#SHAREorg

SHARE
Technology · Connections · Results

# TRACK for VM

## What's happening in *your* Virtual Machine?

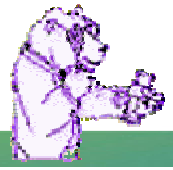
## *BYOL Lab (Bring your own Laptop!)*

James Vincent

March 2012

# Topics

- What is TRACK?
- A tiny bit of history
- A few details for your reference
  - Documentation, installing, usage
- HANDS ON – 90% of session

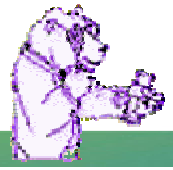


# What is TRACK?

- TRACK examines one or more virtual machines as they execute. It is a full screen program that is PF key driven and displays information in a number of different panels. It shows selected information from the TRACKed user's VMDBK, real and virtual storage, all of a user's devices, details about one device, open console page along with certain information about a CMS user's virtual machine, such as accessed disks, open files, filedefs, etc.
- There are many other panels and commands available!

# A little history of TRACK

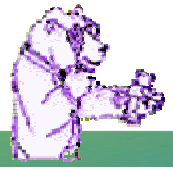
- Developed by Serge Goldstein from Princeton University's Computing Center (PUCC)
  - Around the mid 1980's
- Picked up by Ross Fried at City University of New York (CUNY)
- Other contributor(s)…
  - Perry Ruiter (then with BC Systems, now with IBM Canada)
  - Dave Jones from Sine Nomine Associates
  - Dan Martin now with Rocket Software
- Currently maintained by Jim Vincent. (See last page for contact info)



# Where to find TRACK

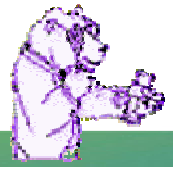
- The *new* TRACK for VM web site!
  - http://vm.marist.edu/track/

    Graciously hosted by Marist College
- Currently "**TRACK for z/VM 6.2**" is available.
  - This will support all VM versions from XA through z/VM 6.2.0
  - The web site will be the location to find new updates, news, babblings, etc, about TRACK
- Postings to the VMESA-L listserv will also be used



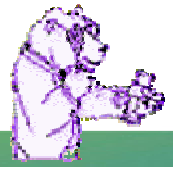
# TRACK documentation

- TRACK has very well documented commands and functions
  - README FIRST and TRACK MEMO will get you started
    - Includes the install instructions and other notes of interest
  - 34 CMS HELP files included in the package
    - A TRACK MENU helps navigate them
  - What about a MANUAL?
    - Got one!  PDF is on the web site



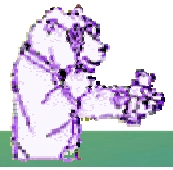
# Pre-built TRACK modules available

- There are **unsupported**, pre-built TRACK modules in the download VMARC package
  - I highly recommend Assembling your own TRACK modules if you can
    - Stuff happens… CP data areas change
  - Assembling your own will ensure TRACK will work correctly for your system
- The pre-built modules are NOT supported!



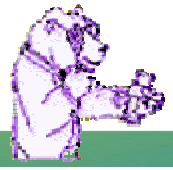
# Installing TRACK

- Start with the README FIRST and TRACK MEMO file
- Installation will depend on your current level of VM
- TRACK is mostly assembler and very dependent on CP source code (MACLIBs)
- Installed and maintained using the standard VM update scheme
  - Xedit Update files, AUX files, etc.



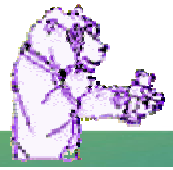
# Installing TRACK (more…)

- CNTRL files help you manage which version of TRACK is built
  - One for each level of VM supported
    - `TRACKV62  z/VM 6.2.0 systems`
    - `TRACKV54  z/VM 5.4.0 and 6.1.0 systems`
    - `TRACKV53  z/VM 5.3.0 64-bit systems`
    - `…etc…`



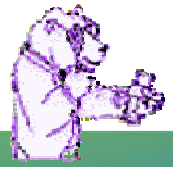
# Installing TRACK (more…)

- Select the right CNTRL file for the release of VM you have
- Verify your current CP control file entry, TEXT MACS to the TRACKVxx CNTRL file you are going to use (HCPVM or HCPVME)
  - Be sure TRAMAC is prefixed on the MACLIB list
  - Not using the right MACLIBs will cause bad things to happen to good people



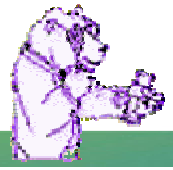
# Installing TRACK (more…)

- You may need to modify some of the AUX files
  - Generally for a current VM, what is there is fine
  - Most updates were for older releases that fixed some things before they were included in the base CP code
  - Be sure to review the TRACK MEMO in the “Modifications” section for updates you may want to include
- Build TRAMAC MACLIB
  - VMFMAC TRAMAC TRACKVxx
    - TRACKVxx is the name of the CNTRL file you will be using



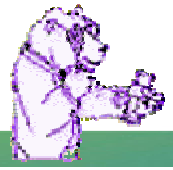
# Installing TRACK (more…)

- Assemble all the parts!
  - Nice little EXEC provided to help you do that
  - TRKASM EXEC VMFHLASM TRACKVxx
    - TRACKVxx is the name of the CNTRL file you will be using
- Modify the load list if you want
  - Handy if you want a ‘lite’ version of TRACK but these days, storage is cheap so build the whole thing!
- Build the TRACK module
  - TRKGEN TRACK TRACKVxx
    - TRACKVxx is the name of the CNTRL file you will be using
    - You can change the name of the module if you want
      - TRKGEN *modulename* TRACKVxx



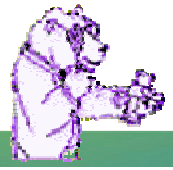
# Installing TRACK (almost done!)

- Copy the TRACK MODULE and all HELP files to a tools disk
- Keep all the files somewhere – you will be updating them as your VM level changes or more updates are made available



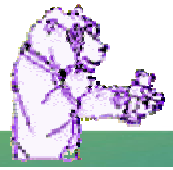
# Using TRACK - requirements

- Required CLASSes
  - You will need CLASS E to use Diag 004 (read CP real storage) and CP LOCATE
  - You will need CLASS D for Diag 290 for view open spool data (optional; z/VM 5.2+ only)
  - You will need CLASS A for the CP LOCK/UNLOCK commands (optional)
  - If you moved these commands to different privileges, adjust accordingly...
- Note – although you need high privileges, TRACK never modifies storage in any way
  - It will only LOCK virtual pages as needed and UNLOCK them immediately if they were not locked in the first place



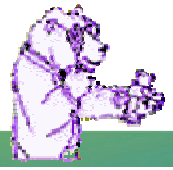
# Using TRACK – a little help

- Nifty front-end routine ZTRACK EXEC included in VMARC
  - Checks for the required privilege classes
  - Executes the correct TRACK module based on the level of z/VM you are running
    - Handy if you have both a 32 and 64 bit system you want to use TRACK on or sharing a tools disk on multiple VM systems with different levels of CP
  - Modify the code to fit your needs



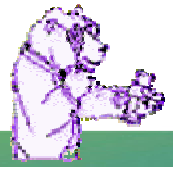
# Using TRACK - notes

- 64 bit support in TRACK needed some panel changes/rearrangements
- Some of the panels will look a bit different depending on what mode you are in (32 vs 64 bit)
  - Registers and data locations expand to 16 bytes for 64 bit
  - Some data condensed to allow room (see VST/RST panels for example)
    - Offset field is only 7 bytes, yet offset could be much more than that



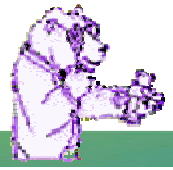
# Using TRACK - notes

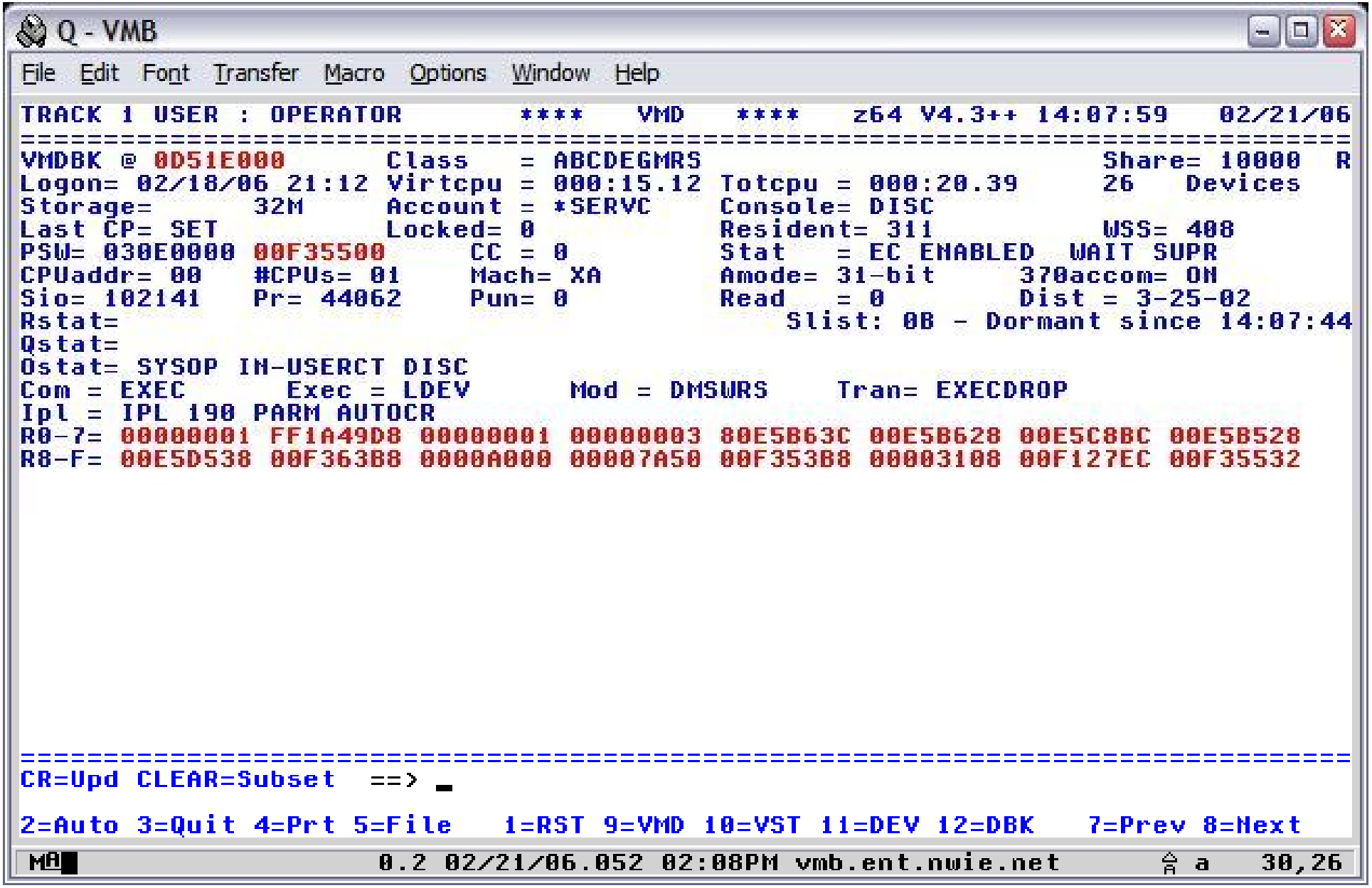
- z/VM 5.2 required a new switch for CP storage
  - Default mode is Host Logical
  - New command:
    - HOSTREAL ON|OFF|?
      - ON makes TRACK look at Host REAL storage
      - OFF looks at Host Logical storage
      - ? Displays the current setting
  - RST will show [ LOGICAL ] or [ REAL ] above the addresses
  - Watch for possible storage errors; if you are set to Logical, Diag 4 may not get what you expect
    - FORMAT, DBK, etc…



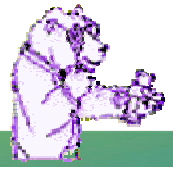
# Really Using TRACK!

