tianocore

UEFI & EDK II TRAINING

UEFI Shell Lab w/ Simics

tianocore.org

Lesson Objective

- Run UEFI Shell with Simics QSP
- Run UEFI Shell Commands
- Run UEFI Shell Scripts

tianocore

UEFI Shell Lab with Simics

Invoke Simics w/ Platform BoardX58Ich10

First Setup for Building EDK II, See Lab Setup then Platform Build Lab for Simics

Details for the following are from the Links above

From the VS command prompt

```
$> cd C:\fw\edk2-ws\edk2-platforms\Platform\Intel
$> python build_bios.py -p BoardX58Ich10  -t VS20XX
```

Where *XX* is 15x86 or 17 or 19

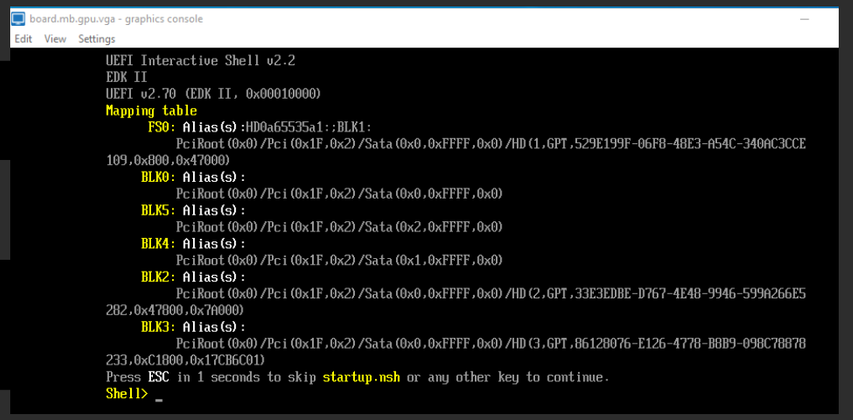
Copy the BOARDX58ICH10.fd to
%USERPROFILE%\AppData\Local\Programs\Simics\simics-qsp-x86-6.0.57\targets\qsp-x86\images

Open a Command Prompt

```
$> cd %USERPROFILE%\simics-projects\my-simics-project-1
```

Run the qsp-modern-core script :

```
$> .\simics targets/qsp-x86/qsp-modern-core.simics
simics> run    (Press “F2” at the logo)
```



tianocore

UEFI SHELL COMMANDS

Commands from the Command Line Interface

Common Shell Commands for Debugging

```
help
mm
mem
memmap
drivers
devices
devtree
dh
load
dmpstore
pci
stall
bcfg
```

"**-b**" is the command line parameter for breaking after each page.

Shell Help

```
Shell> help -b
```

board.mb.gpu.vga - graphics console

Edit View Settings

```
acpiview      - Display ACPI Table information.
alias         - Displays, creates, or deletes UEFI Shell
attrib        - Displays or modifies the attributes of fi
bcfg          - Manages the boot and driver options that
cd            - Displays or changes the current directory
cls           - Clears the console output and optionally
.
comp          - Compares the contents of two files on a b
connect       - Binds a driver to a specific device and s
cp            - Copies one or more files or directories t
date          - Displays and sets the current date for th
dblk          - Displays one or more blocks from a block
devices       - Displays the list of devices managed by U
devtree       - Displays the UEFI Driver Model compliant
dh            - Displays the device handles in the UEFI e
disconnect    - Disconnects one or more drivers from the
dmem          - Displays the contents of system or device
dmpstore      - Manages all UEFI variables.
drivers       - Displays the UEFI driver list.
drvcfg        - Invokes the driver configuration.
drvdiag       - Invokes the Driver Diagnostics Protocol.
echo          - Controls script file command echoing or d
edit          - Provides a full screen text editor for AS
eficompress   - Compresses a file using UEFI Compression
efidecompress - Decompresses a file using UEFI Decompress
else          - Identifies the code executed when 'if' is
endfor        - Ends a 'for' loop.
endif         - Ends the block of a script controlled by
exit          - Exits the UEFI Shell or the current scrip
for           - Starts a loop based on 'for' syntax.
Press ENTER to continue or 'Q' break:_
```

board.mb.gpu.vga - graphics console

Edit View Settings

```
mm            - Displays or modifies MEM/MMIO/IO/PCI/PCIE address space.
mode          - Displays or changes the console output device mode.
mv            - Moves one or more files to a destination within or between file systems.
openinfo      - Displays the protocols and agents associated with a handle.
parse         - Retrieves a value from a standard format output file.
pause         - Pauses a script and waits for an operator to press a key.
pci           - Displays PCI device list or PCI function configuration space and PCIe extended
configuration space.
ping          - Ping the target host with an IPv4 stack.
ping6         - Ping a target machine with UEFI IPv6 network stack.
reconnect     - Reconnects drivers to the specific device.
reset         - Resets the system.
rm            - Deletes one or more files or directories.
sermode       - Sets serial port attributes.
set           - Displays or modifies UEFI Shell environment variables.
setsize       - Adjusts the size of a file.
setvar        - Displays or modifies a UEFI variable.
Press ENTER to continue or 'Q' break:
shift         - Shifts in-script parameter positions.
smbiosview    - Displays SMBIOS information.
stall         - Stalls the operation for a specified number of microseconds.
time          - Displays or sets the current time for the system.
timezone      - Displays or sets time zone information.
touch         - Updates the filename timestamp with the current system date and time.
type          - Sends the contents of a file to the standard output device.
unload        - Unloads a driver image that was already loaded.
ver           - Displays UEFI Firmware version information.
vol           - Displays or modifies information about a disk volume.

Help usage:help [cmd|pattern|special] [-usage] [-verbose] [-section name][-b]
Shell> _
```

