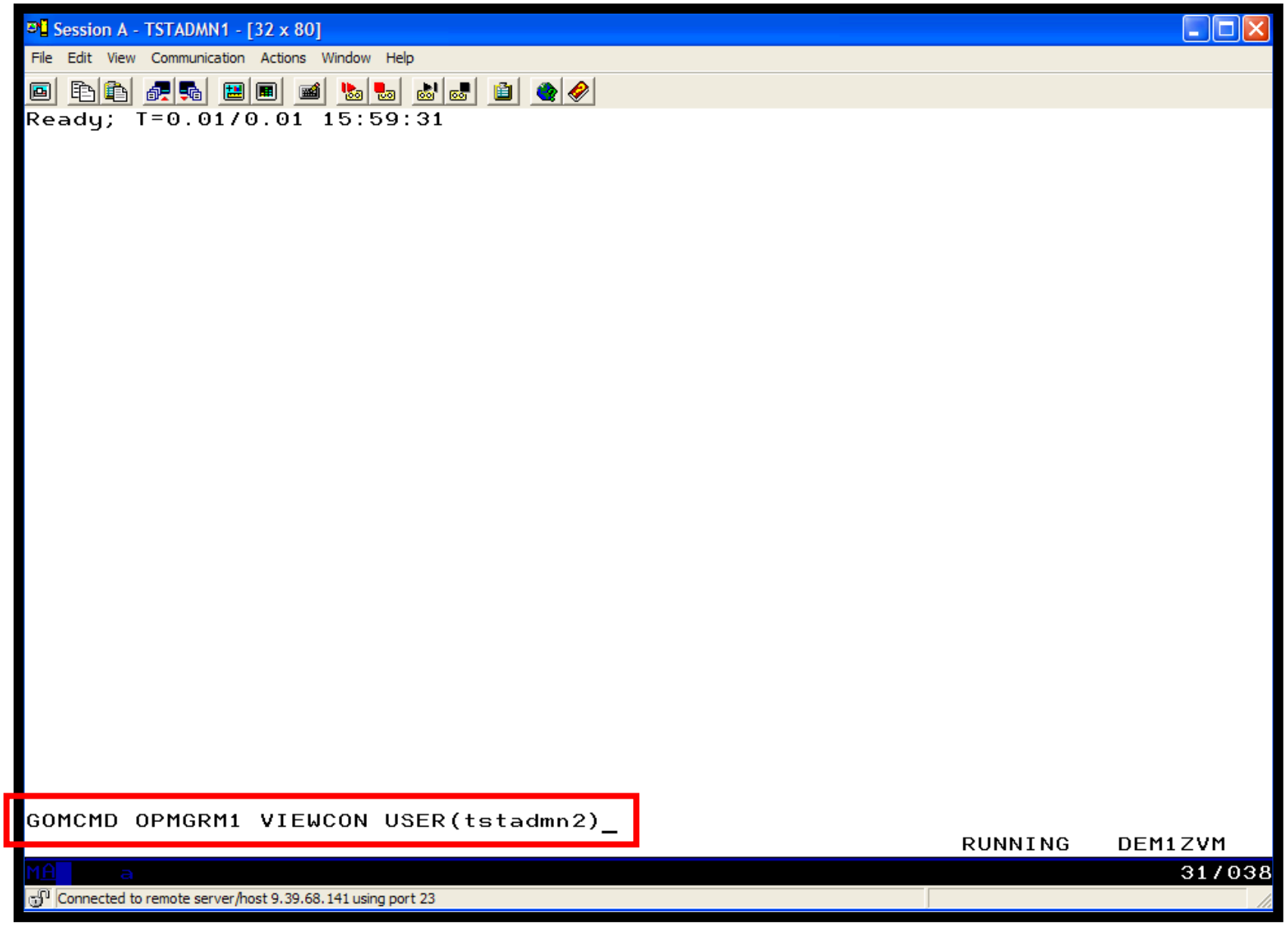


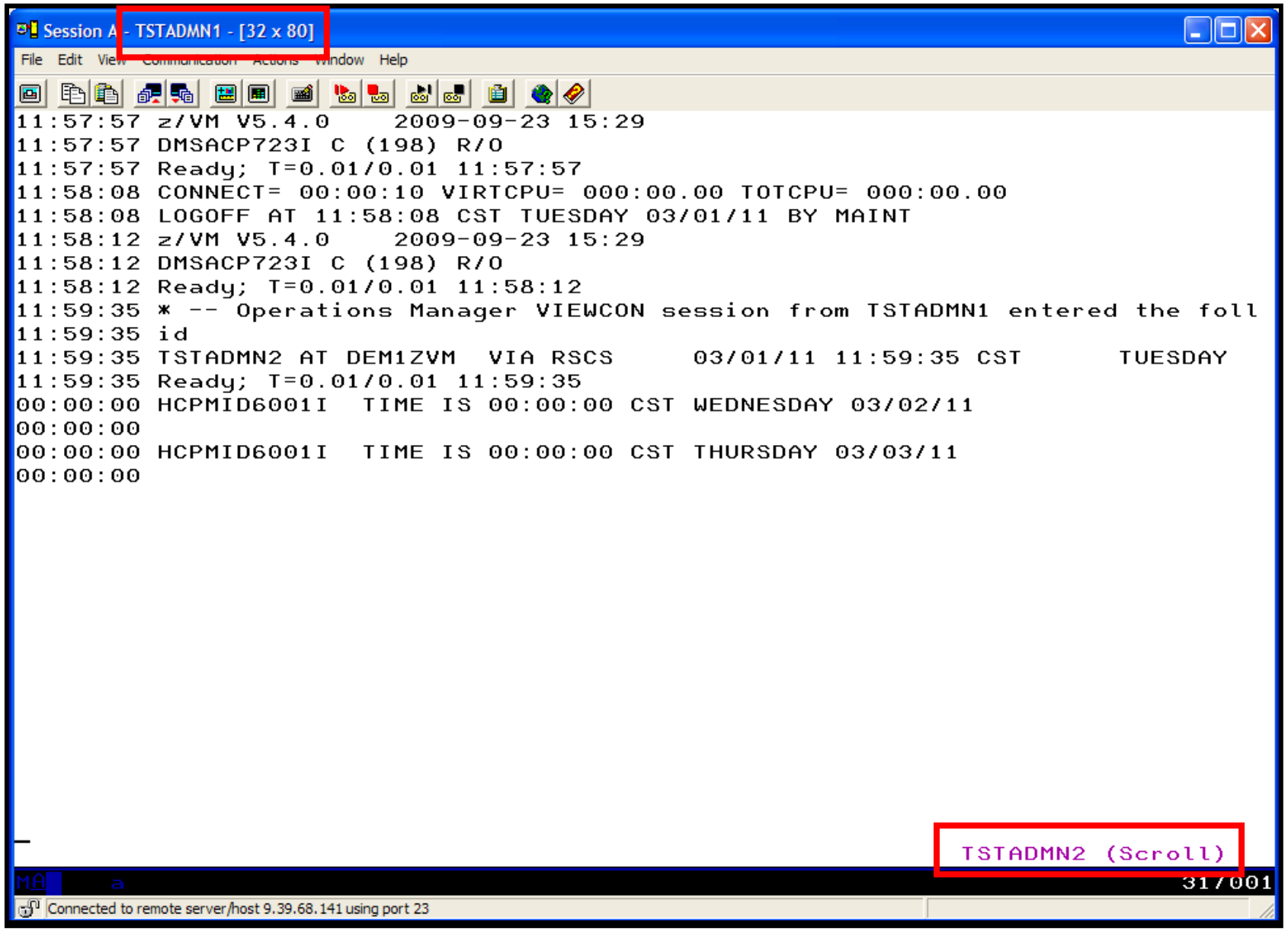
Scenario 11: Detailed Steps
OMEGAMON Configuration

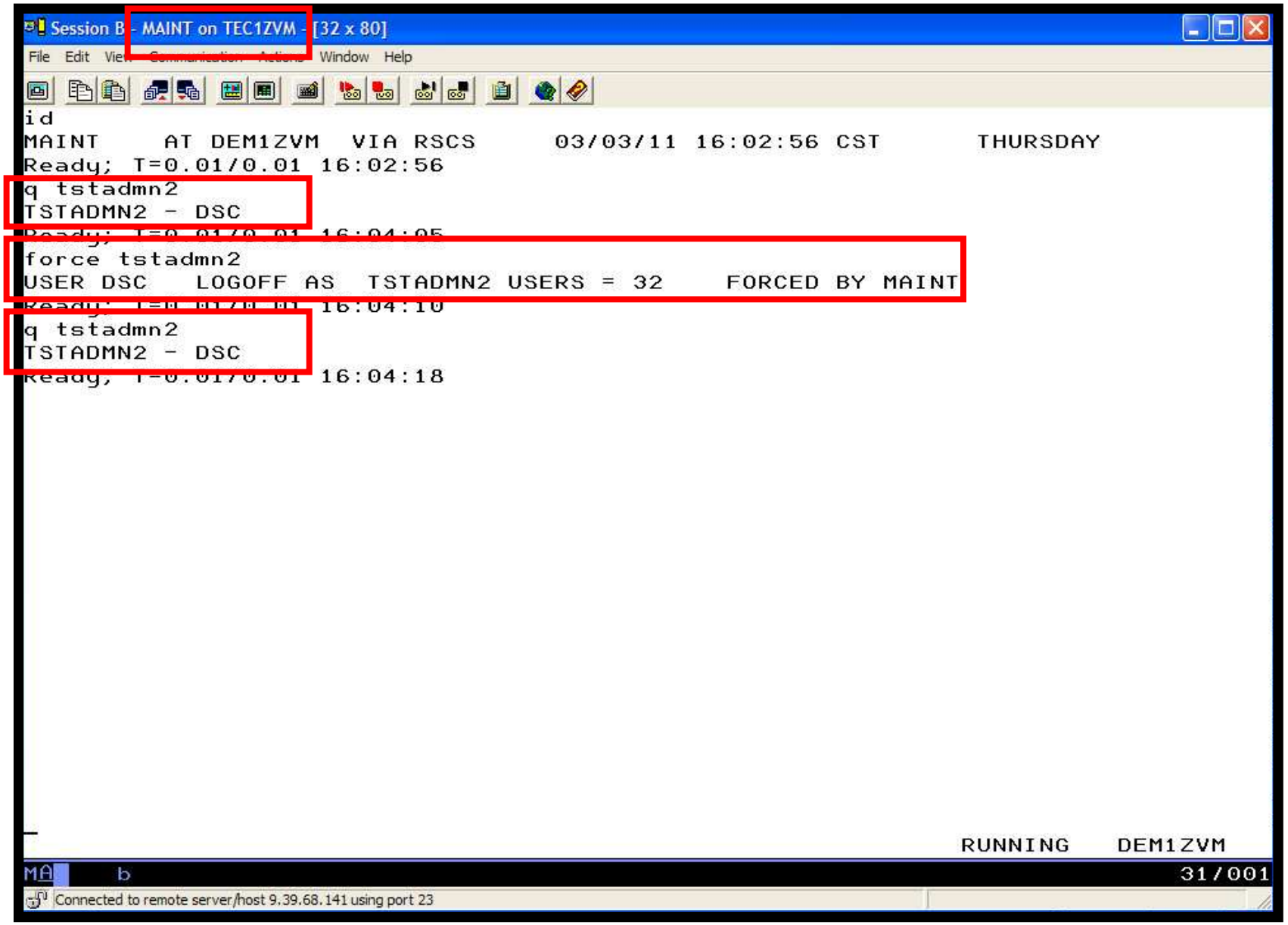


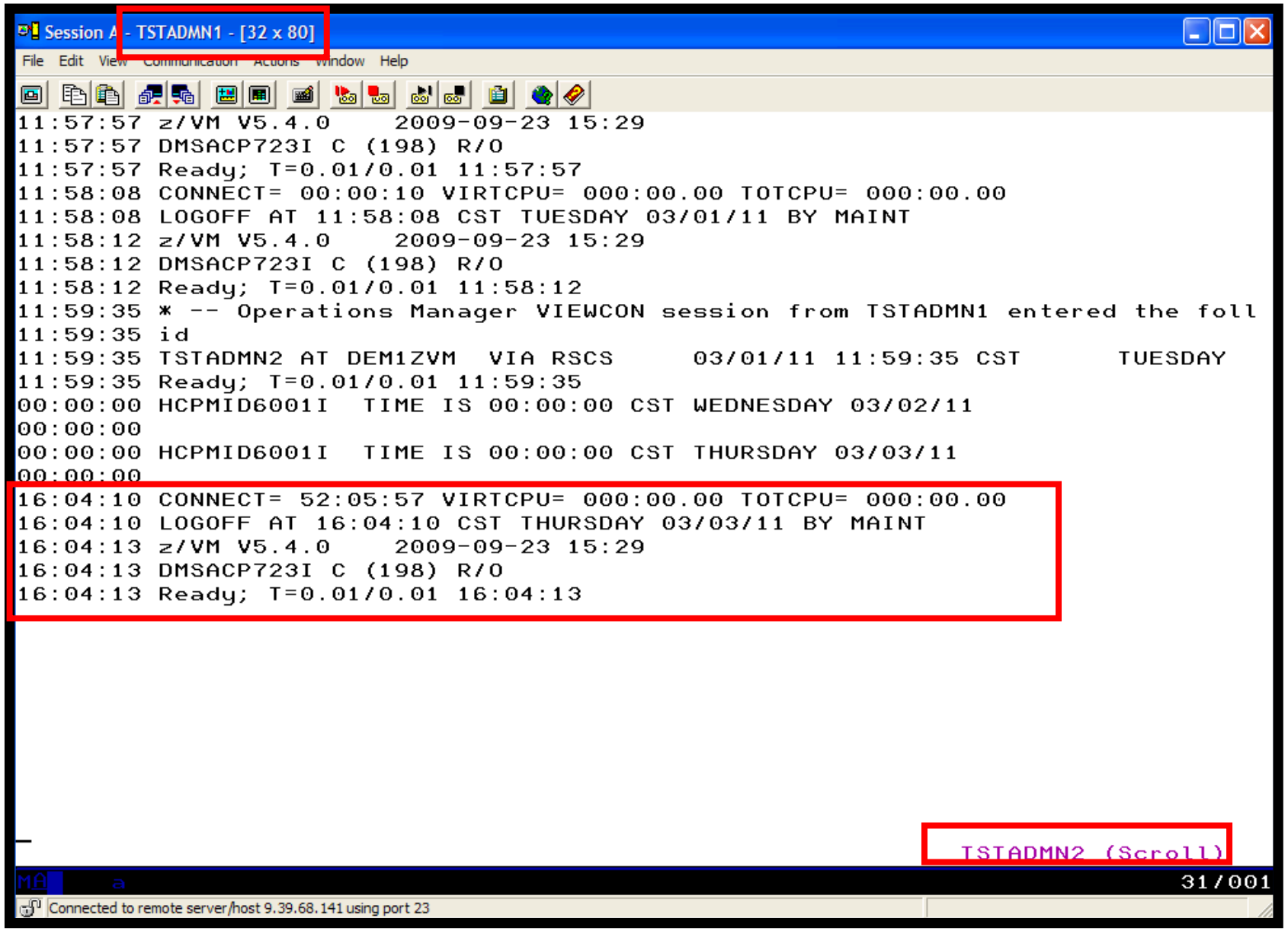
Scenario 12:
Monitor Service Machines for LOGOFF Status – and AUTOLOG them

- **Monitor specific service machines to make sure they stay logged on**
  - Demo will monitor TSTADMN2 user ID
  - 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 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)**
- **Usage exceeds the specified limit**
  - For demo purposes, we'll dynamically resume (re-activate) an existing page space monitor that requires the spool to only be 10% full
- **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:**

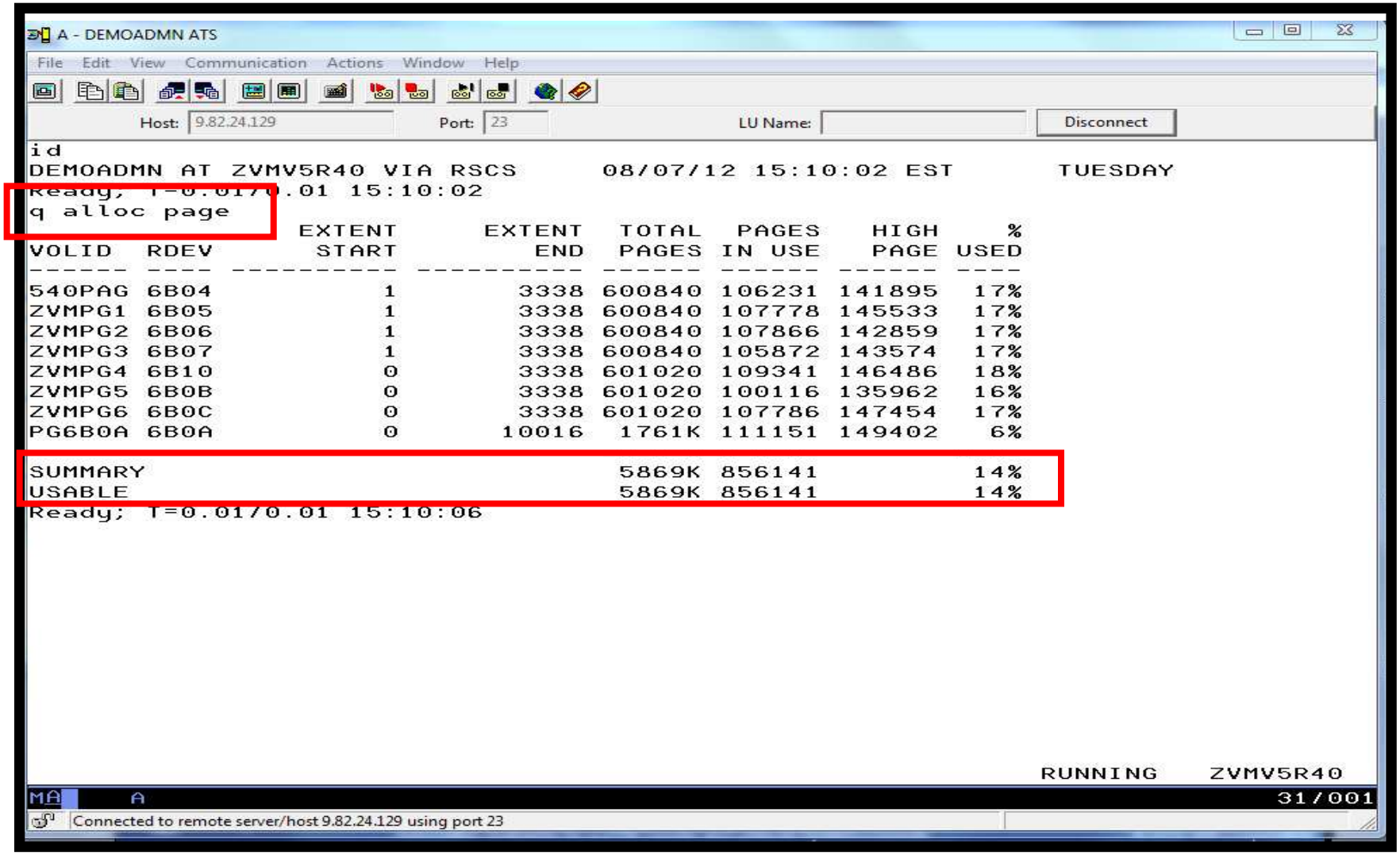
```
gomcmd opmgrm1 resume page(pgfull)
```