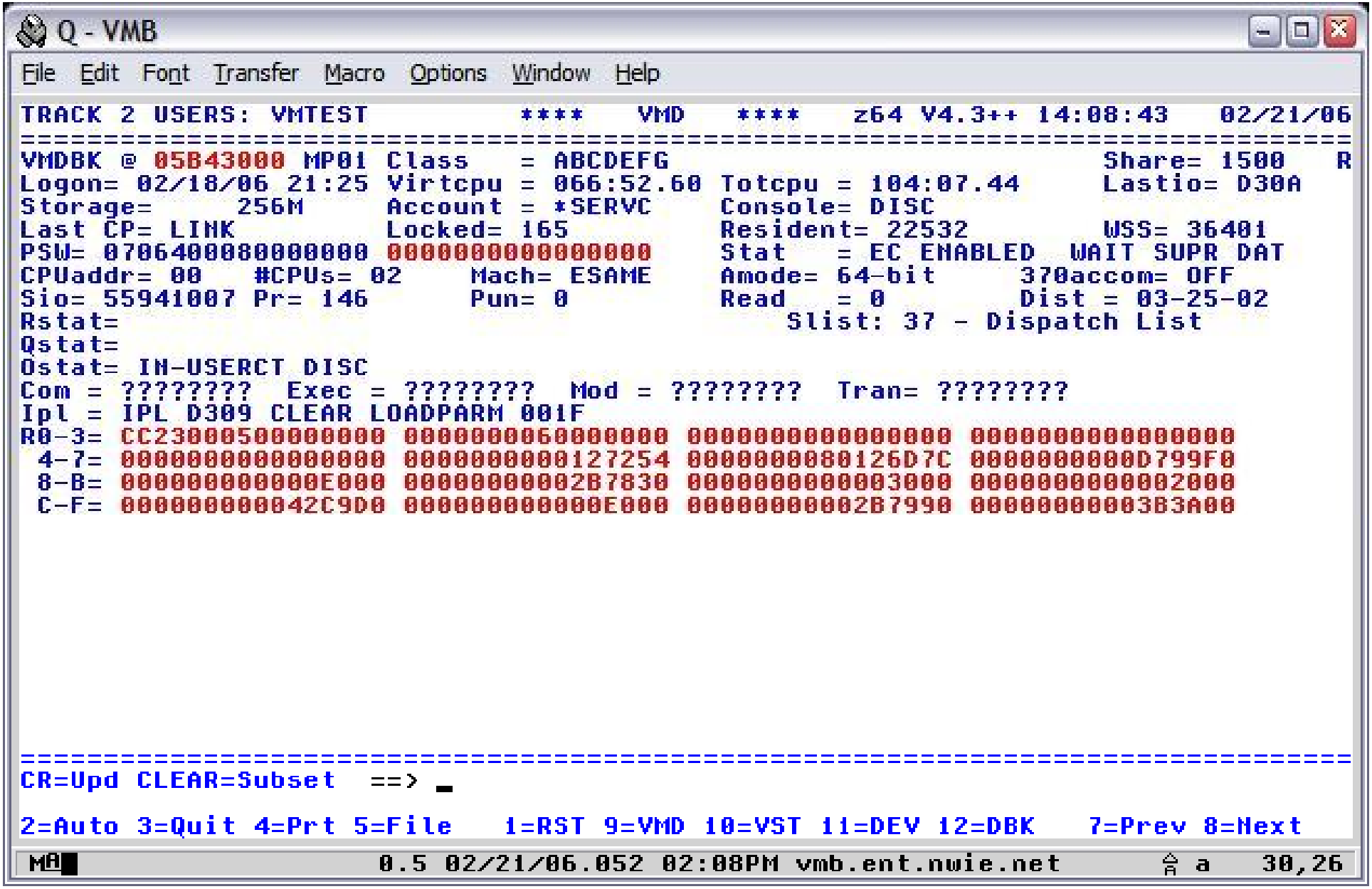
- The simplest invocation:  TRACK userid
  - Example:  TRACK OPERATOR
- Tap ENTER on any panel to update/refresh the contents
  - Or use the AUTO command to set an automatic update time and limit of updates, then use  PF2 (AUTO)
  - Use “AUTO .5” for half-second, almost real-time updates
  - Sit back and watch!
    - Popcorn and ‘adult foamy beverage’ optional

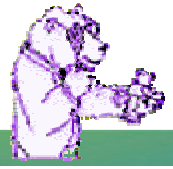





# VMD Panel – 32 bit guest



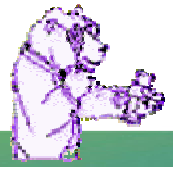


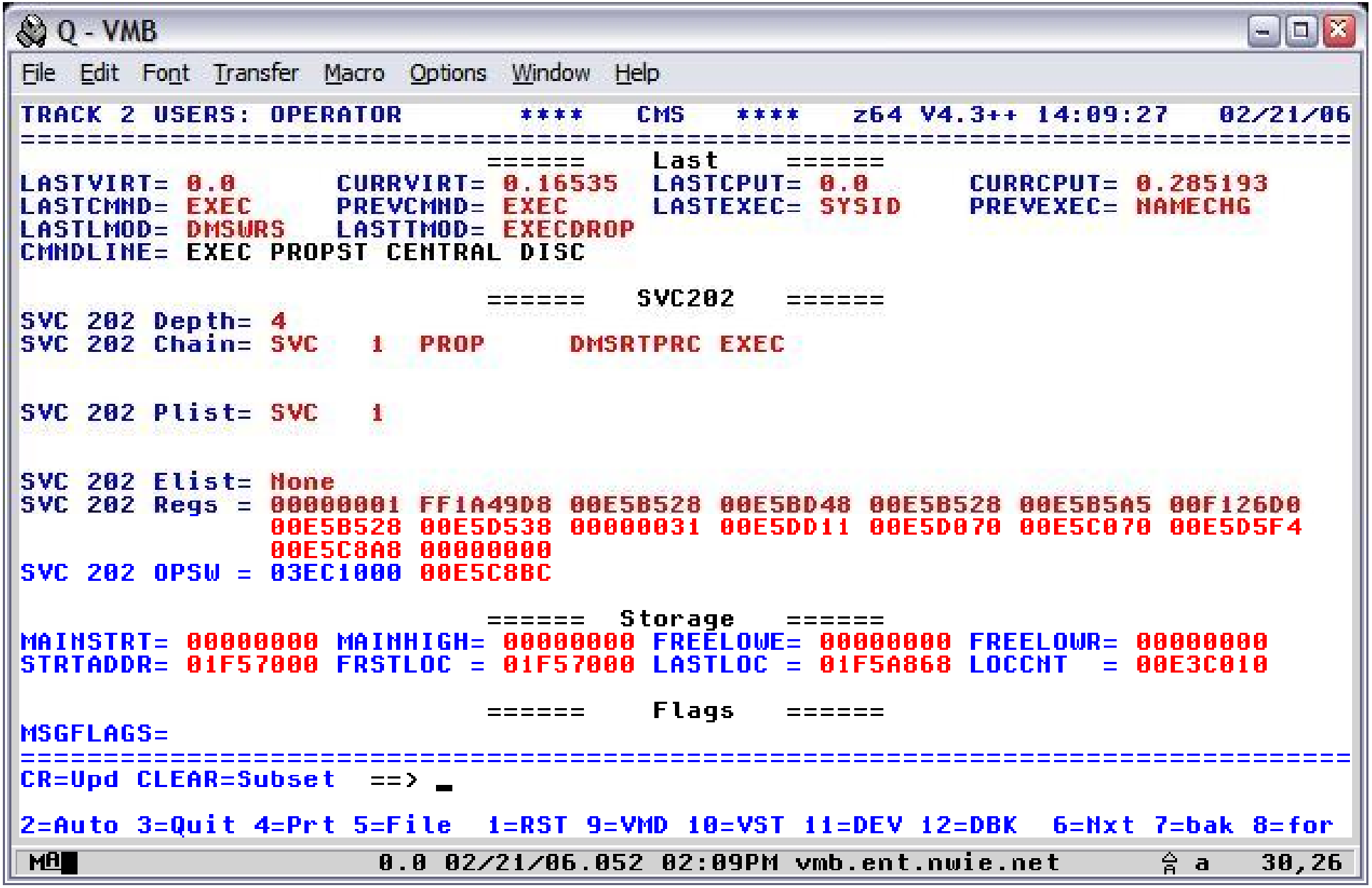



# VMD Panel – 64 bit guest

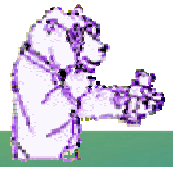
# Using TRACK

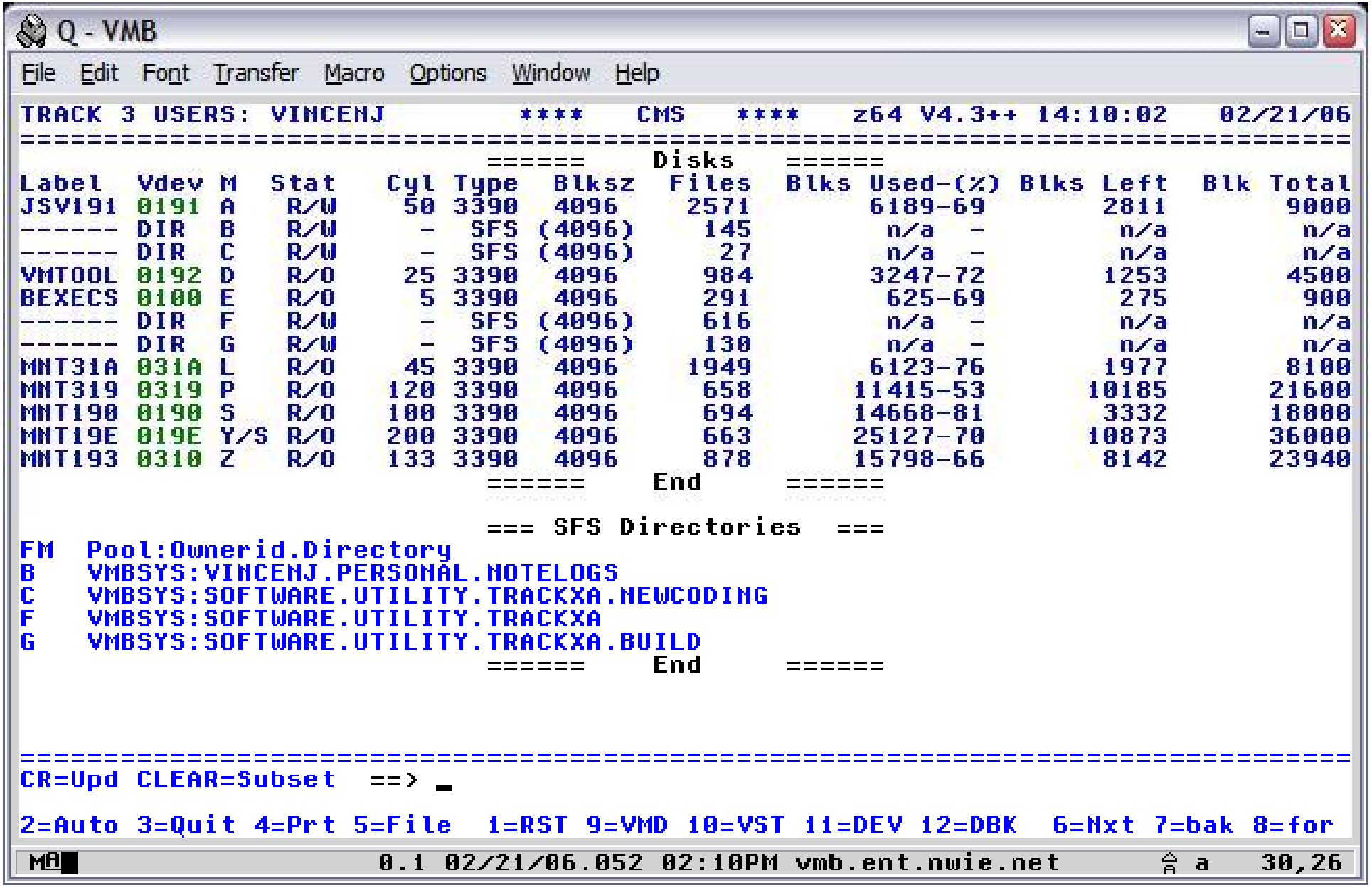
- Let's look at the CMS settings
  - Type CMS on the command line
  - CMS panel can be customized to show any or all of the sections – in any order!
    - Handy to see DISKS and SFS directories accessed
  - CMS XEDIT will show the current Xedit data
  - CMS STACK shows what data is currently stacked by buffer number