Shell “memmap”

```
Shell> memmap
```

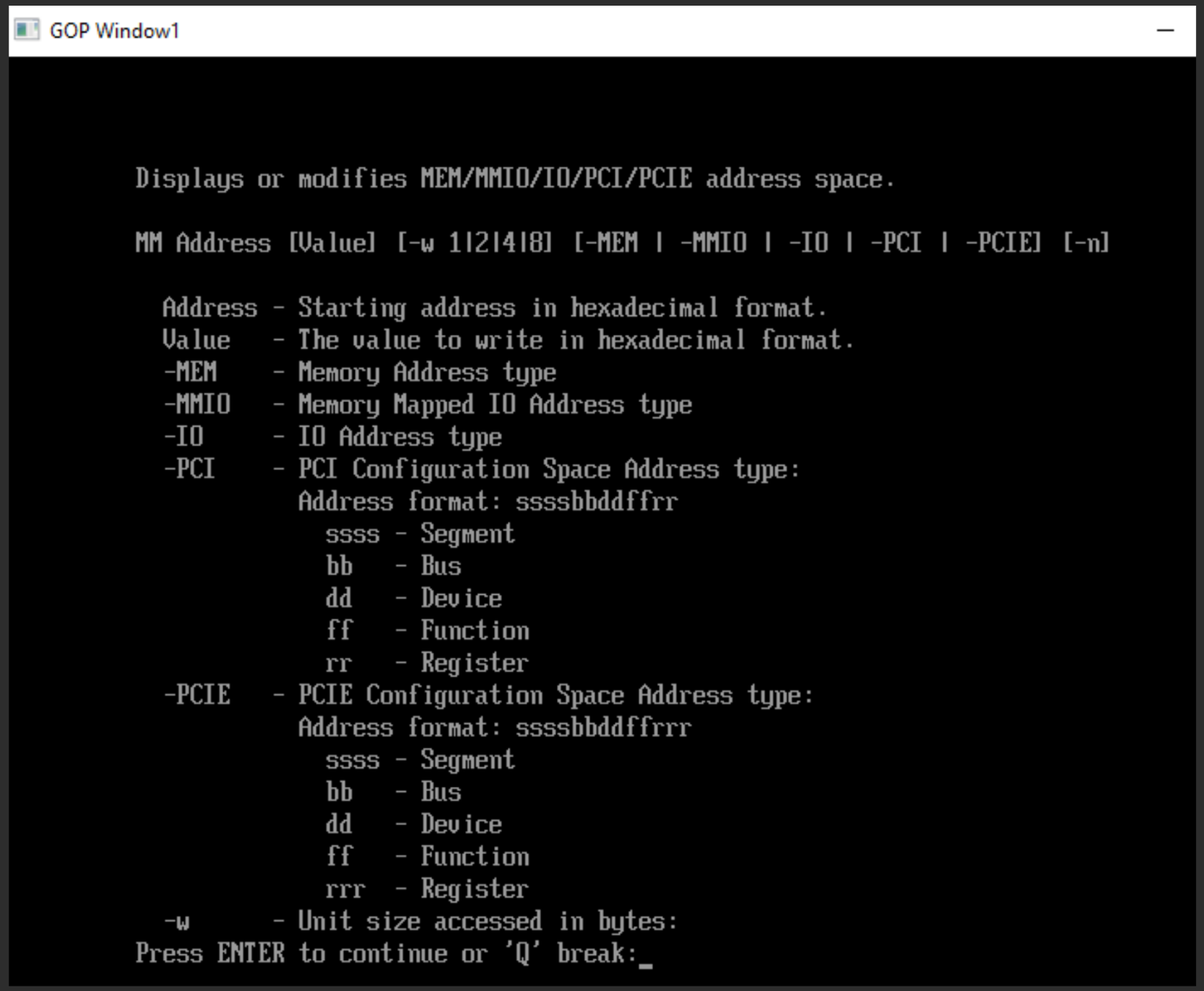
Displays the memory map maintained by the UEFI environment

```
Reserved   00000000DEFEE000-00000000DEFF1FFF 0000000000000004 000000000000000F
ACPI_Recl  00000000DEFF2000-00000000DEFF9FFF 0000000000000008 000000000000000F
ACPI_NVS   00000000DEFFA000-00000000DEFFDFFF 0000000000000004 000000000000000F
BS_Data    00000000DEFFE000-00000000DF1FFFFF 0000000000000202 000000000000000F
Available  00000000DF200000-00000000DF2E3FFF 00000000000000E4 000000000000000F
BS_Data    00000000DF2E4000-00000000DF303FFF 0000000000000020 000000000000000F
BS_Code    00000000DF304000-00000000DF33CFFF 0000000000000039 000000000000000F
BS_Data    00000000DF33D000-00000000DF552FFF 0000000000000216 000000000000000F
BS_Code    00000000DF553000-00000000DF57FFFF 000000000000002D 000000000000000F
ACPI_NVS   00000000DF580000-00000000DF7FFFFF 0000000000000280 000000000000000F
Available  0000000100000000-00000001FFEFFFFF 00000000000FFF00 000000000000000F
Reserved   00000000DF800000-00000000DFFFFFFF 0000000000000800 000000000000000F
Reserved   00000000E0000000-00000000EFFFFFFF 0000000000010000 0000000000000001

  Reserved  :         67,643 Pages (277,065,728 Bytes)
  LoaderCode:            316 Pages (1,294,336 Bytes)
  LoaderData:              0 Pages (0 Bytes)
  BS_Code   :            889 Pages (3,641,344 Bytes)
  BS_Data   :          6,593 Pages (27,004,928 Bytes)
  RT_Code   :             48 Pages (196,608 Bytes)
  RT_Data   :            226 Pages (925,696 Bytes)
  ACPI_Recl :             11 Pages (45,056 Bytes)
  ACPI_NVS  :          3,965 Pages (16,240,640 Bytes)
  MMIO      :              0 Pages (0 Bytes)
  MMIO_Port :              0 Pages (0 Bytes)
  PalCode   :              0 Pages (0 Bytes)
  Available :      1,951,573 Pages (7,993,643,008 Bytes)
  Persistent:              0 Pages (0 Bytes)
             --------------

Total Memory:          7,670 MB (8,042,991,616 Bytes)
Shell> _
```

Shell “mm”

```
Shell> mm -? -b
```

Help for “mm” command shows options for different types of memory and I/O that can be modified



Shell “mm”

```
**Shell> mm df33d000
```

```
Shell> mm df33d000
MEM  0x00000000DF33D000 : 0x00 >

MEM  0x00000000DF33D001 : 0x00 >

MEM  0x00000000DF33D002 : 0x00 >

MEM  0x00000000DF33D003 : 0x00 >

MEM  0x00000000DF33D004 : 0x00 >

MEM  0x00000000DF33D005 : 0x00 >

MEM  0x00000000DF33D006 : 0x00 >

MEM  0x00000000DF33D007 : 0x00 > q

Shell> _
```

**Pick a location from the MemMap command on Previous slide

```
BS_Data   00000000DF33D000-00000000DF552FFF 0000000000000216 000000
```

