
PLEASE STAND BY

OpenVMS on Integrity Servers Part I

Thomas Siebold
Sr. Technology Consultant, HP

Thomas Siebold

Sr. Technology Consultant

Transition Engineering & Consulting

thomas.siebold@hp.com

„The language and cultural barrier…“

- This is a ‚mobile phone‘
- …but in Germay it is called a ‚handy‘
- …but in other countries a ‚handy‘ is a …..

Agenda

- OpenVMS on Itanium®
  - Schedule
  - Status
  - Application/ISV migration





Schedule

Change of Name

OpenVMS on Itanium®

Will be called

„**hp OpenVMS Industry Standard 64**“

(Official Name)

or

„**OpenVMS I64**“

(Informal Name)

HP OpenVMS the Road to Itanium®

OpenVMS V8.2 Production Quality Release
(Alpha & Integrity) 2H2004

Mixed Alpha Integrity Superdome Cluster
January, 2004

16Processor System Boot
January, 2004

OpenVMS V8.1 Evaluation Release
December, 2003

Runs in a Superdome Cell
November, 2003

1st ISV Applications ported
August, 2003

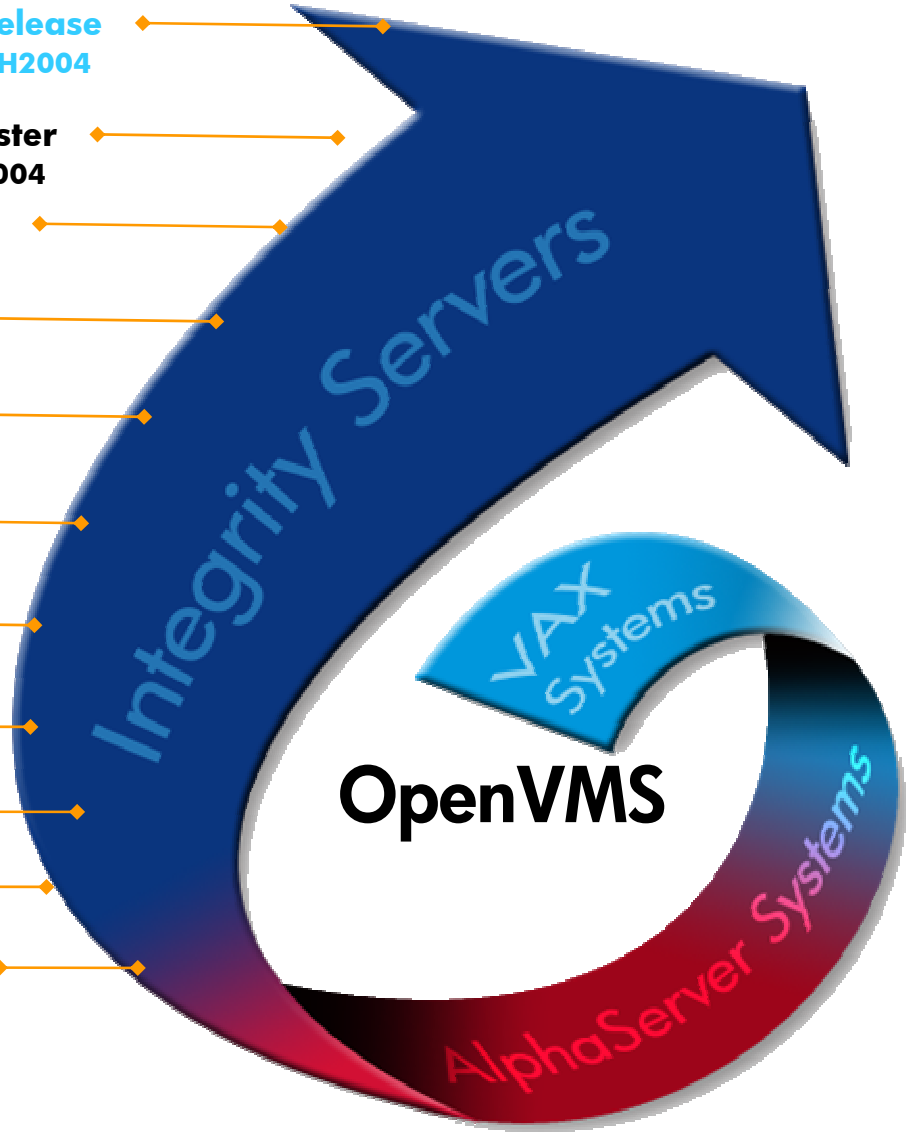
OpenVMS V8.0 in DSPP
August, 2003

OpenVMS V8.0 Evaluation Release
June 30, 2003

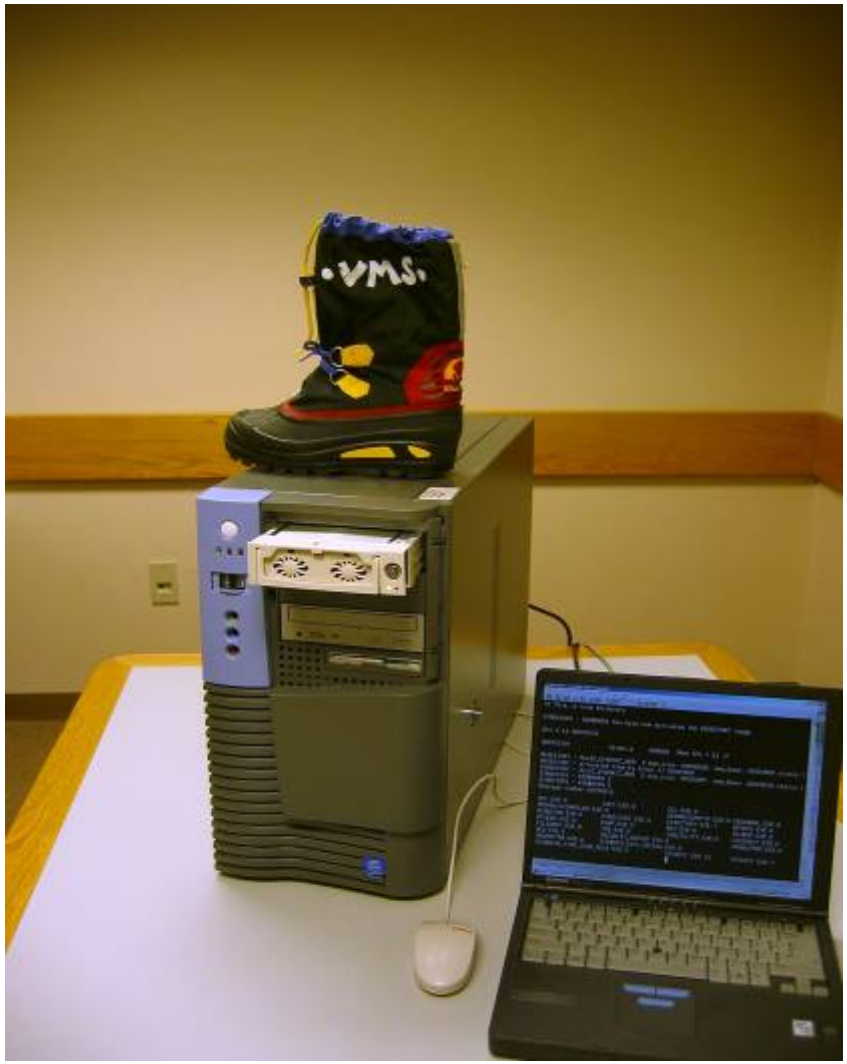
1st Application Port & Mixed Cluster
May 15, 2003

Boot to rx 2600 server
March 17, 2003

1st Boot to Itanium® system
January 31, 2003



OpenVMS on Itanium® -- 31. Januar 2003 15:31



June 25, 2003

(B) TELNET (deion) - PowerTerm 525

File Edit Terminal Communication Options Script Help

View of Cluster from system ID 51202 node: DEION 19-MAY-2003 12:06:17

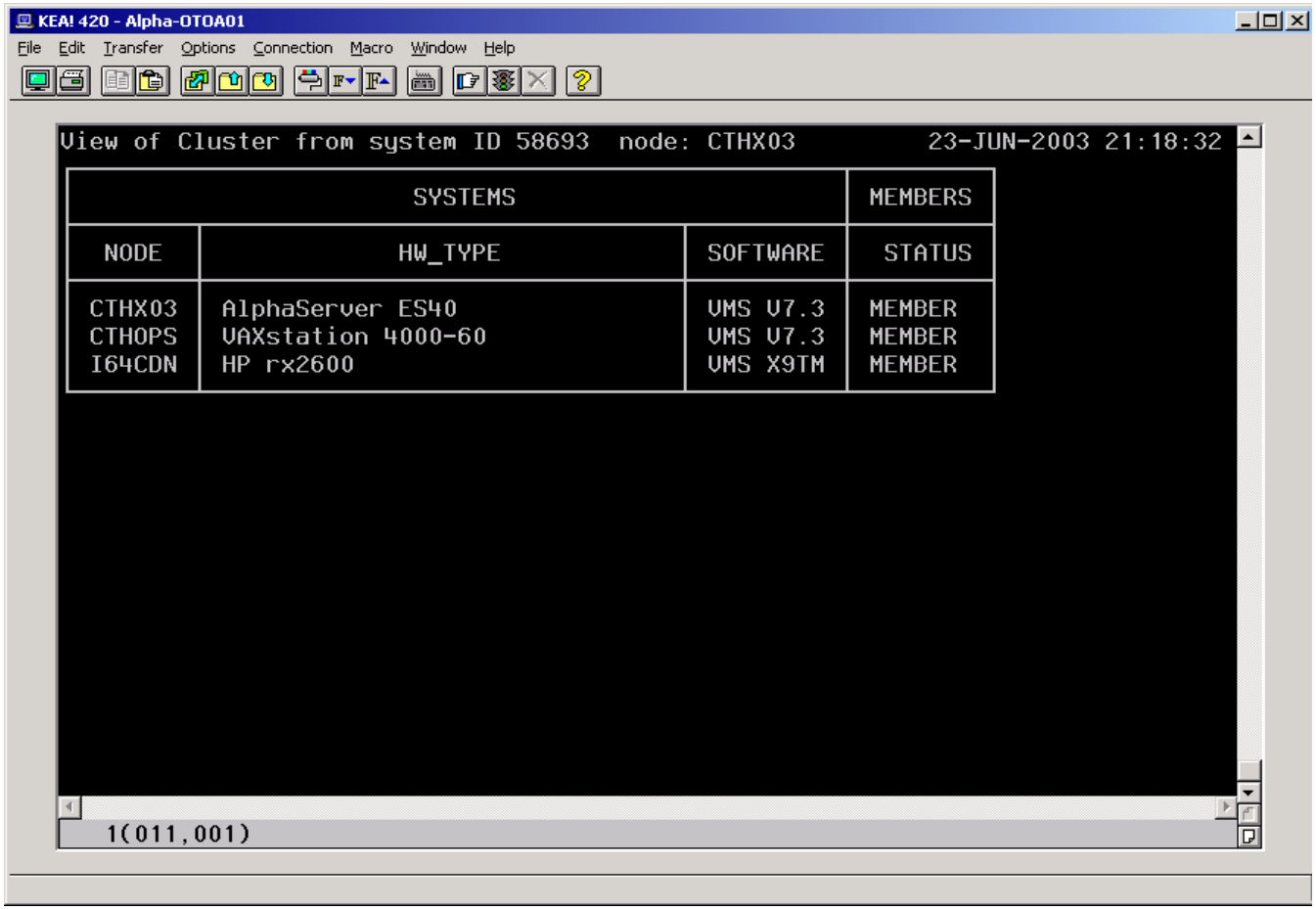
SYSTEMS			MEMBERS	CONNECT
NODE	HW_TYPE	SOFTWARE	STATUS	LOC_PROC_NAME
DEION	hp AlphaServer GS1280 7/1150	VMS V7.3	MEMBER	SCS$DIRECTORY MSCP$TAPE MSCP$DISK VMS$SDA_AXP VMS$VAXcluster SCA$TRANSPORT PATHWORKScluste
IA64	Generic Itanium Platform	VMS X9SG	MEMBER	MSCP$DISK VMS$VAXcluster

CLUSTER						
CL_EXP	CL_QUORUM	CL_VOTES	QF_VOTE	CL_MEMBERS	FORMED	LA
1	1	1	NO	2	18-MAY-2003 18:07	19-

F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12

VT420-7 24:1 Caps Hold On Line

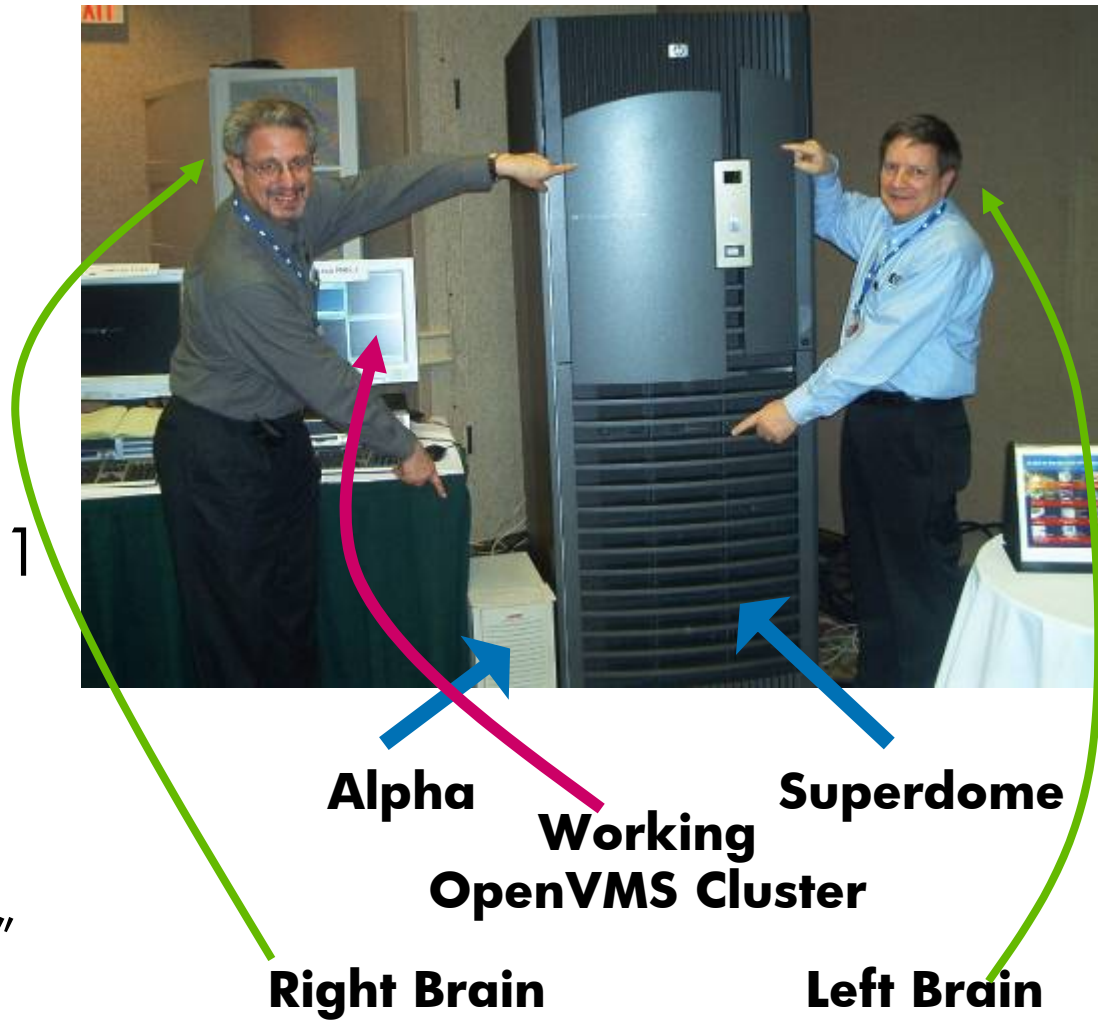
OpenVMS VAX-Alpha-IA64 Cluster Demo



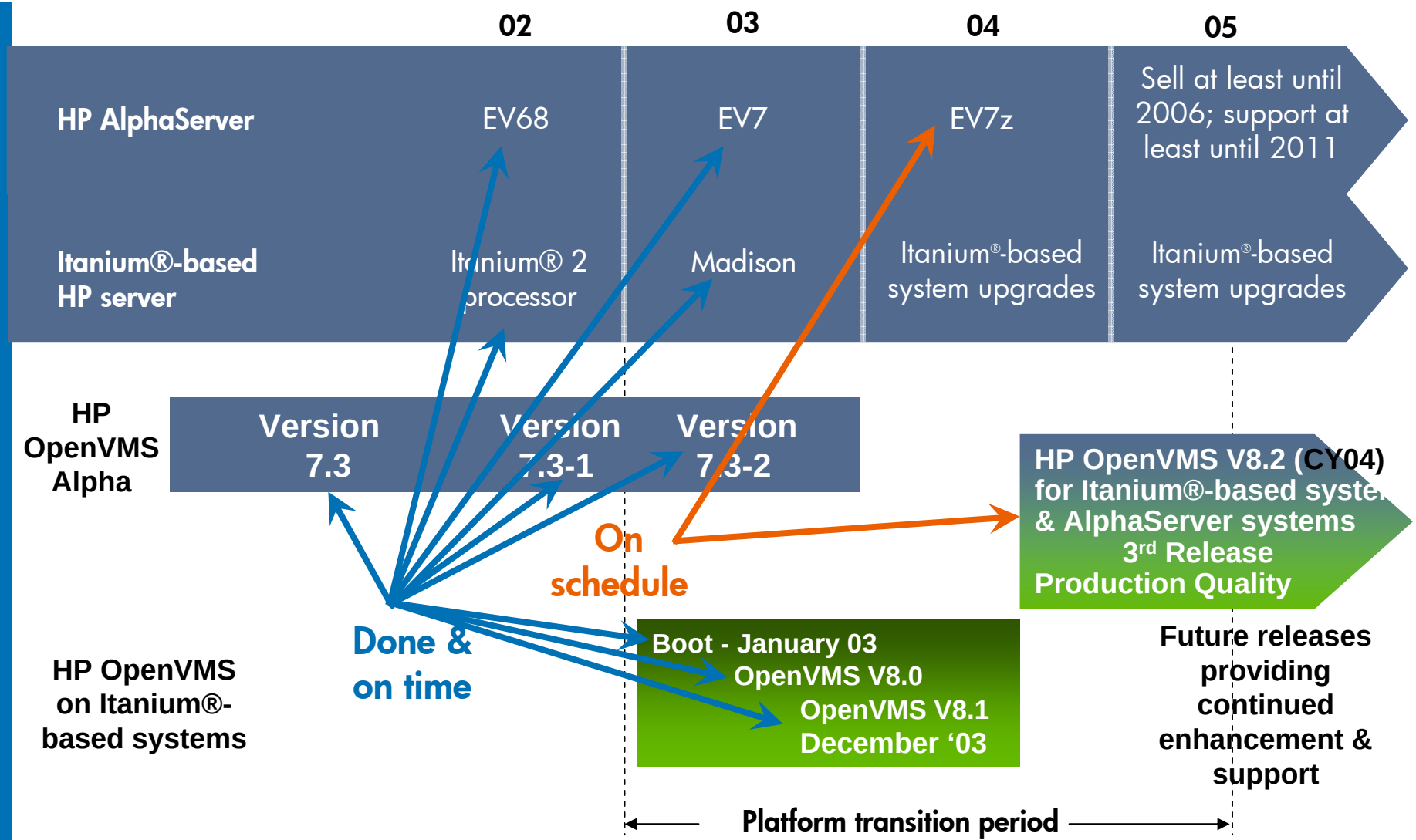
Even though clustering VAX with Alpha-IA64 will not be supported, Engineering is not doing anything to prevent it from working. The above proves it now works. Btw - Clustering three totally different HW architectures with a fully shared read-write active-active-active cluster file system with one OS (OpenVMS) is way cool. ☺

OpenVMS @ Analyst Summit 1/13/04

- OpenVMS on all multi-OS slides
- Rich Marcello discussed OpenVMS Integrity progress and partner support
- Demo with OpenVMS V8.1 and Superdome/Alpha cluster
- Some reactions:
  - "Cool"
  - "Two different OS versions?"
  - "Impressive"
  - "It's good to see OpenVMS back on the front burner."