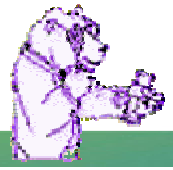




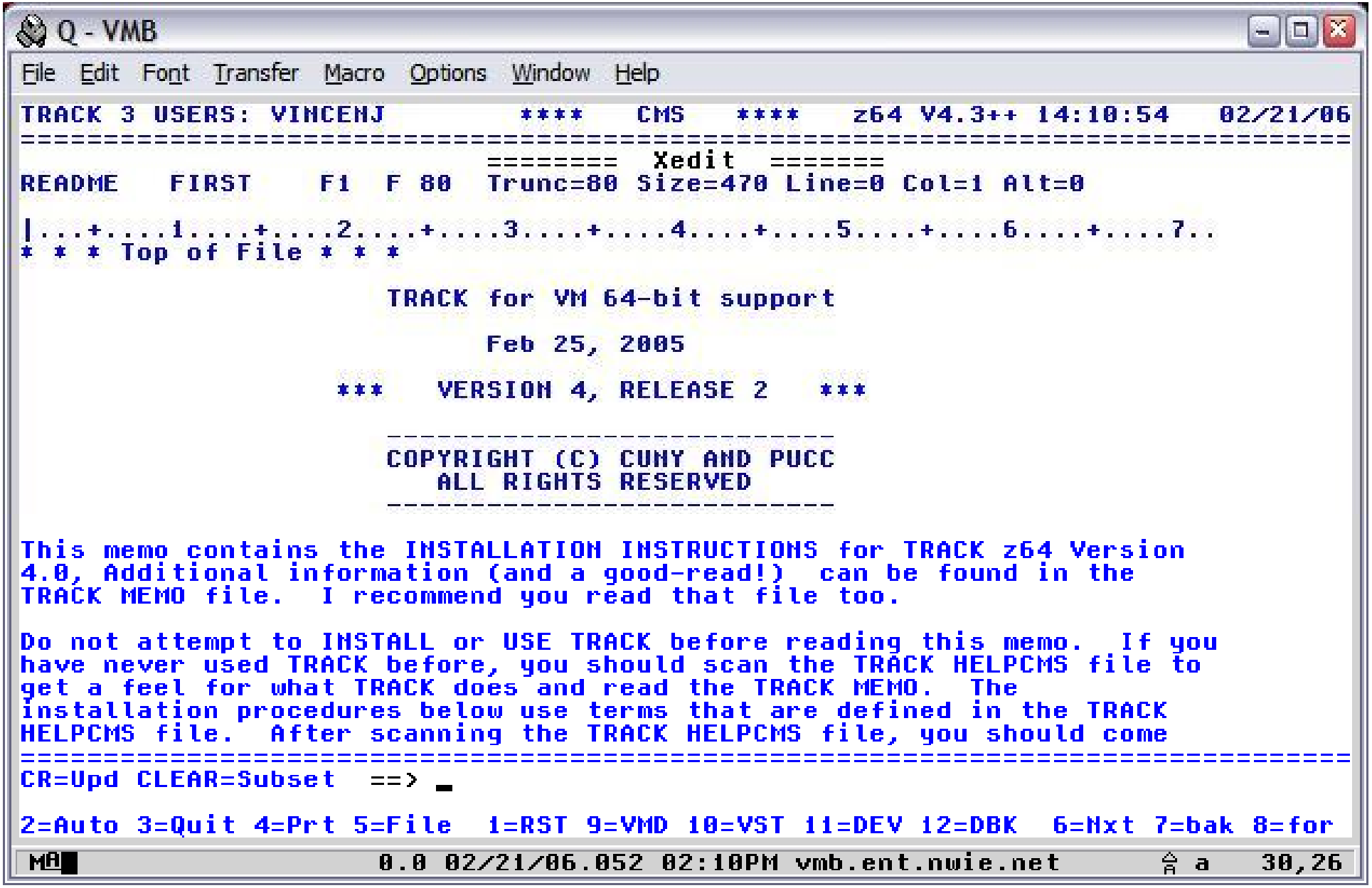

# CMS Panel (default)

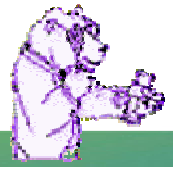




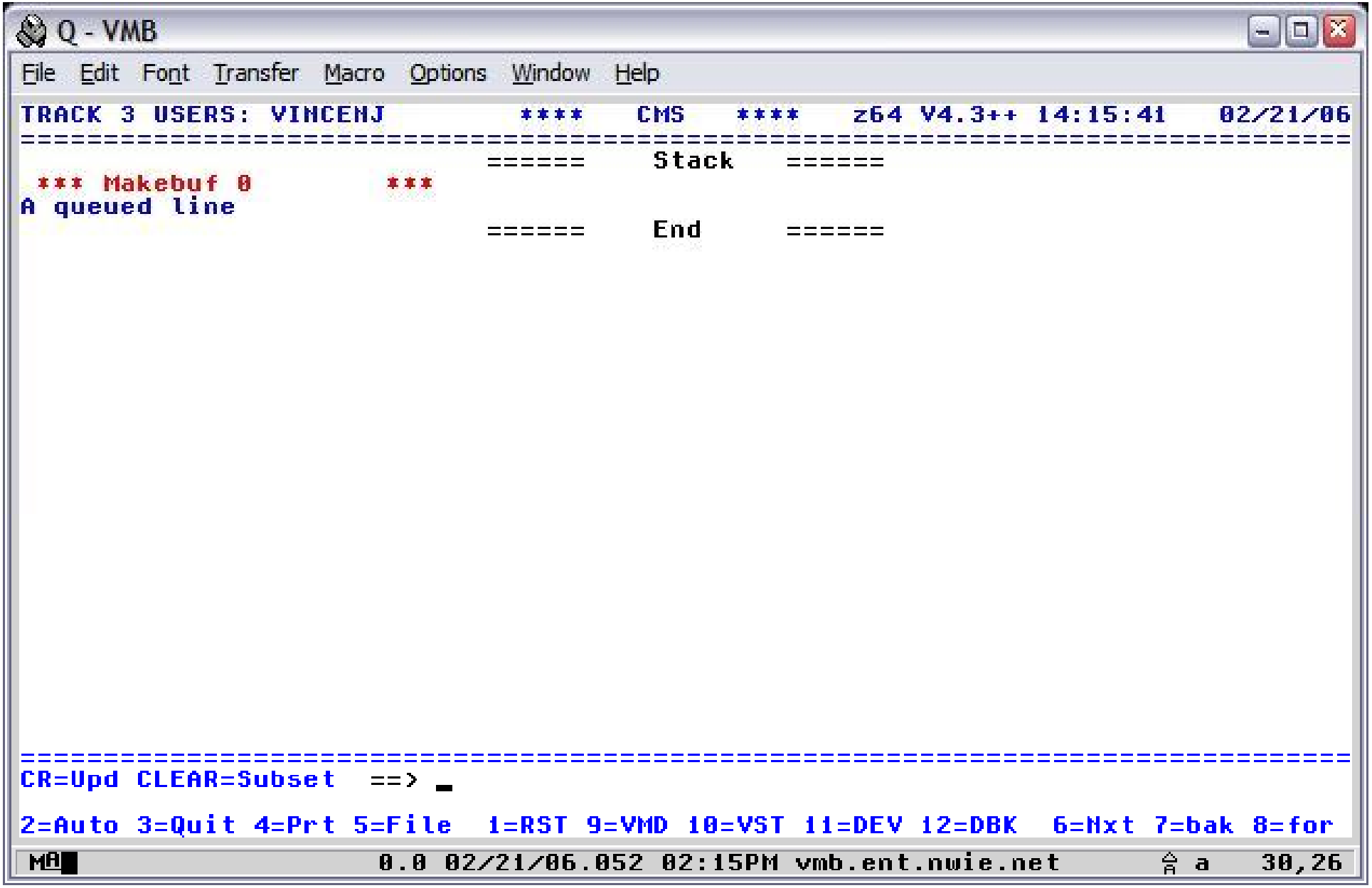



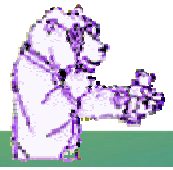
# CMS Panel (custom: TRACK user CMS DISKS SFS)