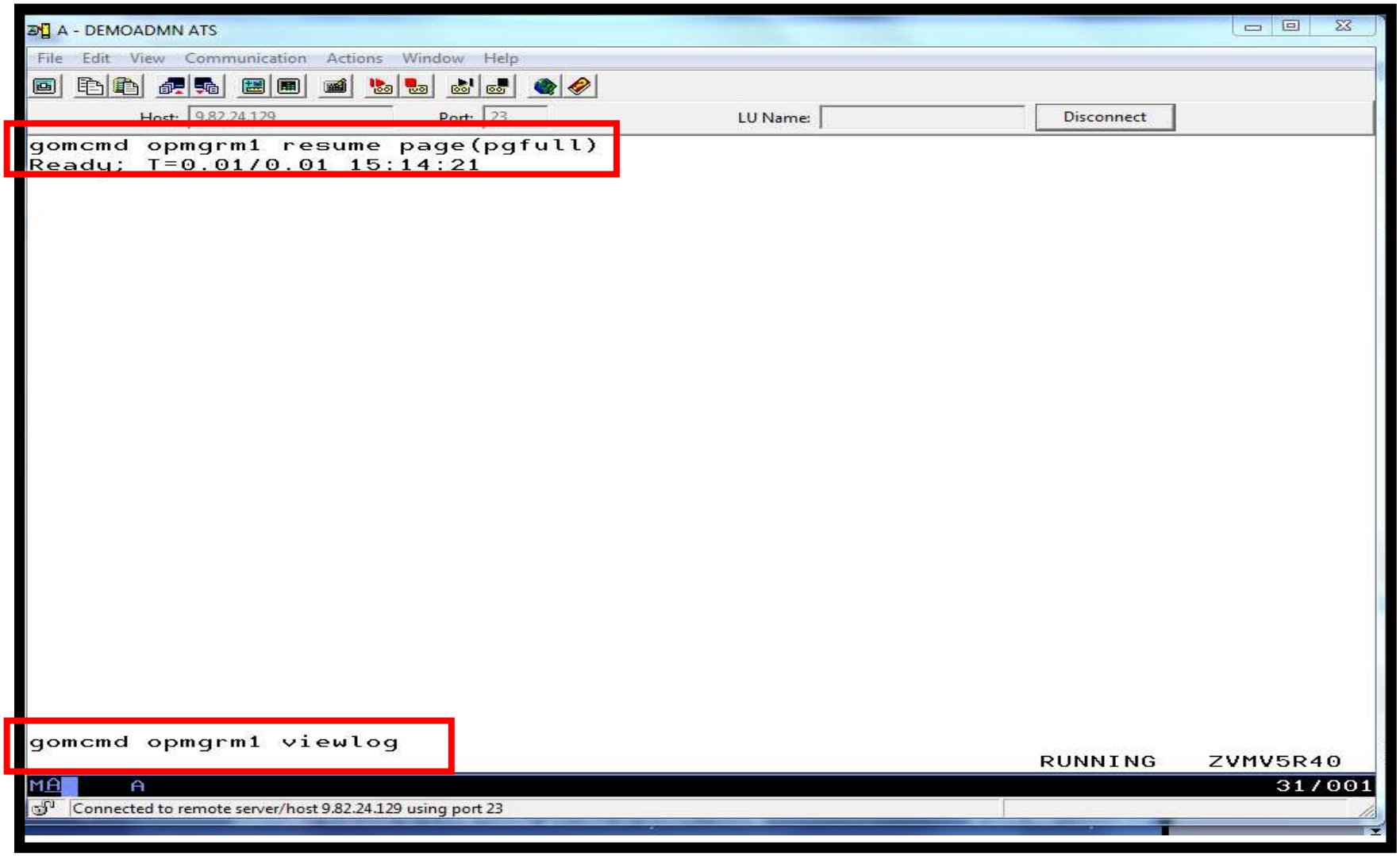
- **Check the Operations Manager log to see the spool monitor triggered:**

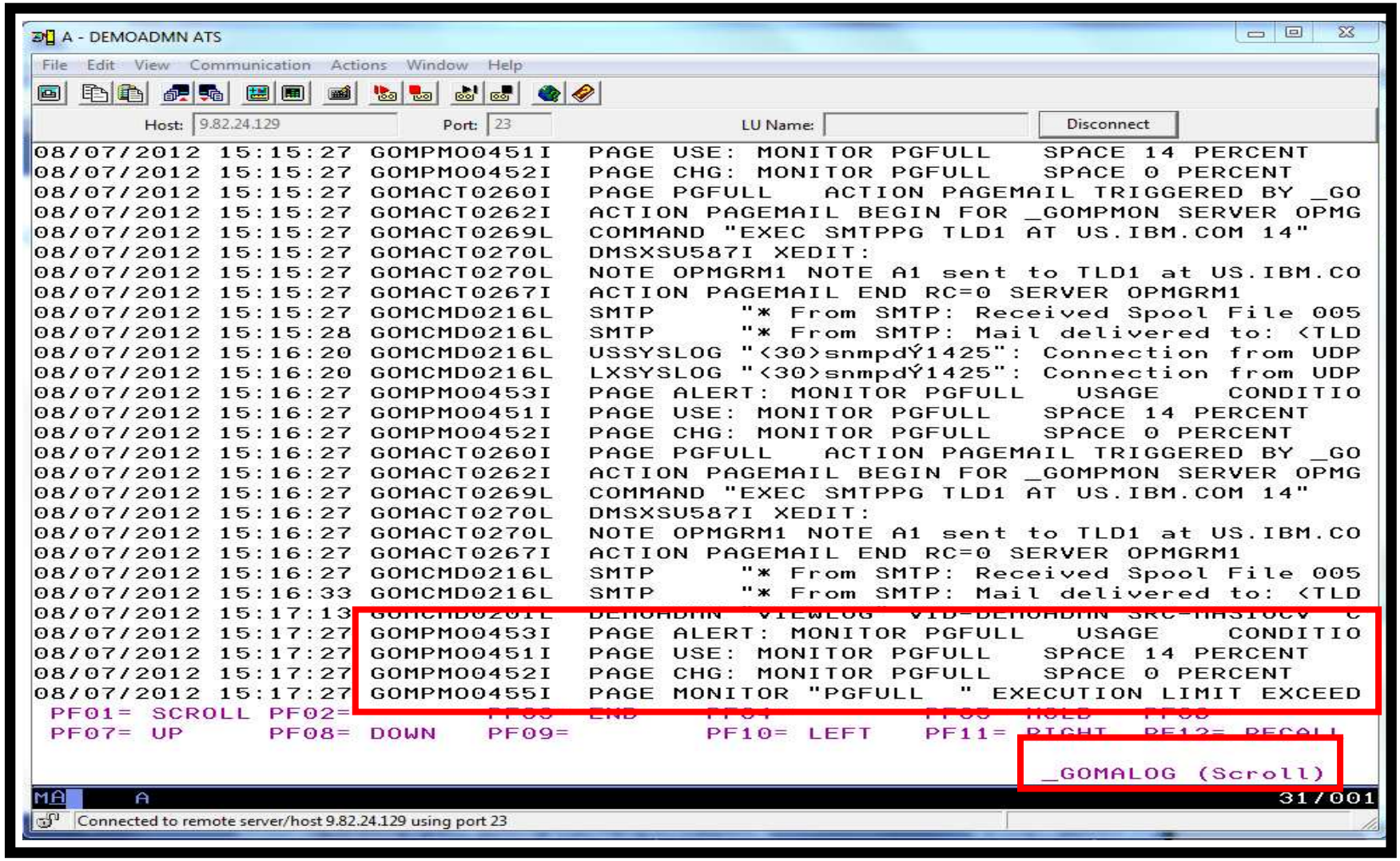
```
gomcmd opmgrm1 viewlog
```

- **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),+
  PARM(SPOOL)
*
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)
```

IBM Software

Backup and Recovery Scenarios
Including Automation

Scenario 16:
Performing an Incremental Backup

- **Administrator previously performed a full backup**
- **Incremental job defined, using last full backup as its base**
- **Change a file on user's A-disk**
- **Submit incremental job for review**
- **Submit incremental job for backup processing**
- **Use Operations Manager to monitor backup servers**

Scenario 16: Detailed Steps

- **From a z/VM user ID, change a file**