OpenVMS Roadmap



OpenVMS on Integrity Servers 2005 Coming Attractions

- V8.2 Code Freeze: April (planned)
- V8.2 External FT: June (planned)
- V8.2 FRS: Q4'CY04 (planned)


What is being ported ??
And how ??

What are porting and How?

- Single source code base to produce the Alpha and Intel® Itanium® architecture variants
  - About 95% of the code is common
  - Support for Itanium® architecture added to OpenVMS AlphaServer code base
  - Releases created from the same sources for both architectures
  - All non-hardware dependent and performance improvements to be incorporated into both versions without multiple changes to the source code and to minimize the time required to perform qualification testing.
- The first Itanium® architecture release will reflect on-going OpenVMS development work
- Allows ISVs and end-user developers to continue using their current and future Alpha systems while migrating to the future Itanium® platforms. Integrating Integrity Servers will be cost effective
- OpenVMS is made more portable and maintainable by replacing VAX assembler
- OpenVMS is made more open to exchanging code with other systems by using new standards

Current Itanium Porting Status

- Native Tools
  - C, Bliss, Cobol, Fortran, DECset, SWS (Apache)
  - Linker, SDA, ….
  - GNV, Kerberos, Freeware Tools, ….
- Console
  - Serial line
  - Management Port (no graphics, keyboard, or mouse yet)
- Booted on
  - I2000, rx2600 (McKinley & Madison), zx2000 (McKinley), rx4640(Madison), rx1600(Deerfield)

Current Itanium Porting status

- What is not yet working
  - Edit/Teco
  - Delta Debugger
  - System Code Debugger (SCD)
  - Security Server
  - Registry Server
  - ACME Server
  - Shadowing
  - Cluster Satellite Booting
  - Java

Challenges

Big Challenges for the Base OS

- No Alpha Console 
  - Booting
  - Device Discovery
  - Interrupts
  - TLB miss handler
- No Alpha PALcode 
  - VAX Queue Instructions
  - VAX Registers
  - IPL and mode change
- Different primitives in CPU 
  - Register Conventions
  - Exception Handling
  - Atomic Instructions
  - Process Context
- Plus, we decided to change 
  - calling standard
  - object language
  - image format

How do we boot VMS ?

The Extensible Firmware Interface

- A new standard from Intel replacing the BIOS
- EFI firmware on the system
  - Includes a user interface called the “shell”
  - EFI commands native in the firmware
  - Interface to the system hardware
- Consists of three major components:
  1. The EFI Firmware core
  2. EFI System Partition (ESP)
     - The OS creates the EFI System Partition
     - OS loader image is in the EFI System Partition
     - Value-add software utilities/tools may be added during install or later
  3. EFI boot manager
     - OS Boot Loader menu

Extended Firmware Interface (EFI)

- An Interface between the Operating System and the platform firmware
- Provides for multiple CPU architectures support
- EFI uses disk storage with a specific FAT file system, identified by a specific FAT type and NVRAM storage
- Introduces a new GUID Partition Table (GPT)
  - [GUID = Globally Unique Identifier]
- Allows legacy MBR methods (boot and partitioning)

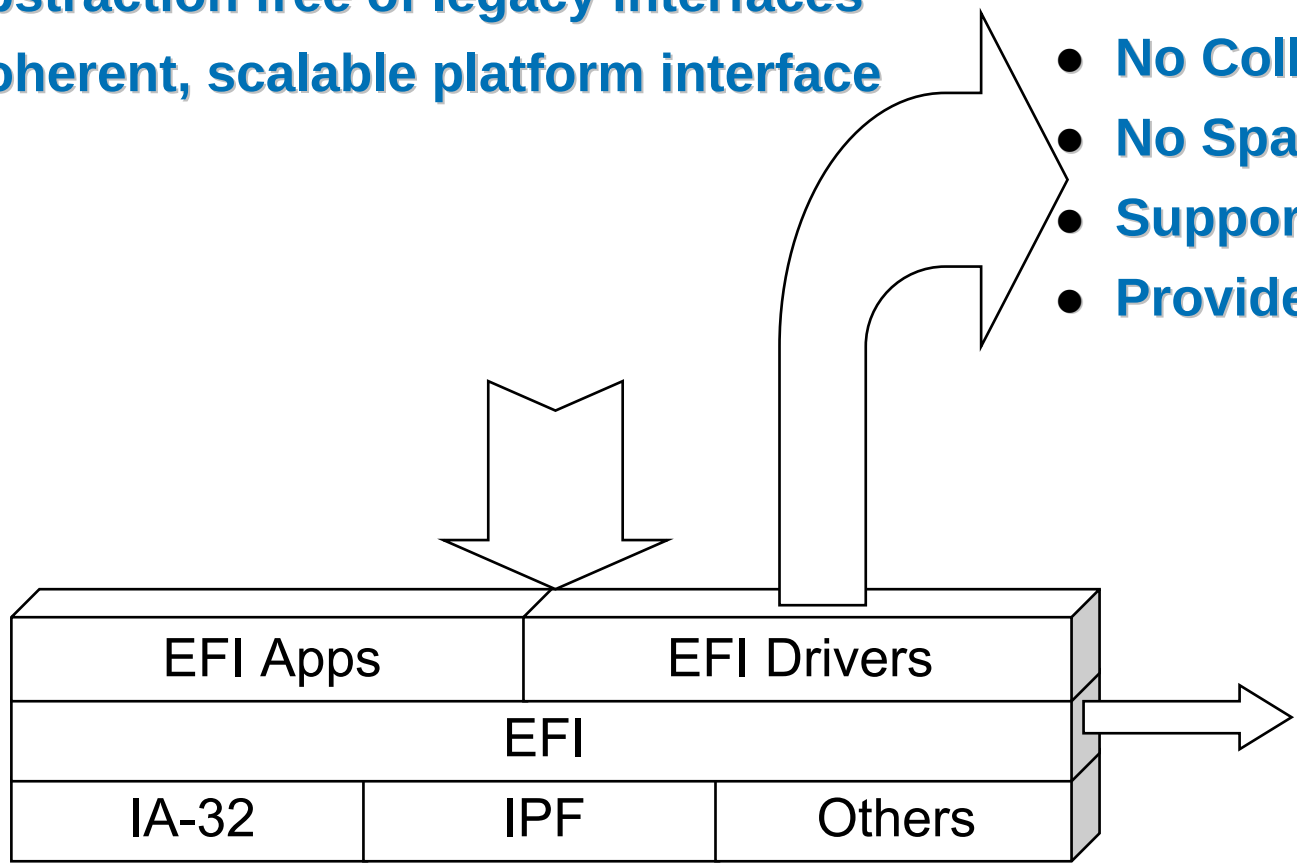
Terminology

- EFI – Extensible Firmware Interface
  - Moving Target - Many code drops from Intel
  - Strategy change as little as possible
- Boot Manager
  - One and Only One User Menu
- Shell
  - Command line Interface (like DOS)
- POSSE – Pre-OS Setup Environment
  - HP's Value add in the Shell & Boot Manager.

Benefits of EFI abstraction

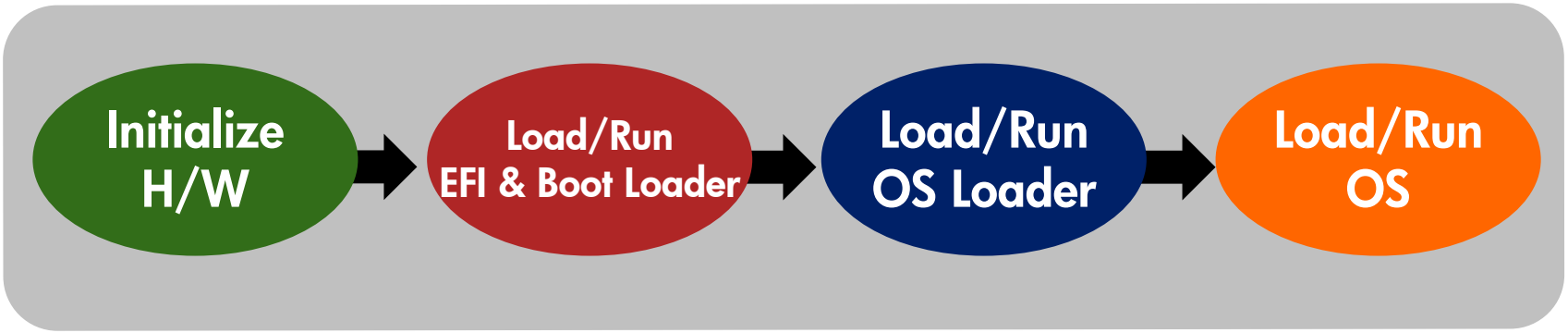
- **Abstraction of OS from firmware**
- **Abstraction free of legacy interfaces**
- **Coherent, scalable platform interface**

- **No Collision**
- **No Space Limitation**
- **Support Speedy Boot**
- **Provide Drivers to OS**

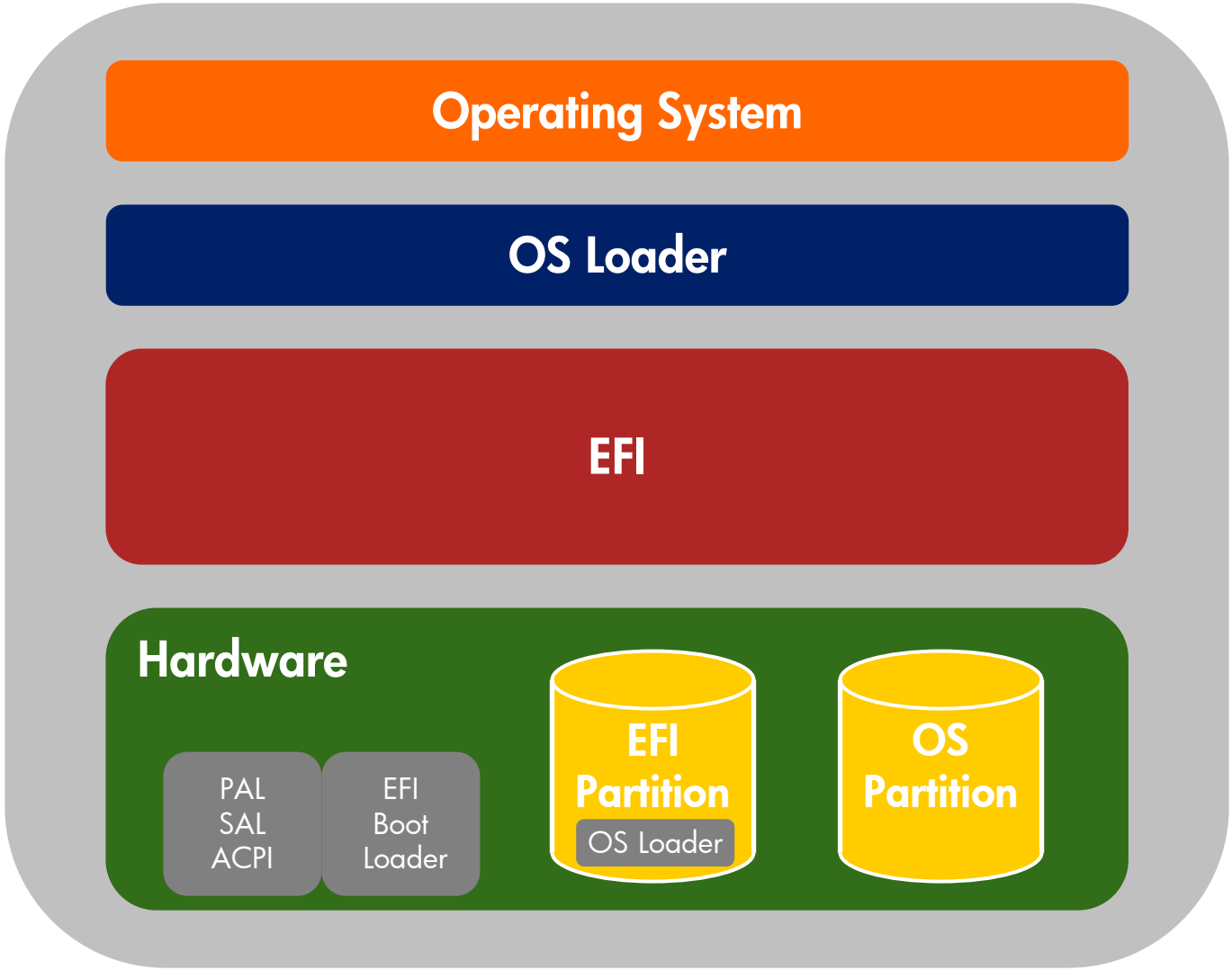


EFI Operational Model

- Boot Starts with Hardware Initialization
- Continues With a Sequence of Loads
  - Each successive loader is a bit “smarter”
- Culminates in the Loading of an OS