# CMS Panel (custom: TRACK user CMS XEDIT)

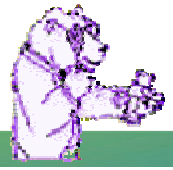


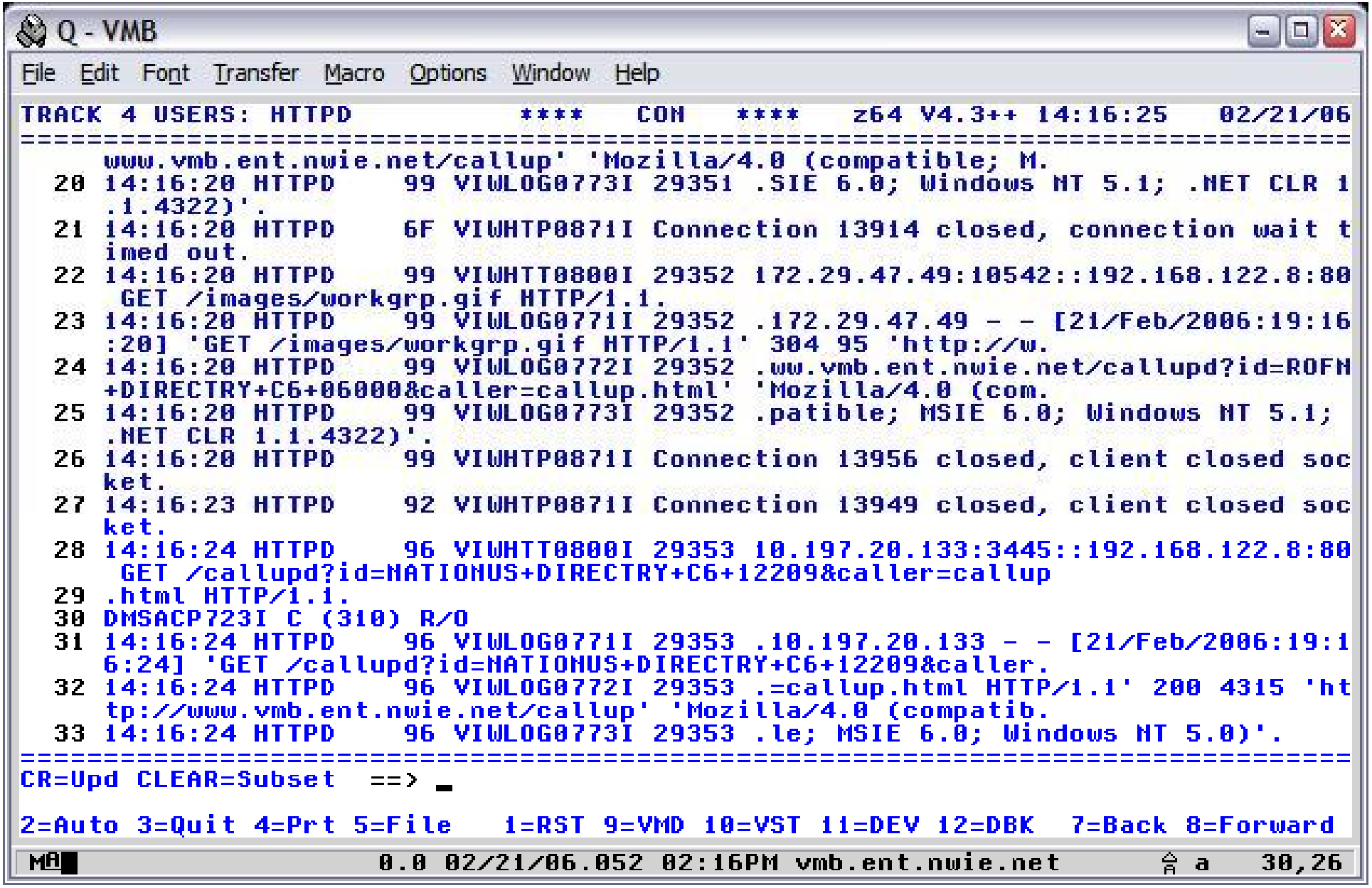



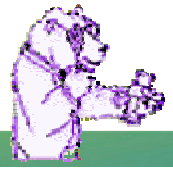
# CMS Panel (custom: TRACK user CMS STACK)

# Using TRACK

- For examples, I will cycle through the following in order:
  - CONS (open console)
  - DEV (via PF11 – virtual devices)
  - NAM (‘query names’)
  - PFX (prefix page)
  - RST (real storage) and VST (virtual storage)
    - Both with “Disassembly mode” and ASCII display toggle
- Note that these are only a sampling of what TRACK can do!





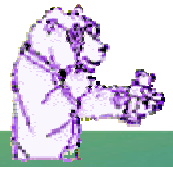



# CONS Panel – last active open console block

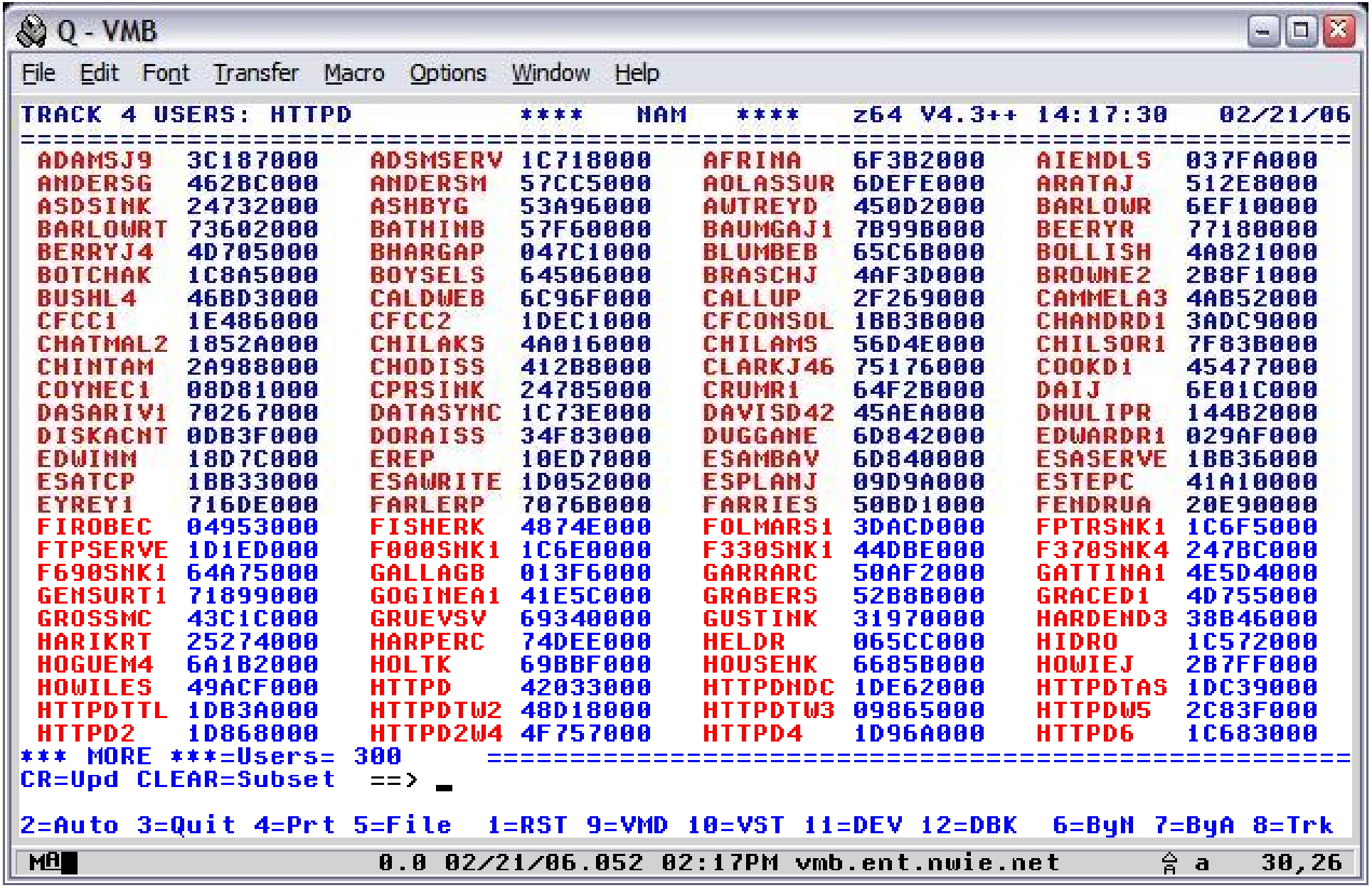
Q - VMB

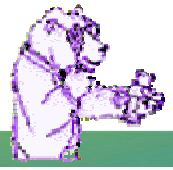
File Edit Font Transfer Macro Options Window Help

```
TRACK 4 USERS: HTTPD          ****   DEV   ****   z64 V4.3++ 14:17:06   02/21/06
================================================================================
TERM 0009  3215     SPOL 000C 2540R     SPOL 000D 2540P     SPOL 000E  3800
DASD 0119  3390     DASD 011A  3390     DASD 0181  3390     DASD 0190  3390
DASD 0191  3390     DASD 0193  3390     DASD 019D  3390     DASD 019E  3390
DASD 01B0  3390     DASD C191  3390     DASD F191  3390


(Chan= 4   Ctls= 7    Devs= 15   )=========================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File  1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=ByC 7=ByA 8=ByT
MA                    0.0 02/21/06.052 02:17PM vmb.ent.nwie.net      a   30,26
```

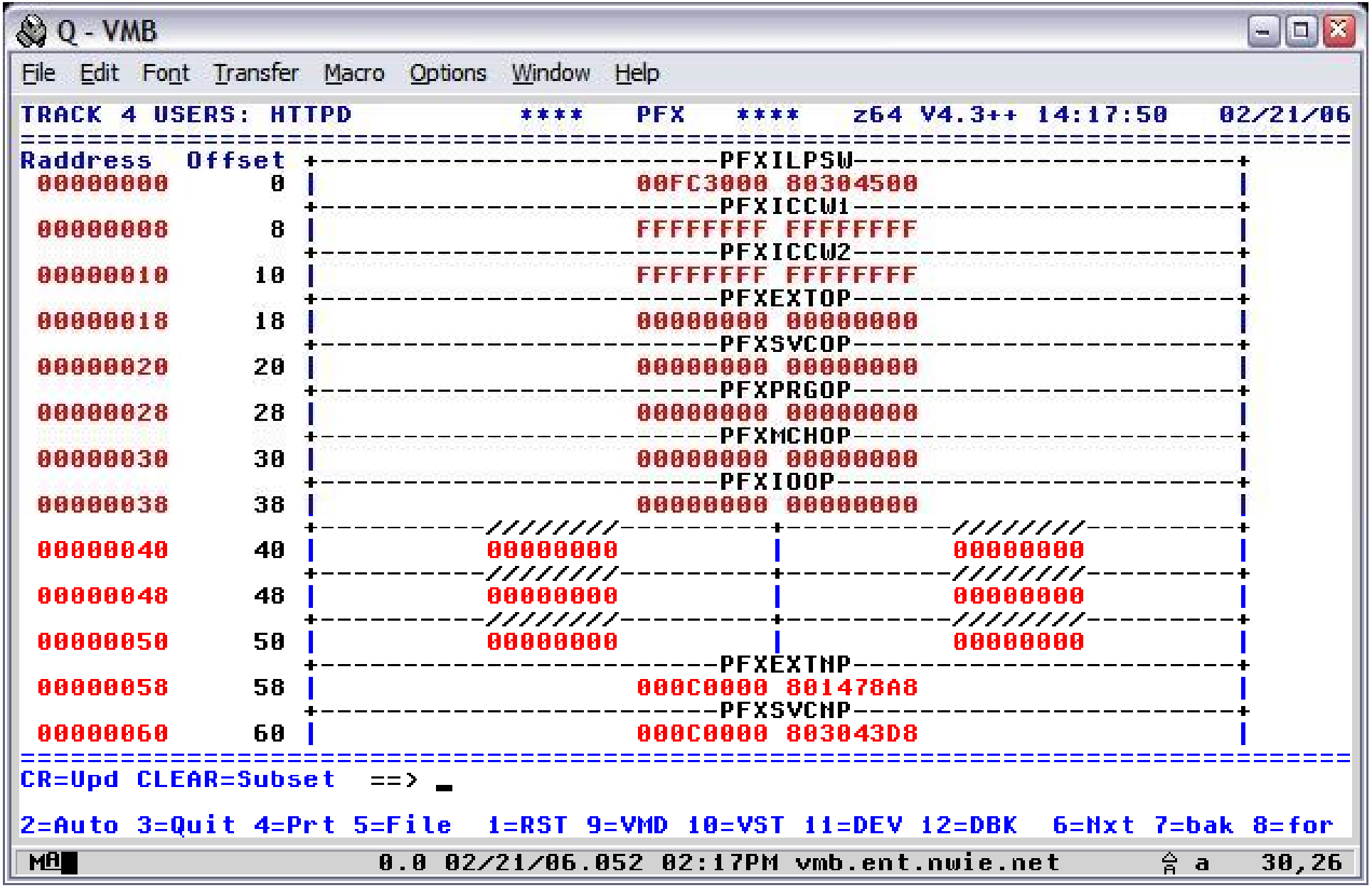


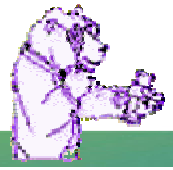
# DEV Panel – all virtual devices for the user.