```
xedit b b a
```

- **From an authorized z/VM user ID, submit a backup job for review**

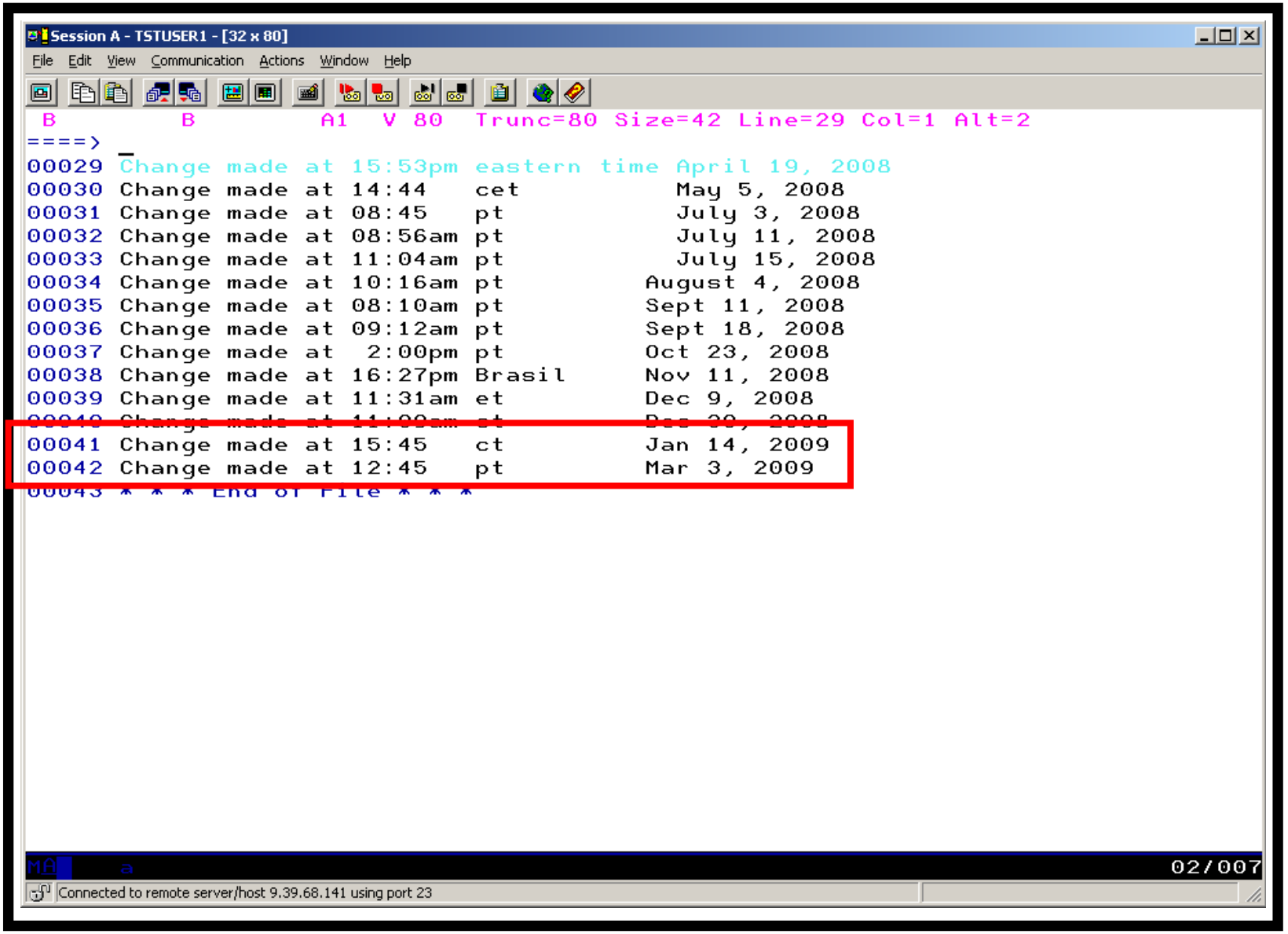
```
smsg bkrbkup review increm01
```

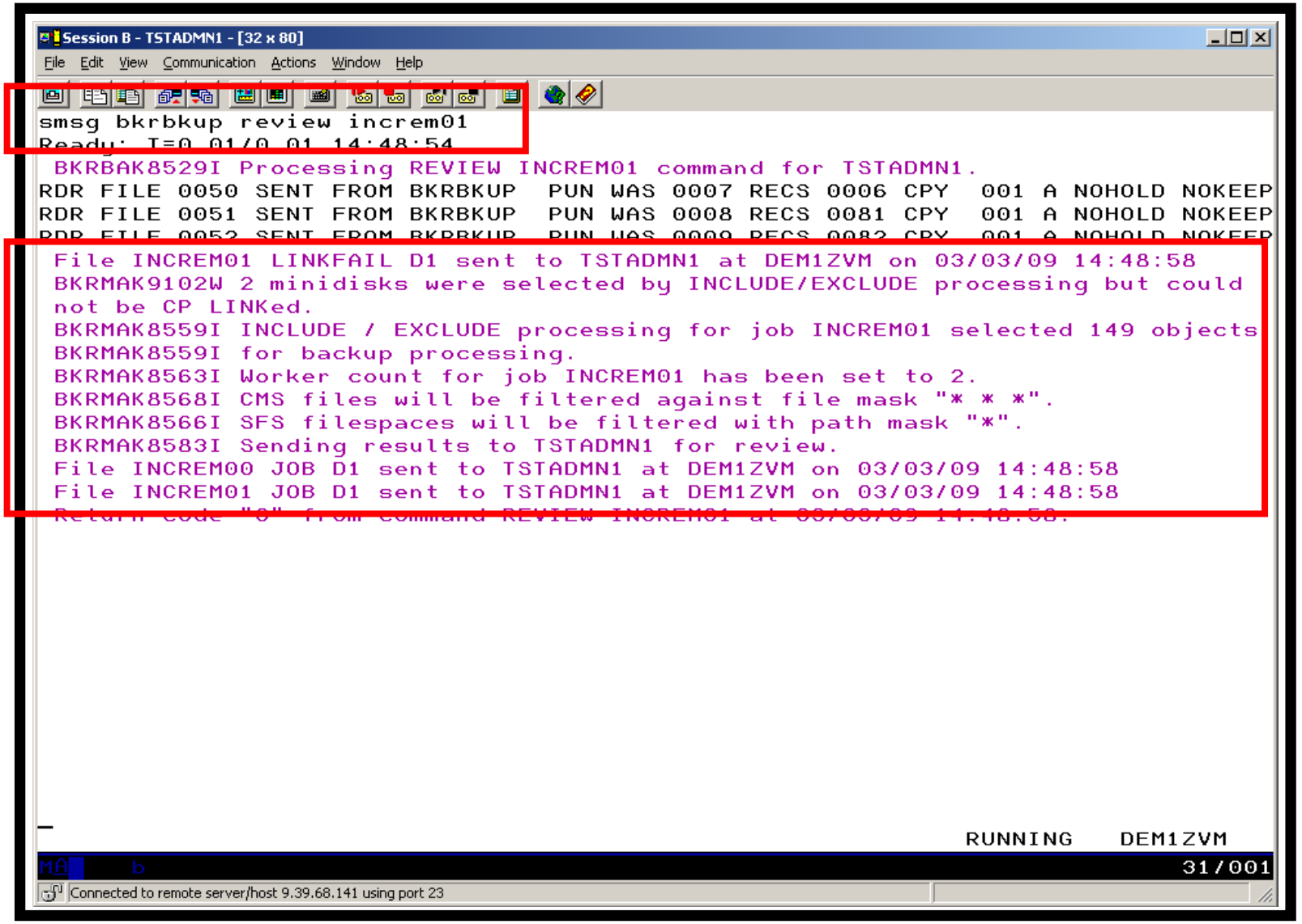
- **Review the resulting files in the reader (LINKFAIL and JOB files)**
- **From an authorized z/VM user ID, submit a backup job for backup processing**

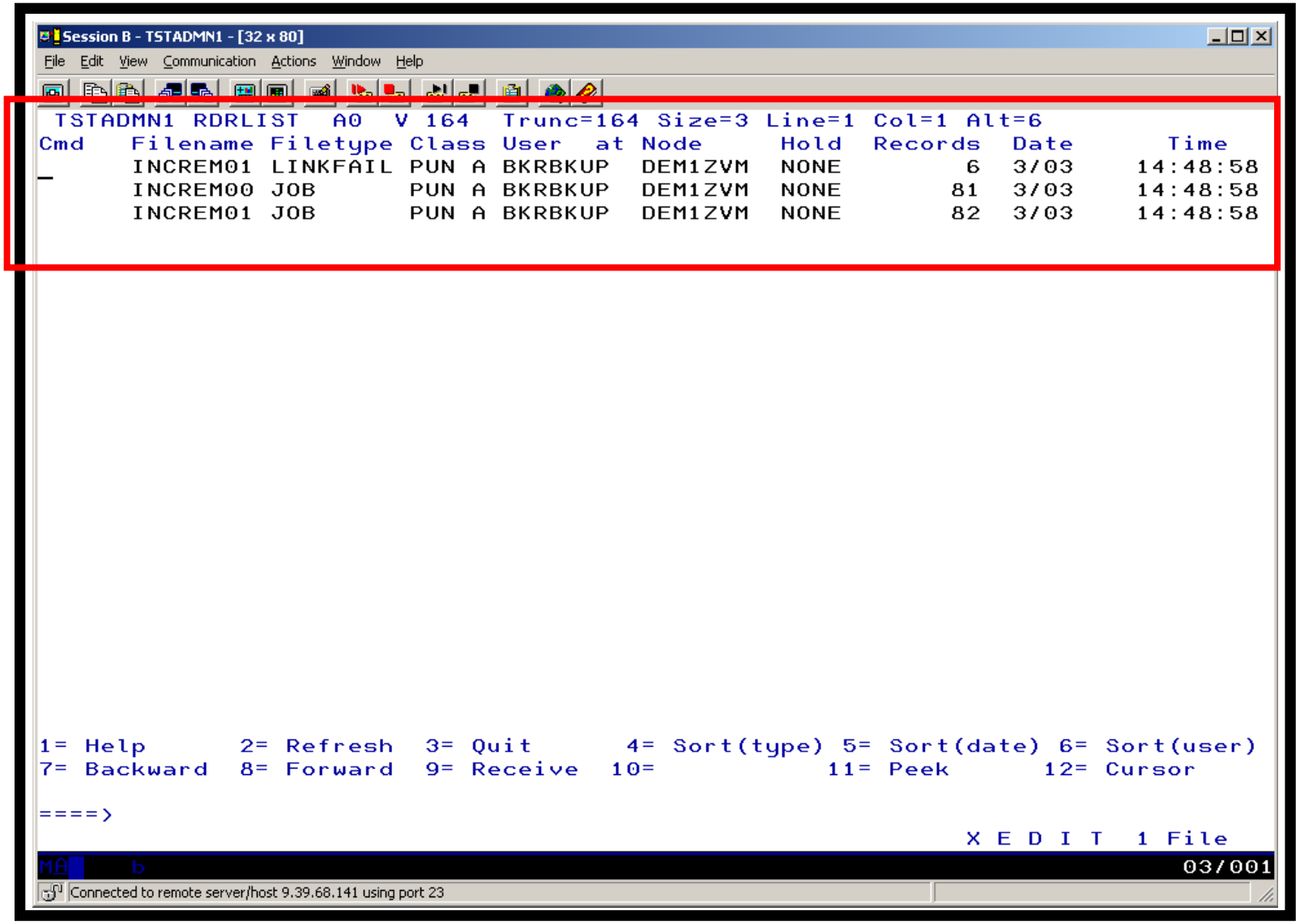
```
smsg bkrbkup submit increm01
```

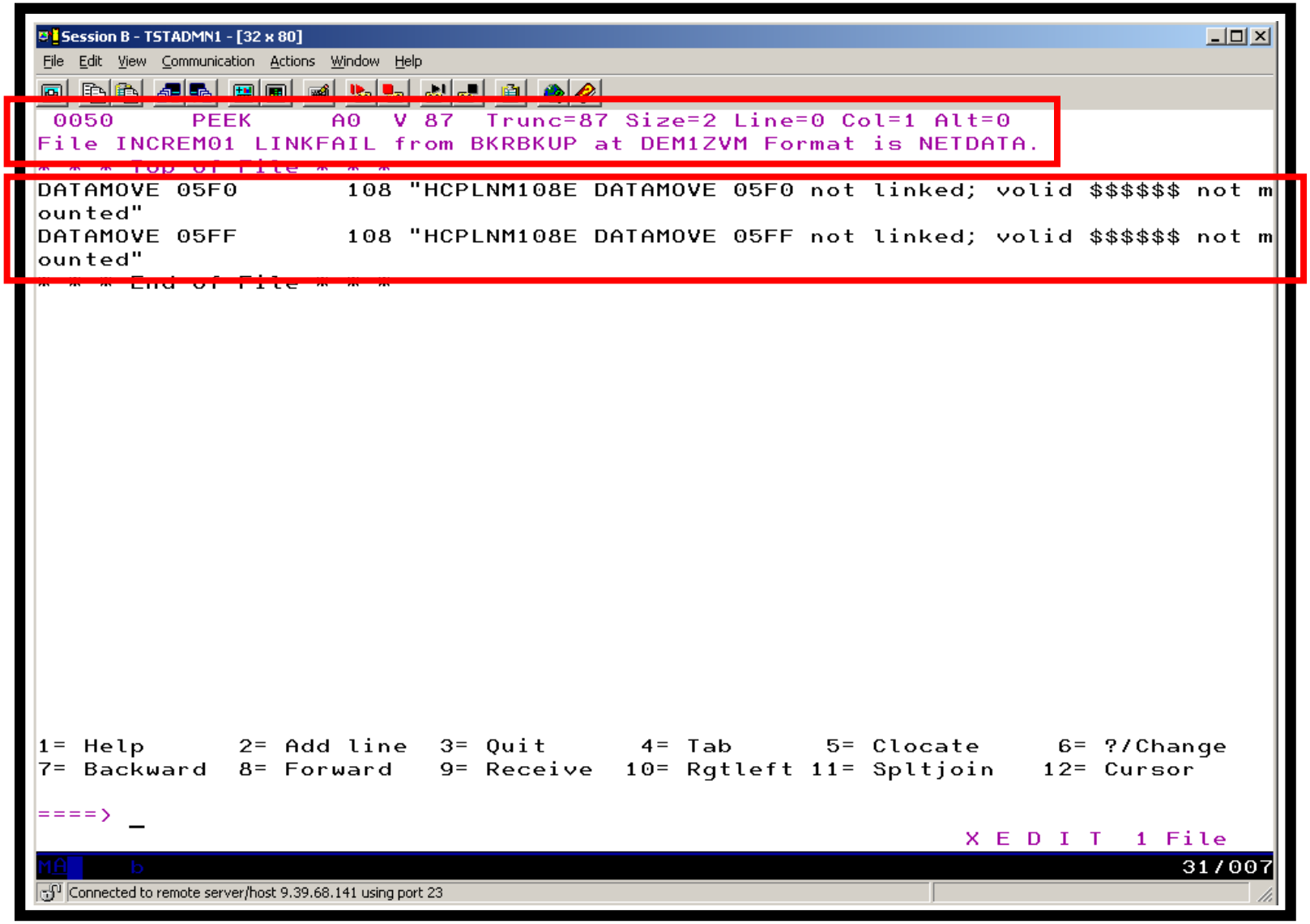
- **View the console of the backup servers to see the processing**

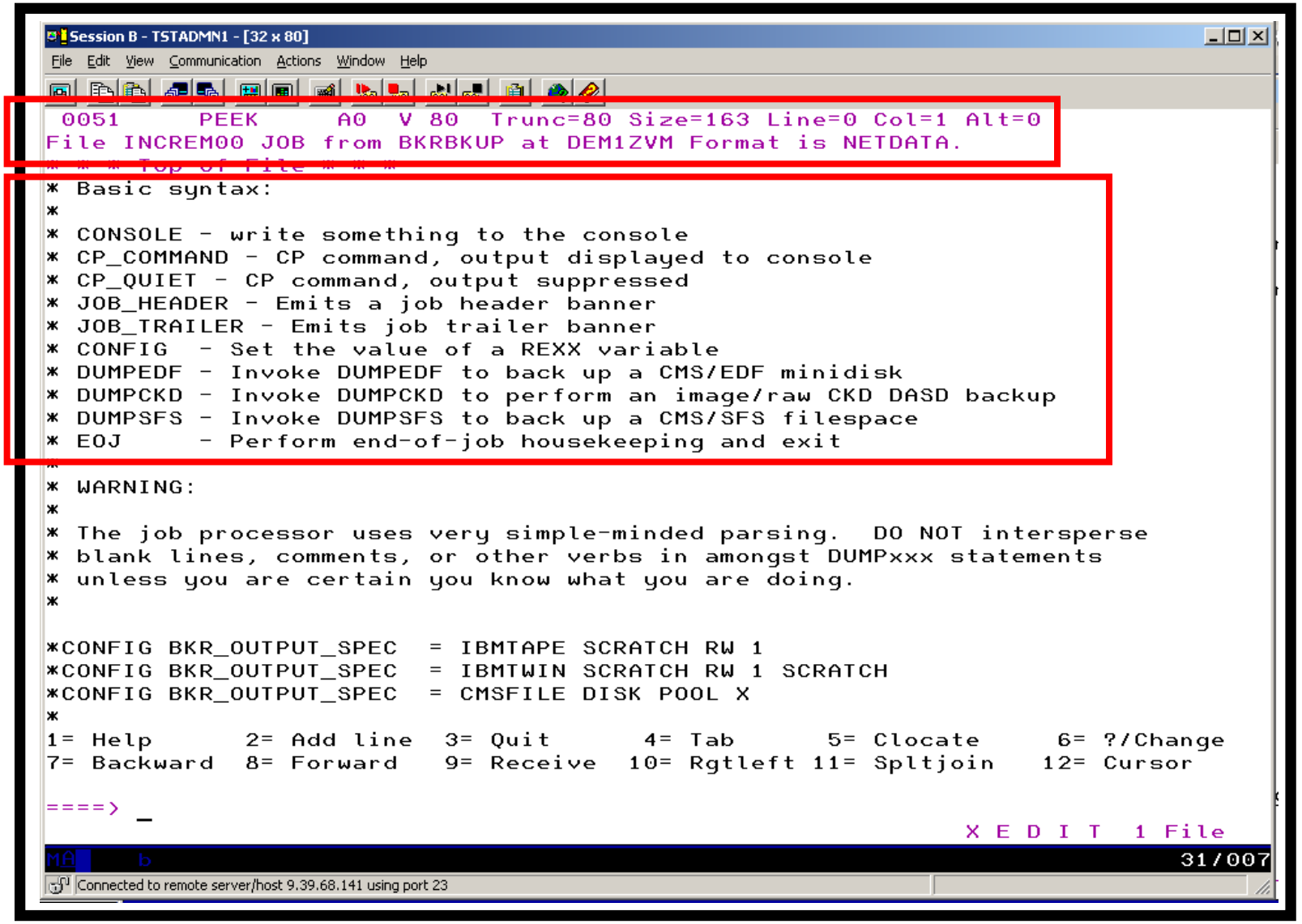
```
gomcmd opmgrm1 viewcon user(backup)
```

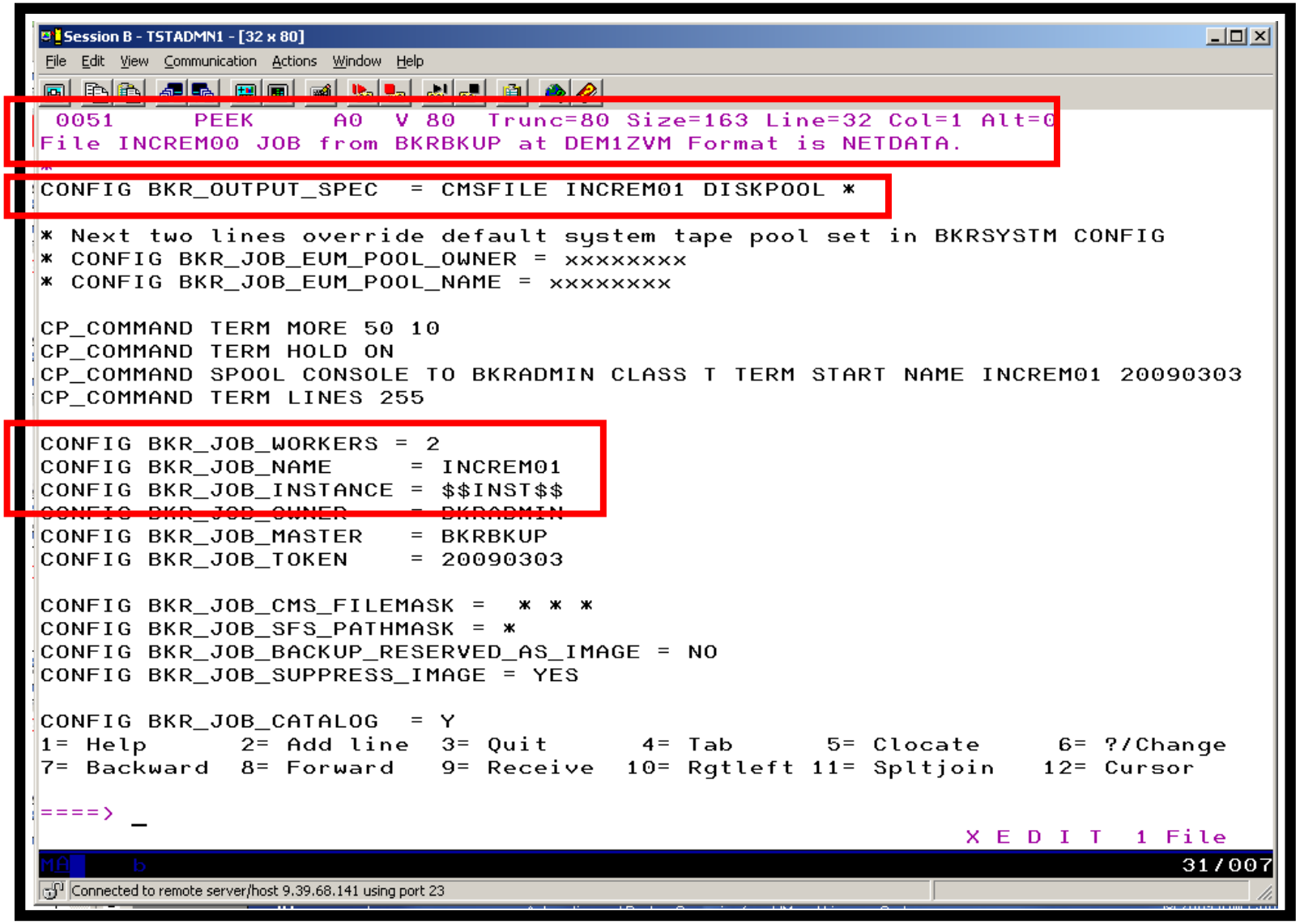


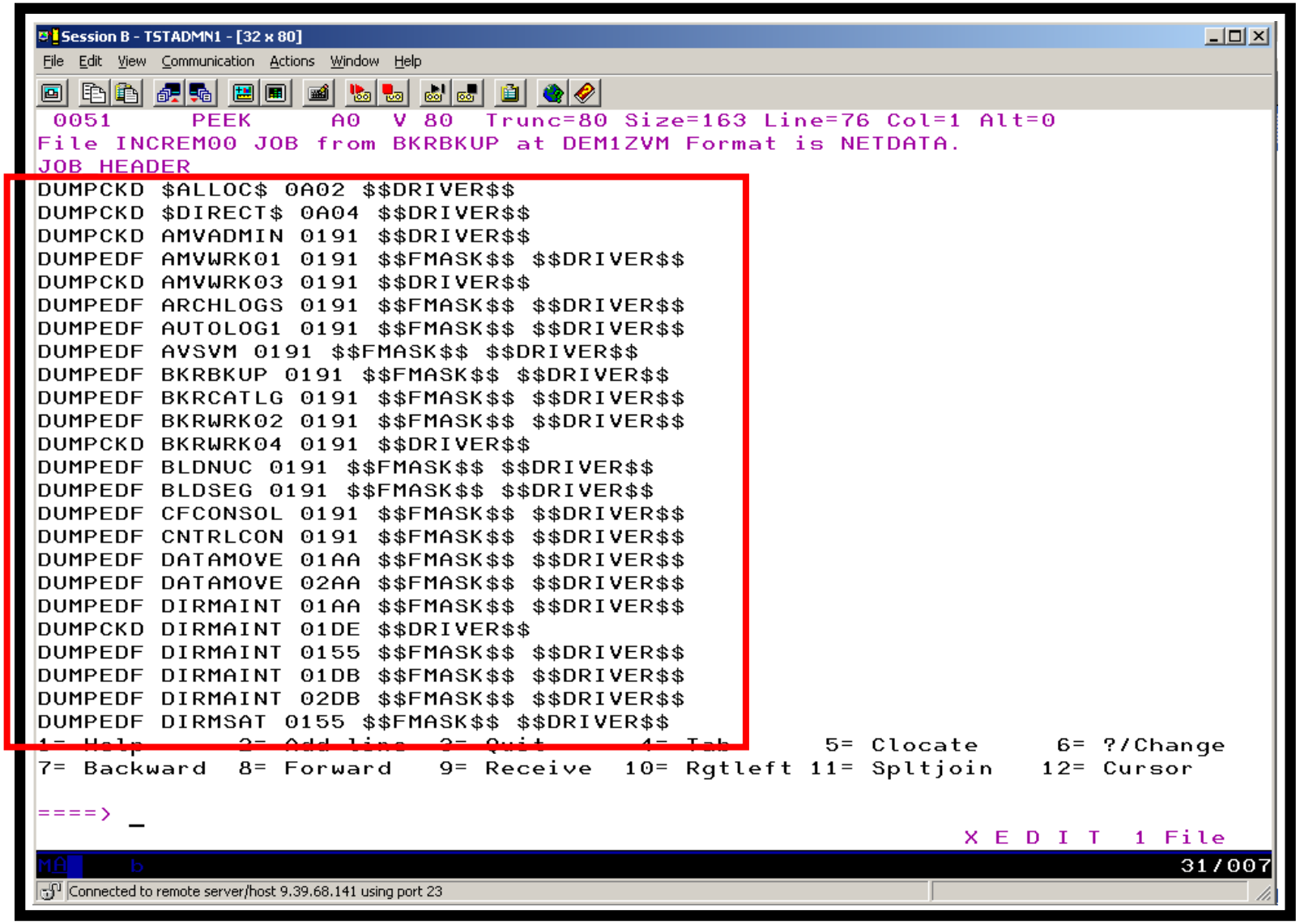


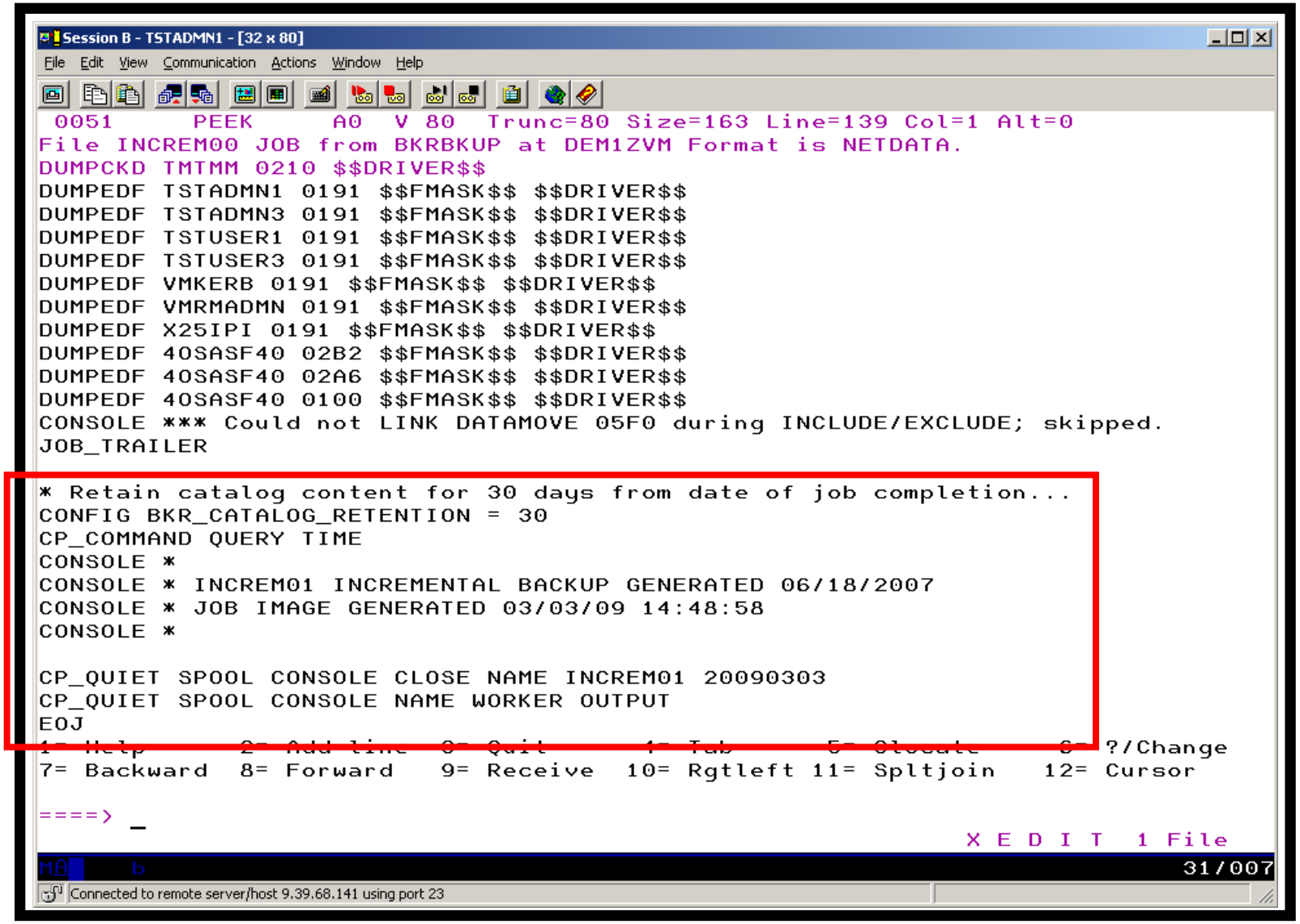


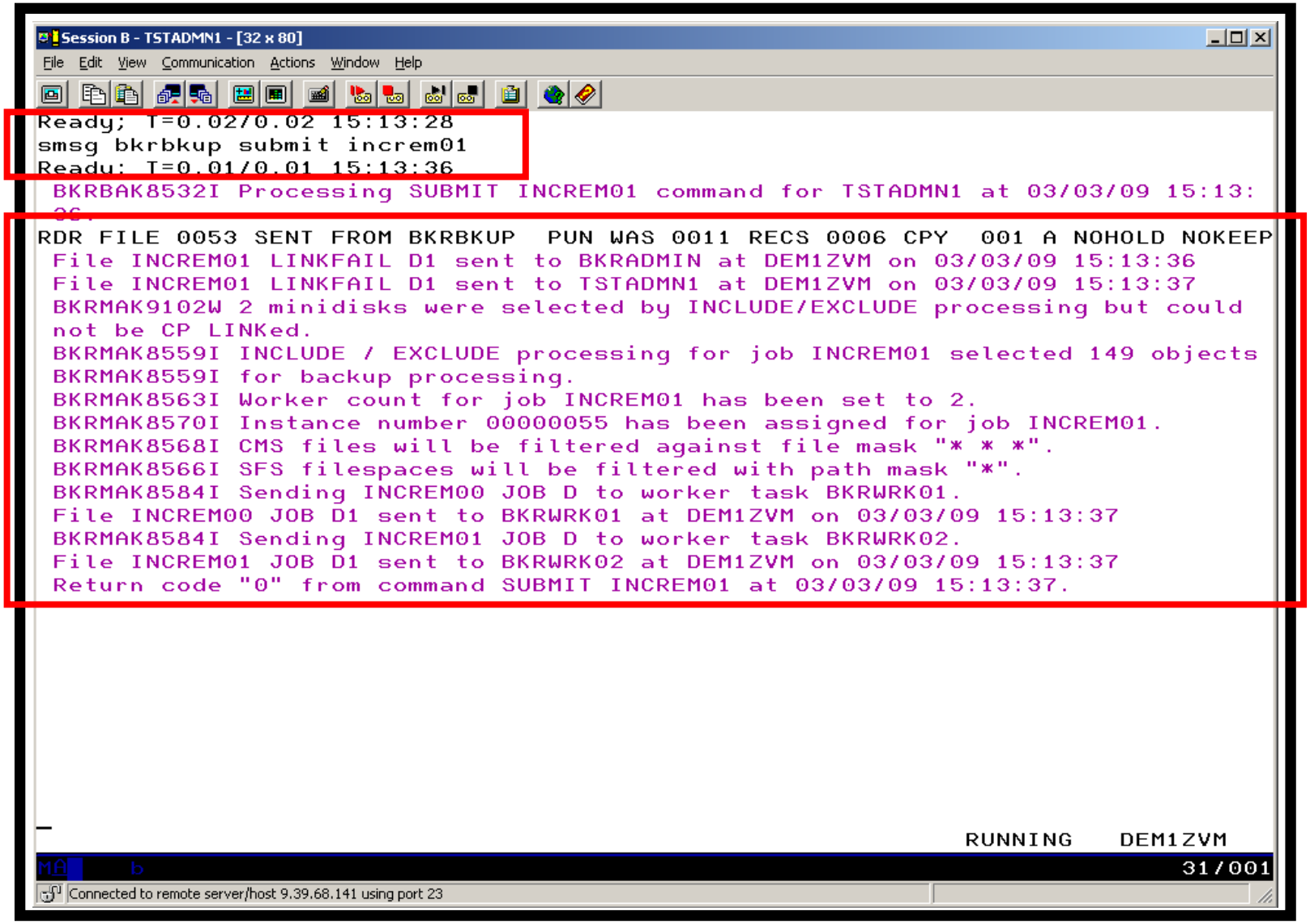


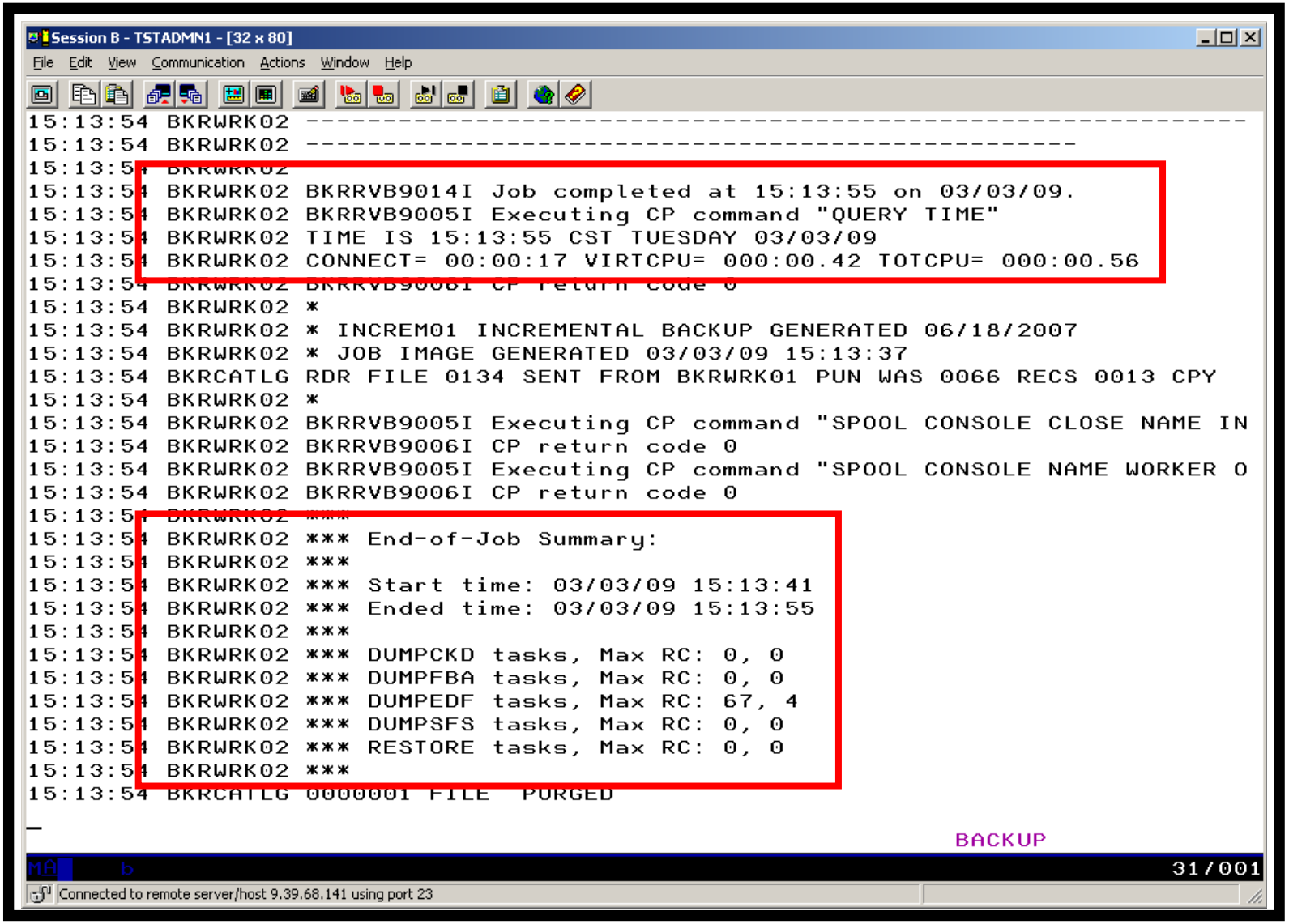












Scenario 17: Restoring Files from Backup

- **Full and incremental backups performed previously**
- **User accidentally erases or corrupts a file**
- **User restores the file from backup**
  - Full screen interface to see all files available in backup
    - Including multiple “versions” of the same file
  - Filters and sorting available to easily find the needed file
  - Request restore directly to disk or to reader
- **No administrator intervention required**

Scenario 17: Detailed Steps

- **From a z/VM user ID, view all catalog data you own**

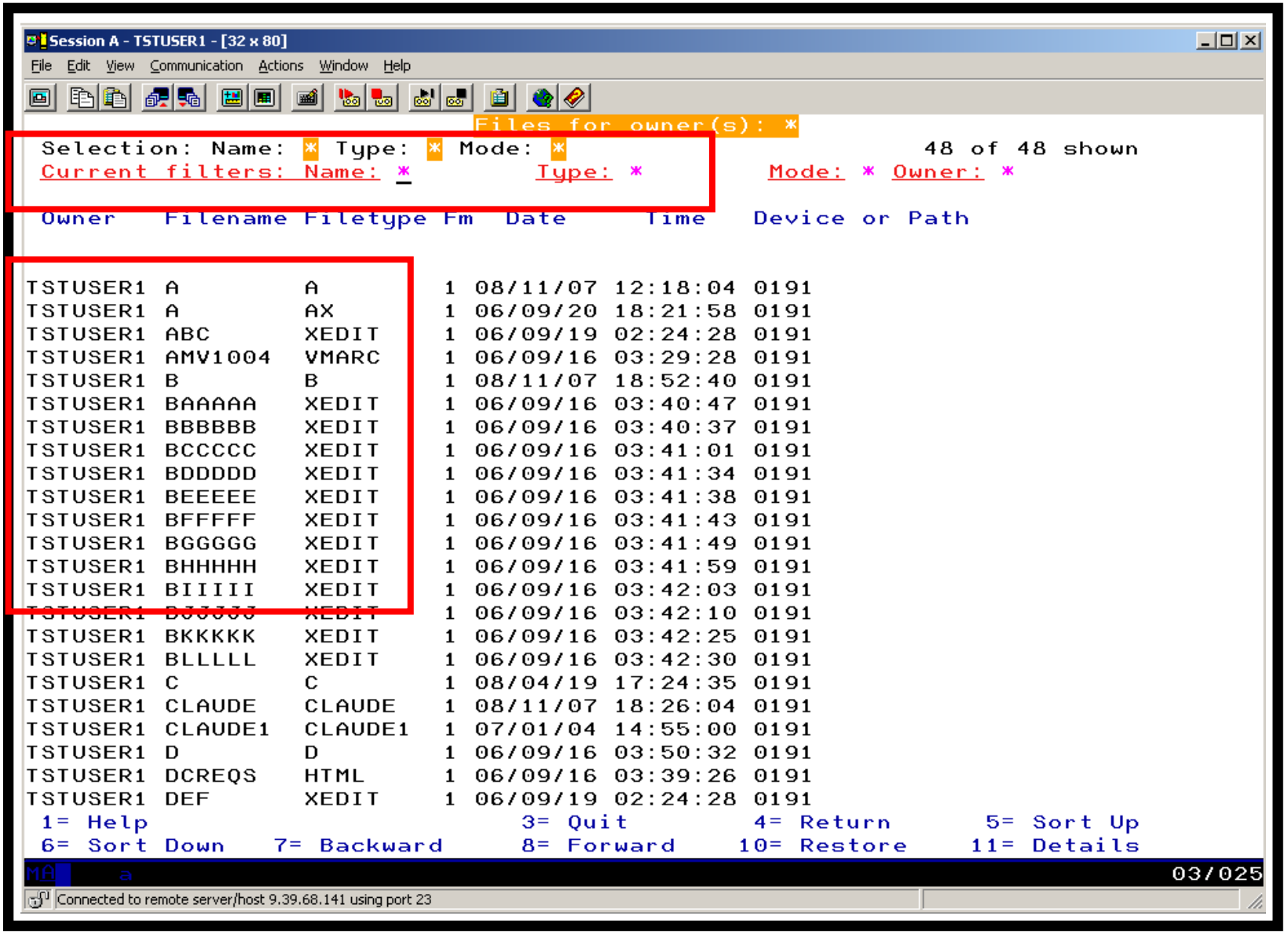
```
bkrlist
```

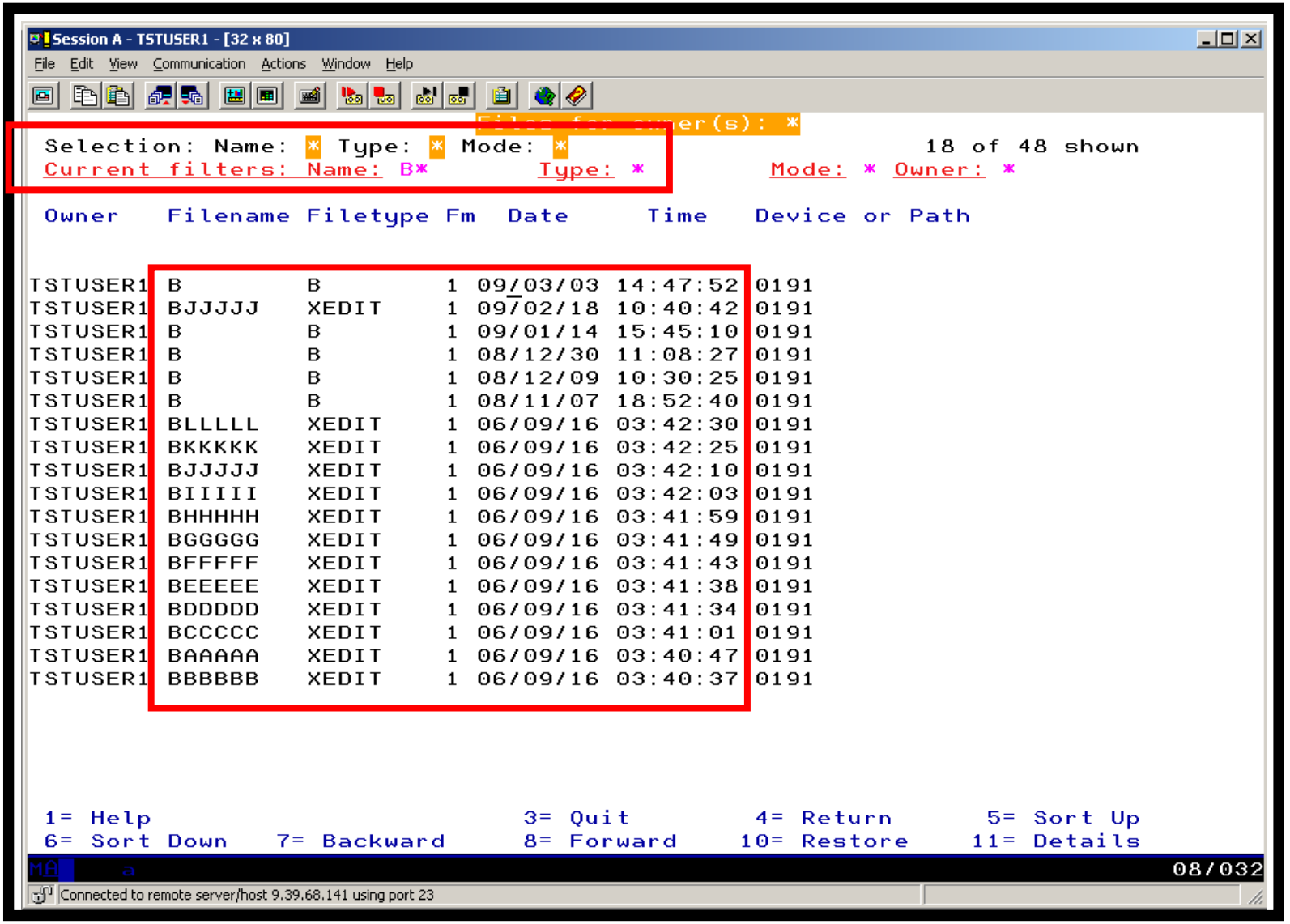