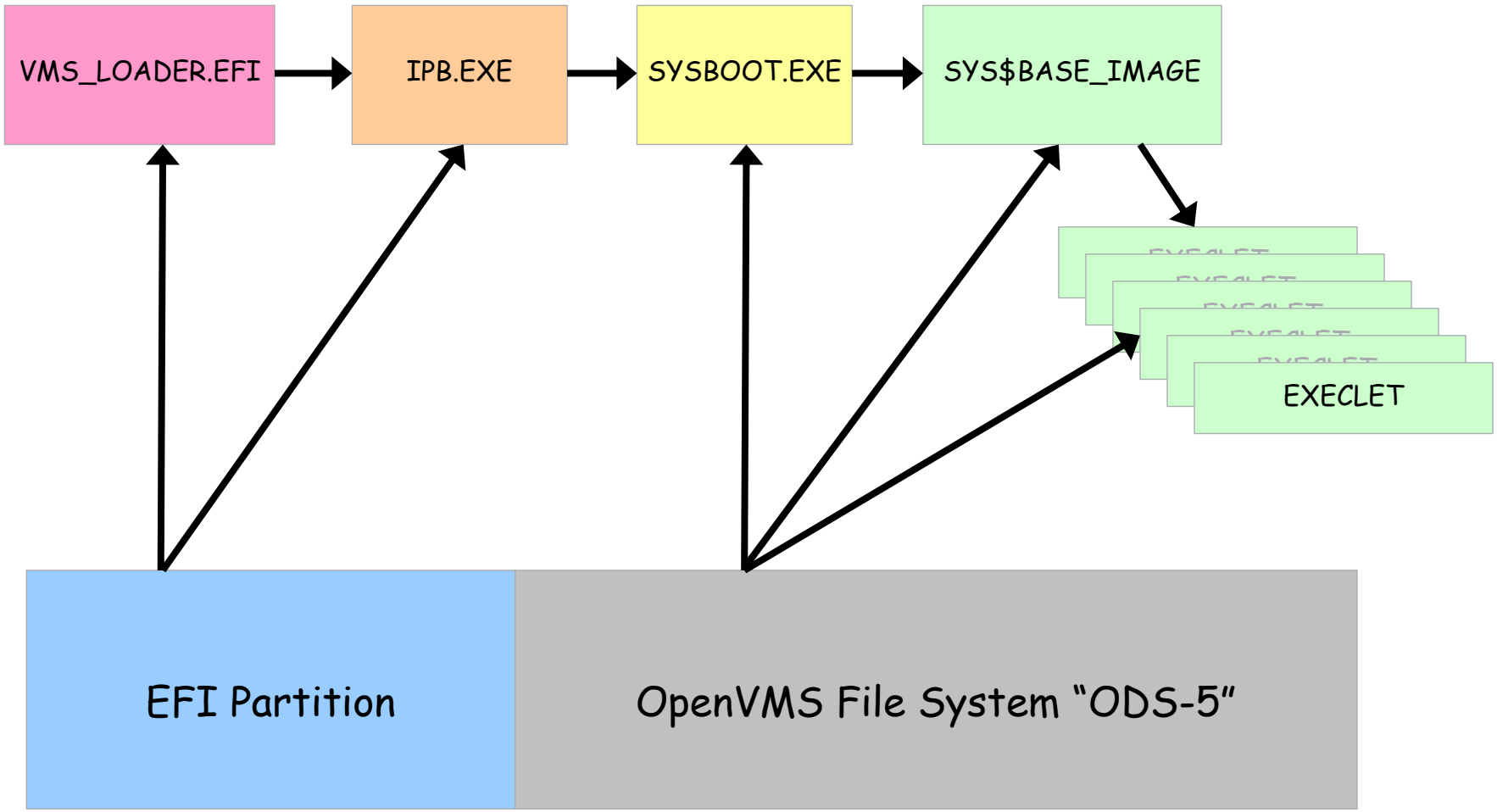
EFI Structural Model

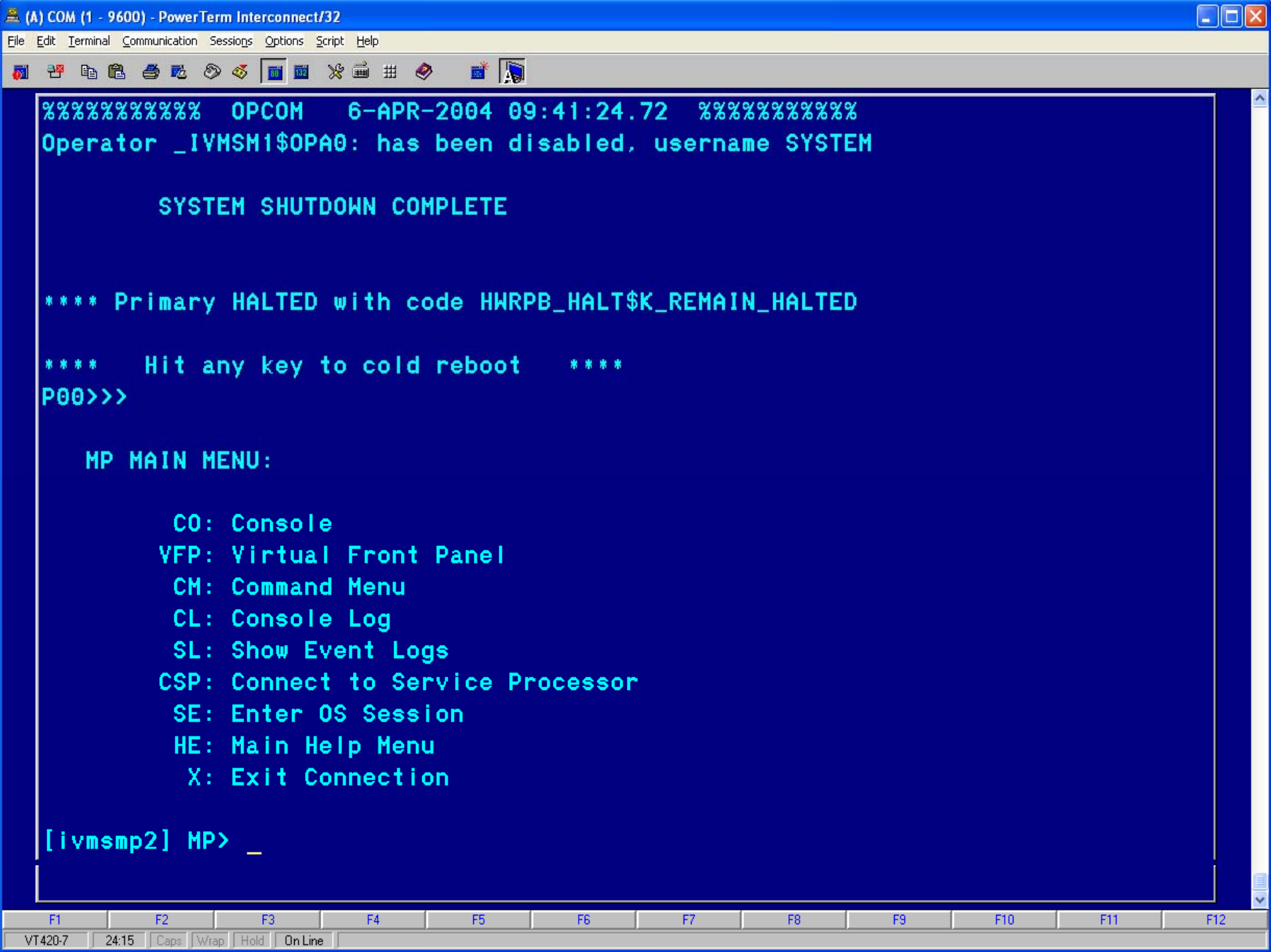


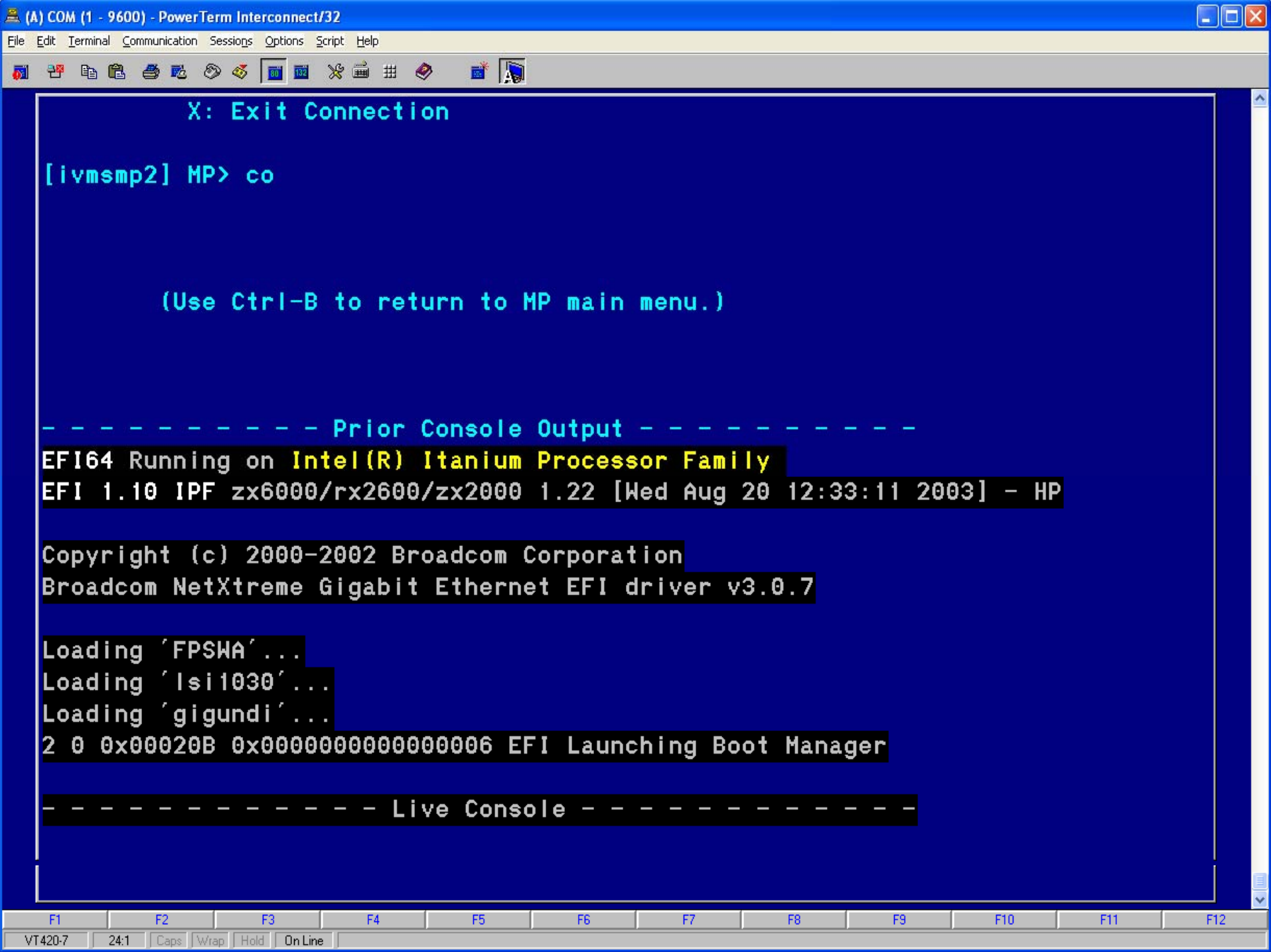
EFI directory structure

- An EFI system partition on a disk contains an EFI directory in the root directory `/EFI`
- Vendors use subdirectories to store their OS loaders and applications.
- On hp OpenVMS I64 systems, the boot loader filename is `vms_loader.efi` and is located in `fs0:\efi\vms`

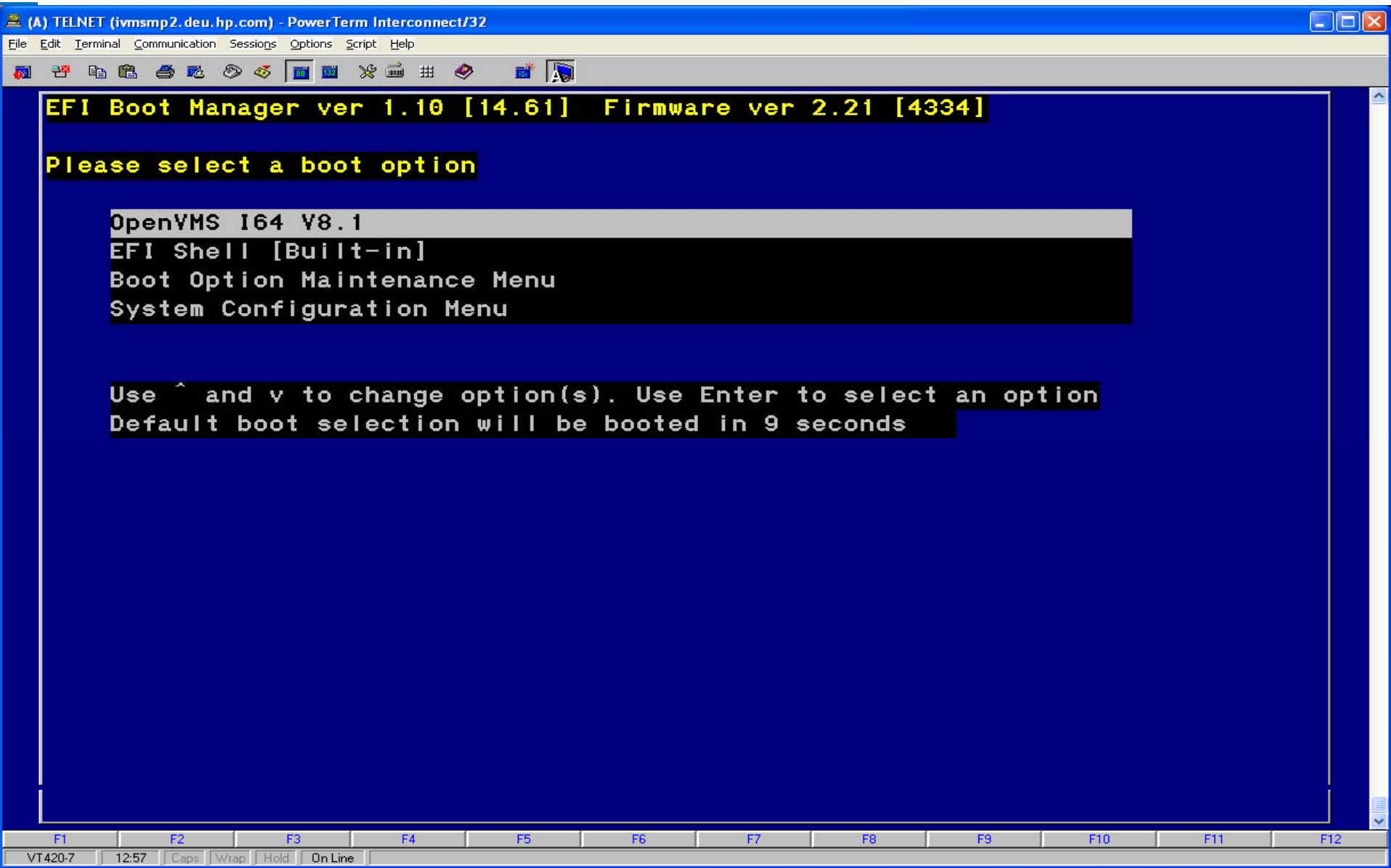
OpenVMS Booting On IPF



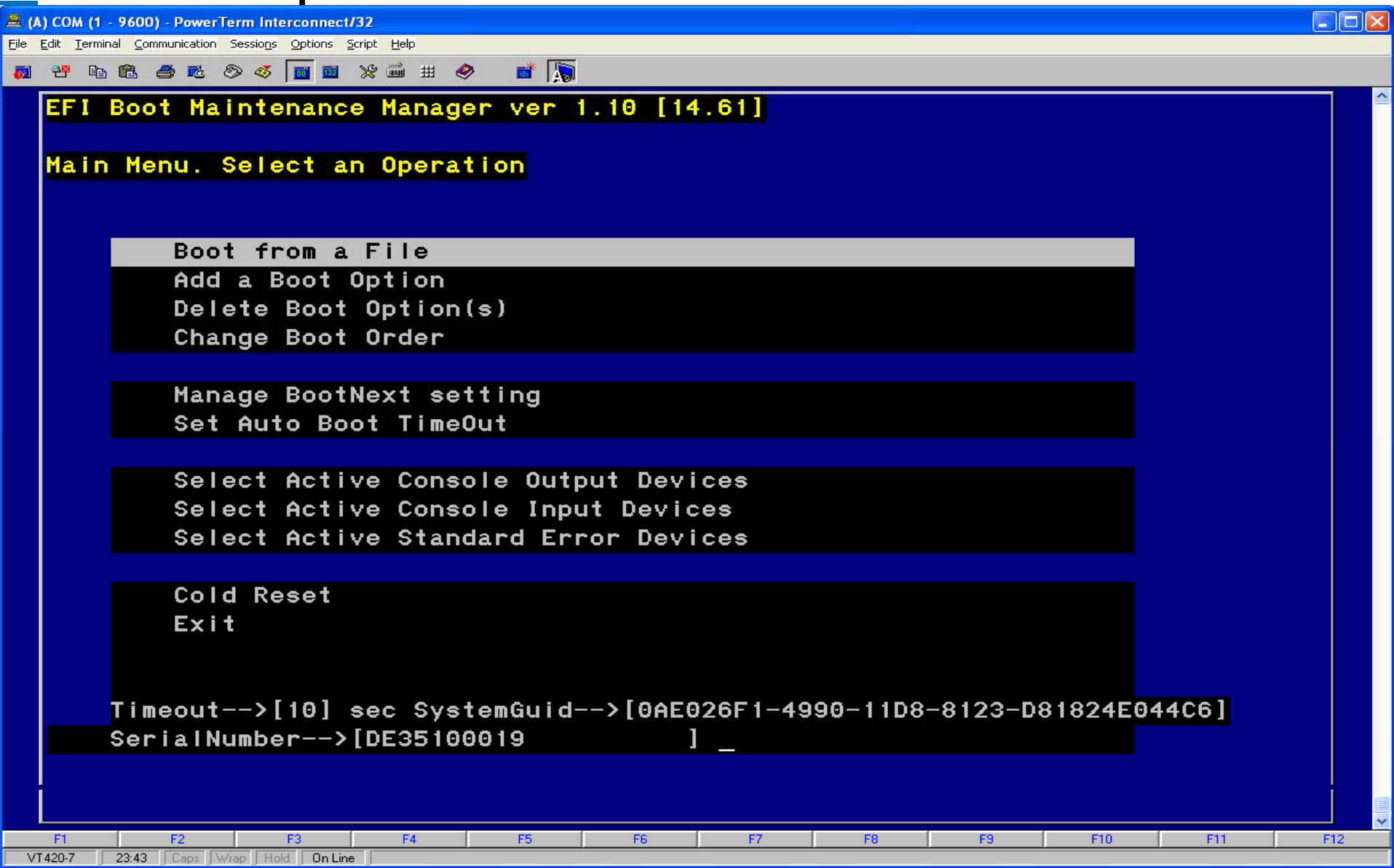

(A) COM (1 - 9600) - PowerTerm Interconnect/32
File Edit Terminal Communication Sessions Options Script Help
%%%%%%%%%% OPCOM   6-APR-2004 09:41:24.72 %%%%%%%%%%
Operator _IVMSM1$OPA0: has been disabled, username SYSTEM
SYSTEM SHUTDOWN COMPLETE
**** Primary HALTED with code HWRPB_HALT$K_REMAIN_HALTED
**** Hit any key to cold reboot ****
P00>>>
MP MAIN MENU:
CO: Console
VFP: Virtual Front Panel
CM: Command Menu
CL: Console Log
SL: Show Event Logs
CSP: Connect to Service Processor
SE: Enter OS Session
HE: Main Help Menu
X: Exit Connection
[ivmsmp2] MP> _
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
VT420-7 24:15 Caps Wrap Hold On Line


(A) COM (1 - 9600) - PowerTerm Interconnect/32
File Edit Terminal Communication Sessions Options Script Help
X: Exit Connection
[ivmsmp2] MP> co
(Use Ctrl-B to return to MP main menu.)
- - - - - - - - - - Prior Console Output - - - - - - - - - -
EFI64 Running on Intel(R) Itanium Processor Family
EFI 1.10 IPF zx6000/rx2600/zx2000 1.22 [Wed Aug 20 12:33:11 2003] - HP
Copyright (c) 2000-2002 Broadcom Corporation
Broadcom NetXtreme Gigabit Ethernet EFI driver v3.0.7
Loading 'FPSWA'...
Loading 'lsi1030'...
Loading 'gigundi'...
2 0 0x00020B 0x0000000000000006 EFI Launching Boot Manager
- - - - - - - - - - - - - Live Console - - - - - - - - - - - - -
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
VT420-7 24:1 Caps Wrap Hold On Line

EFI Selection Menu



Boot Options Maintenance Menu



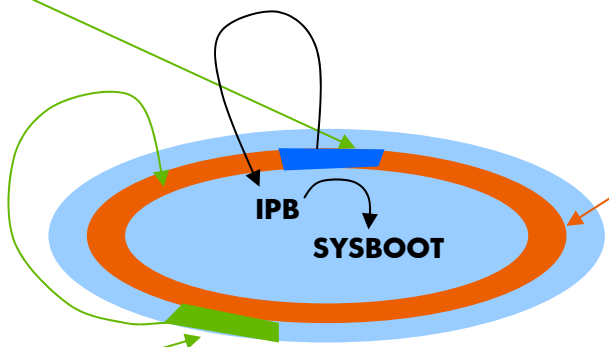
Hybrid Disk Format

- EFI requires GUID Partition Table (GPT) disk format
  - GUID = Globally Unique Identifier
- EFI requires one FAT32 partition
- VMS requires ODS Files-11 disk format
- VMS does not (yet) support partition disks
- Both formats co-exist, independent of each other
- EFI console view
  - GPT format with one 48Mb FAT32 partition
  - Remaining space unallocated
- VMS view
  - ODS Files-11 format disk
  - 48Mb container file allocated for FAT32 partition
- CD-ROMs use ISO9660 format instead of GPT