MM in can display / modify any location

Try

```
Shell> mm 0000
```

“q” to quit

Shell “mem”

```
Shell> mem
```

Displays the contents of the system or device memory

without arguments, displays the system memory configuration.

```
Shell> mem
Memory Address 00000000DEFED018 78 Bytes
  DEFED018: 49 42 49 20 53 59 53 54-46 00 02 00 78 00 00 00  *IBI SYSTF...x...*
  DEFED028: 7E 7B 92 D4 00 00 00 00-98 8A FD DE 00 00 00 00  *~{.............*
  DEFED038: 00 00 01 00 00 00 00 00-98 DB 60 DE 00 00 00 00  *..........`.....*
  DEFED048: C0 20 EC DD 00 00 00 00-98 31 FA DD 00 00 00 00  *. .......1......*
  DEFED058: A0 74 0F DE 00 00 00 00-18 CF 60 DE 00 00 00 00  *.t........`.....*
  DEFED068: 30 24 EC DD 00 00 00 00-98 DB FE DE 00 00 00 00  *0$..............*
  DEFED078: 80 3D 32 DF 00 00 00 00-0B 00 00 00 00 00 00 00  *.=2.............*
  DEFED088: 98 DC FE DE 00 00 00 00-                         *........*

Valid EFI Header at Address 00000000DEFED018
----------------------------------------
System: Table Structure size 00000078 revision 00020046
ConIn (00000000DDEC20C0) ConOut (00000000DE0F74A0) StdErr (00000000DDEC2430)
Runtime Services 00000000DEFEDB98
Boot Services    00000000DF323D80
SAL System Table 0000000000000000
ACPI Table       00000000DEFF9000
ACPI 2.0 Table   00000000DEFF9014
MPS Table        0000000000000000
SMBIOS Table     0000000000000000
Shell> _
```

UEFI System Table Pointer

Shell “Drivers”

```
Shell> drivers -b
```

```
                T   D
D               Y C I
R               P F A
V  VERSION      E G G #D #C DRIVER NAME                          IMAGE NAME
== ======== = = = == == ================================== ==========
5F 0000000A B - -  1 31 PCI Bus Driver                      PciBusDxe
60 00000030 ? - -  -  - Usb Xhci Driver                     XhciDxe
61 00000030 D - -  2  - Usb Ehci Driver                     EhciDxe
62 00000020 D - -  6  - Usb Uhci Driver                     UhciDxe
63 0000000A D - -  8  - Usb Bus Driver                      UsbBusDxe
64 00000011 ? - -  -  - Usb Mass Storage Driver             UsbMassStorageDxe
65 0000000A ? - -  -  - Usb Keyboard Driver                 UsbKbDxe
66 0000000A D - -  6  - Generic Disk I/O Driver             DiskIoDxe
67 0000000B B - -  1  3 Partition Driver(MBR/GPT/El Torito) PartitionDxe
6A 0000000A D - -  1  - FAT File System Driver              Fat
6B 0000000A D - -  1  - Graphics Console Driver             GraphicsConsoleDxe
6C 0000000A D - -  1  - Platform Console Management Driver  ConPlatformDxe
6D 0000000A D - -  1  - Platform Console Management Driver  ConPlatformDxe
6E 0000000A B - -  1  1 Console Splitter Driver             ConSplitterDxe
6F 0000000A ? - -  -  - Console Splitter Driver             ConSplitterDxe
70 0000000A ? - -  -  - Console Splitter Driver             ConSplitterDxe
71 0000000A B - -  1  1 Console Splitter Driver             ConSplitterDxe
72 0000000A ? - -  -  - Console Splitter Driver             ConSplitterDxe
76 0000000A D - -  1  - Sata Controller Init Driver         SataController
79 0000000A D - -  1  - Simple Network Protocol Driver      SnpDxe
7A 0000000A B - -  1  1 VLAN Configuration Driver           VlanConfigDxe
7B 0000000A B - -  1  3 MNP Network Service Driver          MnpDxe
7C 0000000A B - -  1  1 ARP Network Service Driver          ArpDxe
7D 0000000A B - -  1  2 DHCP Protocol Driver                Dhcp4Dxe
7E 0000000A B - -  2 11 IP4 Network Service Driver          Ip4Dxe
Press ENTER to continue or 'Q' break:_
```

Displays the UEFI driver list.

To get a description of each section in the list, (top header)

Use:

```
Shell> drivers -?
```

Shell “Devices”

```
Shell> devices -b
```

Displays a list of devices that UEFI drivers manage.

```
      T   D
      Y C I
      P F A
CTRL  E G G #P  #D  #C  Device Name
==== = = = == == === ==========================================
  35 R - -  0   1   31 PciRoot(0x0)
  73 D - -  1   0    0 Primary Console Input Device
  74 D - -  1   0    0 Primary Console Output Device
  99 D - -  1   0    0 PciRoot(0x0)/Pci(0x0,0x0)
  9A D - -  1   0    0 PciRoot(0x0)/Pci(0x1,0x0)
  9B D - -  1   0    0 PciRoot(0x0)/Pci(0x2,0x0)
  9C D - -  1   0    0 PciRoot(0x0)/Pci(0x3,0x0)
  9D D - -  1   0    0 PciRoot(0x0)/Pci(0x4,0x0)
  9E D - -  1   0    0 PciRoot(0x0)/Pci(0x5,0x0)
  9F D - -  1   0    0 PciRoot(0x0)/Pci(0x7,0x0)
  A0 B - -  1   1    1 QEMU Video PCI Adapter
  A1 D - -  1   0    0 PciRoot(0x0)/Pci(0x10,0x0)
  A2 D - -  1   0    0 PciRoot(0x0)/Pci(0x10,0x1)
  A3 D - -  1   0    0 PciRoot(0x0)/Pci(0x11,0x0)
  A4 D - -  1   0    0 PciRoot(0x0)/Pci(0x11,0x1)
  A5 D - -  1   0    0 PciRoot(0x0)/Pci(0x13,0x0)
  A6 D - -  1   0    0 PciRoot(0x0)/Pci(0x14,0x0)
  A7 D - -  1   0    0 PciRoot(0x0)/Pci(0x14,0x1)
  A8 D - -  1   0    0 PciRoot(0x0)/Pci(0x14,0x2)
  A9 D - -  1   0    0 PciRoot(0x0)/Pci(0x14,0x3)
  AA B X X  1   1    1 PciRoot(0x0)/Pci(0x19,0x0)
  AB D - -  1   2    0 Usb Universal Host Controller
  AC D - -  1   2    0 Usb Universal Host Controller
  AD D - -  1   2    0 Usb Universal Host Controller
  AE D - -  1   2    0 Enhanced Host Controller (USB 2.0)
Press ENTER to continue or 'Q' break:_
```

```
  DC D - -  2   1    0 PXE Controller
  DD D - -  1   1    0 PXE Controller
  DE D - -  1   1    0 PXE Controller
  DF D - -  1   1    0 PciRoot(0x0)/Pci(0x19,0x0)/MAC(0017A0000000,0x0)/IPv4(0.0.
  E0 D - -  1   1    0 PXE Controller
  E1 D - -  1   1    0 PXE Controller
  E2 B - -  1   1    1 IPv6(Not started)
  E3 D - -  2   1    0 PXE Controller
  E4 B - -  1   1    1 IPv6(Not started)
  E5 D - -  2   1    0 PXE Controller
  E6 D - -  1   1    0 PXE Controller
  E7 D - -  1   1    0 PciRoot(0x0)/Pci(0x19,0x0)/MAC(0017A0000000,0x0)/IPv6(0000
ress ENTER to continue or 'Q' break:
:0000:0000:0000)
  E8 D - -  2   1    0 TCPv4 (Not started)
  E9 B - -  1   1    1 IPv4 (Not started)
  EA D - -  2   1    0 TCPv6(Not started)
  EB B - -  1   1    1 IPv6(Not started)
  ED D - -  1   1    0 UEFI Http Boot Controller
  EE D - -  1   0    0 PciRoot(0x0)/Pci(0x19,0x0)/MAC(0017A0000000,0x0)/IPv4(0.0.
  EF D - -  1   1    0 UEFI Http Boot Controller
  F0 D - -  1   1    0 IPv6(Not started)
  F1 D - -  1   0    0 PciRoot(0x0)/Pci(0x19,0x0)/MAC(0017A0000000,0x0)/IPv6(0000
0000:0000:0000)/Uri()
  F2 B - -  1   1    1 IPv6(Not started)
  F3 D - -  2   1    0 UDPv6 (Not started)
  F4 B - -  1   1    1 IPv4 (Not started)
  F5 D - -  2   1    0 UDPv4 (Not started)
  F6 D - -  1   0    0 PciRoot(0x0)/Pci(0x1F,0x0)/Acpi(PNP0F03,0x0)
  F7 D - -  1   0    0 PciRoot(0x0)/Pci(0x1D,0x0)/USB(0x0,0x0)
Shell> _
```