- **Use the filters to find the file you want to restore**
- **Put the cursor on the file and hit F10**
- **Specify the user ID to whom the file should be sent and hit F10**
- **Look at the reader of that user ID to see the restored file and a copy of the console during the restore processing**

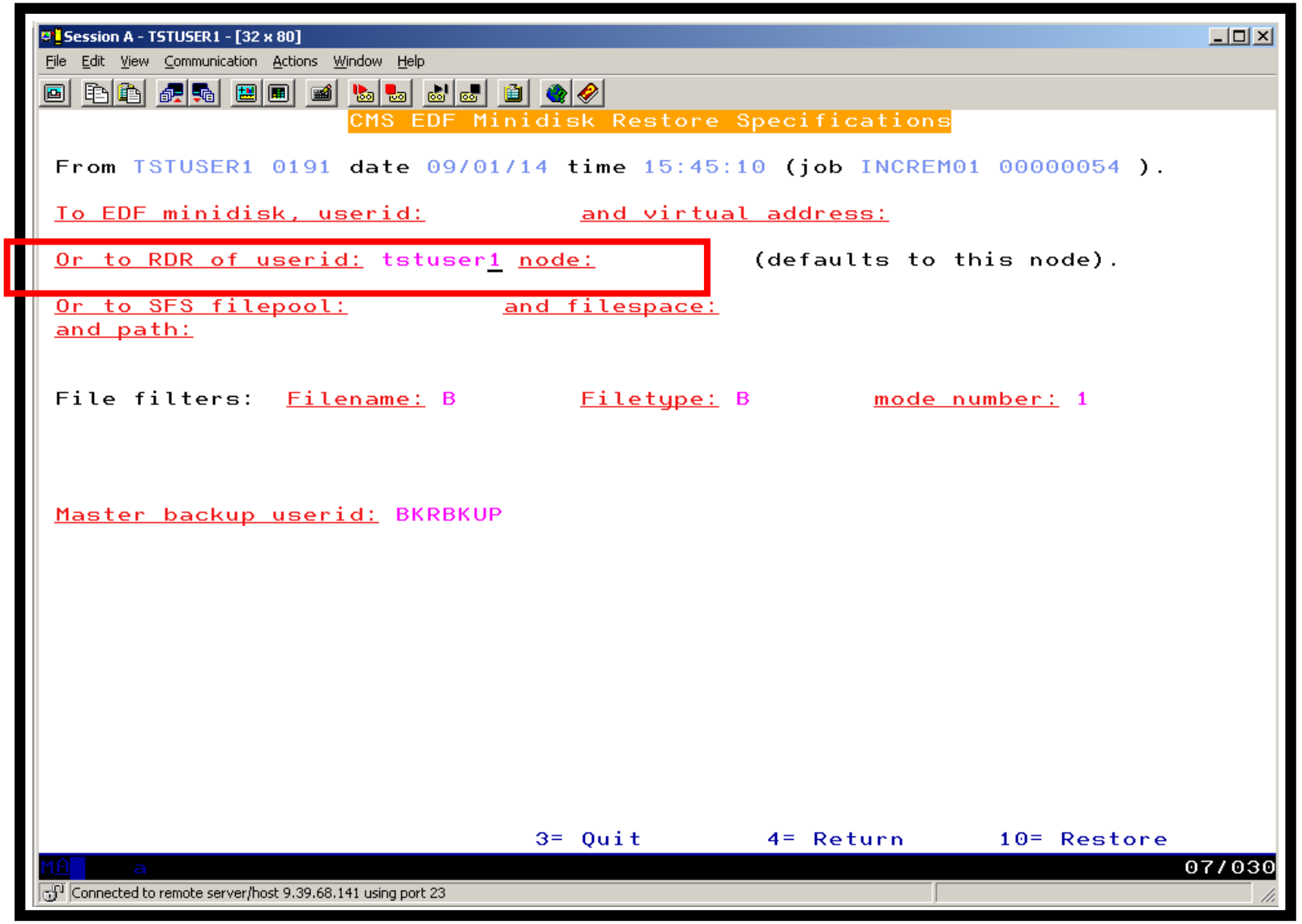
```
rdrlist
```

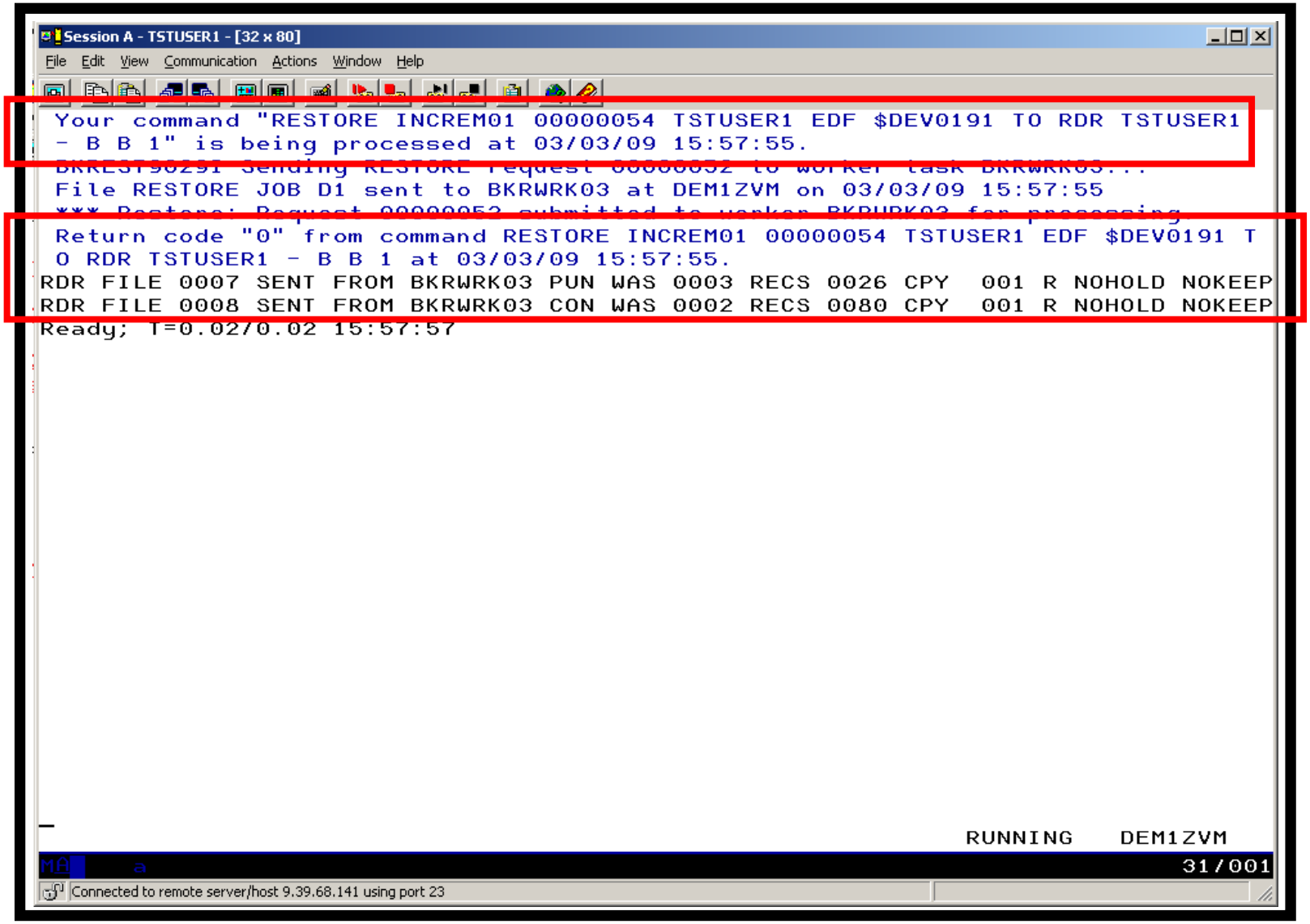
- **View the contents of the file to verify it's the correct version**

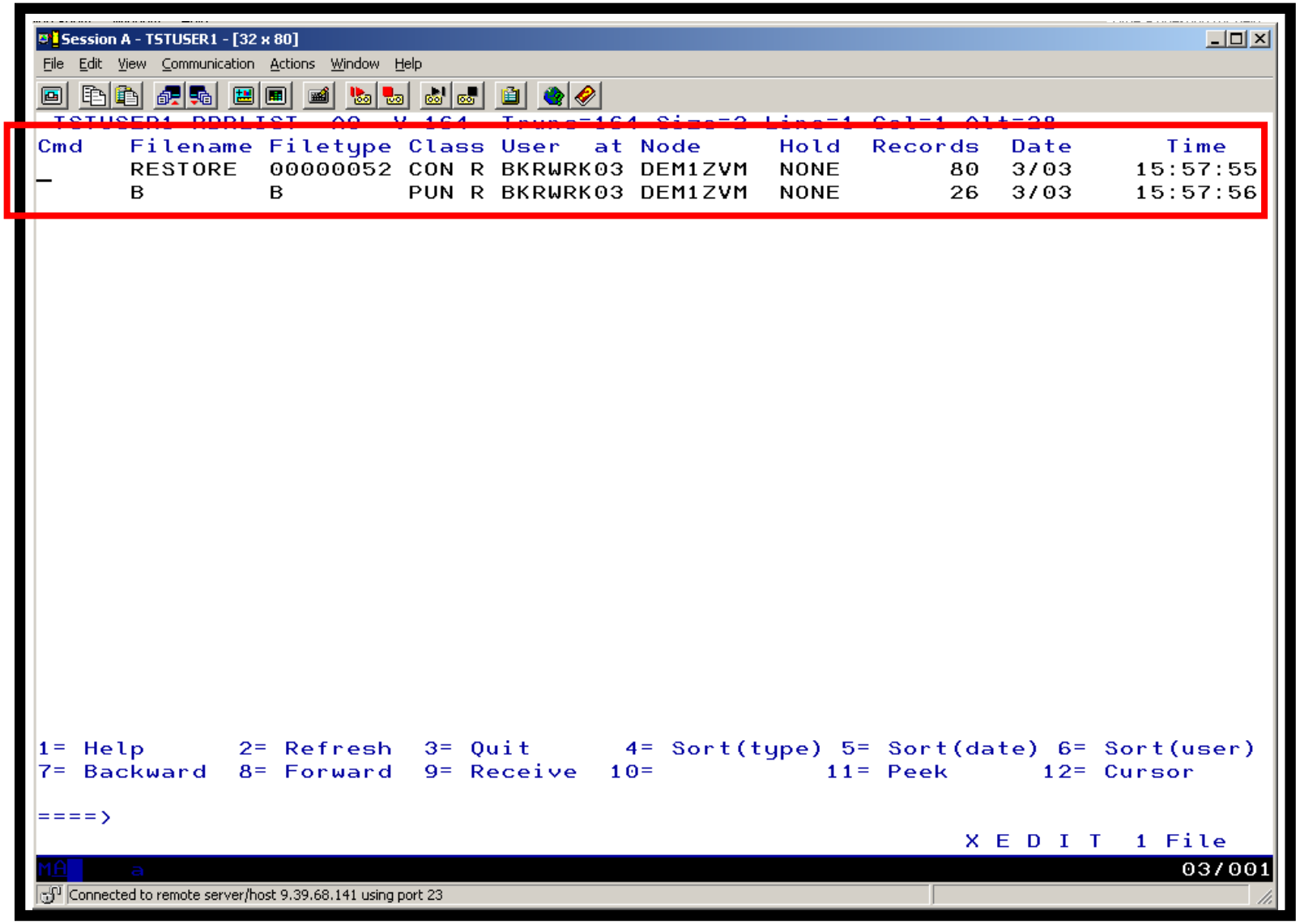
```
peek
```

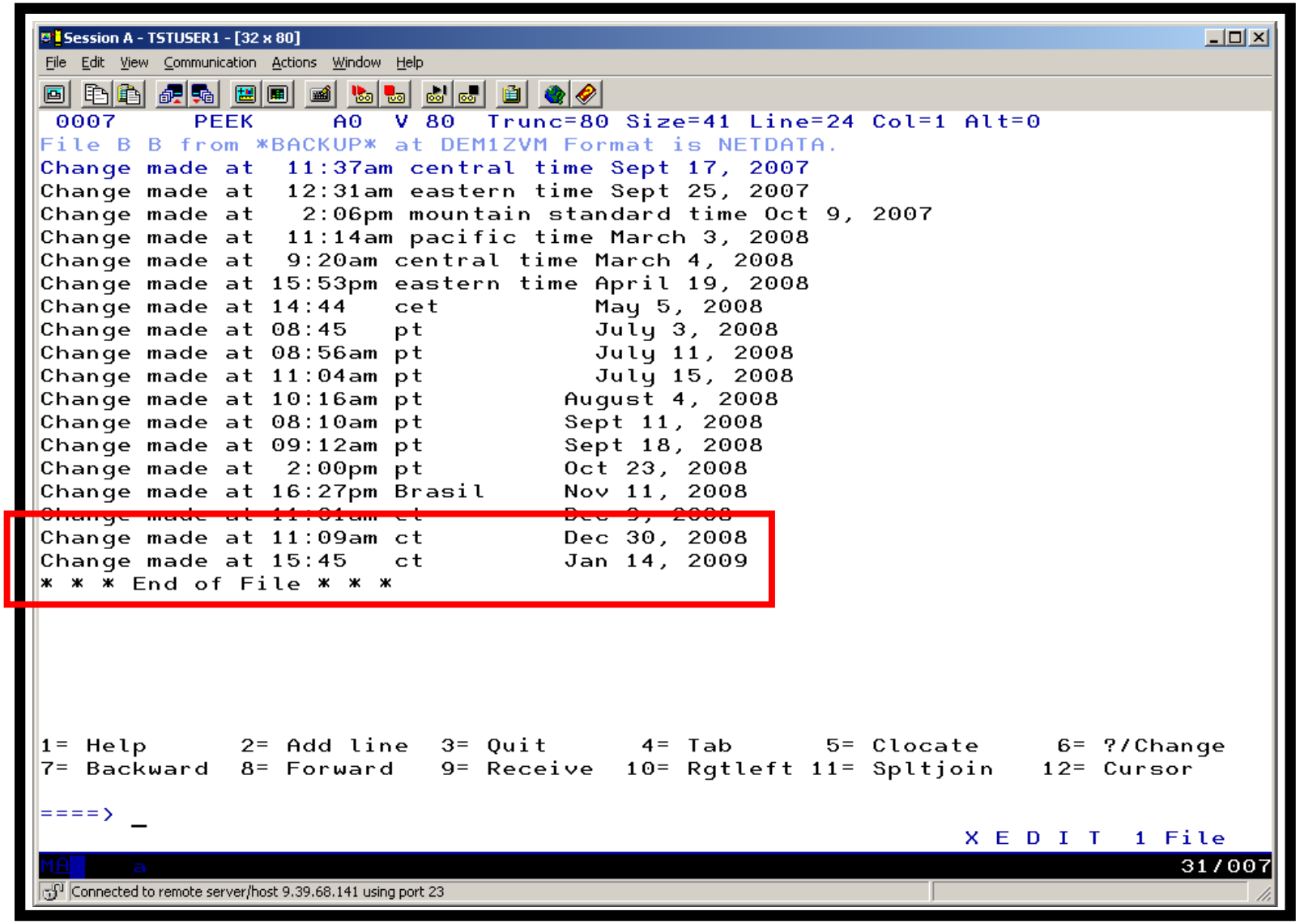












Scenario 18:
Backup and Restore Single and Multiconfiguration Users in SSI

- **Two member SSI cluster**
  - TEST7SSI, TESTCSSI
- **Three backup jobs for full backups**
  - USERFULL – all single configuration users across the SSI cluster
    - Always run from TEST7SSI (for consistency)
  - IDSSI7FL – all multiconfiguration users on TEST7SSI
    - Always run from TEST7SSI (required)
  - IDSSICFL - all multiconfiguration users on TESTCSSI
    - Always run from TESC7SSI (required)
- **Three similar jobs for incremental**
- **Restore files in multiple ways**
  - Single configuration users
    - Restore to disk or reader from any member of the cluster
  - Multiconfiguration users
    - Restore to disk from the local member
    - Restore CMS files to reader from any member

Scenario 18: Detailed Steps

- **From a Backup Manager admin ID (DEMOADMN) on TEST7SSI, view all catalog data for multiconfiguration user OP1**