ODS-2 disk

OS Loader
VMS_LOADER.EFI

Seen from EFI:
FAT32 partition

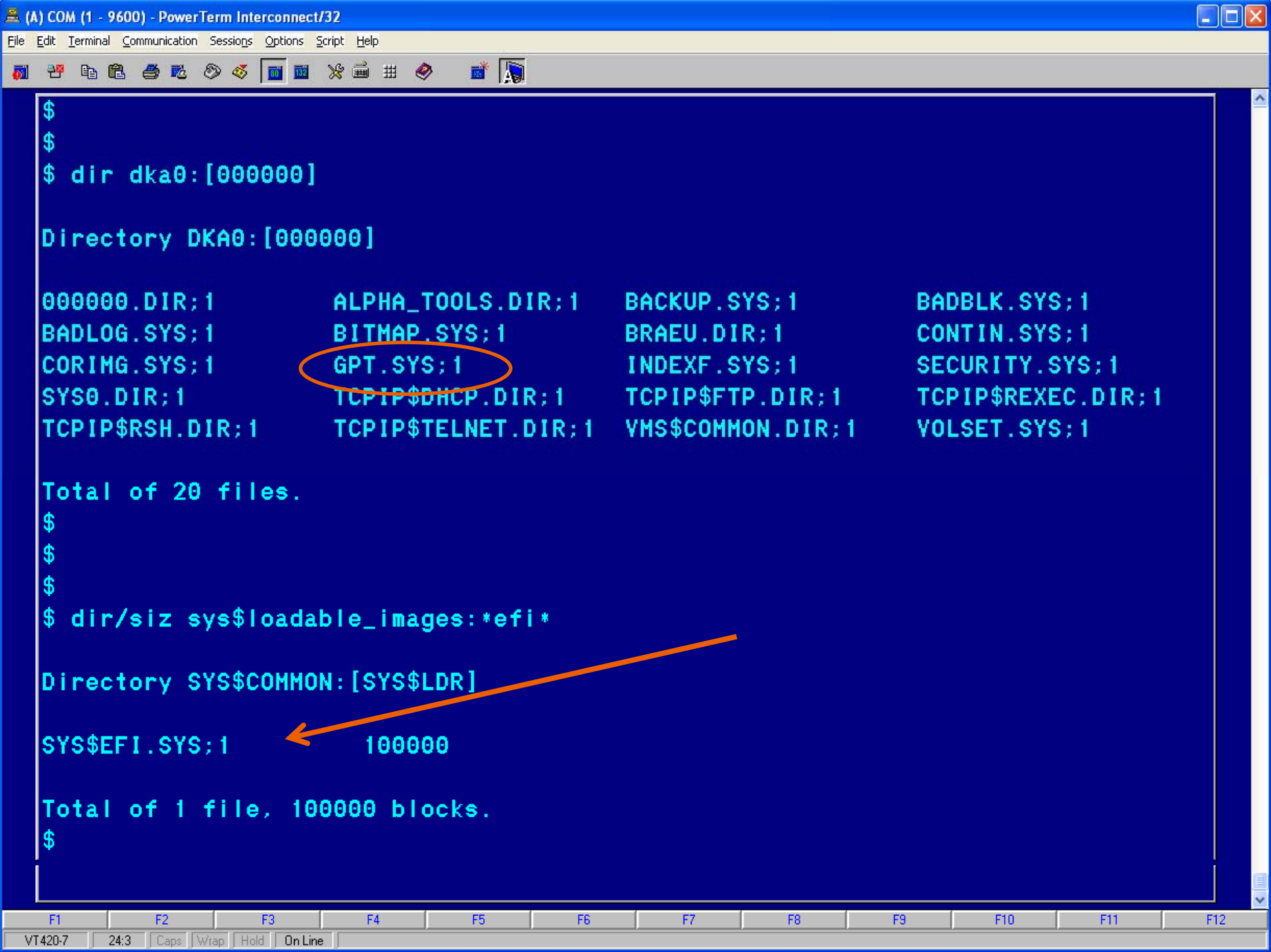


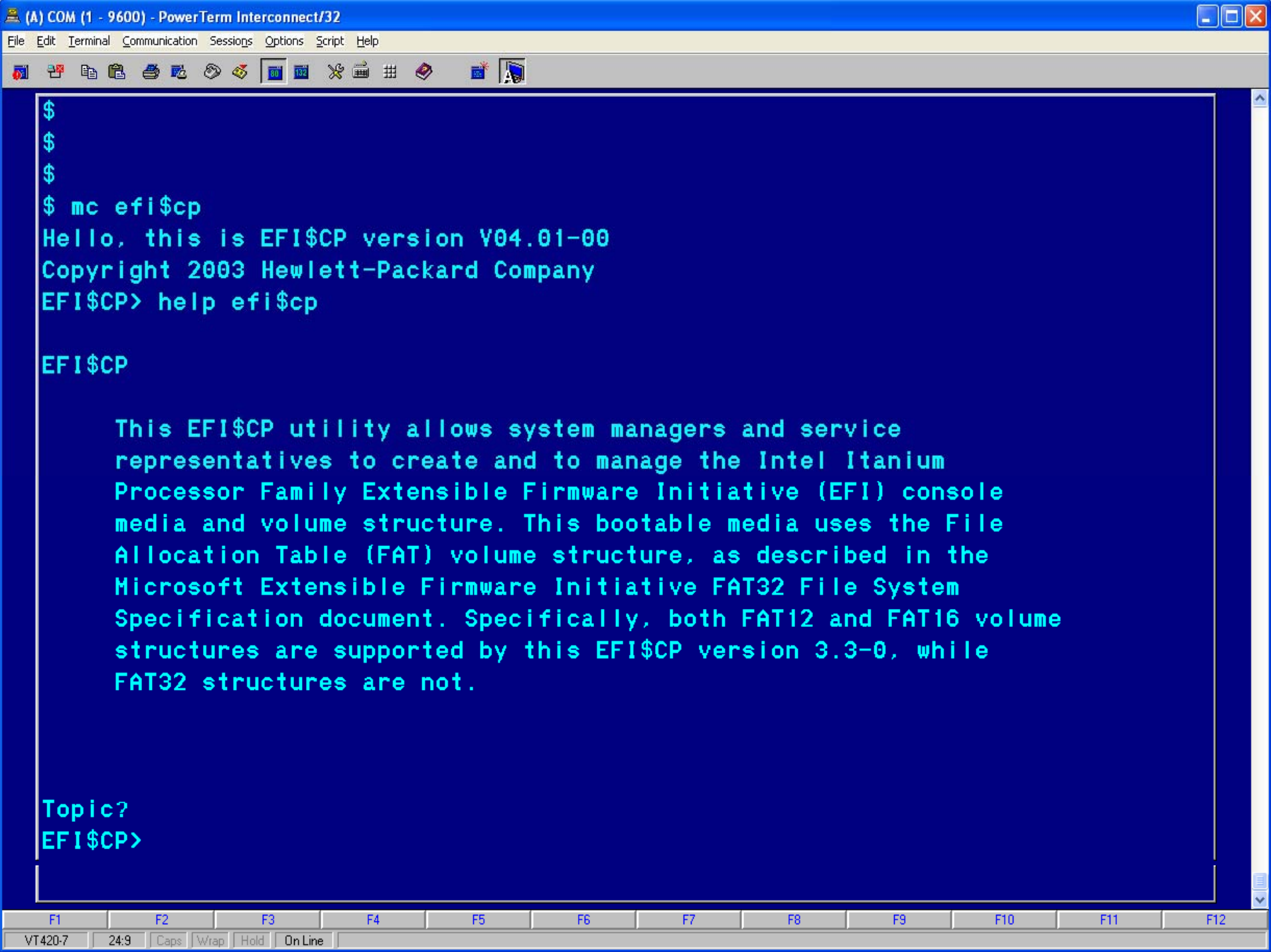
Seen from VMS:
SYS$EFI.SYS

PC style boot block

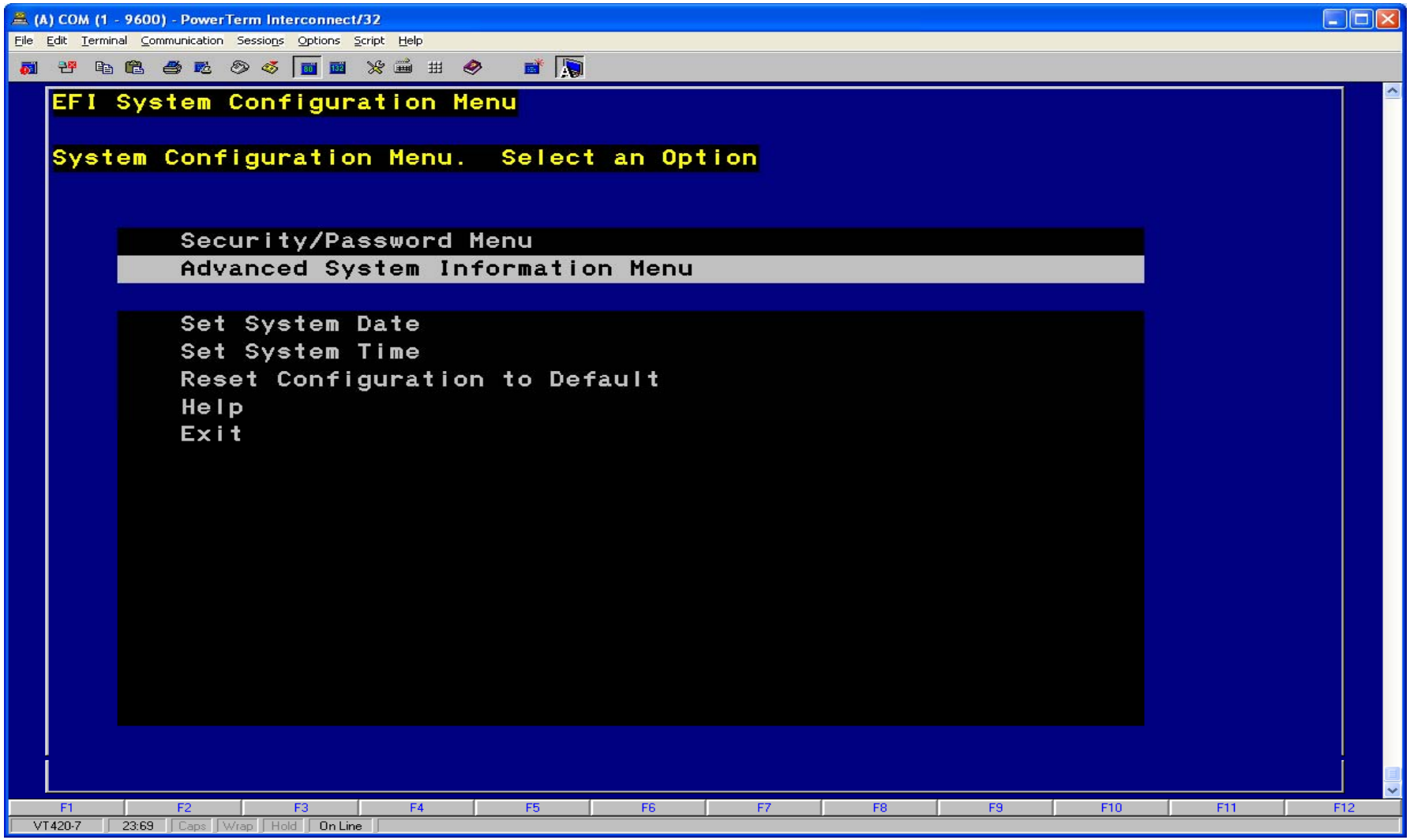
GPT.SYS:

- MBR
- Pointer to EFI partition
- GPT (GUID Partition Table)

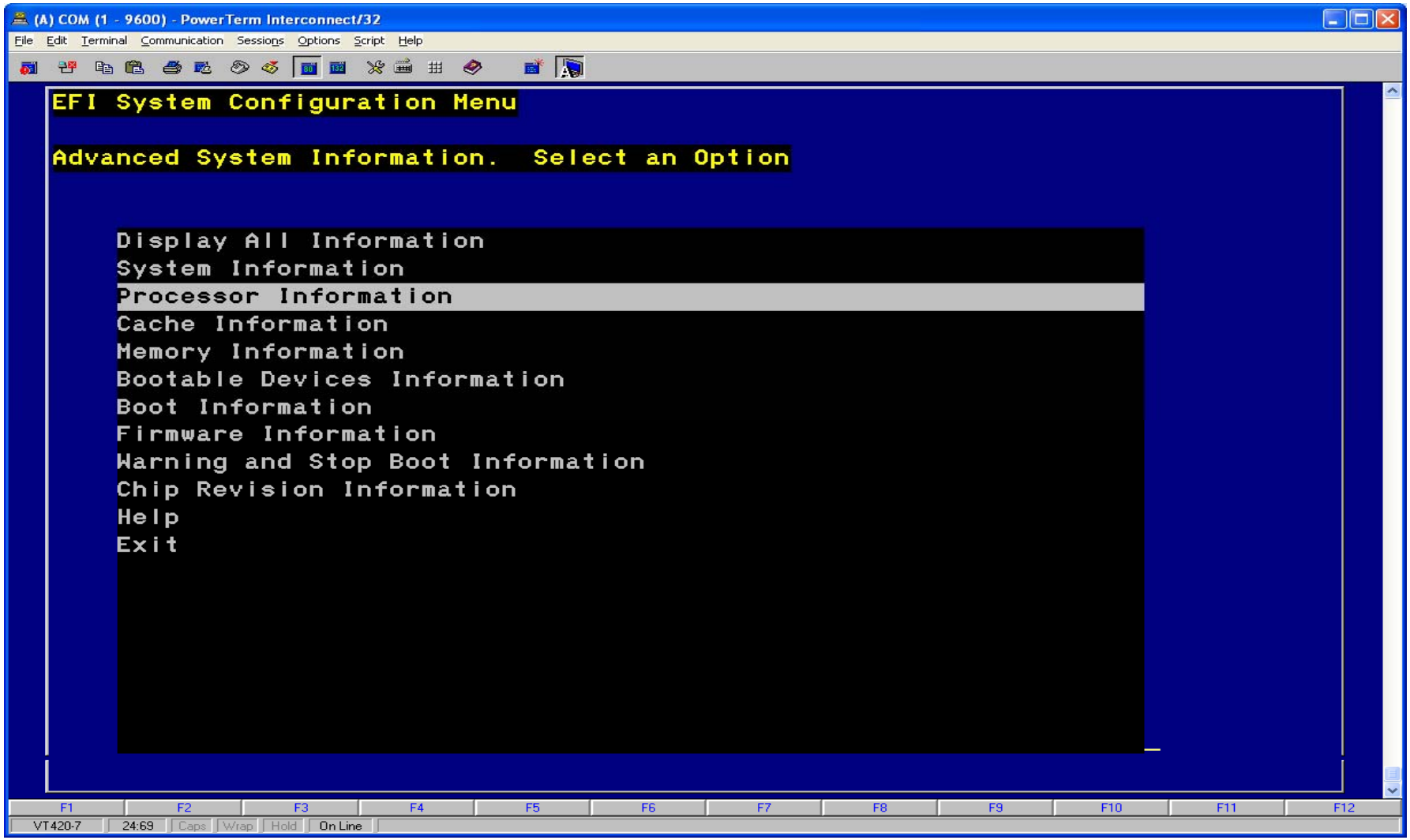

(A) COM (1 - 9600) - PowerTerm Interconnect/32
File Edit Terminal Communication Sessions Options Script Help
$
$
$ dir dka0:[000000]
Directory DKA0:[000000]
000000.DIR;1 ALPHA_TOOLS.DIR;1 BACKUP.SYS;1 BADBLK.SYS;1
BADLOG.SYS;1 BITMAP.SYS;1 BRAEU.DIR;1 CONTIN.SYS;1
CORIMG.SYS;1 GPT.SYS;1 INDEXF.SYS;1 SECURITY.SYS;1
SYS0.DIR;1 TCPIP$DHCP.DIR;1 TCPIP$FTP.DIR;1 TCPIP$REXEC.DIR;1
TCPIP$RSH.DIR;1 TCPIP$TELNET.DIR;1 VMS$COMMON.DIR;1 VOLSET.SYS;1
Total of 20 files.
$
$
$
$ dir/siz sys$loadable_images:*efi*
Directory SYS$COMMON:[SYS$LDR]
SYS$EFI.SYS;1 100000
Total of 1 file, 100000 blocks.
$
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
VT420-7 24:3 Caps Wrap Hold On Line


(A) COM (1 - 9600) - PowerTerm Interconnect/32
File Edit Terminal Communication Sessions Options Script Help
$
$
$
$ mc efi$cp
Hello, this is EFI$CP version V04.01-00
Copyright 2003 Hewlett-Packard Company
EFI$CP> help efi$cp
EFI$CP
This EFI$CP utility allows system managers and service representatives to create and to manage the Intel Itanium Processor Family Extensible Firmware Initiative (EFI) console media and volume structure. This bootable media uses the File Allocation Table (FAT) volume structure, as described in the Microsoft Extensible Firmware Initiative FAT32 File System Specification document. Specifically, both FAT12 and FAT16 volume structures are supported by this EFI$CP version 3.3-0, while FAT32 structures are not.
Topic?
EFI$CP>
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
VT420-7 24:9 Caps Wrap Hold On Line