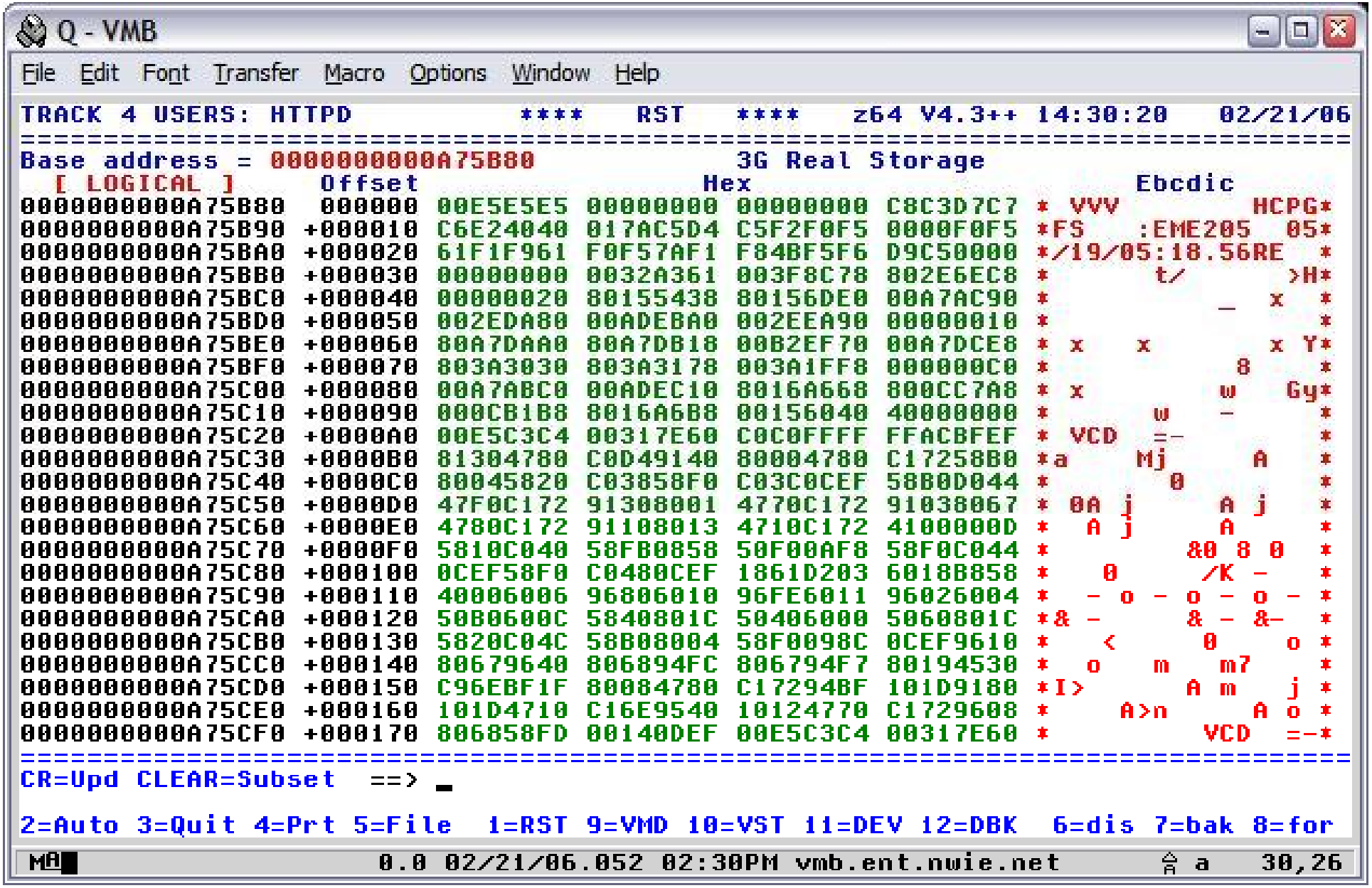



# NAMES Panel – lists all logged on userids

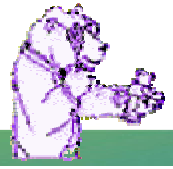


# PFX Panel – prefix page storage mapping



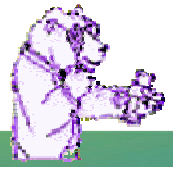

# RST Panel – real storage display



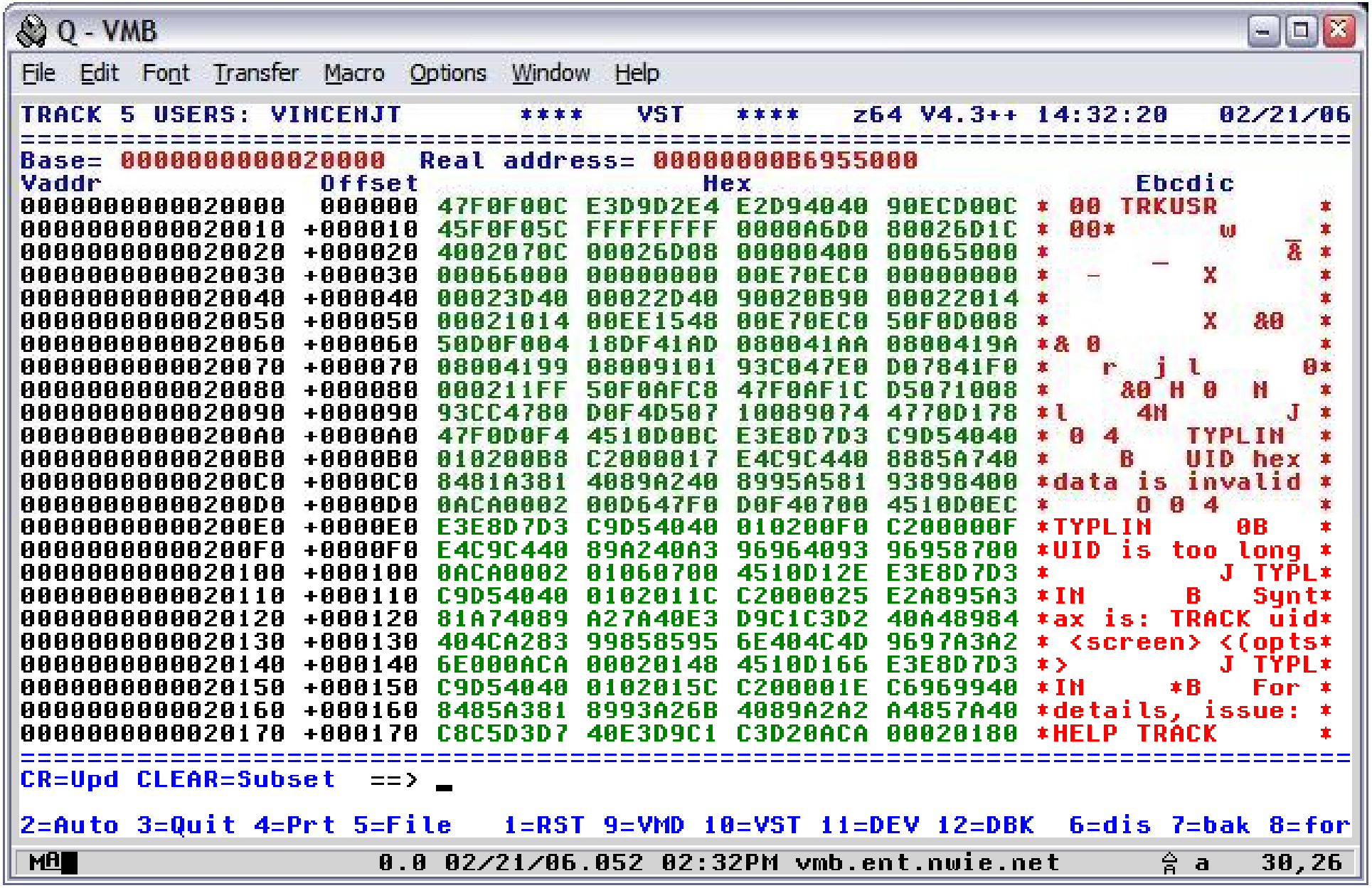
Q - VMB

File Edit Font Transfer Macro Options Window Help

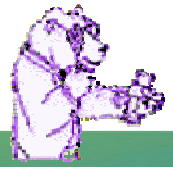
```
TRACK 4 USERS: HTTPD          ****   RST   ****   z64 V4.3++ 14:31:41   02/21/06
================================================================================
Base address = 0000000000A75C2E                3G Real Storage
  [ LOGICAL ]      Hex data      Ins   Operands (in hex)                  Ebcdic   E
0000000000A75C2E    BFEF8130      ICM   R14,B'1111',304(R8)              *    a     *
0000000000A75C32    4780C0D4      BZ    212(,R12)                        *     M    *
0000000000A75C36    91408000      TM    0(R8),X'40'                      * j        *
0000000000A75C3A    4780C172      BZ    370(,R12)                        *    A     *
0000000000A75C3E    58B08004      L     R11,4(,R8)                       *          *
0000000000A75C42    5820C038      L     R2,56(,R12)                      *          *
0000000000A75C46    58F0C03C      L     R15,60(,R12)                     *  0       *
0000000000A75C4A    0CEF          BASSM R14,R15                          *          *
0000000000A75C4C    58B0D044      L     R11,68(,R13)                     *          *
0000000000A75C50    47F0C172      B     370(,R12)                        *  0A      *
0000000000A75C54    91308001      TM    1(R8),X'30'                      * j        *
0000000000A75C58    4770C172      BNZ   370(,R12)                        *    A     *
0000000000A75C5C    91038067      TM    103(R8),X'03'                    * j        *
0000000000A75C60    4780C172      BZ    370(,R12)                        *    A     *
0000000000A75C64    91108013      TM    19(R8),X'10'                     * j        *
0000000000A75C68    4710C172      BO    370(,R12)                        *    A     *
0000000000A75C6C    4100000D      LA    R0,13                            *          *
0000000000A75C70    5810C040      L     R1,64(,R12)                      *          *
0000000000A75C74    58FB0858      L     R15,2136(R11)                    *          *
0000000000A75C78    50F00AF8      ST    R15,2808                         * &0  8    *
0000000000A75C7C    58F0C044      L     R15,68(,R12)                     *  0       *
0000000000A75C80    0CEF          BASSM R14,R15                          *          *
0000000000A75C82    58F0C048      L     R15,72(,R12)                     *  0       *
0000000000A75C86    0CEF          BASSM R14,R15                          *          *
================================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File  1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=dis 7=bak 8=for
MA                    0.0 02/21/06.052 02:31PM vmb.ent.nwie.net      a    30,26
```