Shell “Devtree”

```
Shell> devtree -b
```

Displays tree of devices currently managed by UEFI drivers.

```
 Ctrl[03] Fv(EACAB9EA-C3C6-4438-8FD7-2270826DC0BB)
 Ctrl[35] PciRoot(0x0)
   Ctrl[99] PciRoot(0x0)/Pci(0x0,0x0)
   Ctrl[9A] PciRoot(0x0)/Pci(0x1,0x0)
   Ctrl[9B] PciRoot(0x0)/Pci(0x2,0x0)
   Ctrl[9C] PciRoot(0x0)/Pci(0x3,0x0)
   Ctrl[9D] PciRoot(0x0)/Pci(0x4,0x0)
   Ctrl[9E] PciRoot(0x0)/Pci(0x5,0x0)
   Ctrl[9F] PciRoot(0x0)/Pci(0x7,0x0)
   Ctrl[A0] QEMU Video PCI Adapter
     Ctrl[B8] PciRoot(0x0)/Pci(0xF,0x0)/AcpiAdr(0x80010100)
       Ctrl[74] Primary Console Output Device
   Ctrl[A1] PciRoot(0x0)/Pci(0x10,0x0)
   Ctrl[A2] PciRoot(0x0)/Pci(0x10,0x1)
   Ctrl[A3] PciRoot(0x0)/Pci(0x11,0x0)
   Ctrl[A4] PciRoot(0x0)/Pci(0x11,0x1)
   Ctrl[A5] PciRoot(0x0)/Pci(0x13,0x0)
   Ctrl[A6] PciRoot(0x0)/Pci(0x14,0x0)
   Ctrl[A7] PciRoot(0x0)/Pci(0x14,0x1)
   Ctrl[A8] PciRoot(0x0)/Pci(0x14,0x2)
   Ctrl[A9] PciRoot(0x0)/Pci(0x14,0x3)
   Ctrl[AA] PciRoot(0x0)/Pci(0x19,0x0)
     Ctrl[C2] Intel(R) 82567LF-2 Gigabit Network Connection
       Ctrl[C3] PciRoot(0x0)/Pci(0x19,0x0)/MAC(0017A0000000,0x0)/VenHw(D79]
BCF5FA8)
       Ctrl[C4] MNP (MAC=00-17-A0-00-00-00, ProtocolType=0x806, VlanId=0)
         Ctrl[DD] PXE Controller
       Ctrl[C5] MNP (MAC=00-17-A0-00-00-00, ProtocolType=0x800, VlanId=0)
         Ctrl[C7] IPv4 (SrcIP=0.0.0.0)
           Ctrl[C9] UDPv4 (SrcPort=68, DestPort=67)
Press ENTER to continue or 'Q' break:_
```

```
   Ctrl[AF] Usb Universal Host Controller
     Ctrl[F7] PciRoot(0x0)/Pci(0x1D,0x0)/USB(0x0,0x0)
   Ctrl[B0] Usb Universal Host Controller
   Ctrl[B1] Usb Universal Host Controller
   Ctrl[B2] Enhanced Host Controller (USB 2.0)
   Ctrl[B3] PciRoot(0x0)/Pci(0x1E,0x0)
   Ctrl[B4] Super I/O Controller
     Ctrl[BB] PS/2 Keyboard Device
       Ctrl[73] Primary Console Input Device
     Ctrl[F6] PciRoot(0x0)/Pci(0x1F,0x0)/Acpi(PNP0F03,0x0)
   Ctrl[B5] Sata Controller
     Ctrl[BE] SCSI Disk Device
     Ctrl[BC] Simics Turbo Harddrive
       Ctrl[BF] FAT File System
       Ctrl[C0] PciRoot(0x0)/Pci(0x1F,0x2)/Sata(0x0,0xFFFF,0x0)/HD(
A266E5282,0x47800,0x7A000)
       Ctrl[C1] PciRoot(0x0)/Pci(0x1F,0x2)/Sata(0x0,0xFFFF,0x0)/HD(
ress ENTER to continue or 'Q' break:
8C78878233,0xC1800,0x17CB6C01)
     Ctrl[BD] Simics Turbo Harddrive
   Ctrl[B6] PciRoot(0x0)/Pci(0x1F,0x3)
   Ctrl[B7] PciRoot(0x0)/Pci(0x1F,0x6)
 Ctrl[4E] VenHw(EBF8ED7C-0DD1-4787-84F1-F48D537DCACF)
 Ctrl[EC] PciRoot(0x0)/Pci(0x19,0x0)/MAC(0017A0000000,0x0)/VenHw(EC
3)
 Ctrl[F9] VenHw(462CAA21-7614-4503-836E-8AB6F4662331)
 Ctrl[FA] VenHw(102579A0-3686-466E-ACD8-80C087044F4A)
 Ctrl[FB] VenHw(1DDDBE15-481D-4D2B-8277-B191EAF66525)
 Ctrl[FC] VenHw(165A028F-0BB2-4B5F-8747-77592E3F6499)
 Ctrl[FD] VenHw(8E6D99EE-7531-48F8-8745-7F6144468FF2)
Shell> _
```