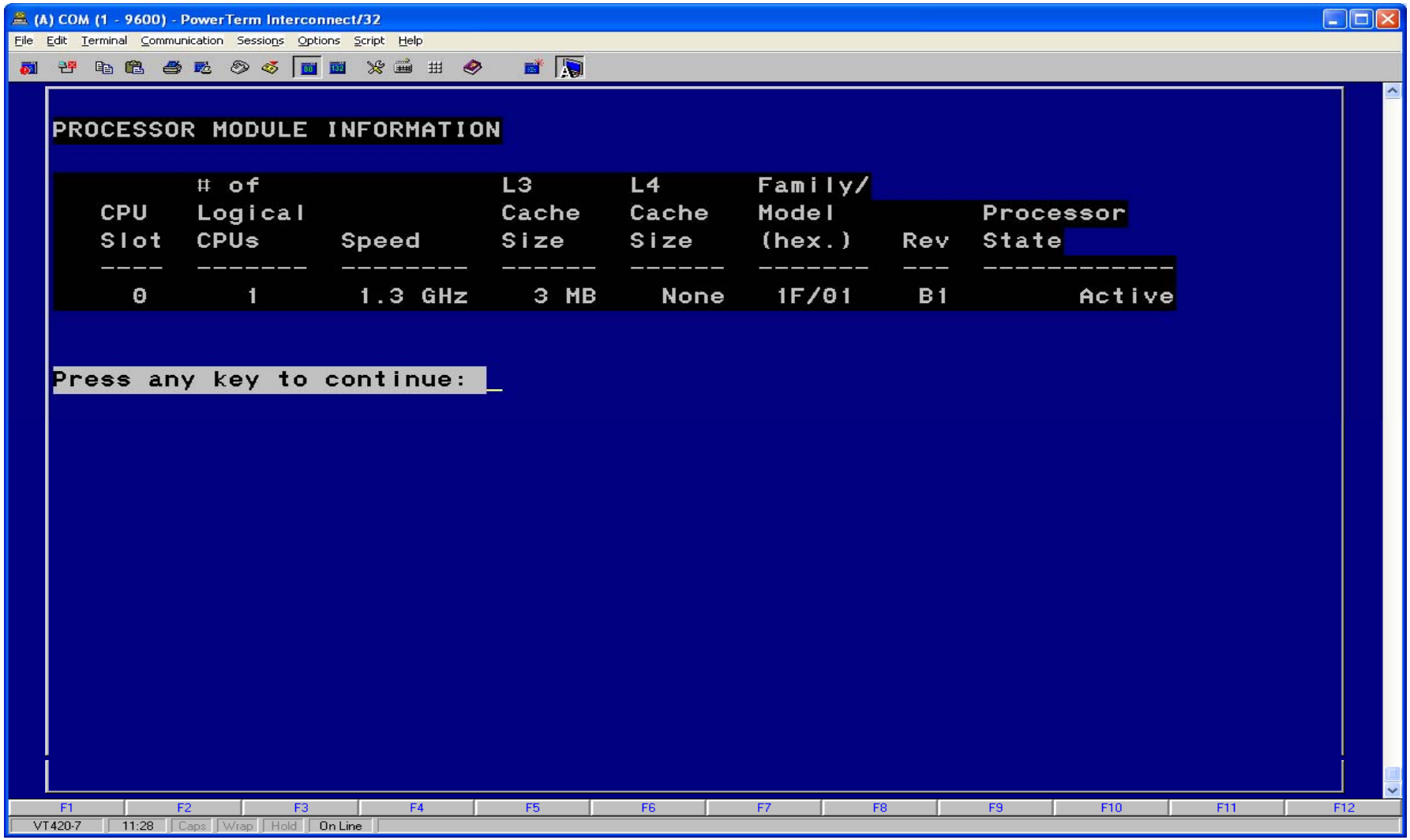
System Configuration Menu



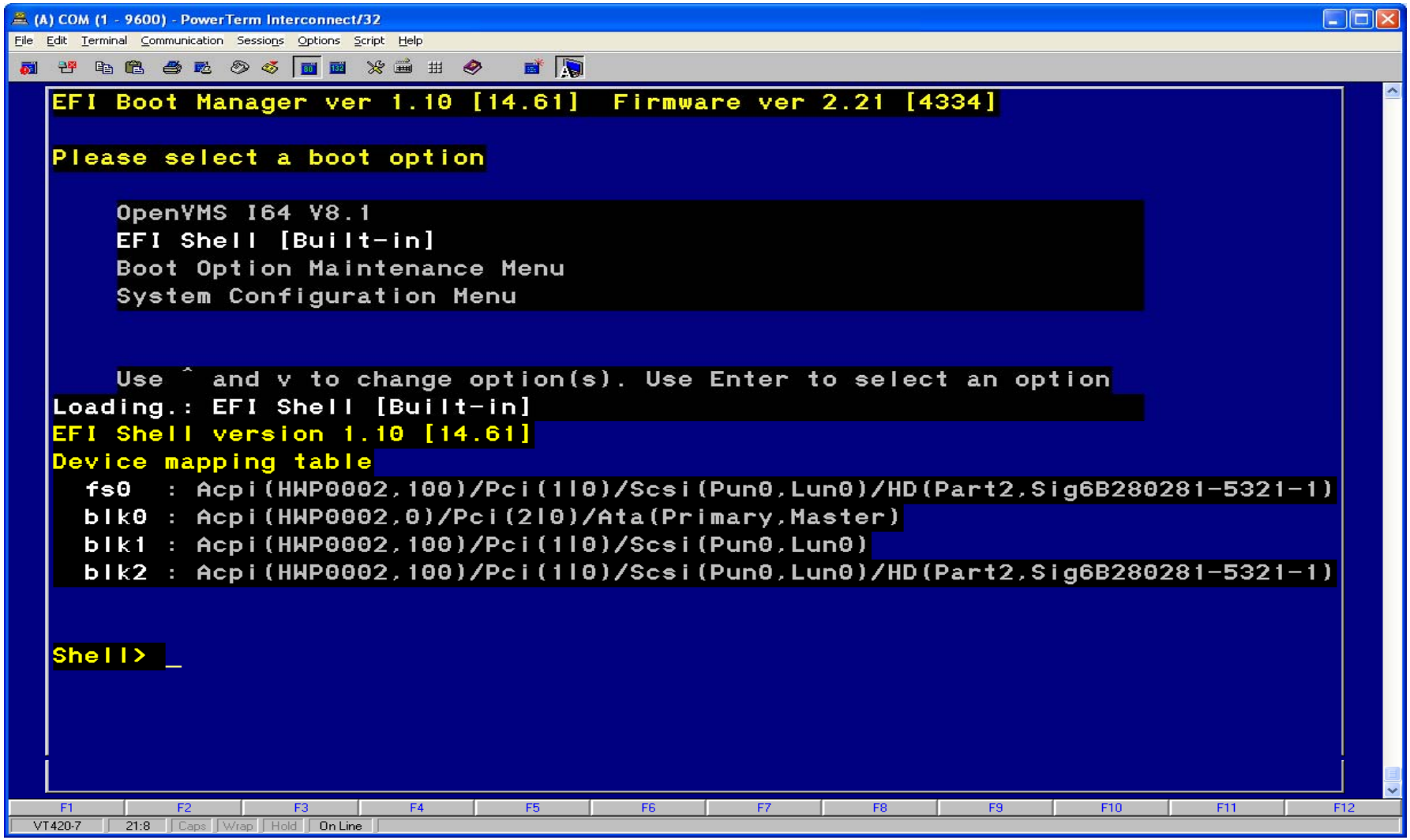
Advanced System Information

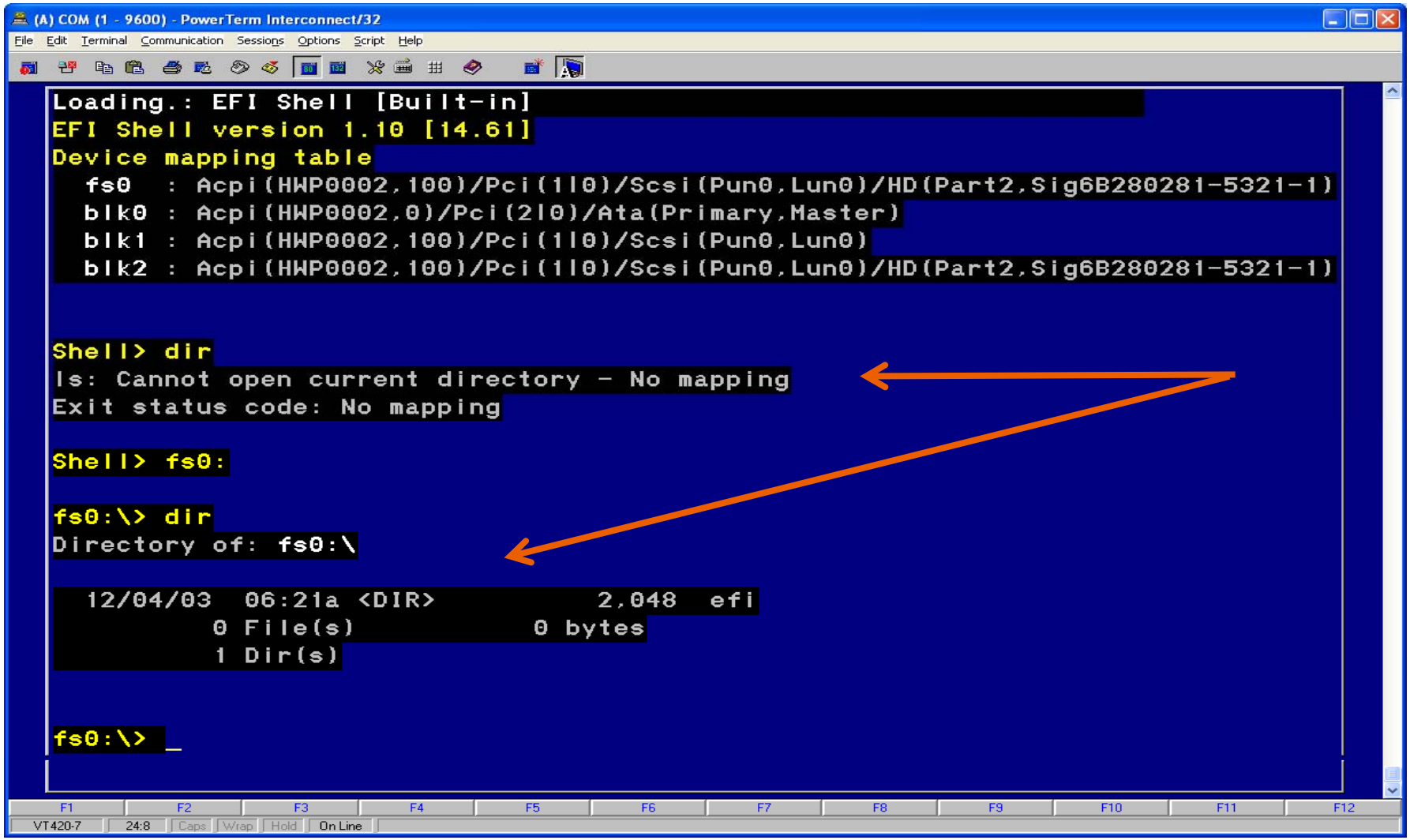


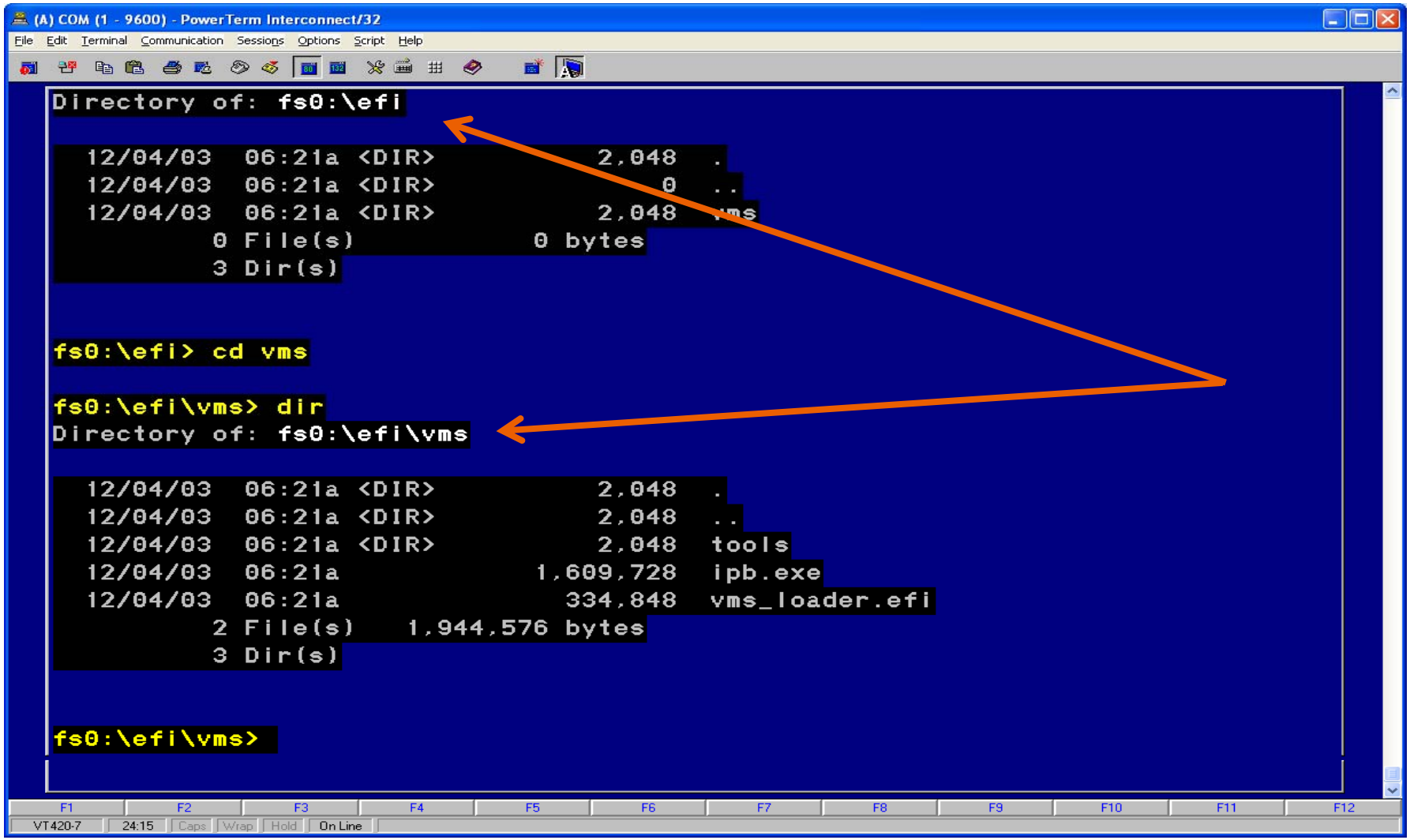
Processor Information Module

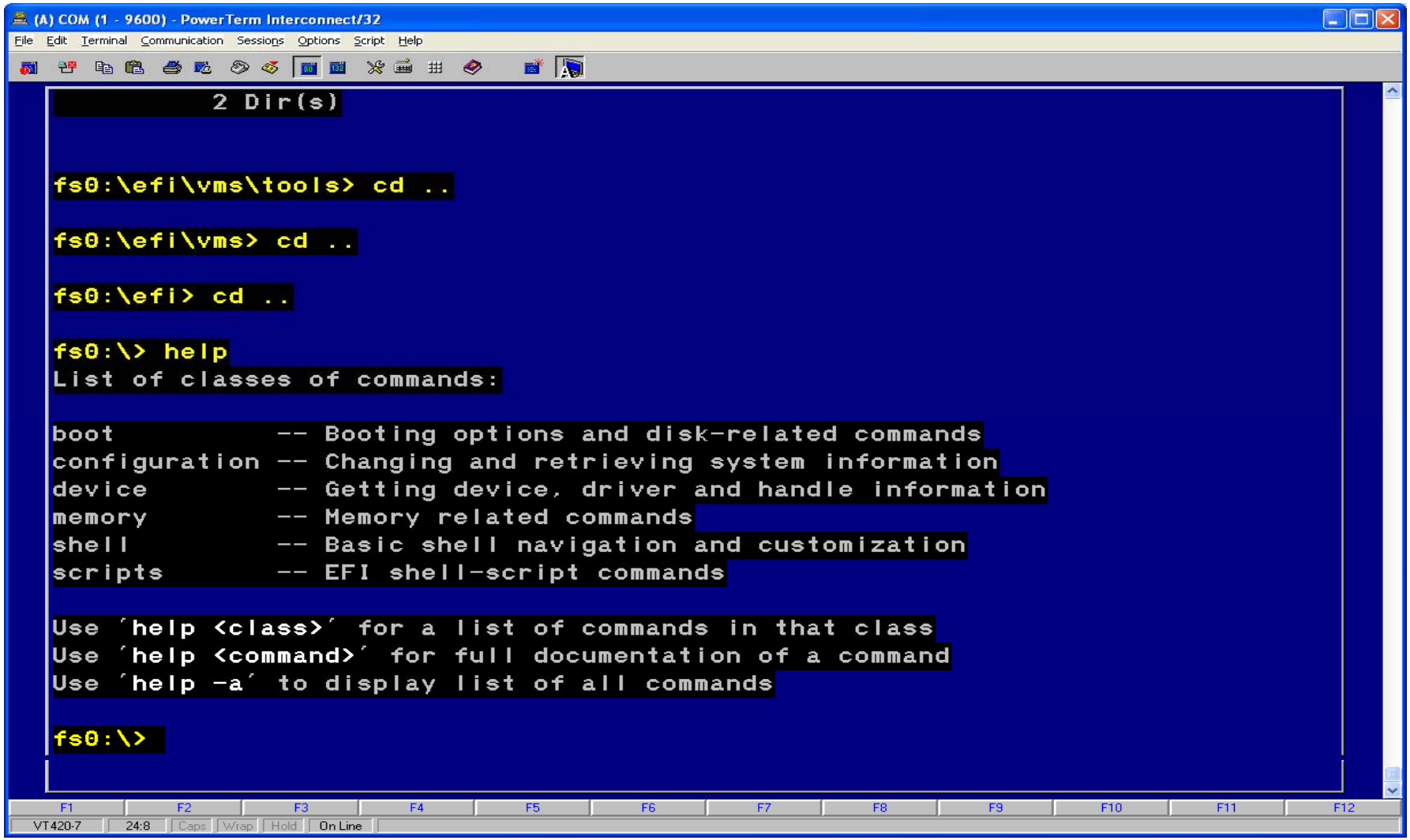


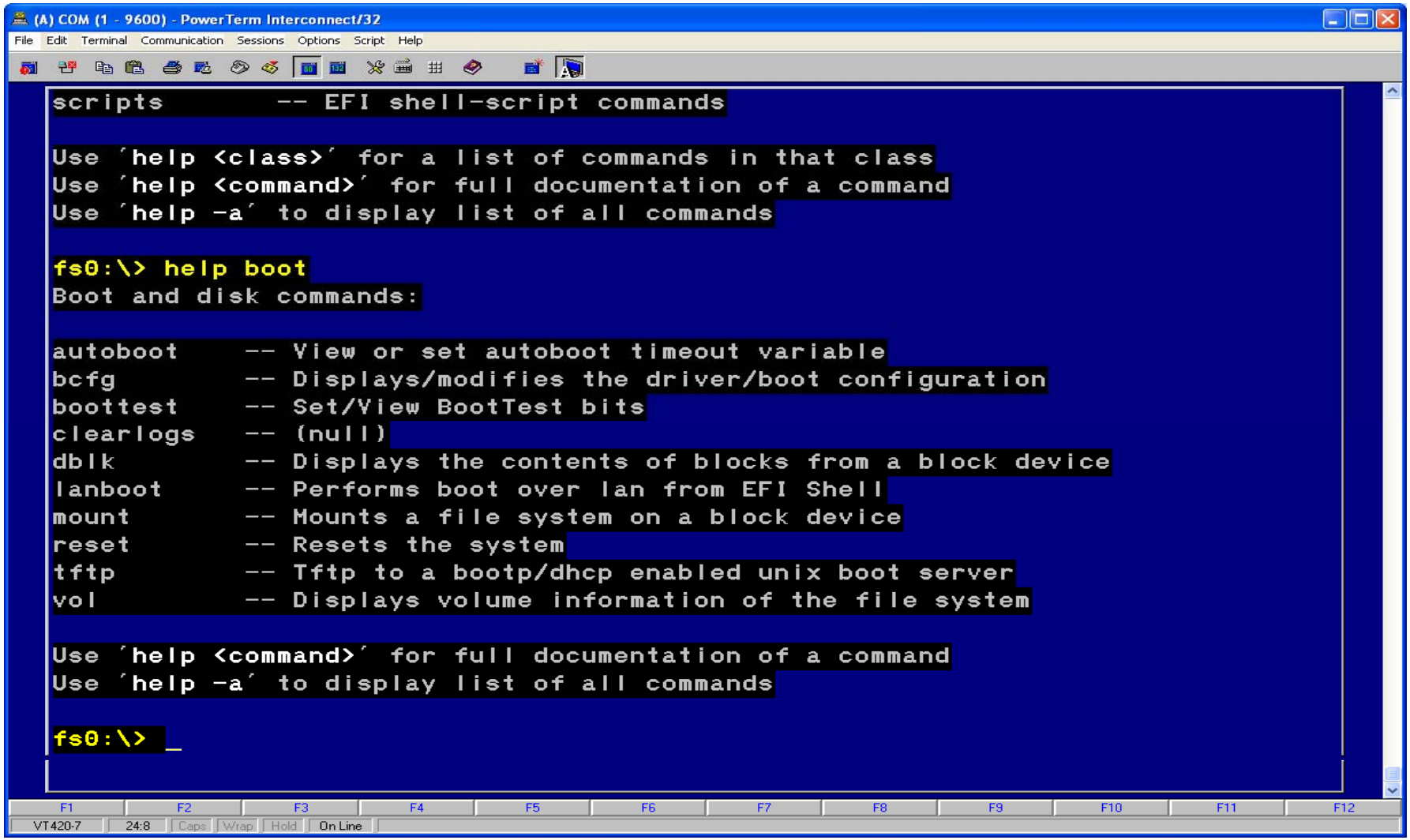
EFI Shell [Built-in]