```
bkruser
```

- **Use the filters to find all files for OP1's 191 disk**
  - Note files exist from both TEST7SSI and TESTCSSI
- **F4 to return and then find all files for single configuration user DEMOADMN**
  - Note files only exist in the USERxxxx jobs – not member specific
- **Update a file on OP1 191 disk**

```
link op1 191 333 mr
acc 333 z
x test op1 z
```

Add a new line to the file

```
file
rel z (det
```

- **Similarly update a file on DEMOADMN 191 disk**

```
x test demoadmn a
```

- **Perform a review of the incremental backup for multiconfiguration users on TEST7SSI**

```
smsg bkrbkup review idss7in
```

Scenario 18: Detailed Steps

- **Perform a backup for multiconfiguration users on TEST7SSI**

```
smsg bkrbkup submit idss7in
```

- **View the console of the worker(s) assigned**

```
Gomcmd opmgrm1 viewcon user(bkrwrkxx)
```

- **Perform a backup for single configuration users in the TEST7SSI and TESTCSSI cluster**

```
smsg bkrbkup submit userincr
```

- **When jobs are complete find the updated test files for OP1 and DEMOADMN in the catalog**

```
bkrlist
```

- **Once a file is chosen, use F10 to restore the file to the reader**
- **View the files in the reader**

```
rdrlist
```

- **Logoff DEMOADMN (do not disconnect – must logoff)**

```
logoff
```

Scenario 18: Detailed Steps

- **Logon DEMOADMN on the other member of the cluster TESTCSSI**
- **Fine the test files for DEMOADMN in the catalog**