Shell Handle Database - “Dh”

```
Shell> dh -b
```

Dump Handle - Displays the device handles associated with UEFI drivers

```
Handle dump
01: LoadedImage(DxeCore)
02: Decompress
03: FirmwareVolume2 DevicePath(..C3C6-4438-8FD7-2270826DC0BB)) FirmwareVolumeBlock
04: EE4E5898-3914-4259-9D6E-DC7BD79403CF
05: ImageDevicePath(..87AB-47F9-A3FE-D50B76D89541)) LoadedImage(PcdDxe)
06: GetPcdInfo GetPcdInfoProtocol Pcd Pcd
07: ImageDevicePath(..A7EB-4730-8C8E-CC466A9ECC3C)) LoadedImage(ReportStatusCodeRouterRuntimeDxe)
08: SmartCardReader RscHandler
09: ImageDevicePath(..C1BC-49F8-875F-54A5D542443F)) LoadedImage(CpuIo2Dxe)
0A: CpuIo2
0B: ImageDevicePath(..A563-4561-B858-D8476F9DEFC4)) LoadedImage(Metronome)
0C: MetronomeArch
0D: ImageDevicePath(..43B7-4784-95B1-F4226CB40CEE)) LoadedImage(RuntimeDxe)
0E: RuntimeArch
0F: ImageDevicePath(..7FD6-4665-8646-88E33EF71DFC)) LoadedImage(SecurityStubDxe)
10: SecurityArch Security2Arch
11: DeferredImageLoad
12: ImageDevicePath(..FF36-4E10-93CF-A82159E777C5)) LoadedImage(ResetSystemRuntimeDxe)
13: 2DF6BA0B-7092-440D-BD04-FB091EC3F3C1 695D7835-8D47-4C11-AB22-FA8ACCE7AE7A ResetNotification Rese
tArch
14: ImageDevicePath(..AD6B-4F3A-B60B-F59899003443)) LoadedImage(DevicePathDxe)
15: DevicePathFromText DevicePathToText DevicePathUtilities
16: ImageDevicePath(..DF4C-4B6E-8232-438DCF448D0E)) LoadedImage(NullMemoryTestDxe)
17: 309DE7F1-7F5E-4ACE-B49C-531BE5AA95EF
18: ImageDevicePath(..BFBD-4882-9ECE-C80BB1C4783B)) LoadedImage(HiiDatabase)
19: HiiImageEx HIIImage ConfigKeywordHandler HIIConfigRouting HIIDatabase HIIString HIIFont
1A: SmmAccess2 ImageDevicePath(..4366-44BF-9A62-E4B29D7A2206)) LoadedImage(SmmAccess2Dxe)
1B: SmmControl2 ImageDevicePath(..A475-4624-A83E-E6FC9BB38E49)) LoadedImage(SmmControl2Dxe)
1C: DebugSupport 96F46153-97A7-4793-ACC1-FA19BF78EA97 EBCInterpreter ImageDevicePath(..73D0-11D4-B0P
ress ENTER to continue or 'Q' break:_
```

Also try `dh -d` with handle number to get more information on that handle.

Shell Handle Database - “Dh -d”

```
Shell> dh -d 76
```

Dump Handle of Device “76” - Dumps UEFI Driver Model Information

```
Shell> dh -d 76
76: ComponentName2 ComponentName DriverBinding ImageDevicePath(..274C-43B2
dImage(SataController)
   Driver Name [76]    : Sata Controller Init Driver
   Driver Image Name   : FvFile(820C59BB-274C-43B2-83EA-DAC673035A59)
   Driver Version      : 0000000A
   Driver Type         : Device
   Configuration       : NO
   Diagnostics         : NO
   Managing            :
     Ctrl[B5]          : Sata Controller
Shell> dh -d b5
B5: 0167CCC4-D0F7-4F21-A3EF-9E64B7CDCE8B 19DF145A-B1D4-453F-8507-38816676D
taPassThru IdeControllerInit PCIIO DevicePath(PciRoot(0x0)/Pci(0x1F,0x2))
   Controller Name     : Sata Controller
   Device Path         : PciRoot(0x0)/Pci(0x1F,0x2)
   Controller Type     : BUS
   Configuration       : NO
   Diagnostics         : NO
   Managed by          :
     Drv[76]           : Sata Controller Init Driver
     Drv[90]           : SCSI Bus Driver
     Drv[92]           : AtaAtapiPassThru Driver
     Drv[93]           : ATA Bus Driver
   Parent Controllers  :
     Parent[35]        : PciRoot(0x0)
   Child Controllers   :
     Child[BE]         : SCSI Disk Device
     Child[BC]         : Simics Turbo Harddrive
     Child[BD]         : Simics Turbo Harddrive
Shell> _
```

Also try `dh -d ##` with handle number of the device this driver is handling, e.g., `dh -d b5`
to get more information on that device being managed

Shell “Load”

```
Shell> load -?
```

Loads a UEFI driver into memory

```
Shell> load -? -b
Loads a UEFI driver into memory.

LOAD [-nc] file [file...]

  -nc  - Loads the driver, but does not connect the driver.
  File - Specifies a file that contains the image of the UEFI driver (wildcards
are
         permitted).

NOTES:
  1. This command loads a driver into memory. It can load multiple files at
     one time. The file name supports wildcards.
  2. If the -nc flag is not specified, this command attempts to connect the
     driver to a proper device. It might also cause previously loaded drivers
     to be connected to their corresponding devices.
  3. Use the 'UNLOAD' command to unload a driver.

EXAMPLES:
  * To load a driver:
```