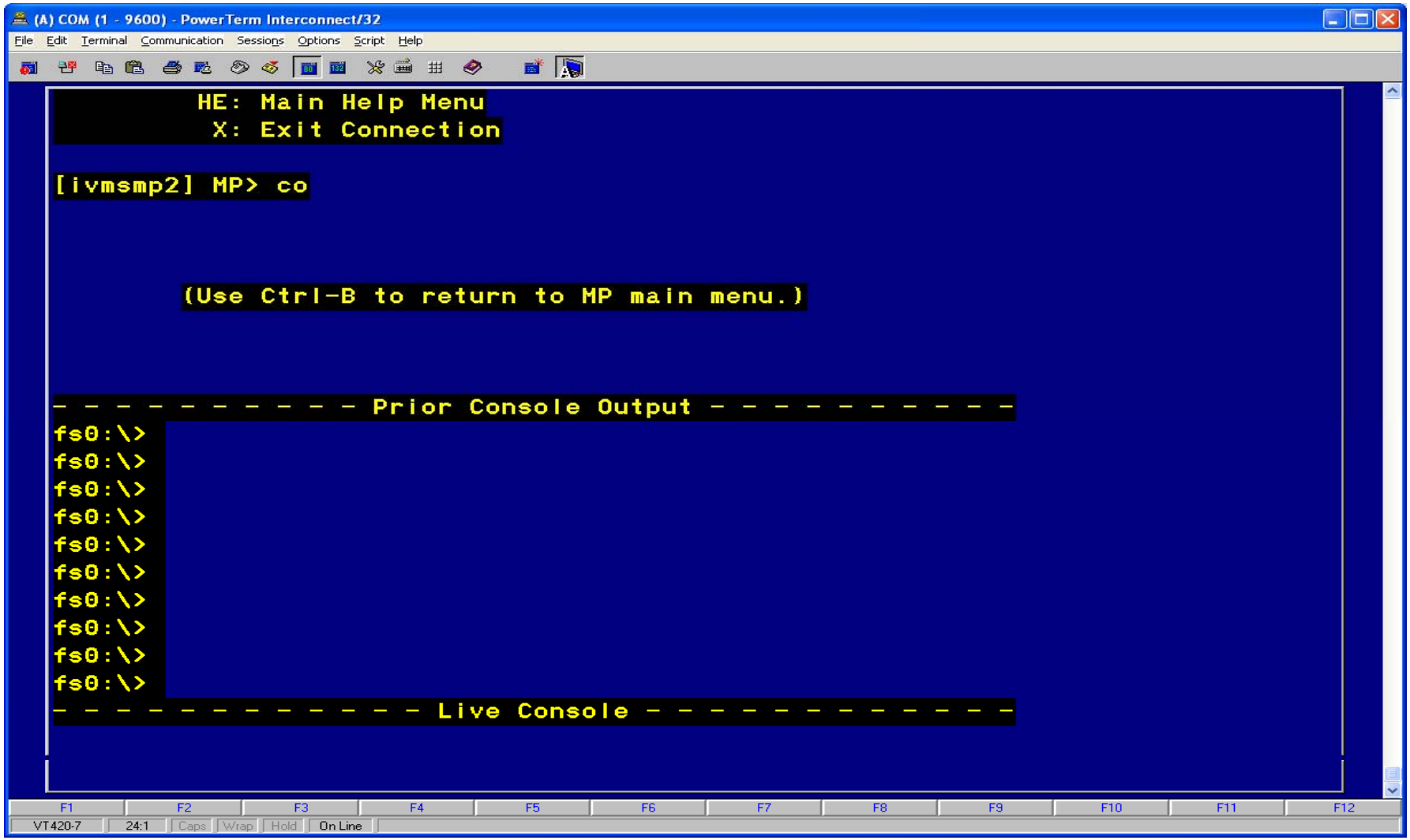

(A) COM (1 - 9600) - PowerTerm Interconnect/32
File Edit Terminal Communication Sessions Options Script Help
Loading.: EFI Shell [Built-in]
EFI Shell version 1.10 [14.61]
Device mapping table
fs0 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)/HD(Part2,Sig6B280281-5321-1)
blk0 : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)
blk1 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)
blk2 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)/HD(Part2,Sig6B280281-5321-1)
Shell> dir
ls: Cannot open current directory - No mapping
Exit status code: No mapping
Shell> fs0:
fs0:\> dir
Directory of: fs0:\
12/04/03 06:21a <DIR> 2,048 efi
0 File(s) 0 bytes
1 Dir(s)
fs0:\> _
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
VT420-7 24:8 Caps Wrap Hold On Line


(A) COM (1 - 9600) - PowerTerm Interconnect/32
File Edit Terminal Communication Sessions Options Script Help
Directory of: fs0:\efi
12/04/03 06:21a <DIR> 2,048 .
12/04/03 06:21a <DIR> 0 ..
12/04/03 06:21a <DIR> 2,048 vms
0 File(s) 0 bytes
3 Dir(s)
fs0:\efi> cd vms
fs0:\efi\vms> dir
Directory of: fs0:\efi\vms
12/04/03 06:21a <DIR> 2,048 .
12/04/03 06:21a <DIR> 2,048 ..
12/04/03 06:21a <DIR> 2,048 tools
12/04/03 06:21a 1,609,728 ipb.exe
12/04/03 06:21a 334,848 vms_loader.efi
2 File(s) 1,944,576 bytes
3 Dir(s)
fs0:\efi\vms>
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
VT420-7 24:15 Caps Wrap Hold On Line

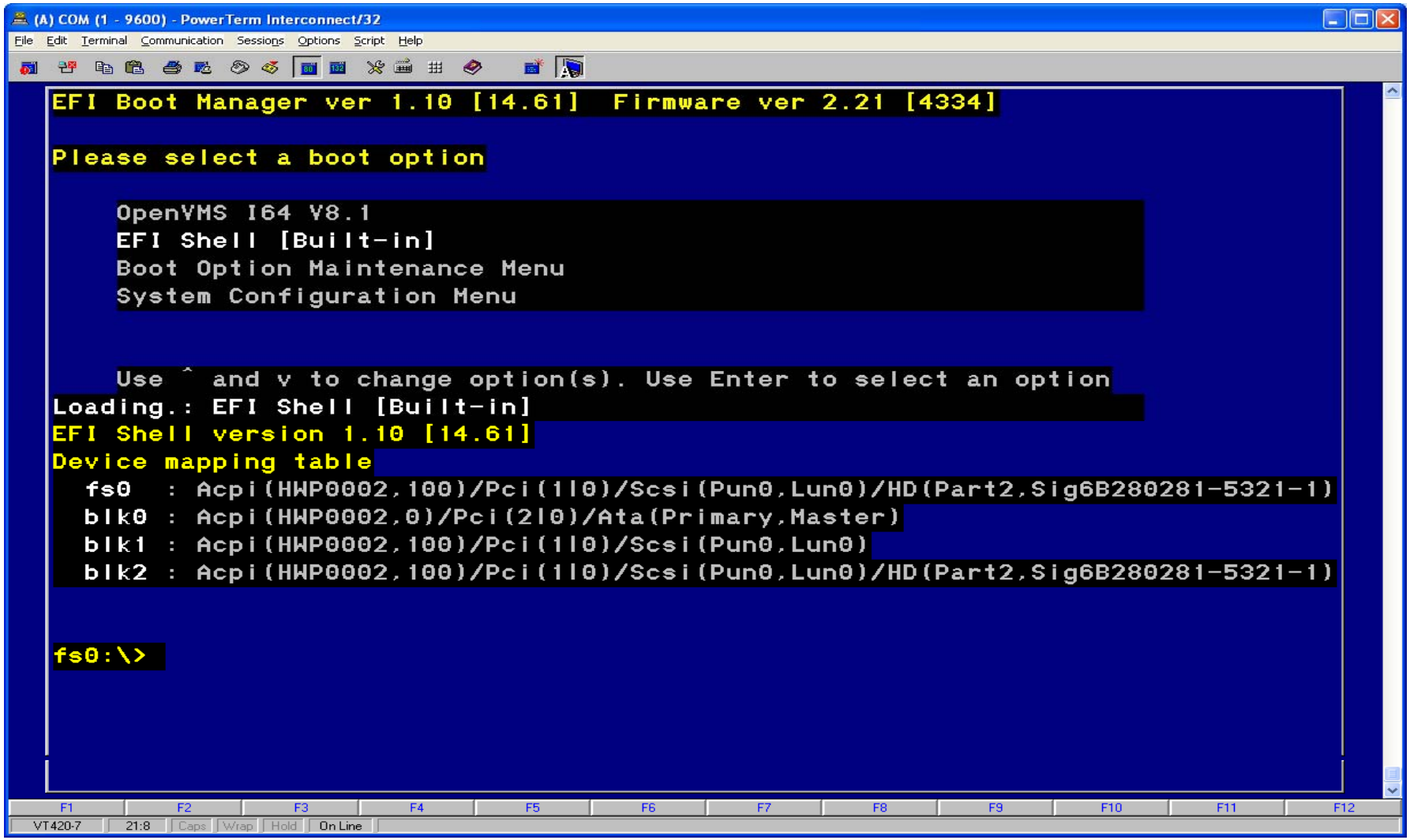

(A) COM (1 - 9600) - PowerTerm Interconnect/32
File Edit Terminal Communication Sessions Options Script Help
2 Dir(s)
fs0:\efi\vms\tools> cd ..
fs0:\efi\vms> cd ..
fs0:\efi> cd ..
fs0:\> help
List of classes of commands:
boot -- Booting options and disk-related commands
configuration -- Changing and retrieving system information
device -- Getting device, driver and handle information
memory -- Memory related commands
shell -- Basic shell navigation and customization
scripts -- EFI shell-script commands
Use 'help <class>' for a list of commands in that class
Use 'help <command>' for full documentation of a command
Use 'help -a' to display list of all commands
fs0:\>
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
VT420-7 24:8 Caps Wrap Hold On Line


(A) COM (1 - 9600) - PowerTerm Interconnect/32
File Edit Terminal Communication Sessions Options Script Help
scripts -- EFI shell-script commands
Use 'help <class>' for a list of commands in that class
Use 'help <command>' for full documentation of a command
Use 'help -a' to display list of all commands
fs0:\> help boot
Boot and disk commands:
autoboot -- View or set autoboot timeout variable
bcfg -- Displays/modifies the driver/boot configuration
boottest -- Set/View BootTest bits
clearlogs -- (null)
dblk -- Displays the contents of blocks from a block device
lanboot -- Performs boot over lan from EFI Shell
mount -- Mounts a file system on a block device
reset -- Resets the system
tftp -- Tftp to a bootp/dhcp enabled unix boot server
vol -- Displays volume information of the file system
Use 'help <command>' for full documentation of a command
Use 'help -a' to display list of all commands
fs0:\> _
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
VT420-7 24:8 Caps Wrap Hold On Line

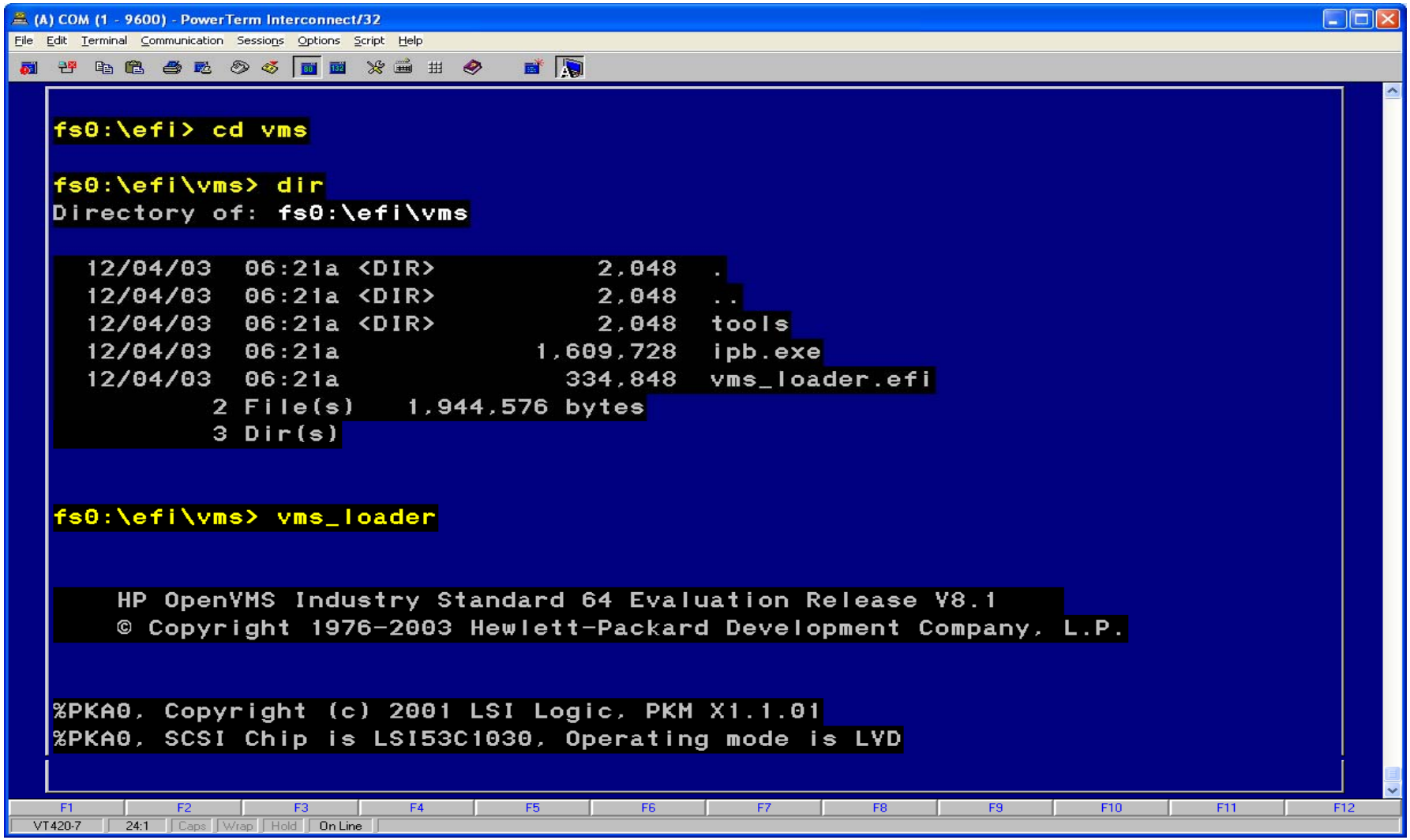
EFI and MCP


(A) COM (1 - 9600) - PowerTerm Interconnect/32
File Edit Terminal Communication Sessions Options Script Help
HE: Main Help Menu
X: Exit Connection
[ivmsmp2] MP> co
(Use Ctrl-B to return to MP main menu.)
- - - - - - - - - - - Prior Console Output - - - - - - - - - - -
fs0:\>
fs0:\>
fs0:\>
fs0:\>
fs0:\>
fs0:\>
fs0:\>
fs0:\>
fs0:\>
fs0:\>
- - - - - - - - - - - - - Live Console - - - - - - - - - - - - -
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
VT420-7 24:1 Caps Wrap Hold On Line

Booting VMS


(A) COM (1 - 9600) - PowerTerm Interconnect/32
File Edit Terminal Communication Sessions Options Script Help
EFI Boot Manager ver 1.10 [14.61] Firmware ver 2.21 [4334]
Please select a boot option
OpenVMS I64 V8.1
EFI Shell [Built-in]
Boot Option Maintenance Menu
System Configuration Menu
Use ^ and v to change option(s). Use Enter to select an option
Loading.: EFI Shell [Built-in]
EFI Shell version 1.10 [14.61]
Device mapping table
fs0 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)/HD(Part2,Sig6B280281-5321-1)
blk0 : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)
blk1 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)
blk2 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)/HD(Part2,Sig6B280281-5321-1)
fs0:\>
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
VT420-7 21:8 Caps Wrap Hold On Line

Booting VMS, cont'd.



Booting VMS from the EFI Shell

- Select EFI Shell from the boot menu
- Set Boot flags environment variable, stored in NVRAM
  - IA64 flag values are generally the same as Alpha/VAX
  - Shell> set vms_flags “0,0”
- Select disk and directory
  - Shell> fs0:
  - fs0:> cd efi\vms
- Start the boot of VMS
  - fs0:> vms_loader
- Override environment variable
  - fs0:> vms_loader –flags 0,1

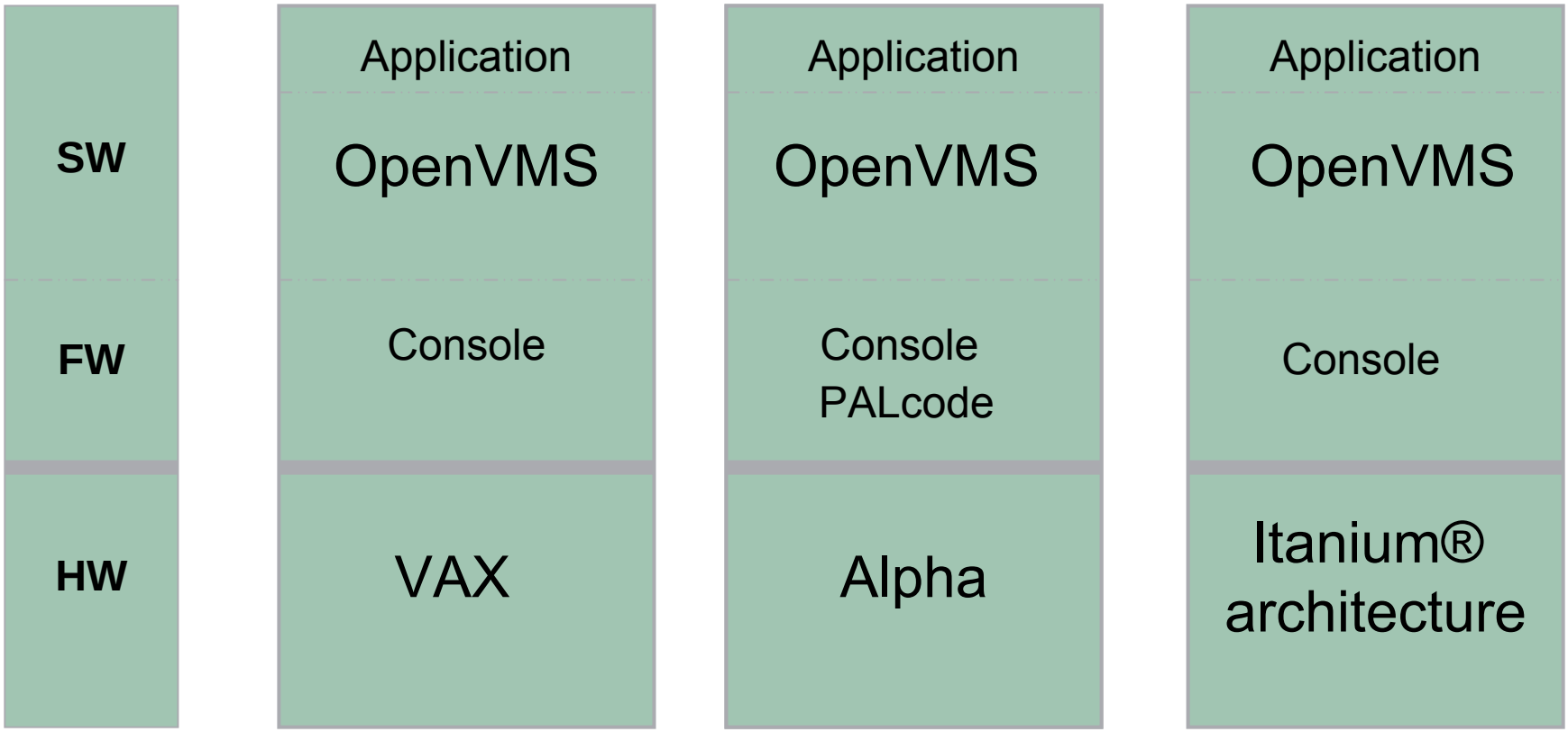
VMS_LOADER.EFI

- Implements functionality of SRM BOOT command:
  - Initializes HWRPB structure
  - Loads IPB.EXE (ELF-format image)
  - Shuts down the EFI console
  - Transfers to IPB in Virtual Address mode
- EFI application, built using MS Visual Studio with:
  - IA-64 cross compilers
  - EFI tool kit provided by Intel
  - MS linker produces image format
- Runs in Physical Address mode as a console command

What's wrong with the Backspace key?