```
bkrlist
```

- **Once a file is chosen, use F10 to restore the file to the reader**
- **View the files in the reader**

```
rdrlist
```

- **Notice you can restore files for DEMOADMN from either member of the cluster**

Scenario 19: Scheduling Image Backups of Linux Guests

- **Initiated or scheduled by Operations Manager**
  - Schedule defined in Operations Manager to initiate backups at specific times/intervals
  - Action associated with each schedule
    - Linux guest is shut down
    - Operations Manager watches for shutdown complete
    - Sends request to Backup and Restore Manager to back up the specific DASD/minidisks associated with the guest
      - Alternatively use FLASHCOPY to copy DASD, restart guest, then perform backup of copy of DASD.
    - Operations Manager watches for backup complete message
    - Restarts Linux guest
  - Guest is down for minimum time required for backup

Scenario 19: Detailed Steps

- **Define a schedule to start the automated backup process**

```
gomcmd opmgrm1 defschd name(demo),action(stoplnx),when(now)
```

- **View the Operations Manager log to see the schedule trigger**

```
gomcmd opmgrm1 viewlog
```

- **View the console of the Linux guest to see it shut down**

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

- **View the console of the backup server to see the backup start**

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

- **Find the worker that has been assigned and view its console**

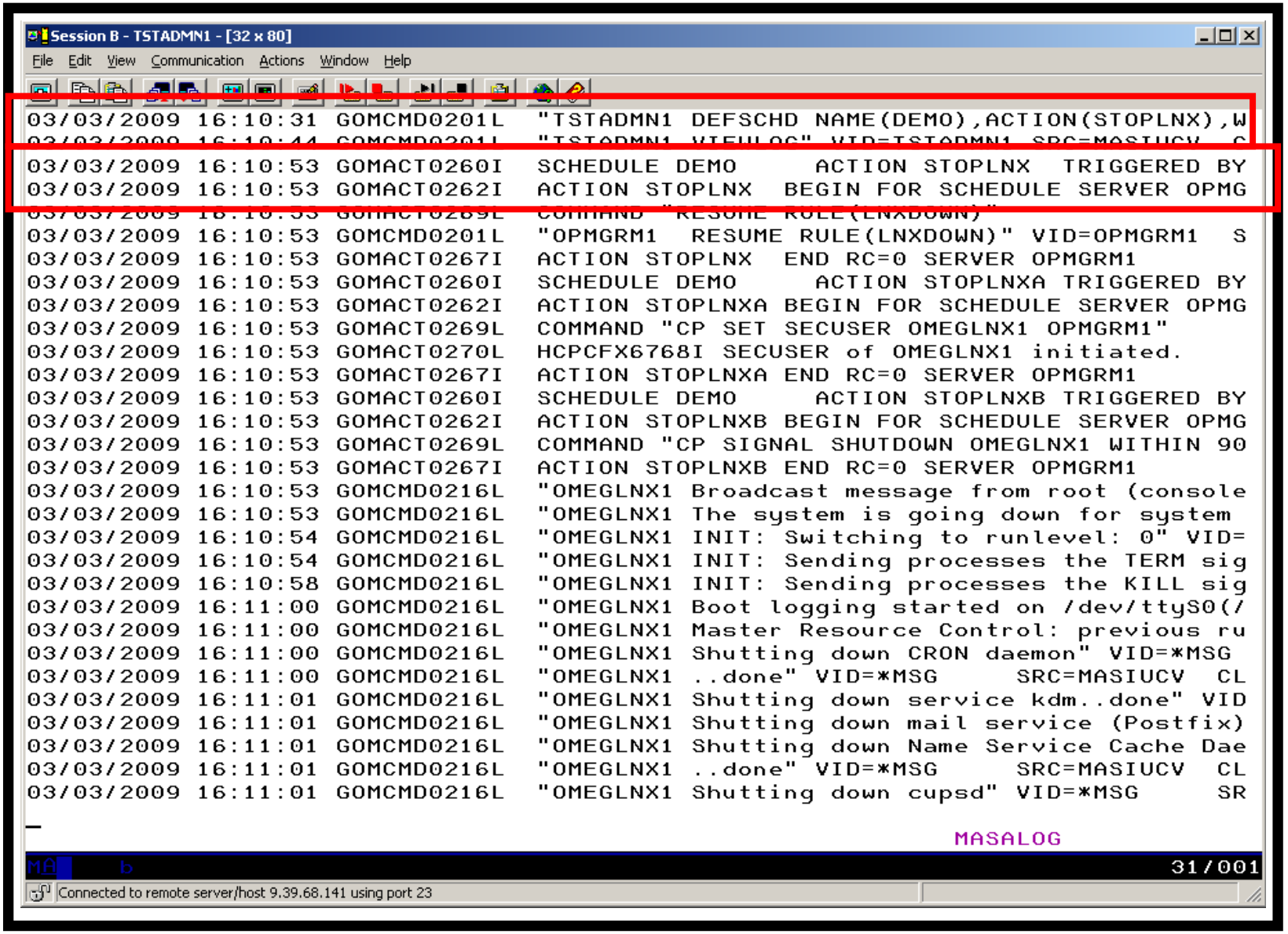
```
gomcmd opmgrm1 viewcon user(bkrwrkxx)
```

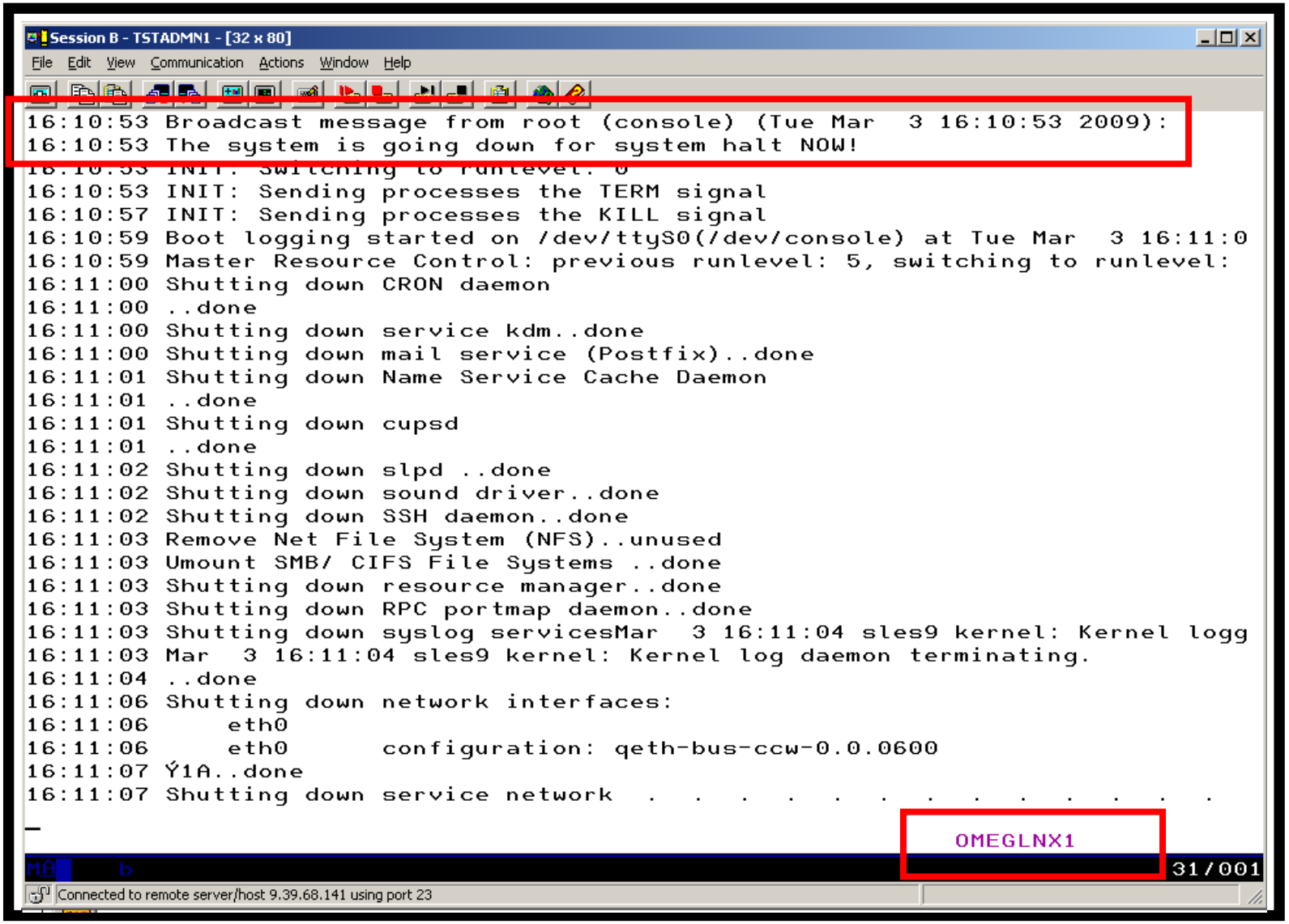
- **View the console of the Linux guest to see it restart**

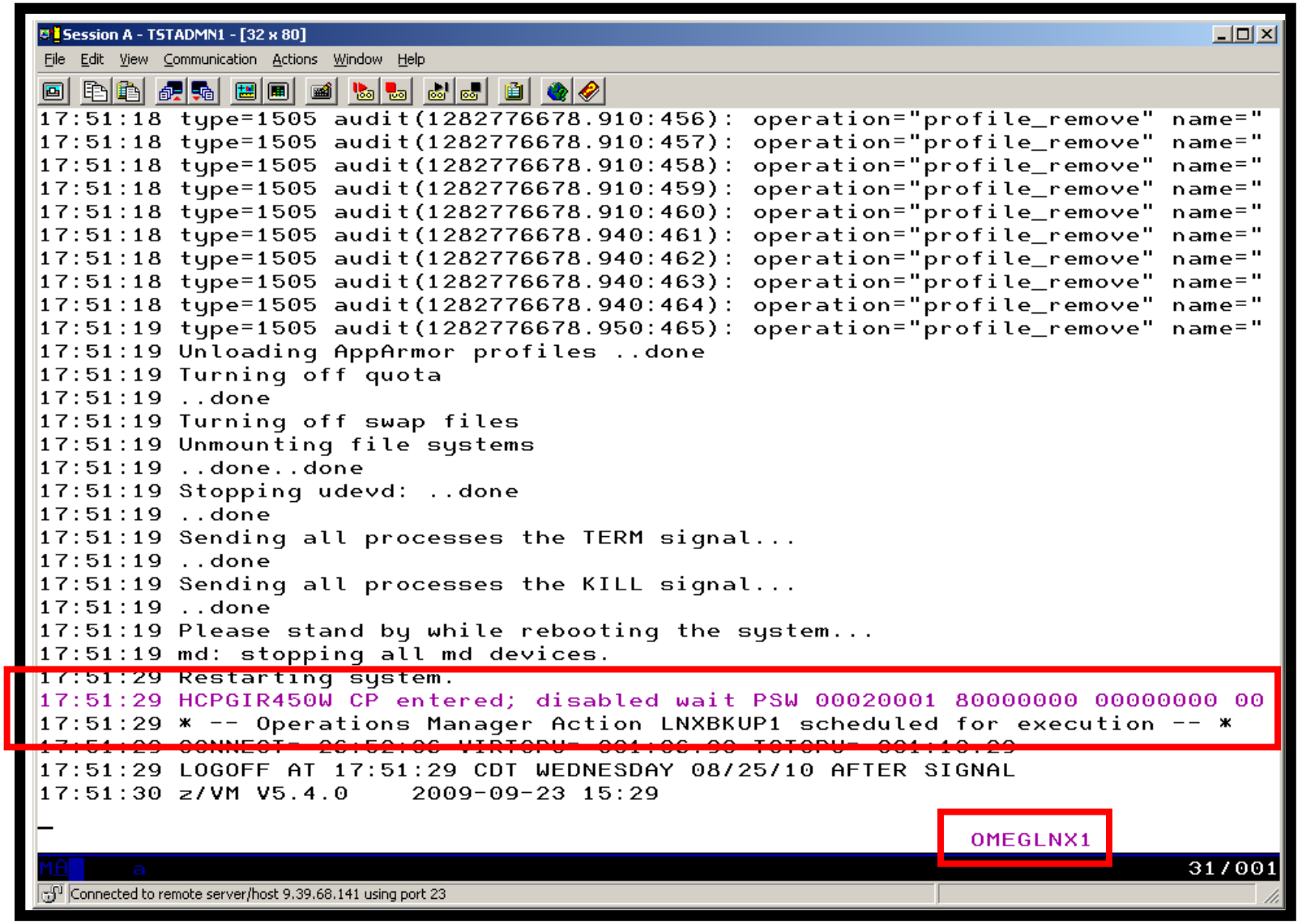
```
gomcmd opmgrm1 viewcon user(omeglnx1)
```

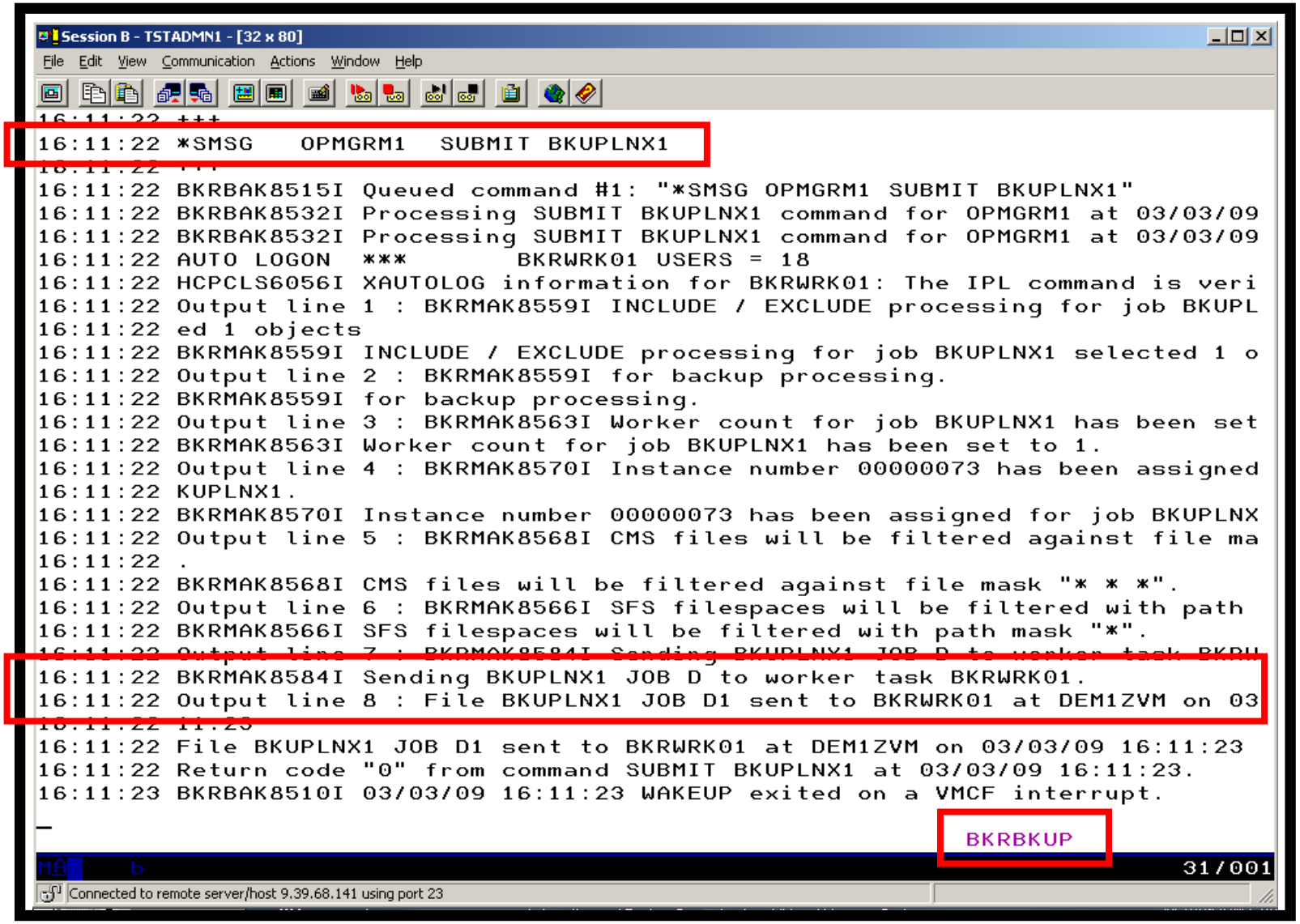
- **View the backup catalog to see the completed job**

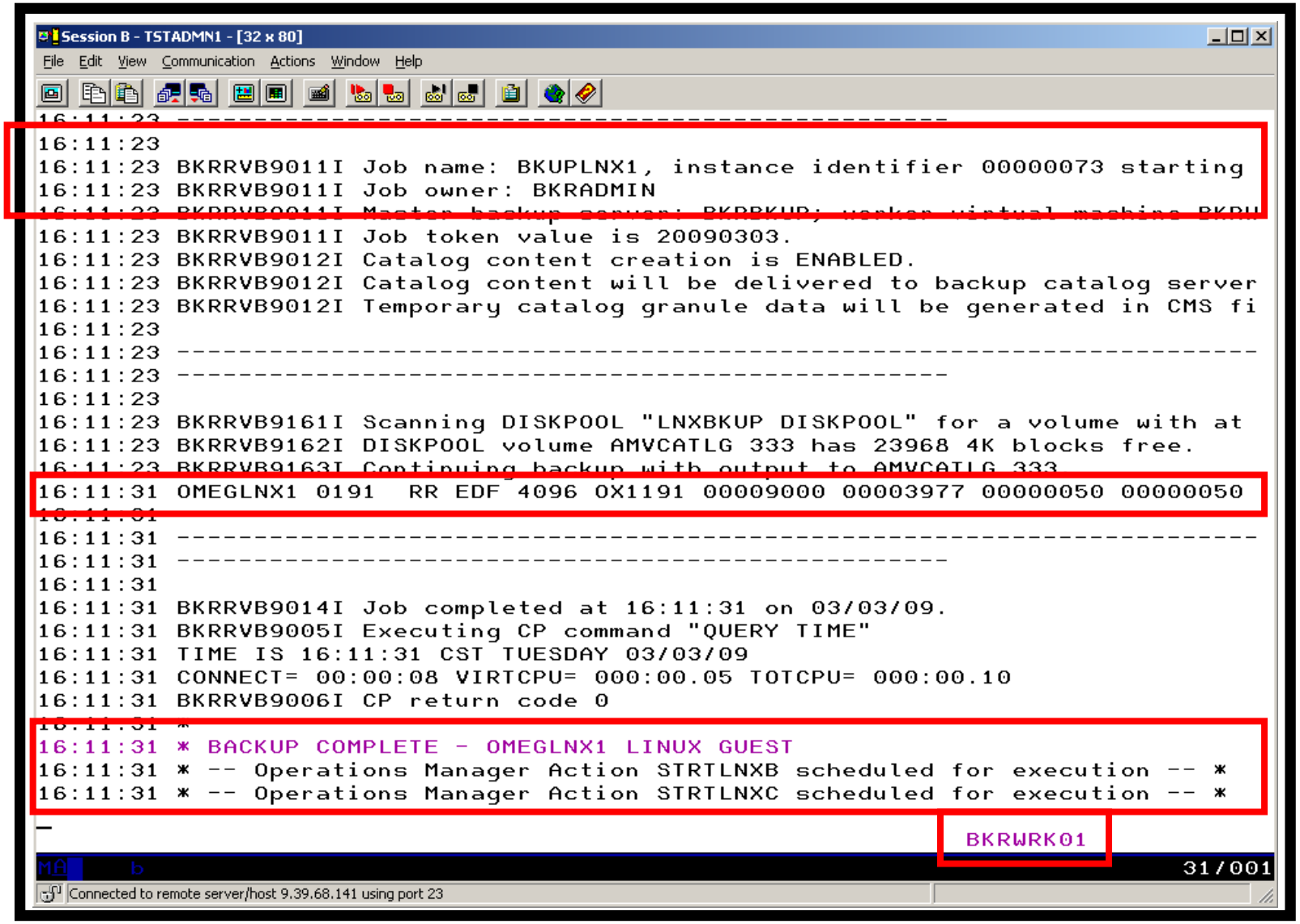
```
bkrjob
```

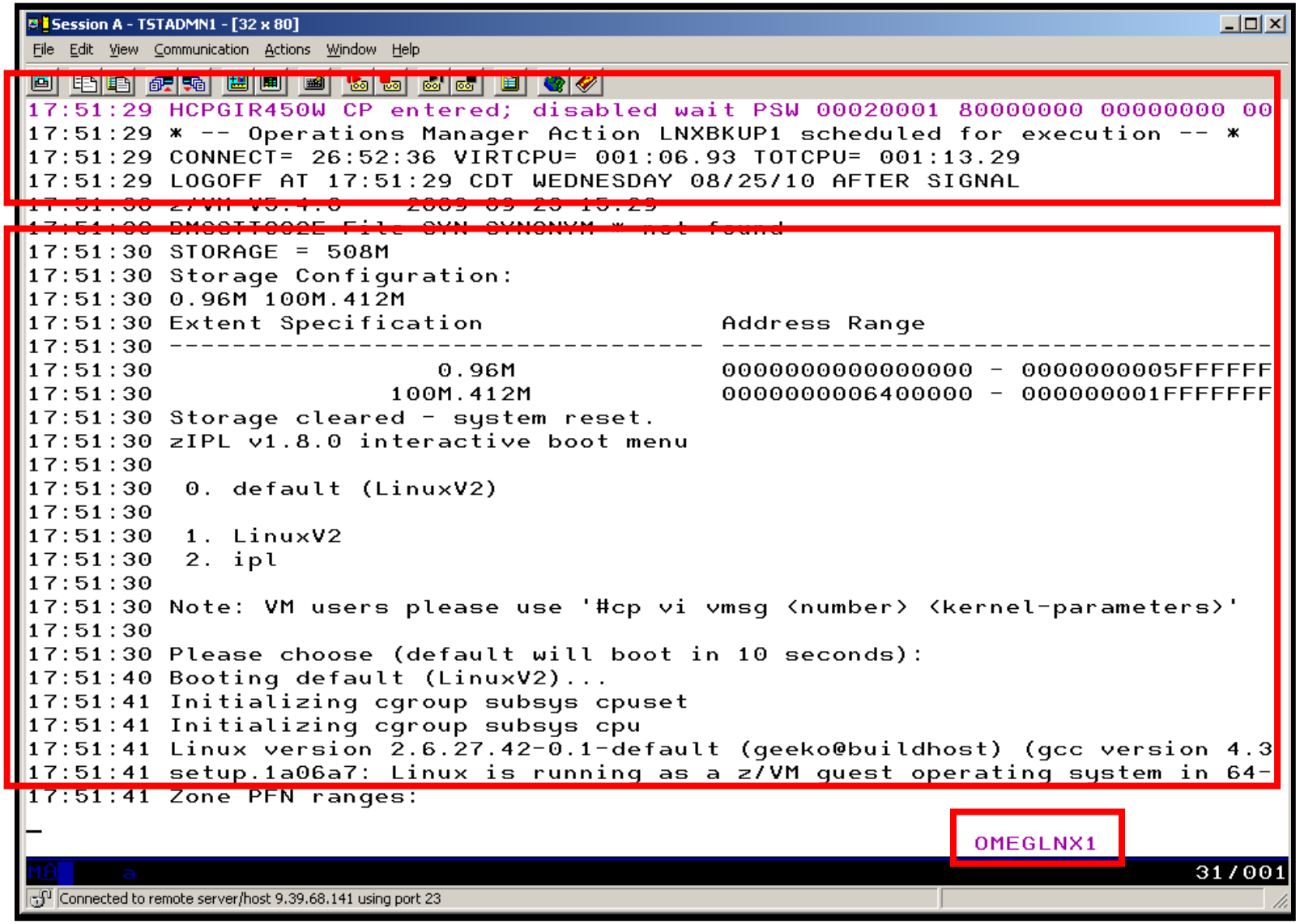


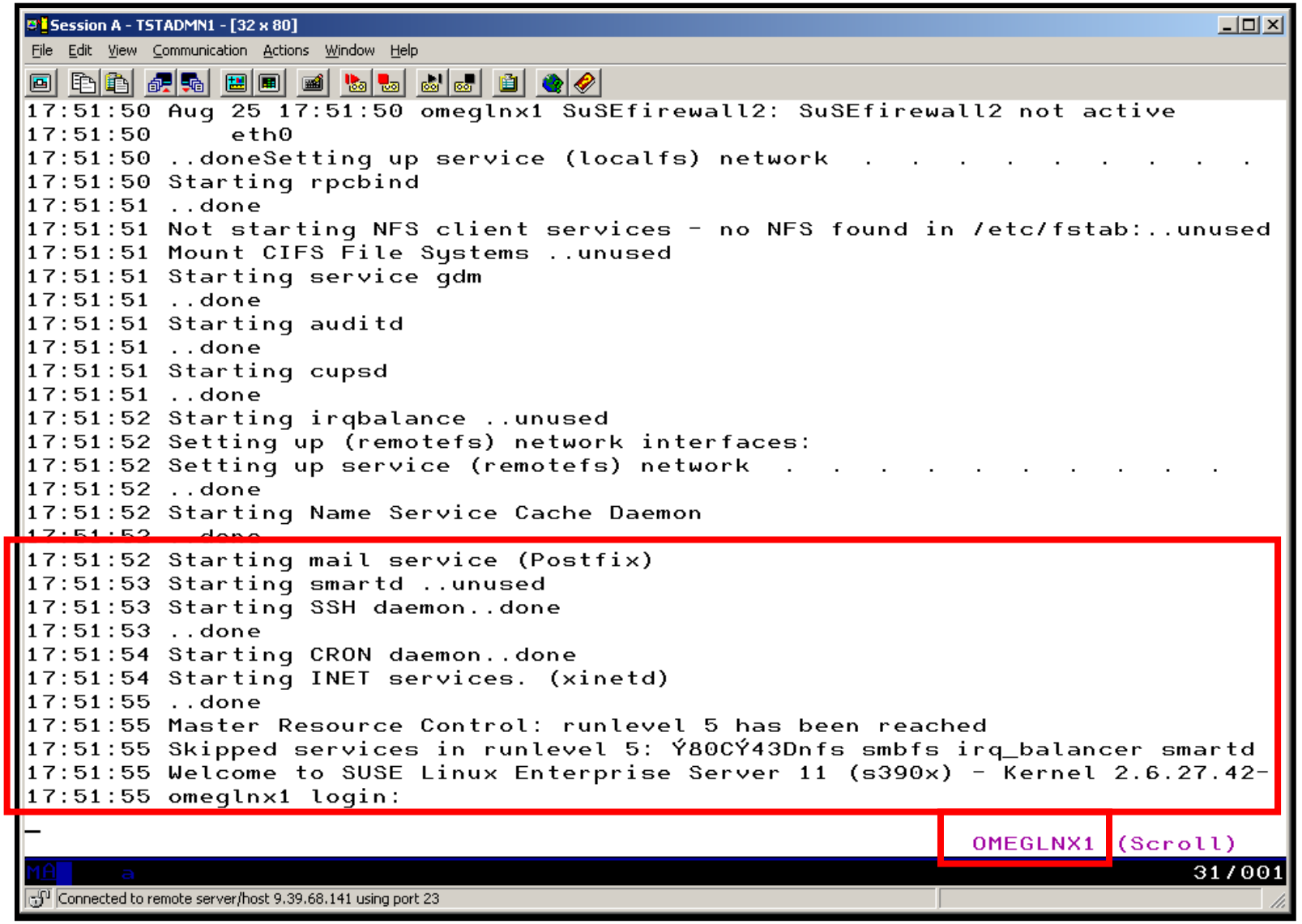












Scenario 19: How Do You Do That?

Console rule in Operations Manager:

```
*
* Watch for shutdown complete message on Linux guest
DEFRULE NAME(LNXDOWN),+
  MATCH(*HCPGIR450%*),+
  USER(OMEGLNX1),+
  ACTION(LNXBKUP)
* Turn off the rule in general
SUSPEND RULE(LNXDOWN)
```

Scenario 19: How Do You Do That?

Chain of actions in Operations Manager, triggered by schedule

```
*
* Start of guest backup scenario, resume rule for guest shutdown msg
DEFACTN NAME(STOPLNX),+
  COMMAND('RESUME RULE(LNXDOWN)'),+
  ENV(GOM),+
  NEXTACTN(STOPLNXA)
*
* Change SECUSER to Operations Manager before shutting it down
DEFACTN NAME(STOPLNXA),+
  COMMAND(CP SET SECUSER OMEGLNX1 OPMGRM1),+
  ENV(LVM),+
  NEXTACTN(STOPLNXB)
*
* Action to shut down Linux guest in prep for backup
DEFACTN NAME(STOPLNXB),+
  COMMAND(CP SIGNAL SHUTDOWN OMEGLNX1 WITHIN 90),+
  ENV(LVM)
```

Scenario 19: How Do You Do That?

Chain of actions and rules in Operations Manager:

```
* Highlight message and submit backup job for a specific Linux guest
DEFACTN NAME(LNXBKUP),+
  INPUT(AHI),+
  NEXTACTN(LNXBKUPB)
*
DEFACTN NAME(LNXBKUPB),+
  COMMAND(CP SMSG BKRBKUP SUBMIT BKUPLNX1),+
  ENV(LVM)
*
* Define all Backup Manager workers as a group
DEFGROUP NAME(BKRWRKRS),+
  USER(BKRWRK0*)
*
* Restart Linux guest when Backup is complete
DEFRULE NAME(BKUPDONE),+
  MATCH(*BACKUP COMPLETE - OMEGLNX1*),+
  GROUP(BKRWRKRS),+
  ACTION(STRTLNX)
```

Scenario 19: How Do You Do That?

Suspend rule in Operations Manager (don't back up the guest every time it is shut down)

```
* Suspend rule for backing up Linux guest
DEFACTN NAME(DELBKUP),+
  COMMAND(SUSPEND RULE(LNXDOWN)),+
  ENV(GOM)
```

Scenario 20:
Reviewing a Disaster Recovery Backup

- **Create a backup job based on sample provided**
- **Perform image backup of DASD volumes for Disaster Recovery (DR) purposes**
  - Can include z/VM and Linux guests
- **Output of backup is a DDR tape**
  - Compatible with DDR for restore at recovery site
- **Submit DR job for review**
- **Review output of review processing**

Scenario 20: Detailed Steps

- **From an authorized z/VM user ID, copy the sample DDR template from the sample disk to a new backup job**
- **Edit the new job and make necessary changes**