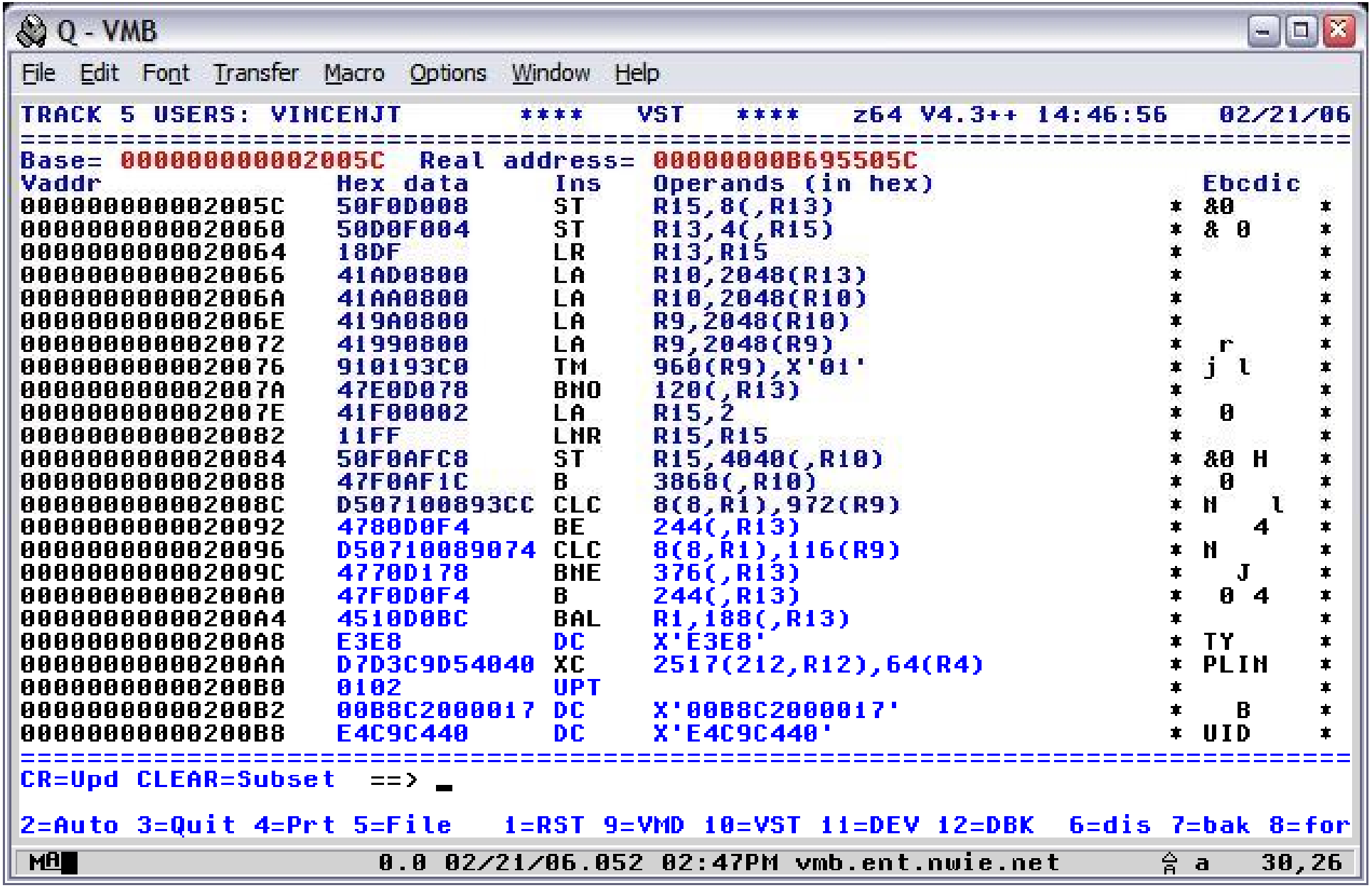


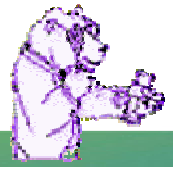
# RST Panel – real storage disassembled



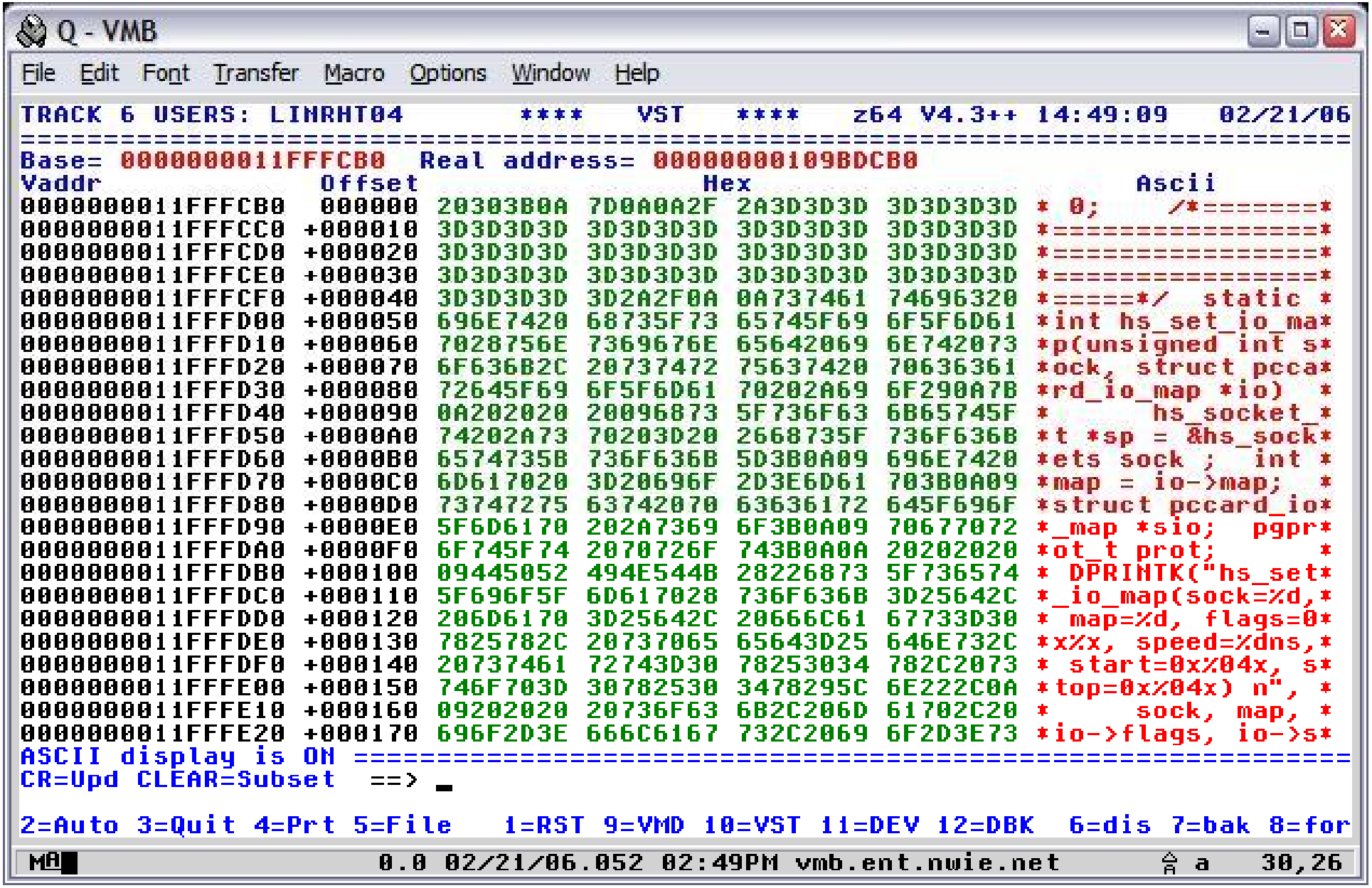

# VST Panel – virtual storage display

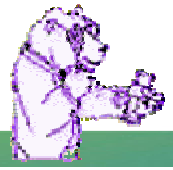




# VST Panel – virtual storage disassembled

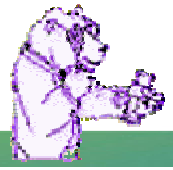


# VST Panel – virtual storage in ASCII

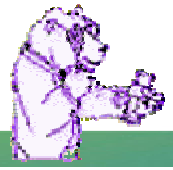
# Using TRACK

- Internal TRACK command
  - Track userid <command> <(<NEW> <DUMP fn>>
  - For example,
    - Track operator
    - Track pvm (new
    - Track pvm format nucon v0
    - Track pvm format nucon v0 (new
    - Track pvm con (new
    - Track pvm ret off
    - T pvm track rscs
    - Track pvm t rscs c
    - Track pvm aup
    - Track * (new
    - Track = VMD



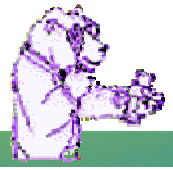
# Using TRACK

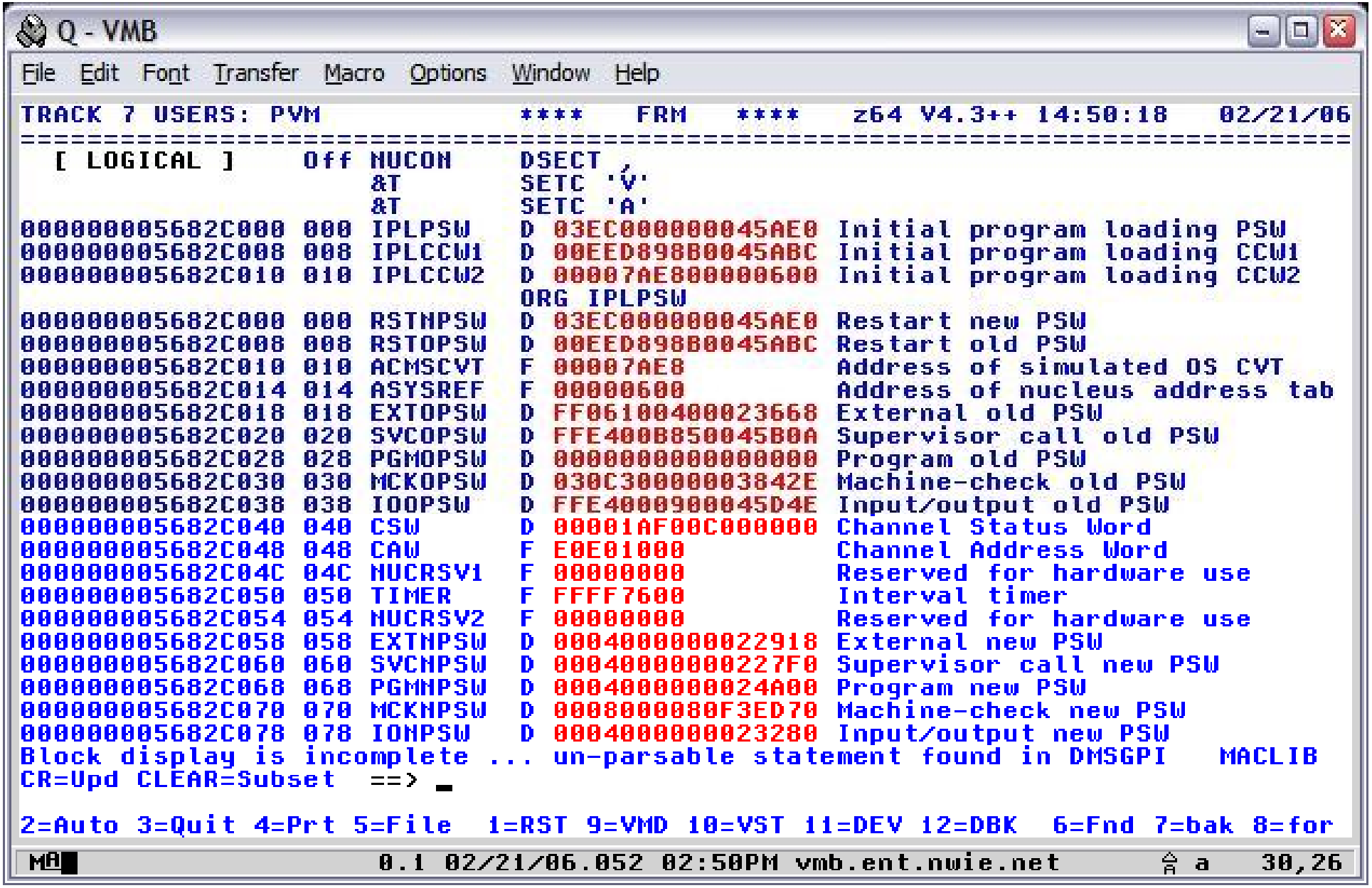
- The internal TRACK command adds the userid to TRACK along with the one(s) you are already TRACKing
  - The “new” option starts a new instance of the tracked user so you can toggle between two different panels for the same userid
  - Enter **T**RACK on the command line by itself to toggle through them all one-by-one
- FORMAT – another of many cool commands and very powerful/helpful
  - Allows you to display storage (real, virtual) as mapped by a DSECT