tianocore

Shell “dmpstore”

```
Shell> dmpstore –all -b
```

Display the contents of the NVRAM variables

```
Variable NV+BS ’4C19049F-4137-4DD3-9C10-8B97A83FFDFA:MemoryTypeInformation’ DataSize = 0x40
  00000000: 0A 00 00 00 04 00 00 00-09 00 00 00 0D 00 00 00  *................*
  00000010: 00 00 00 00 48 00 00 00-06 00 00 00 98 00 00 00  *....H...........*
  00000020: 05 00 00 00 30 00 00 00-03 00 00 00 D6 03 00 00  *....0...........*
  00000030: 04 00 00 00 61 19 00 00-0F 00 00 00 00 00 00 00  *....a...........*
Variable NV+RT+BS ’EFIGlobalVariable:BootOrder’ DataSize = 0x12
  00000000: 00 00 01 00 02 00 03 00-04 00 05 00 06 00 07 00  *................*
  00000010: 09 00                                            *..*
Variable NV+RT+BS ’EFIGlobalVariable:Boot0009’ DataSize = 0x58
  00000000: 01 00 00 00 2C 00 45 00-46 00 49 00 20 00 49 00  *....,.E.F.I. .I.*
  00000010: 6E 00 74 00 65 00 72 00-6E 00 61 00 6C 00 20 00  *n.t.e.r.n.a.l. .*
  00000020: 53 00 68 00 65 00 6C 00-6C 00 00 00 04 07 14 00  *S.h.e.l.l.......*
  00000030: EA B9 CA EA C6 C3 38 44-8F D7 22 70 82 6D C0 BB  *......8D.."p.m..*
  00000040: 04 06 14 00 83 A5 04 7C-3E 9E 1C 4F AD 65 E0 52  *.......|>..O.e.R*
  00000050: 68 D0 B4 D1 7F FF 04 00-                         *h.......*
Variable NV+RT+BS ’EFIGlobalVariable:Boot0007’ DataSize = 0xCF
  00000000: 01 00 00 00 7B 00 55 00-45 00 46 00 49 00 20 00  *....{.U.E.F.I. .*
  00000010: 48 00 54 00 54 00 50 00-76 00 36 00 20 00 28 00  *H.T.T.P.v.6. .(.*
  00000020: 4D 00 41 00 43 00 3A 00-30 00 30 00 31 00 37 00  *M.A.C.:.0.0.1.7.*
  00000030: 41 00 30 00 30 00 30 00-30 00 30 00 30 00 30 00  *A.0.0.0.0.0.0.0.*
  00000040: 29 00 00 00 02 01 0C 00-D0 41 03 0A 00 00 00 00  *).......A......*
  00000050: 01 01 06 00 00 19 03 0B-25 00 00 17 A0 00 00 00  *........%.......*
  00000060: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  00000070: 00 00 00 00 00 00 00 00-00 00 00 03 0D 3C 00 00  *.............<..*
  00000080: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  00000090: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  000000A0: 00 00 00 00 00 00 40 00-00 00 00 00 00 00 00 00  *......@.........*
  000000B0: 00 00 00 00 00 00 00 03-18 04 00 7F FF 04 00 4E  *...............N*
  000000C0: AC 08 81 11 9F 59 4D 85-0E E2 1A 52 2C 59 B2     *.....YM....R,Y.*
Variable NV+RT+BS ’EFIGlobalVariable:Boot0006’ DataSize = 0xAE
Press ENTER to continue or ’Q’ break:_
```

Shell “pci”

```
Shell> pci -? -b
```

Display the help for the PCI command

```
Shell> pci -? -b
Displays PCI device list or PCI function configuration space and PCIe extended
configuration space.

PCI [Bus Dev [Func] [-s Seg] [-i [-ec ID]]]

  -s   - Specifies optional segment number (hexadecimal number).
  -i   - Displays interpreted information.
  -ec  - Displays detailed interpretation of specified PCIe extended capability
         ID (hexadecimal number).
  Bus  - Specifies a bus number (hexadecimal number).
  Dev  - Specifies a device number (hexadecimal number).
  Func - Specifies a function number (hexadecimal number).

NOTES:
  1. This command displays a list of all the PCI devices found in the system. It

     also displays the configuration space of a PCI device according to the
     specified bus (Bus), device (Dev), and function (Func) addresses. If the
     function address is not specified, it defaults to 0.
  2. The -i option displays verbose information for the specified PCI
     device. The PCI configuration space for the device is displayed with
     a detailed interpretation.
  3. If no parameters are specified, all PCI devices are listed.
Press ENTER to continue or 'Q' break:_
```

tianocore

Shell “pci”

```
Shell> pci -b
```

Display the list of PCI devices . . .

```
   Seg  Bus  Dev  Func
   ---  ---  ---  ----
    00   00   00    00 ==> Base System Peripherals - PIC
                Vendor 8086 Device 3400 Prog Interface 0
    00   00   01    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 8086 Device 3408 Prog Interface 0
    00   00   02    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 8086 Device 3409 Prog Interface 0
    00   00   03    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 8086 Device 340A Prog Interface 0
    00   00   04    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 8086 Device 340B Prog Interface 0
    00   00   05    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 8086 Device 340C Prog Interface 0
    00   00   07    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 8086 Device 340E Prog Interface 0
    00   00   0F    00 ==> Display Controller - VGA/8514 controller
                Vendor 4321 Device 1111 Prog Interface 0
    00   00   10    00 ==> Base System Peripherals - PIC
                Vendor 8086 Device 3425 Prog Interface 0
    00   00   10    01 ==> Base System Peripherals - PIC
                Vendor 8086 Device 3426 Prog Interface 0
    00   00   11    00 ==> Base System Peripherals - PIC
                Vendor 8086 Device 3427 Prog Interface 0
    00   00   11    01 ==> Base System Peripherals - PIC
                Vendor 8086 Device 3428 Prog Interface 0
    00   00   13    00 ==> Base System Peripherals - PIC
                Vendor 8086 Device 342D Prog Interface 20
    00   00   14    00 ==> Base System Peripherals - PIC
                Vendor 8086 Device 342E Prog Interface 0
Press ENTER to continue or 'Q' break:_
```

```
    00   00   19    00 ==> Network Controller - Ethernet controller
                Vendor 8086 Device 10CD Prog Interface 0
    00   00   1A    00 ==> Serial Bus Controllers - USB
                Vendor 8086 Device 3A67 Prog Interface 0
    00   00   1A    01 ==> Serial Bus Controllers - USB
                Vendor 8086 Device 3A68 Prog Interface 0
    00   00   1A    02 ==> Serial Bus Controllers - USB
                Vendor 8086 Device 3A69 Prog Interface 0
    00   00   1A    07 ==> Serial Bus Controllers - USB
                Vendor 8086 Device 3A6C Prog Interface 20
    00   00   1D    00 ==> Serial Bus Controllers - USB
                Vendor 8086 Device 3A64 Prog Interface 0
    00   00   1D    01 ==> Serial Bus Controllers - USB
                Vendor 8086 Device 3A65 Prog Interface 0
    00   00   1D    02 ==> Serial Bus Controllers - USB
                Vendor 8086 Device 3A66 Prog Interface 0
    00   00   1D    07 ==> Serial Bus Controllers - USB
                Vendor 8086 Device 3A6A Prog Interface 20
    00   00   1E    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 8086 Device 244E Prog Interface 1
    00   00   1F    00 ==> Bridge Device - PCI/ISA bridge
                Vendor 8086 Device 3A16 Prog Interface 0
    00   00   1F    02 ==> Mass Storage Controller - Serial ATA controller
                Vendor 8086 Device 3A22 Prog Interface 1
Press ENTER to continue or 'Q' break:
    00   00   1F    03 ==> Serial Bus Controllers - System Management Bus
                Vendor 8086 Device 3A30 Prog Interface 0
    00   00   1F    06 ==> Data Acquisition & Signal Processing Controllers
lers
                Vendor 8086 Device 3A32 Prog Interface 0
Shell> _
```