```
xedit ddrdemo template c
```

- **If not using SFS for templates disk, tell Backup Manager to reaccess the disk**

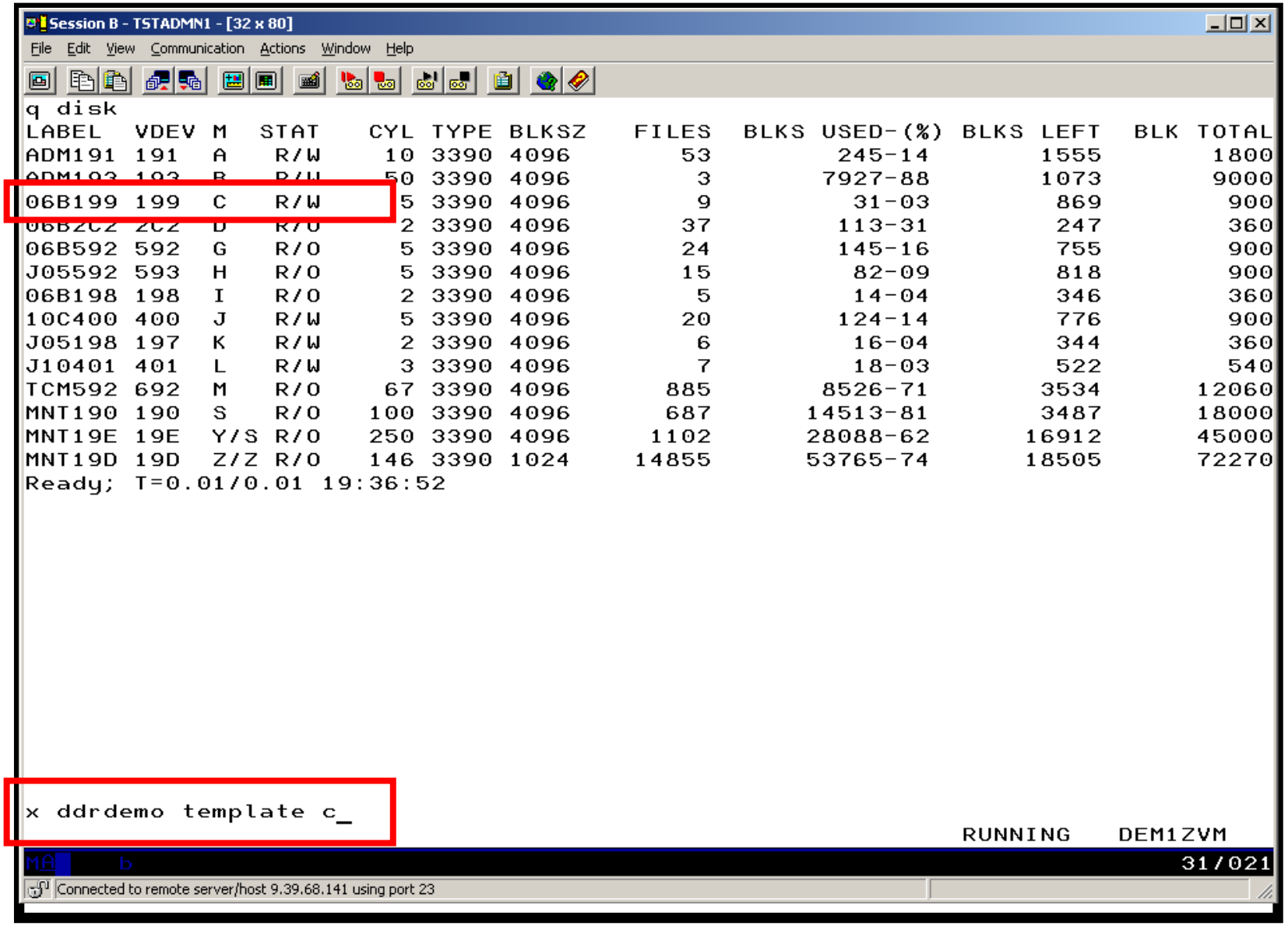
```
smsg bkrbkup cms acc 199 e/e
```

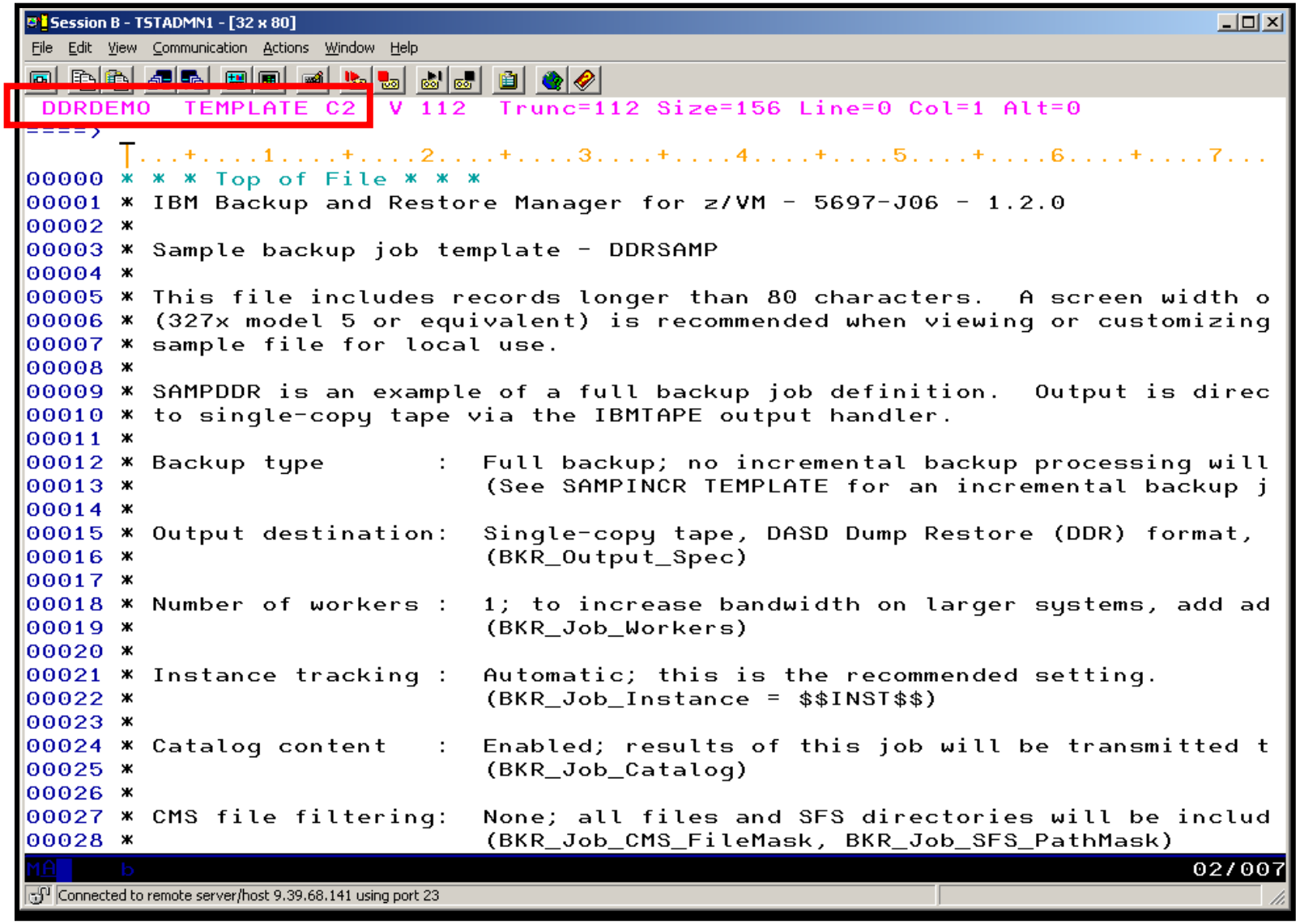
- **From an authorized z/VM user ID, submit the backup job for review processing**

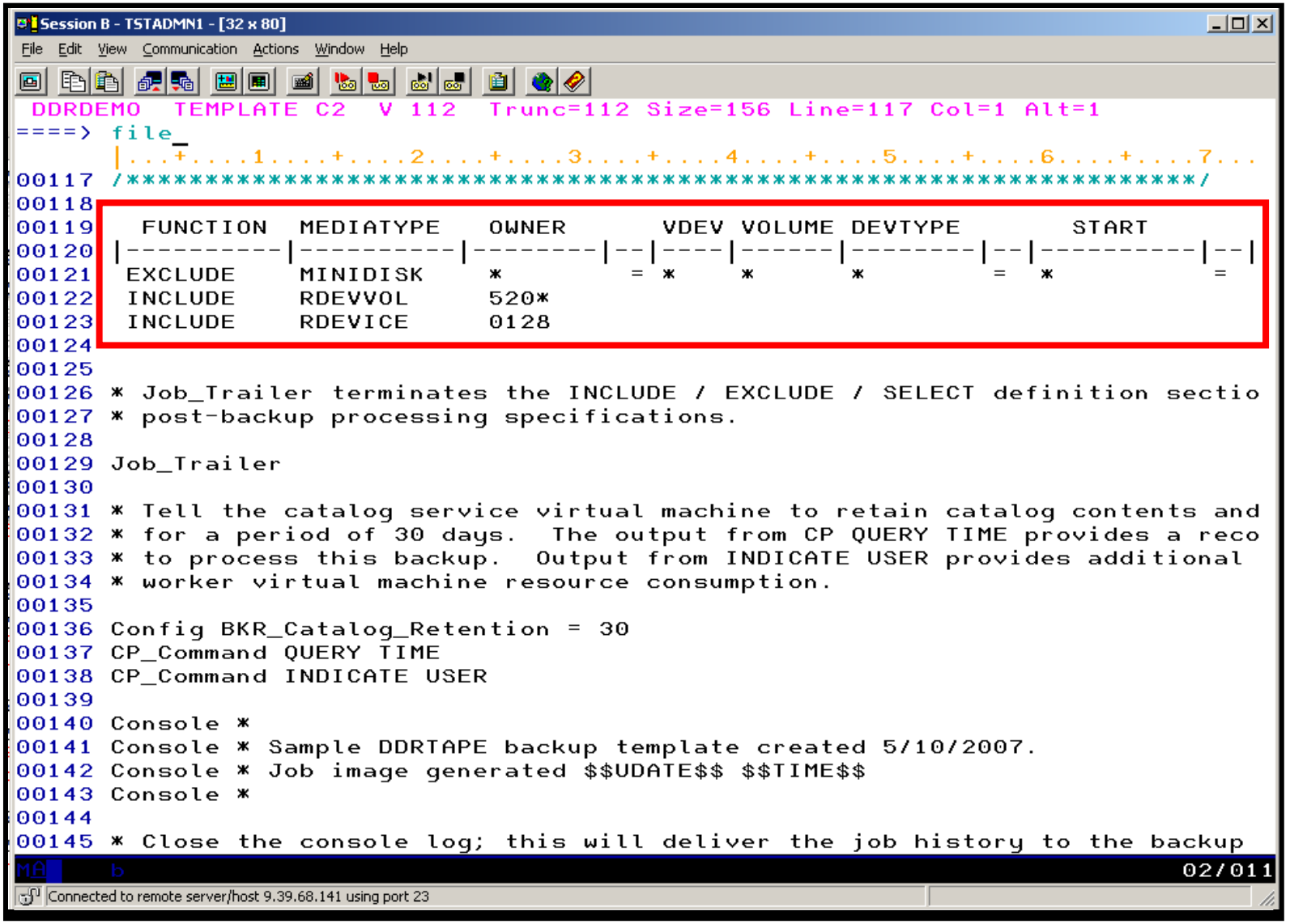
```
smsg bkrbkup review ddrdemo
```

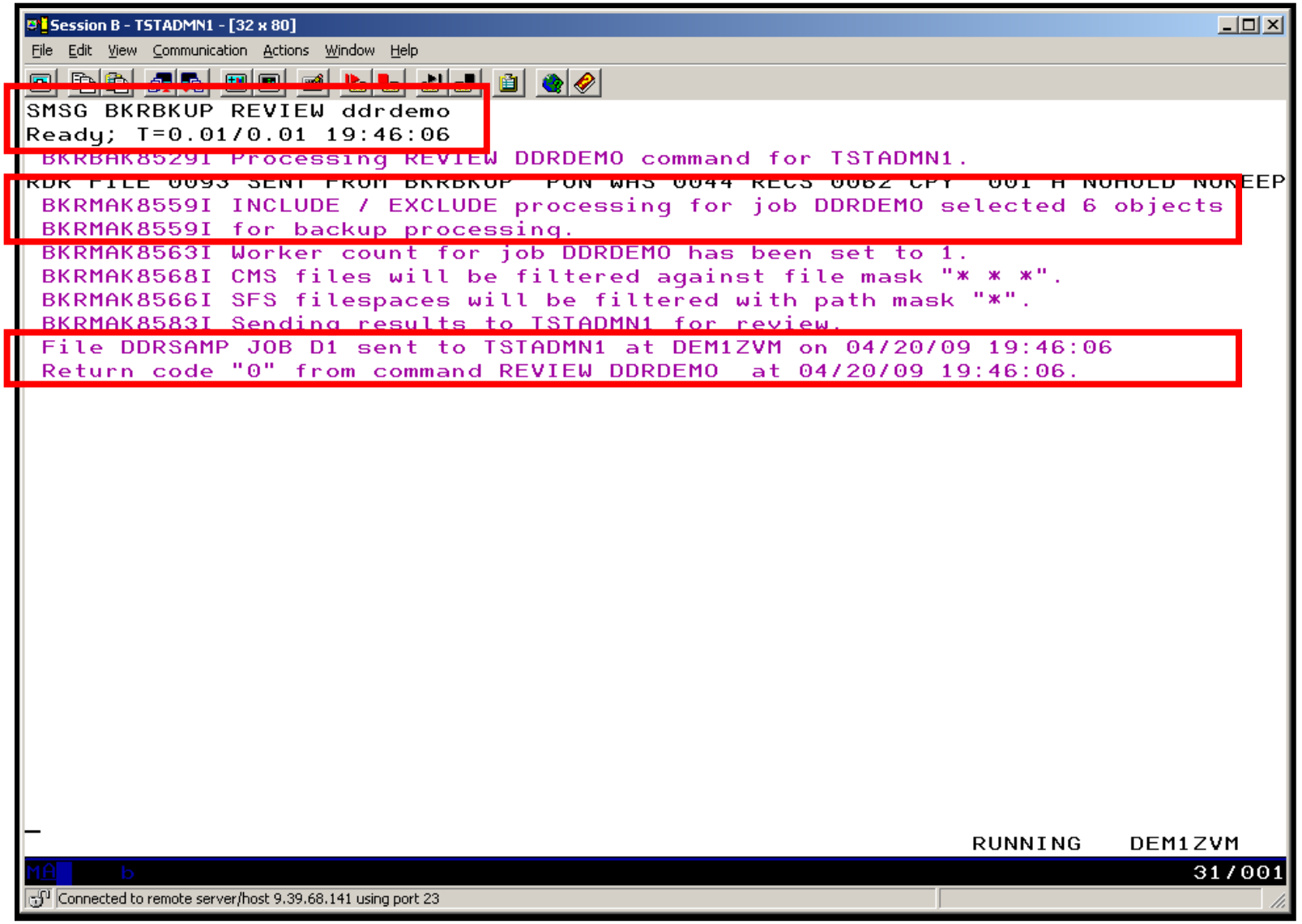
- **View the file(s) returned to you by Backup Manager**

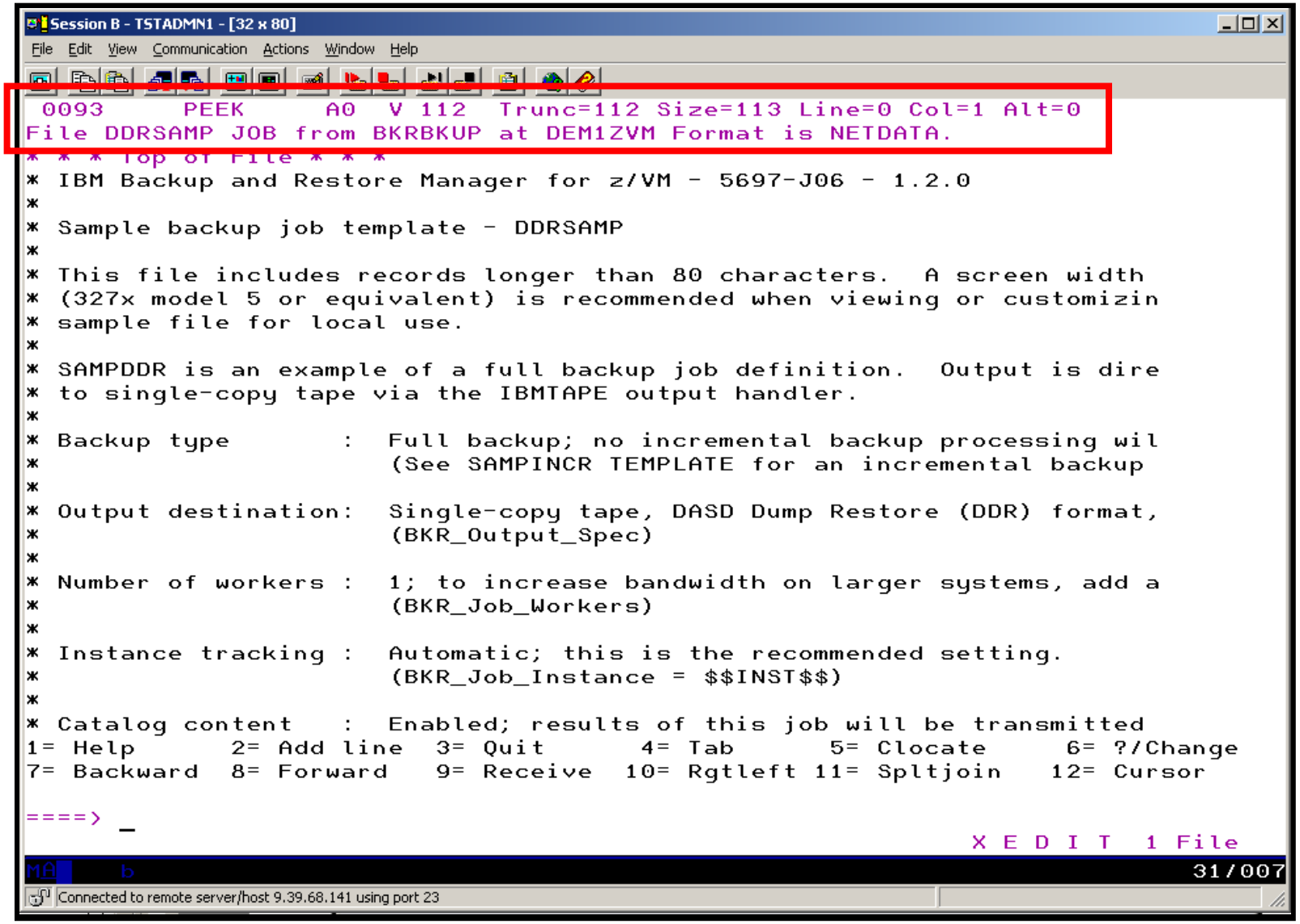
```
peek <rdrfile>
```

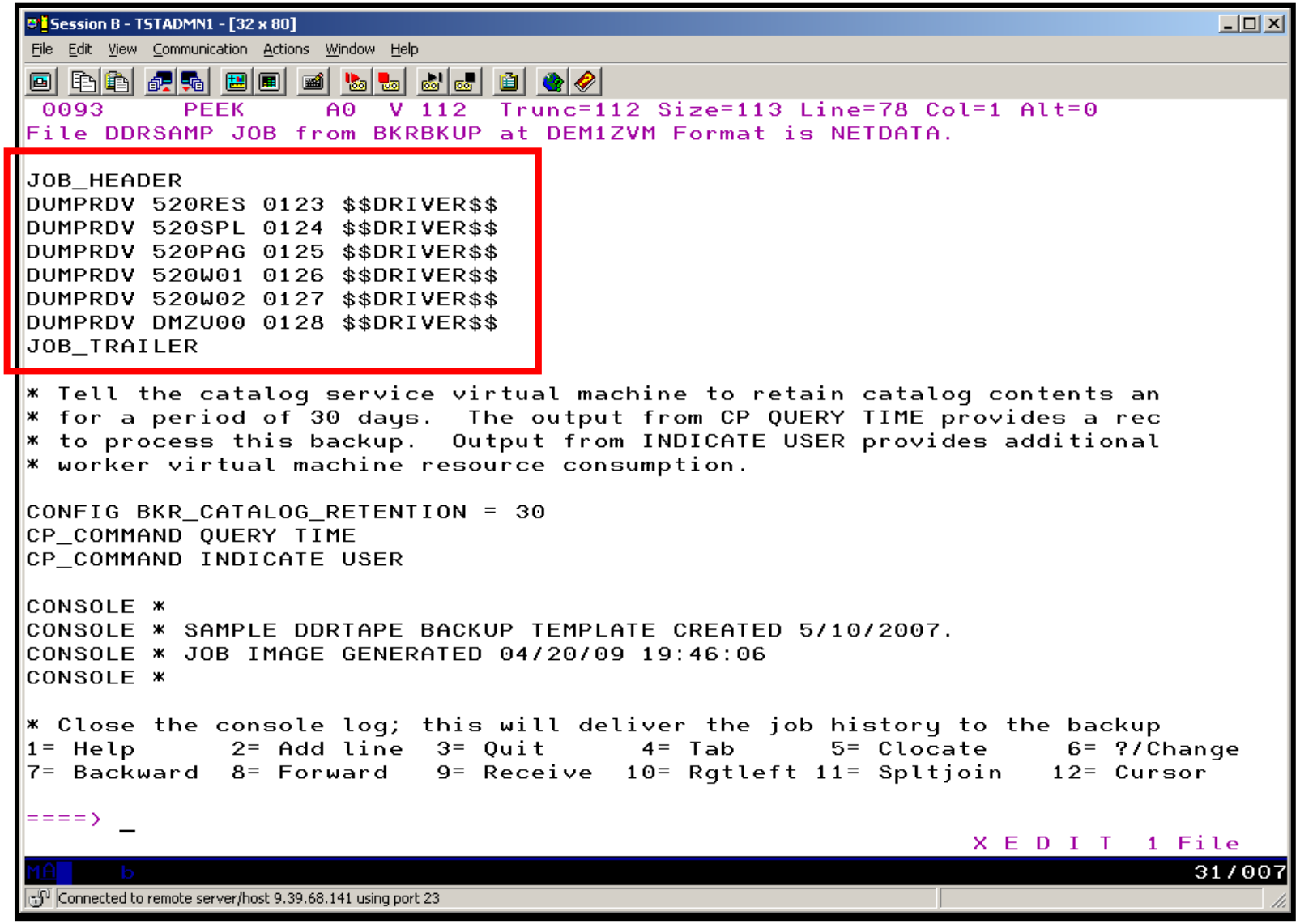












Scenario 21:
Reviewing data in the Backup catalog for recovery

- **Various backup jobs have previously been submitted and completed**
- **Full screen interfaces available for searching the backup catalog and finding data available for recovery**
  - BKRLIST
    - Useful when looking for a specific file or set of files owned by a specific user ID
    - Users with ADMIN authority beware of size
      - Use parameters to narrow the search
  - BKRUSER
    - Useful when looking for backup jobs associated with a specific user ID
  - BKRJOB
    - Useful when looking for backup jobs by job name
  - BKRVOL
    - Useful when looking for backup jobs associated with a specific DASD volume

Scenario 21: Detailed Steps

- **From an authorized z/VM user ID, issue one of the following commands to browse the catalog**

```
bkrlist
bkruser
bkrjob
bkrvol
```

- **Use F11 to drill down through details**
- **Use F10 to restore data**

धन्यवाद
Hindi

多謝
Traditional Chinese

감사합니다
Korean

Спасибо
Russian

Gracias
Spanish

Thank You
English

Obrigado
Brazilian Portuguese

شكراً
Arabic

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