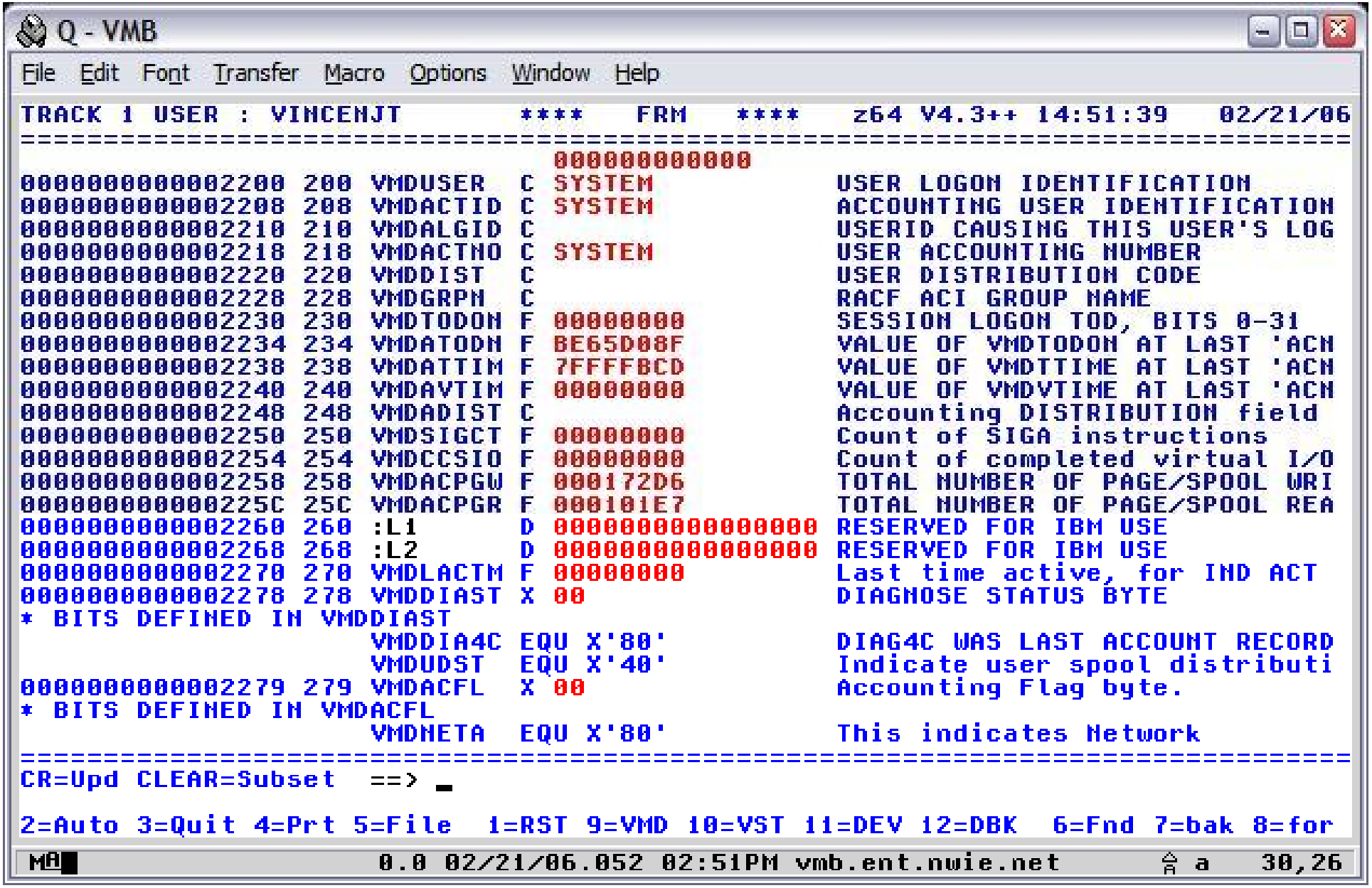
# Using TRACK - FORMAT

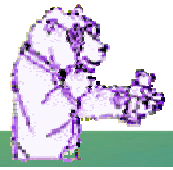
- Read the FORMAT HELP for all the details!
- FORMAT with CP DSECTs
  - You need to have access to MAINT's 193 disk for the MACLIBs
  - Usually best to specify the MACLIB explicitly
    - FORMAT VMDBK 0E8F0000 HCPOM1
- A couple of specific examples…
  - TRACK OPERATOR FORMAT NUCON V0
  - TRACK * FORMAT VMDBK 00002000 HCPOM1
- FORMAT allows a "point and shoot" option (V4.2)
  - Type the FORMAT command you need using a * as the address
  - Place your cursor on the address you want to use that is displayed on the panel
  - Tap Enter
    **Ex:** FORMAT VMDBK * HCPOM1 MACLIB



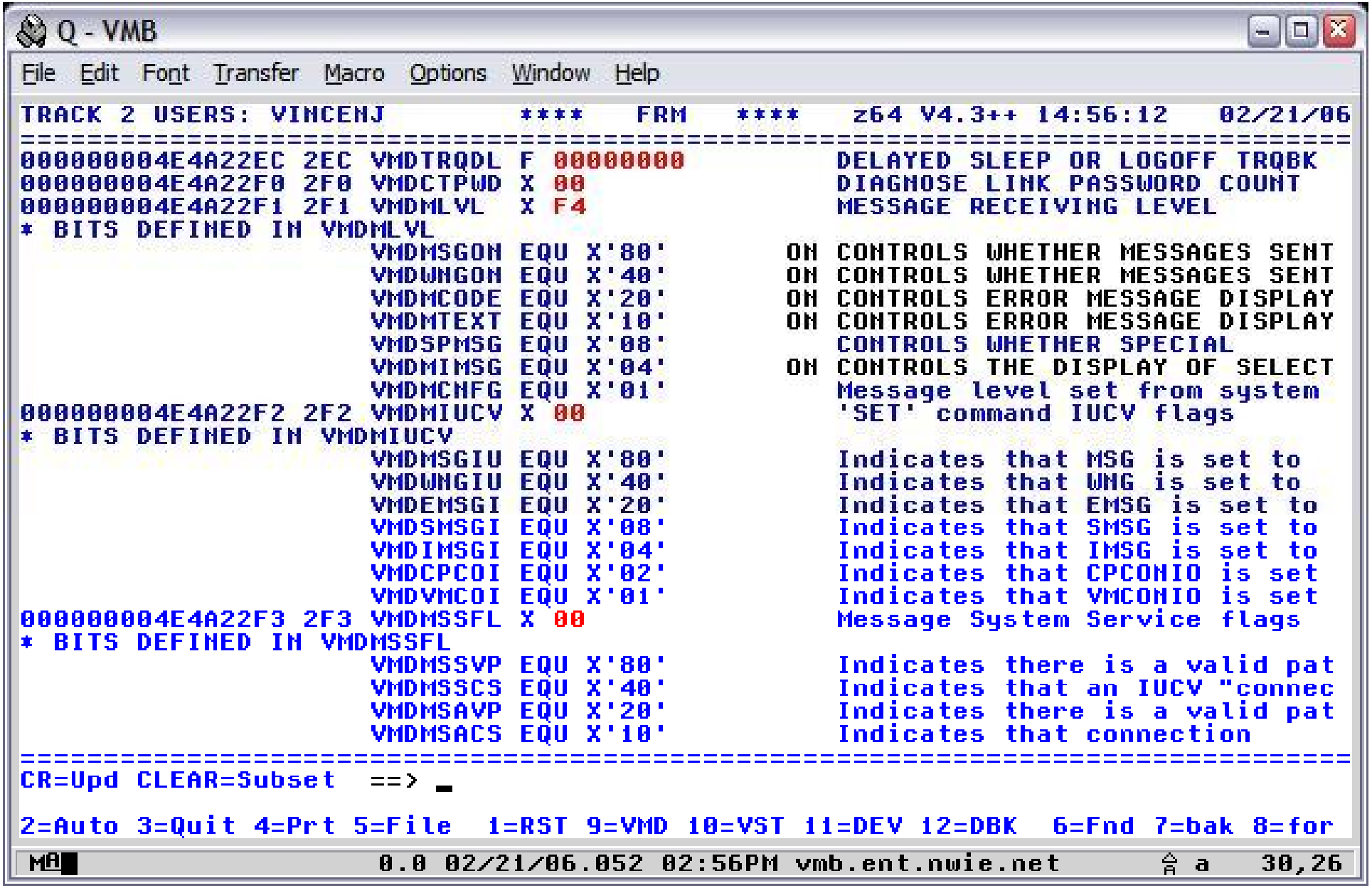



# FORMAT Panel – TRACK PVM FORMAT NUCON V0



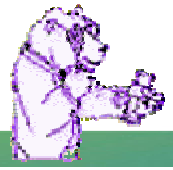
# FORMAT Panel – TRACK * FORMAT VMDBK 00002000 HCPOM1