tianocore

Shell “pci 00 00 00 -i”

```
Shell> pci 00 00 00 -i
```

Display the configuration space of Bus 0, Device 0, Function 0.

```
  PCI Segment 00 Bus 00 Device 00 Func 00 [EFI 0000000000]
  00000000: 86 80 00 34 07 00 10 00-13 00 00 08 00 00 00 00  *...4............*
  00000010: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  00000020: 00 00 00 00 00 00 00 00-00 00 00 00 86 80 00 00  *................*
  00000030: 00 00 00 00 00 00 00 00-00 00 00 00 FF 00 00 00  *................*

  00000040: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  00000050: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  00000060: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  00000070: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  00000080: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  00000090: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  000000A0: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  000000B0: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  000000C0: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  000000D0: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  000000E0: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*
  000000F0: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  *................*

Vendor ID(0): 8086     Device ID(2): 3400
Command(4): 0007
  (00)I/O space access enabled:       1  (01)Memory space access enabled:    1
  (02)Behave as bus master:           1  (03)Monitor special cycle enabled:  0
  (04)Mem Write & Invalidate enabled: 0  (05)Palette snooping is enabled:    0
  (06)Assert PERR# when parity error: 0  (07)Do address/data stepping:       0
  (08)SERR# driver enabled:           0  (09)Fast back-to-back transact...:  0

Status(6): 0010
  (04)New Capabilities linked list:   1  (05)66MHz Capable:                  0
  (07)Fast Back-to-Back Capable:      0  (08)Master Data Parity Error:       0
Press ENTER to continue or 'Q' break:_
```

```
  (09)DEVSEL timing:                Fast  (11)Signaled Target Abort:        0
  (12)Received Target Abort:           0  (13)Received Master Abort:        0
  (14)Signaled System Error:           0  (15)Detected Parity Error:        0

Revision ID(8): 13 BIST(0F):  Incapable
Cache Line Size(C): 00                  Latency Timer(D): 00
Header Type(0E):    00, Single function, PCI device
Class: Base System Peripherals - PIC - Generic 8259
Shell> _
```

Shell “Stall”

```
Shell> stall 10000000
```

Stalls the operation for a specified number of microseconds

```
Shell> stall 10000000
Shell> _
```

tianocore

UEFI SHELL SCRIPTS

Use Scripting with UEFI Shell

UEFI Shell Scripts

The UEFI Shell can execute commands from a file, which is called a batch script file (**.nsh** files).

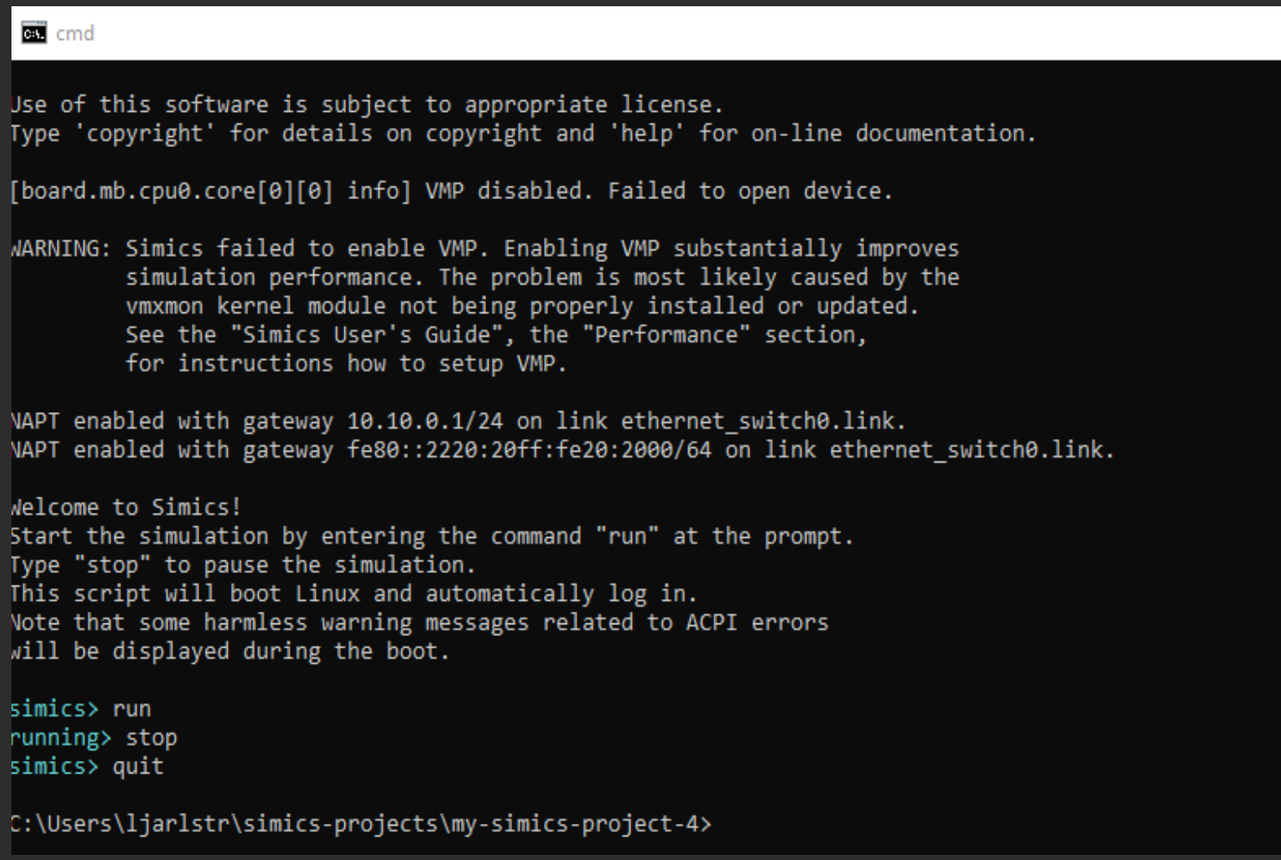
Benefits: These files allow users to simplify routine or repetitive tasks.