- EFI console requires ^H (ASCII 0x08) for backspace
- VMS traditionally uses DEL (ASCII 0x7F) for backspace
- Two methods for using the backspace key
  1. Set terminal emulator to send DEL for backspace key
     - Use ^H in EFI Shell and Backspace key in VMS
  2. Set terminal emulator to always send ^H for backspace key
     - In LOGIN.COM:
       - $ SET TERM/BACKSPACE=DELETE
     - Backspace key sends ^H and works for EFI Shell and VMS
     - Use ^V ^H ^H for beginning of line

It's All in the Software



PALcall Builtins -- Replacement

- Most, but not all, PALcall builtins result in system service calls on IPF
  - [C] __PAL_BPT(); => [asm] break
  - [C] rd_ps = __PAL_RD_PS(); => [asm] br.call br0 = SYS$PAL_RD_PS
- Some service calls are generated directly by compilers
- Otherwise, there are definition files
    - C - builtins.h ➔ pal_builtins.h ➔ pal_services.h
    - BLISS - builtins.b32
    - MACRO - ia64_macros.mar
- We can determine whatever is best in each case
- Changes can be made anytime

Remove from head of queue, interlocked

- **VAX**: microcoded instruction REMQHI
- **Alpha**: CALL_PAL REMQHIL
- **Itanium® Architecture**: *OpenVMS* system service SYS$PAL_REMQHIL

Infrastructure changes in OpenVMS V8.2

- We're making changes to some system level data structures in OpenVMS V8.2 (Alpha and I64)
- Benefits
  - Laying the foundation for scalability and performance improvements in future releases of OpenVMS
- Impact to applications
  - **Non-privileged applications are not affected**
  - Applications that access the modified data structures in non-standard ways may need to be modified
    - Examples: hard-coded data structure sizes and assumptions about the relative locations of fields within a data structure

Infrastructure changes in OpenVMS V8.2

- Impact to applications (continued)
  - Some privileged applications (such as device drivers) will need to be recompiled and relinked
    - Privileged applications in this case are images linked against the system using the /SYSEXE qualifier and reference the changed data structures or related structures and routines
    - Attempting to execute or load such an image that has not been rebuilt will result in an error during image activation of SYSVERDIF – "System Version Mismatch".

WHAT did you say ? Or The nomenclature game

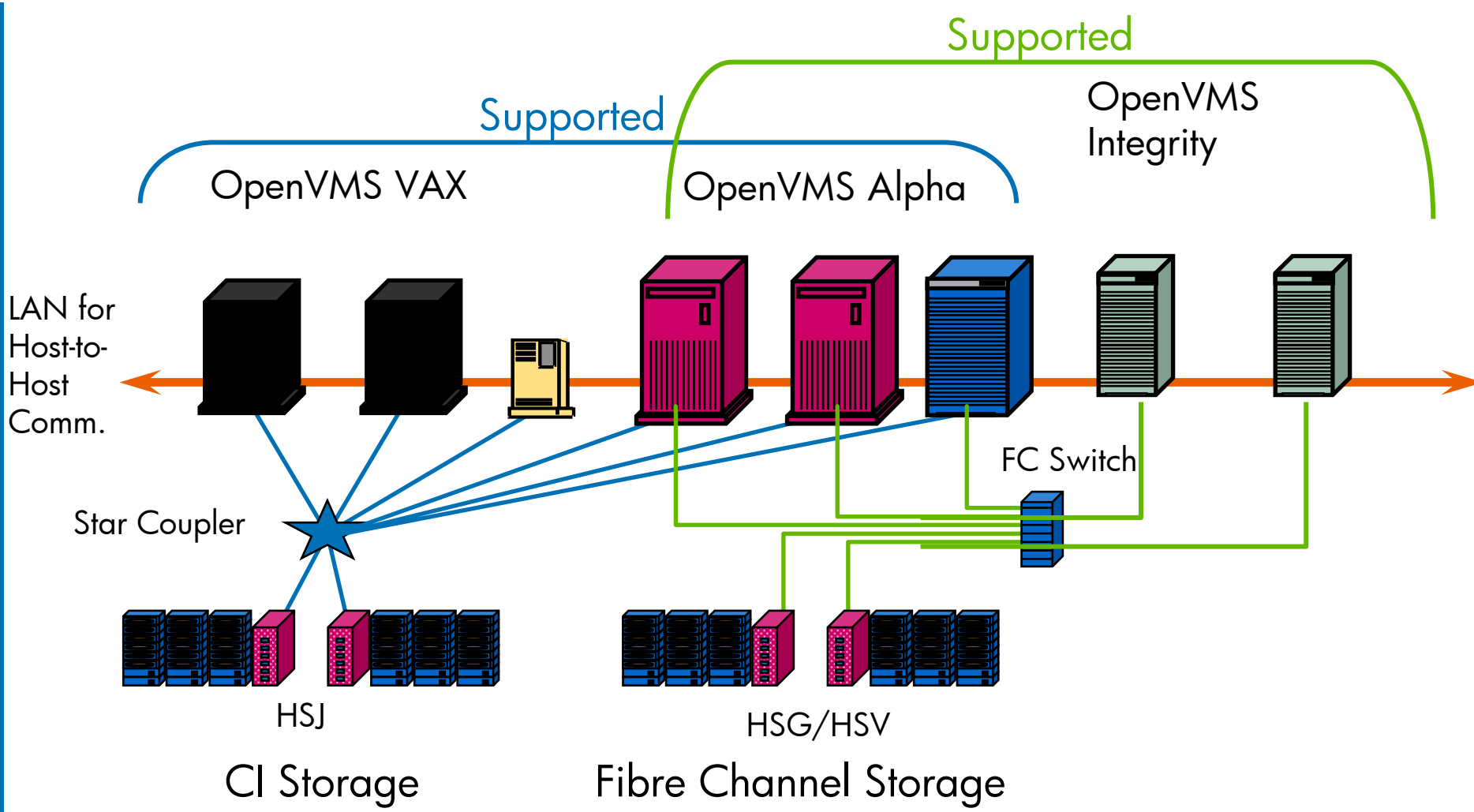
Bytes	Intel®	Alpha
1	***byte***	***byte***
2	halfword	***word***
4	***word***	longword
8	doubleword	***quadword***
16	***quadword***	octaword


And….
What about clustering ?

OpenVMS mixed architecture clusters

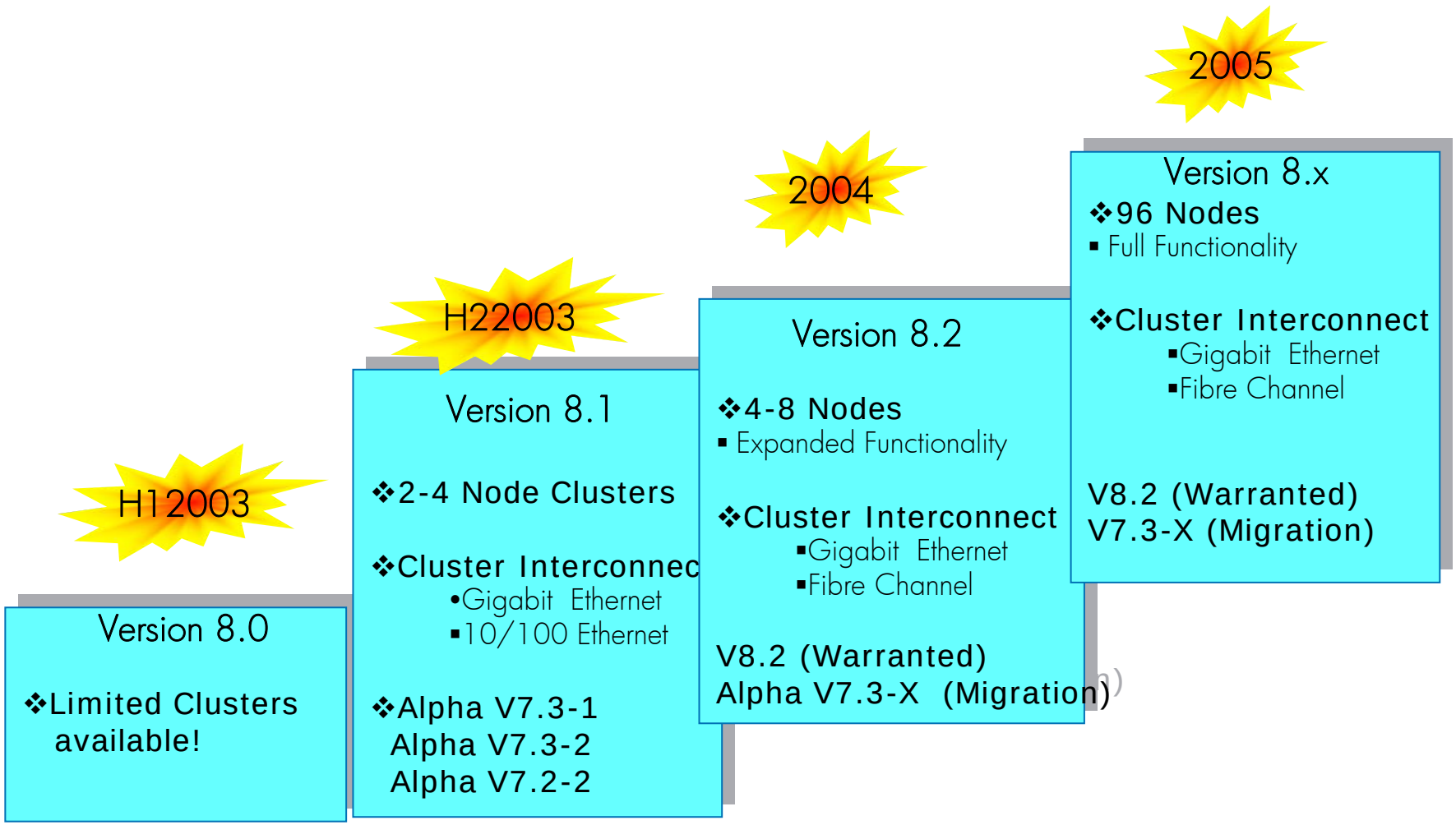
- Clustering is a software architecture
  - underlying chip is easy to deal with
- Will support mixed OpenVMS Alpha and OpenVMS Itanium®-based clusters in a phased roll out
- Do you need VAXes in the same cluster?

Continuing evolution of OpenVMS Clusters



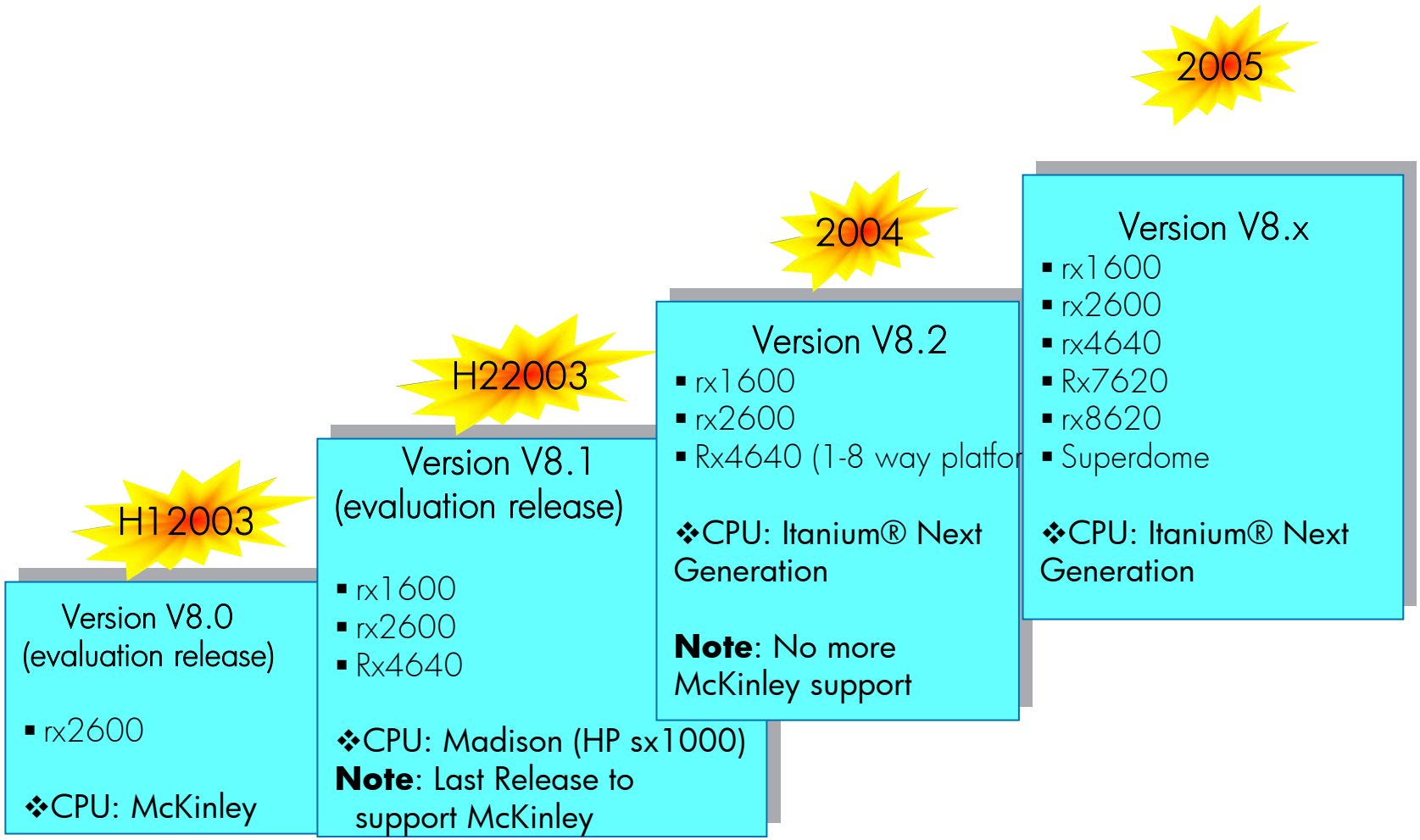
NOTE: Support for VAX and Integrity mixed environment is not currently planned.

OpenVMS I64 Clusters Rollout Plan



Roll-out plans

OpenVMS for Integrity Servers Rollout Plan



HP AlphaServer evolution

GS80
GS160
GS320

GS1280
8p, 16p

GS1280
32p

GS1280
64p

CPU: EV7 1.15GHz
Type: new-box
OS: HP Tru64 UNIX®, HP OpenVMS

CPU: EV7z
Type: in-box
OS: HP Tru64 UNIX, HP OpenVMS

HP Integrity Superdome

CPU: Itanium® next gen
Type: new box
OS: OpenVMS
HP-UX 11i v3
Linux
.NET

HP Integrity rx86xx

CPU: Itanium next gen
Type: new box
OS: OpenVMS
HP-UX 11i v3
Linux
.NET

ES45

ES80

CPU: EV7 1GHz
Type: new-box
OS: Tru64 UNIX, OpenVMS

ES47

CPU: EV7 1GHz
Type: new-box
OS: Tru64 UNIX, OpenVMS

CPU: EV7z
Type: in-box
OS: Tru64 UNIX, OpenVMS

HP Integrity rx76xx

CPU: Itanium next gen
Type: new box
OS: OpenVMS
HP-UX 11i v3
Linux
.NET

DS10, DS10L
DS20E
DS20L
DS25

DS15

CPU: EV68 1GHz
Type: new-box
OS: Tru64 UNIX, OpenVMS

HP Integrity rx2600

CPU: Itanium next gen
Type: new box
OS: OpenVMS, Linux
HP-UX 11i v2, .NET

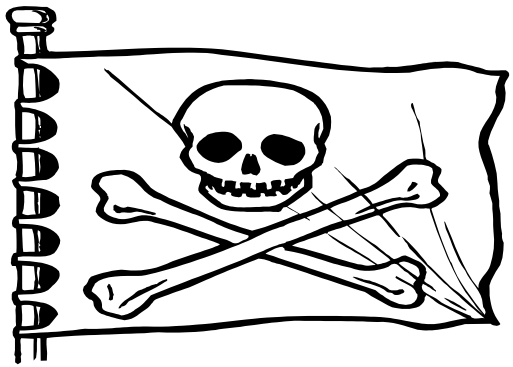
CPU: Itanium next gen
Type: in-box
OS: OpenVMS
HP-UX 11i v3
Linux
.NET

OpenVMS for Integrity Servers Rollout Plan

- V8.2:
  - rx1600, rx2600, rx4640 (4-8 CPUs!)
  - No cell based systems
  - Madison 6M CPU & dual CPU module (Hondo)
  - No more McKinley support!
  - Q4 CY04
- V8.x:
  - Add more systems
  - Cell based systems: rx7620, rx8620, Superdome
  - 50bit physical addressing needed for cell based systems
  - Madison 9M and Montecito CPUs, Arches chip set
  - „Performance“

Packaging

Don't kill the messenger !