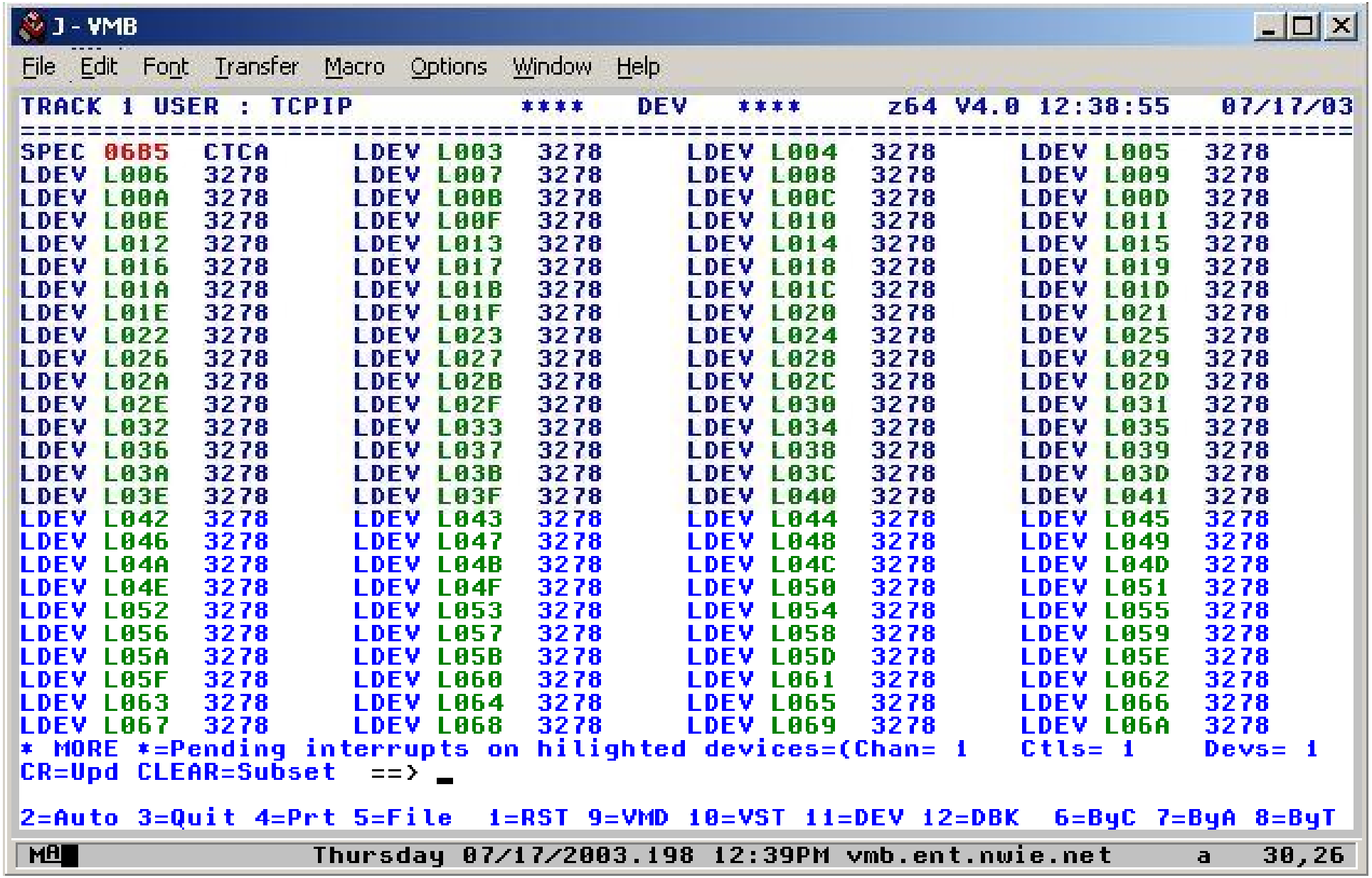


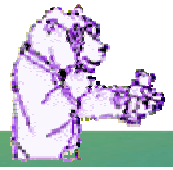

# FORMAT Panel – FORMAT mapping of flags

# Using TRACK

- There is so much more to TRACK
  - Most commands have a ton of options!
- Okay okay… just ONE more then you have to go to bed!
  - DEV – you can check on IO pendings or active
    - track tcpip dev class * io pending
    - track tcpip dev class * io active



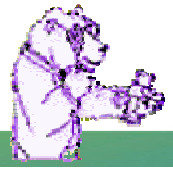


# DEV Panel – showing pending interrupt device(s)

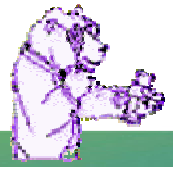
# Advanced TRACK features - Macros

- TRACK Macros
  - You can design your own macros to enhance how TRACK functions
  - Build your own “panels”
- Some of the included TRACK macros:
  - QIUCV displays IUCV paths for user, CP, or a specific CP system service – but this is now an internal panel!
  - STORMAP maps all virtual storage and shows allocated subpools
  - NUCXMAP lists all nucleus loaded extensions



# Advanced TRACK features - Macros

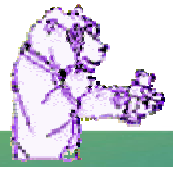
- TRACK Macro sub-commands:
  - **READSTRG <H>address <flag> <length> <( VAR <var>>**
    Read virtual or real storage
  - **TRACKEE <USER|VMDBK> <(VAR <var>>**
    Get the userid or VMDBK of the TRACKed user
  - **REGS <Gn|Rn Cn An Fn> <(VAR <var>>**
    Return the TRACKed user's registers
  - **CMSG string**
    Write contents of string to TRACK command line
  - **NOTE string**
    Write contents of string to the console
    (use REXX Say instead)



# Advanced TRACK features - Macros

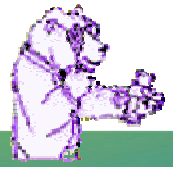
- A TRACK MACRO: MYVMD

```
/* Test the TRACK macro facility */
Numeric Digits 16
'TRACKEE VMDBK'
Say 'TRACKEE returned:' result

Parse Value Diag(8,'Q CPLEVEL') with system .
If system = 'z/VM'
  Then vmduseroff = '200'
  Else vmduseroff = '080'

me = d2x(x2d(result) + x2d(vmduseroff))
Say '... VMDUSER is located at:' me
'READSTRG H'me '8 (VAR VMDUSER'
Say 'VMDUSER is:' vmduser
vmduser = x2c(vmduser)
Say 'VMDUSER field for this VMDBK is "'vmduser'"'
Exit
```



# Advanced TRACK features - Macros

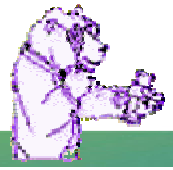
- Two ways to execute
  - TRACK * MACRO MYVMD
    - Inits TRACK environment, calls MYVMD and exits TRACK – no panel is displayed
  - TRACK * MYVMD
    - Inits TRACK, calls MYVMD and remains in TRACK panel
- Either MACRO MYVMD or just MYVMD can also be entered on the TRACK command line. The same actions will occur as above except for MACRO: If more than one userid is being TRACKed, only the current userid is dropped and you remain in the TRACK panel for the other userids.



# Advanced TRACK features - Macros

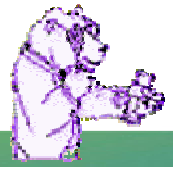
- Running MYVMD TRACK on my userid produces:

```
TRACKEE returned: 6CA00000
... VMDUSER is located at: 6CA00200
VMDUSER is: E5C9D5C3C5D5D140
VMDUSER field for this VMDBK is "VINCENJ "
```



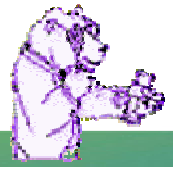
# Advanced TRACK features – FILE opt

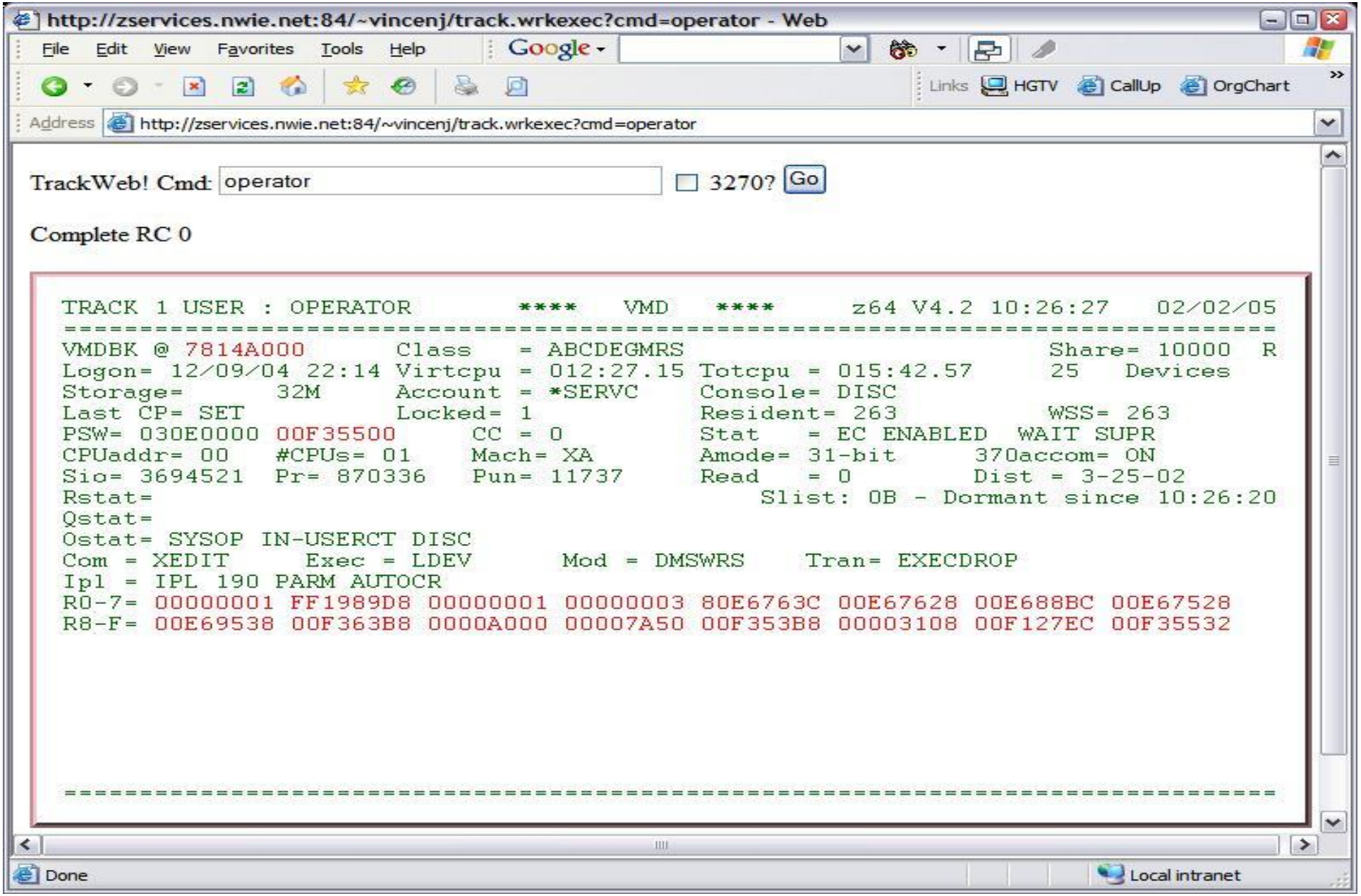
- New FILE option in TRACK z64 V4.2
  **TRACK userid track-parms ( FILE**
  - Saves the contents of the screen into a file on the A disk called “userid TRACKFIL” and then exits TRACK
  - No TRACK panel(s) are displayed
  - Data written is 60 lines by 80 LRECL and contains the 3270 data stream used to display the panel; **no** non-display character translations are done
  - Data is appended to the file if the file exists already
  - File is automatically CLOSED when TRACK exits

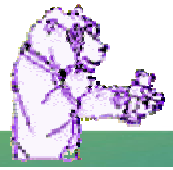


# Advanced TRACK features – FILE opt

- New FILE option – why?
  - What could you possibly do with this file?
  - Just imagine the possibilities...



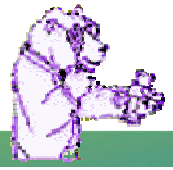


# Make TRACK available via the Web!

# Keeping VM on TRACK!

- TRACK is a very powerful tool for the VM Systems Programmer
- Extremely handy for debugging virtual machine or system problems
- Makes displaying virtual and real storage/devices a snap
- Once you get used to it, navigating through and checking key areas in a userid or storage area for problems will become second nature
- Use it!



# Contact Info

Light travels faster than sound, that's why people seem bright until you hear them...


**Jim Vincent**
**Senior z/VM Systems Programmer**

**Phone: (614) 249-5547**
**Internet: James.Vincent@nationwide.com**