OpenVMS I64 Operating Environments

Operating Environment Packaging

- **Introduce OpenVMS packaging consistent with HP-UX OEs**
- **Provides a 3 tier pricing paradigm (good, better, best)**

OpenVMS I64 Operating Environments:

- **Foundation OE (FOE) Base**
  - An **internet ready**, feature rich feature set for **price sensitive** customer
- **Enterprise OE (EOE)**
  - A **higher cost** feature set that enhances the customer experience in areas of **manageability, single system availability and performance**
- **Mission Critical OE (MCOE)**
  - Has the **highest cost**, but delivers the ultimate customer experience in terms of **multi-system availability and workload management**

HP OpenVMS I64 Operating Environments Revision 2.8

OpenVMS I64 Mission Critical Operating Environment (MCOE)

OpenVMS I64 Enterprise Operating Environment (EOE)

OpenVMS I64 Foundation Operating Environment (FOE)

- OpenVMS Operating System
- OpenVMS Unlimited User Licensing
- TCP/IP Services for OpenVMS
- DECnet-Plus for OpenVMS End System
- DECwindows Motif for OpenVMS
- DECnet IV
- Performance Data Collector

Integration Technologies

- BridgeWorks
- COM for OpenVMS
- Secure Web Server (SWS)
- Secure Web Browser (SWB)
- SDK for the Java™ Platform
- XML Technology
- NetBeans
- Simple Object Access Protocol (SOAP) Toolkit
- Kerberos
- Enterprise Directory
- CDSA
- SSL
- OpenSource Tools

Add to Foundation:

- **RMSjournaling**
- **VolumeShadowing**
- **DECram**
- **OpenVMS System Management Tools**

-OVMS Management Station
-Enterprise Capacity Planner - ECP
- Availability Manager
- OpenVMS Web Agents
- OpenVMS WEBM/CIM

Add to Enterprise:

- **OpenVMS Clusters**
- **OpenVMS RTR Backend**

One DVD media for all 3 OE's

Easier to order

Easier license management

Straight forward installation of OE's from a single DVD

Simpler support contracts

➔ higher customer satisfaction

OpenVMS I64 Licensing/Packaging Approach

- All three OE bundles are on one DVD
- PPL licensing for each level (FOE, EOE, MCOE)
  - One LMF PAK for the OE bundle purchased.
- EOE and MCOE components are also available a-la-carte
  - LMF PPL license for each
  - Delivered on the OE DVD and/or OE Delta Disk
- Non-OE layered products
  - Use LMF license

OpenVMS Integrity Operating Environment Phase Rollout Plan

	Q4 2004	Q1 2005	Q2 2005
Foundation Operating Environment (FOE)	• OpenVMS Operating System w/ unlimited users • TCP/IP Services • DECnet-Plus End System • Decnet Phase IV • DECwindows Motif • Secure Web Server (SWS) • Java SDK (Classic VM) • XML Technology • SOAP Toolkit • Enterprise Directory • Kerberos • CDSA • SSL (Secure Socket Layer)	• SWS Tomcat • SWS PHP • Secure Web Browser • Java SDK (Hotspot) • Netbeans • TDC2 Data Collector	• Bridgeworks • COM

Enterprise Operating Environment (EOE)	• RMS Journaling
	• Volume Shadowing
	• DECram
	• Management Tools: Web Agents, Management Station Availability Mgr.
		• Management Tools:
		  • WEBM/CM
		  • Enterprise Capacity Planner	

	• OpenVMS Clusters (available separately)		
Mission Critical Operating Environment (MCOE)			• Reliable Transaction Router – Backend • OpenVMS Clusters

OpenVMS Integrity Layered Product Phase Rollout Plan

Q4/2004	Q1/2005	Q2/2005	Q3/2005
• Compilers: BASIC, Fortran, C, C++, COBOL, Pascal • DECset: CMS, MMS, LSE, DTM, PCA & SCA • Distributed File System • DECprint Supervisor • DQS • WEBES • DCE • Archive Backup System • Data Cartridge Server • Disk File Optimizer (DFO) • Hierarchical Storage Mgmt. • Media Robot Utility • RAID Software • Save Set Manager (SSM) • GKS • Phigs • FMS • BASEstar Family • Datatrieve • Device Access Software • OMNI API/MMS		• Reliable Transaction Router (RTR) • X.25	• ACMS (including TP Web & TP Desktop Connectors) • Advanced Server • DECforms
			Q4/2005
			• Soft Partitioning (ie. Galaxy/vPars) • Storage Library System (SLS)

Software Licensing Overview

Per-processor licensing (PPL)

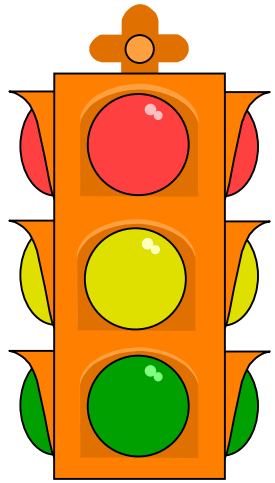
- Very **flexible** licensing design
- Purchase software based on the **# of CPUs** in a partition or system

Benefits of PPL

- **More granular** – customer pays for exactly what they need
- **More flexible** – licensing is not per box, but per-processor so customer can move assets as needed
- **Accommodates partitioning** – allows use of different types of OEs in different hardware partitions and different Operating Systems
- **Expandable** – customers can purchase processors and software to meet needs over time

SOFT COMPLIANCE

- PPL is based on licensing CPUs
- 1 unit per running CPU on node
- 3 states
  - Red - PAK is not loaded
    product cannot run (failure status returned)
  - Yellow - short 1 or more units as compared to # of CPUs
    product can run (success status returned, PPL tool will flag)
  - Green - units equal or greater than # of CPUs
    product can run (success status returned)



Soft Compliance

- On VAX/Alpha a PAK will not be loaded if

  Required units != Available PAK units

- On I64 a PAK will be loaded if there is at least one unit available
  - RX4640 with 4 CPUs may load a 2 units FOE PAK
  - SHOW LICENSE/USAGE displays compliance info
  - Compliance report tool will tell customers if they are in compliance or not (remember the 3 states, red, yellow and green)
- We may provide ISVs with the ability to force hard compliance for their products (P3)

SHOW LICENSE/USAGE

```
IPL31> show license/usage

View of loaded licenses from node IPL31                    27-JAN-2004
  09:57:06.51

------- Product ID --------   ---- Unit usage information ----------------
Product             Producer       Loaded  Allocated  Available  Compliance
C                   HP                  2          2          0  Yes
OPENVMS-I64-FOE     HP                 10          2          8  Yes
OPENVMS-I64-MCOE    HP                  1          1          0  No

PPL license HP OPENVMS-I64-FOE usage information:
        Per Processor License
        Activity: 0
        Version:  0.0
        Release Date: (none)
        Termination Date:  1-JUL-2004
        Product Token: *ENGINEERING_INTERNAL_USE_ONLY*
    Units   Node
        2   IPL31
    Units loaded: 10    Units allocated: 2    Units available: 8
```

Migrating Applications

Development Tools

- All development tools and utilities that ship with OpenVMS are being ported
- Developers can use existing procedures for developing, debugging, testing, and deploying their applications
- DECset tools shipped with V8.1
  - Language-Sensitive Editor/Source Code Analyzer (LSE/SCA)
  - Code Management System (CMS)
  - Module Management System (MMS)
  - Digital Test Manager
  - Performance and Coverage Analyzer (PCA) (ships H2/04)

Compiler Version Mapping Alpha vs. Itanium(r)

Compiler	***Alpha***	***Itanium***
Basic	V1.5	tbs
Bliss	V1.10-030	T1.1-049
Cobol	V2.8-1286	T2.8-1340
Fortran 77	--	na (Alpha only)
Fortran 90	V7.5	T8.0
C	V6.5	T7.0
C++	V6.5	tbs
Java	1.4.2-beta	1.4.2-beta1
Macro-32	V4.1-18	T1.0-77
Macro-64	V1.2	na (Alpha only)
Pascal	V5.8A	tbs

Example 1: Database vendor

- Application 1: written in C; no problems at all
- Application 2a: written in VAX assembler
  - Using HW knowledge in code
  - Hand coded kernel threads
  - Use calling standard knowledge
  - Hand coded save/restore of stack
    - VAX: ok
    - Alpha: using AMACRO, luckily it worked
    - Itanium(r): using IMACRO, very large effort
- Application 2b: written in C
  - Issue: uses functionality not yet implemented under UNIX Portability Initiative (fork, semaphore handling,…)

Example 2: Cadture

- 801 Fortran modules, about 2500 routines, 6 needed /nowarning
- Successful run after first link
- Found one programming error (status code)
- Compile time 10min total
- Dynamics:
  - Alpha Fortran noopt/opt 1:3
  - Itanium Fortran noopt/opt 1:5

Example 2: Cadture

- „VMS-bound“, virtual Fortran arrays, system services, IMG-services, X11 und Motif
- Conflicts: „Classical Fortran (Dispatch)“: Computed/Assigned Goto results in too many warnings: „Possible illegal jump into code block“.
- Program uses Floating, Integer, Character and Byte.
- To start only a text file is necessary, no floating conversion of old data

Code Changes necessary

Code that will require changes

- Alpha Macro 64 Assembler code.
  - This code must be rewritten in another language.
- Conditionalized code for Alpha or VAX systems.
  - This code must be revised to express an I64 condition.
- Code that uses OpenVMS system services that have dependencies on the Alpha architecture.
- Code with other dependencies on the Alpha architecture.
- Code that uses floating point data types.
- Code that uses threads, in particular, custom-written tasking or stack switching.
- Privileged code.

Alpha Macro 64 Code

- Rewrite in another language!

Conditionalized Code

- Old:
- #ifdef __vax
- …
- #endif
- #ifdef __alpha
- …
- #endif

- New:
- #ifdef __vax
- … 32bit path
- #else
- … 64bit
- … Path (alpha & I64)
- #endif

System Services & Alpha dependencies

- SYS$GOTO_UNWIND
- uses 32bit invocation context handle
- Change to:
  - SYS$GOTO_UNWIND_64
  - uses 64bit invocation context handle
  - Different set of library routines to return a 64bit invocation context handle
  - See *HP OpenVMS Calling Standard*

System Services & Alpha dependencies, cont'd...

- SYS$LKWSET & SYS$LKWSET_64
- SYS$ULWSET & SYS$ULWSET_64
- Replace with LIB$LOCK_IMAGE, LIB$UNLOCK_IMAGE
  - Only on Alpha and IA64!
  - No need for code that finds code, data and linkage sections and locks them
  - Addresses for these difficult to find on IA64

Alpha Architecture Dependancy

- Condition handling using SS$_HPARITH
  - Alpha: signaled for several arithmetic error conditions
  - I64: never signaled for arithmetic error conditions
  - I64: use SS$_FLTINV or SS$_FLTDIV instead
- Mechanism Array Data structure
  - Content is different
- Alpha Object/Image File Format
  - I64 uses a different formats
    - Object: Executable and Linkable Format (64bit version)
      - http://www.caldera.com/developers/gabi
    - Image & DST: DWARF V3
      - http://www.egercon.com/dwarf/dwarf3std.htm

Floating Point Data Type Usage

- Float wait_time = 2.0;
- Lib$wait (&wait_time);

- IA64: sends S_FLOATING to routine
- LIB$WAIT expects F_FLOATING -> FLTINV condition
- Better:
  - #ifdef __ia64
  - Int float_type = LIB$K_IEEE_S;
  - #else
  - Int float_type = LIB$K_VAX_F;
  - #endif
  - Float wait_time = 2.0;
  - Lib$wait (&wait_time,0,&float_type);

Code using threading

- All thread interfaces are supported on OpenVMS I64
- I64 code use much more stack space than Alpha code
  - may receive stack overflow as ACCVIO (V8.1) STKOVF (V8.2)
- I64: default stack size larger
- I64: may need to increase size if application requests specific stack size

Unaligned Data

- Unaligned data seriously degrades performance
- No difference for OpenVMS Alpha and I64

Reliance on Alpha Calling Standard

- OpenVMS I64 calling standard based on Intel calling standard with modification
- Different from Alpha
- Differences include:
  - Register numbers are different
  - No frame pointer (FP)
  - Multiple stacks
  - Only 4 registers preserved across calls

Privileged Code

- See SYS$LKWSET example
- Terminal drivers
  - Interface changed from JSB to call based interface
  - (JSB uses registers to pass arguments)

OpenVMS Infrastructure Changes

- IPF and Alpha only
- Privileged Images only (link against system [/SYSEXE])
- Dependancy on following subsystems
  - SYS$K_VERSION_IO
  - SYS$K_VERSION_MEMORY_MANAGEMENT
  - SYS$K_VERSION_CLUSTERS_LOCKMGR
  - SYS$K_VERSION_FILES_VOLUMES
  - SYS$K_VERSION_CPU
  - SYS$K_VERSION_MULTI_PROCESSING
- Increase of version number
- How to find out dependancy:
  - $ ANAL/IMAGE your_image.exe/OUT=image.txt
  - $ SEARCH image.txt "SYS$K"

Kernel Process Extensions

- Usage of Kernel Processes now allowed in outer modes and all IPLs
- **Alpha and IPF only change**
- Code with private threading packages can now make use of Kernel Processes
- Some changes to the KPB$ data structure were necessary
- No source changes necessary for existing Alpha code
- Recompile and relink required (image has "SYS$K" matches)

CPU Name Space

- OpenVMS current architectural limit of maximum CPU Id of 31
- Increase this limit to
  - maximum of 64 for Alpha
  - Maximum of 1024 for IPF
- V8.2 release will not support any systems (IPF or Alpha) with CPU Ids larger than 31
- Some kernel data structures maintain 32-bit CPU Id masks
- Increase the space allocated for these CPU Id masks
- Existing longword symbols for CPU masks will continue to be maintained
- With the exception of rebuilding, there should be no impact to privileged images and drivers.
- Recompile and relink required (image has "SYS$K" matches)

64Bit Logical Block Number (LBN)

- OpenVMS today supports LBNs of only 31 bits
- This limits a disk volume to 1 terabyte
- Various LBN fields in data structures are promoted from longwords to quadwords
- Longword symbols will continue to be maintained
- This will allow for future operating system support of volumes larger than 1 terabyte
- No plans to support volumes larger than 1 terabyte for V8.2
- Recompile and relink required (image has "SYS$K" matches)

Forking to Dynamic Spinlock

- In order to scale the OpenVMS operating system on large SMP systems, a number of areas in the operating system have been using dynamic spinlocks as opposed to the very limited number of static spinlocks. The ability to FORK and have the fork dispatcher obtain synchronization with a dynamic spinlock is very desirable. We are adding this capability to OpenVMS V8.2 by extending the size of the FKB$ data structure and adding a FKB$L_SPINLOCK field. This spinlock field will only be referenced if FKB$B_FLCK contains the value SPL$C_DYNAMIC.
- Recompile and relink is required if a search of your image as outlined above results in any "SYS$K" matches. A very small subset of applications may need to make code changes. We recommend privileged code be checked for cases of allocating FKB structures and using a hard coded value of the old structure size of 32 bytes. Code should use the symbol FKB$C_LENGTH for the size of a FKB structure. Also, if privileged use the FKB$B_FLCK from a VMS supplied FKB structure it may be necessary to insure the FKB$L_SPINLOCK field is also taken into account - for example if copying the FKB$B_FLCK field to another FKB structure.

Fast Device Create/Delete

- Device list (UCBs) associated with a controller (DDB) is a zero terminated singularly linked list
- When creating and deleting a UCB, these lists must be walked until the appropriate location is found in order to add or remove a UCB from the list
- Will now be a doubly linked list (still zero terminated) to avoid the sequential search when creating and deleting a UCB
- This requires the addition of some new cells in the UCB and DDB.
- Recompile and relink (image has "SYS$K" matches)
- Code which modifies the list of UCBs associated with a DDB should be updated to utilize VMS provided routines
  - IOC_STD$CLONE_UCB, IOC_STD$COPY_UCB, IOC_STD$LINK_UCB, IOC_STD$DELETE_UCB
- Code walking the list of UCBs still works correctly without any changes

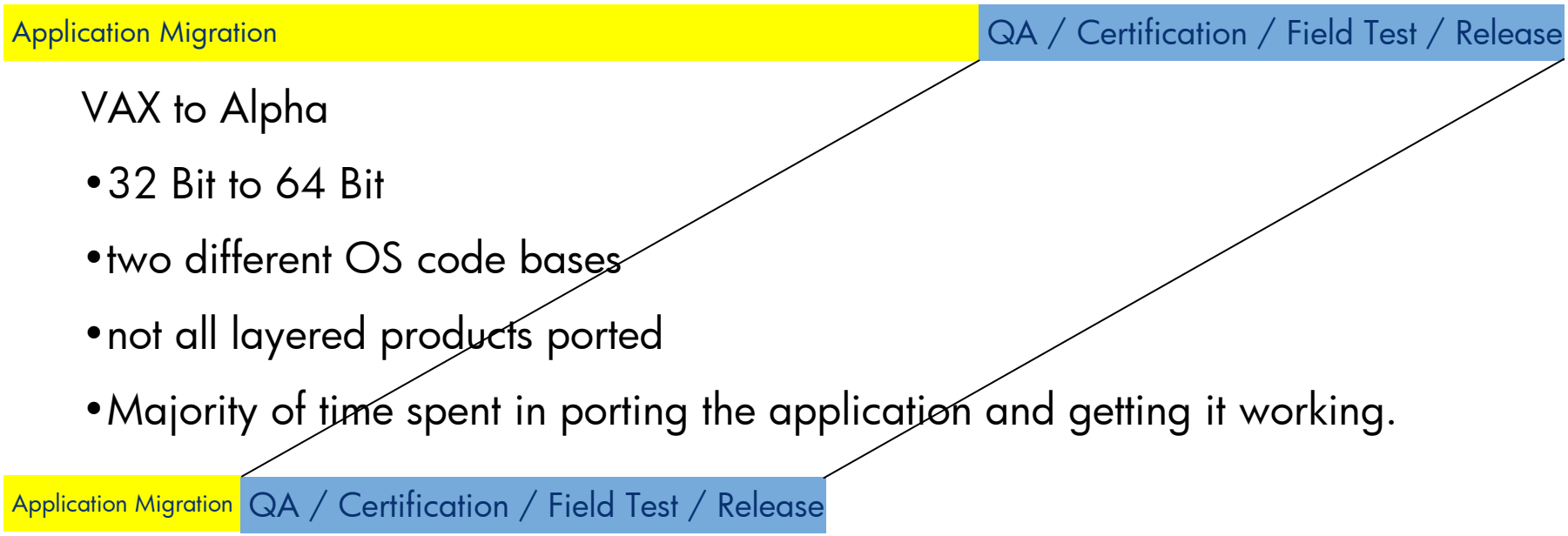
UCB Field Promotions

- The UCB$W_UNIT field promoted to a longword
- Support more than 64k unit numbers for a device
- The UCB$W_UNIT field will still be maintained
- Recompile and relink (image "SYS$K" matches)

Terminal Driver Updates

- Fields in the terminal driver's UCB extension will be promoted from bytes and words to longwords
- Existing field names will continue to overlay the promoted fields
- Recompile and relink (image has "SYS$K" matches)

Porting OpenVMS applications VAX to Alpha to Itanium



Cross-section of leading OpenVMS ISVs committed to HP Integrity servers


©JOGIN


invent