- Perform basic flow control.
- Allow branching and looping in a script.
- Allow users to control input and output and call other batch programs (known as script nesting).

Exit QSP UEFI Shell & Simics

- To Stop the QSP Simulation, from the Simics Command Line Prompt Window, Type: “`stop`”
  - This will stop the Simics simulation of the QSP board
  - To continue, type: “`run`”
- To Exit this Simulation, type: “`quit`”
  - This will remove all other Simics windows

```
simics> stop
simics> quit
```



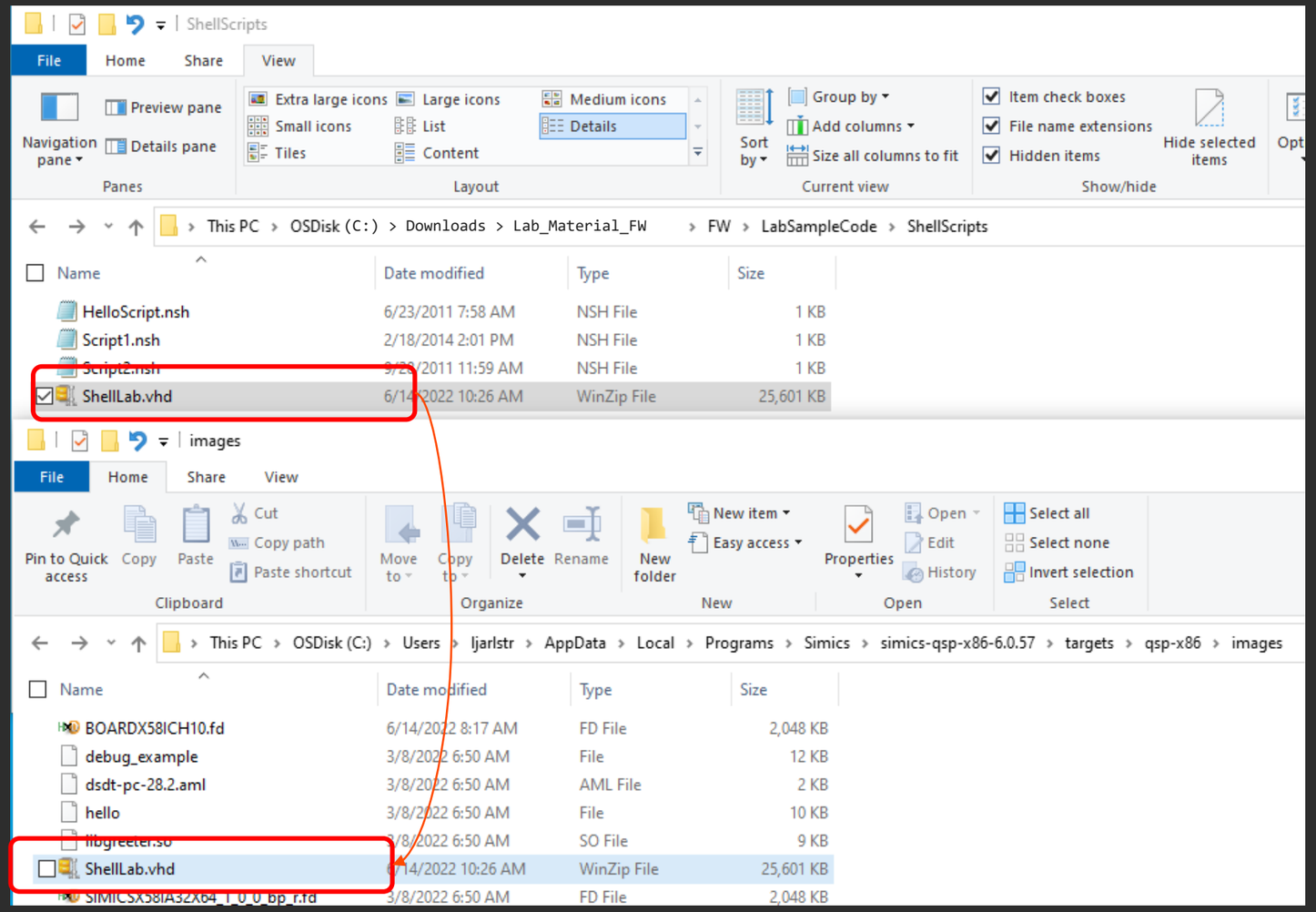
Copy ShallLab.vhd file

Copy the ShellLab.vhd

From:
.../Lab_Material_FW/FW/LabSampleCode/ShellScripts/ShallLab.vhd

to

%USERPROFILE%\AppData\Local\Programs\Simics\simics-qsp-x86-6.0.57\targets\qsp-x86\images



Update the Simics Script

Update the Simics Script to Use the ShellLab.vhd image as a file system

Edit the file: qsp-modern-core.simics from

%USERPROFILE%\
\AppData\Local\Programs\Simics\simics-qsp-cpu-6.0.4\targets\qsp-x86\qsp-modern-core.simics

Add the following Line:

`$disk1_image="%simics%/targets/qsp-x86/images/ShellLab.vhd"`

Before the "`run-command-file`" line

Save qsp-modern-core.simics

File: qsp-modern-core.simics

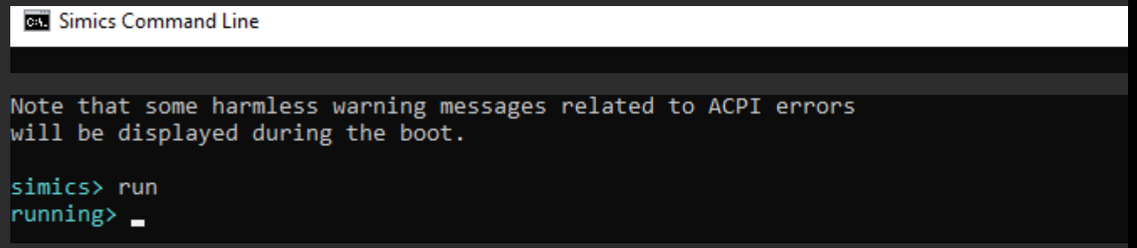
```
Decl{
decl {
! Script that runs the Quick Start Platform (QSP) with a modern
!   processor core.

params from "%simics%/targets/qsp-x86/qsp-clear-linux.simics"
default cpu_comp_class = "x86QSP2"
default num_cores = 2
default num_threads = 2
}
$disk1_image="%simics%/targets/qsp-x86/images/ShallLab.vhd“

run-command-file "%simics%/targets/qsp-x86/qsp-clear-linux.simics"
```

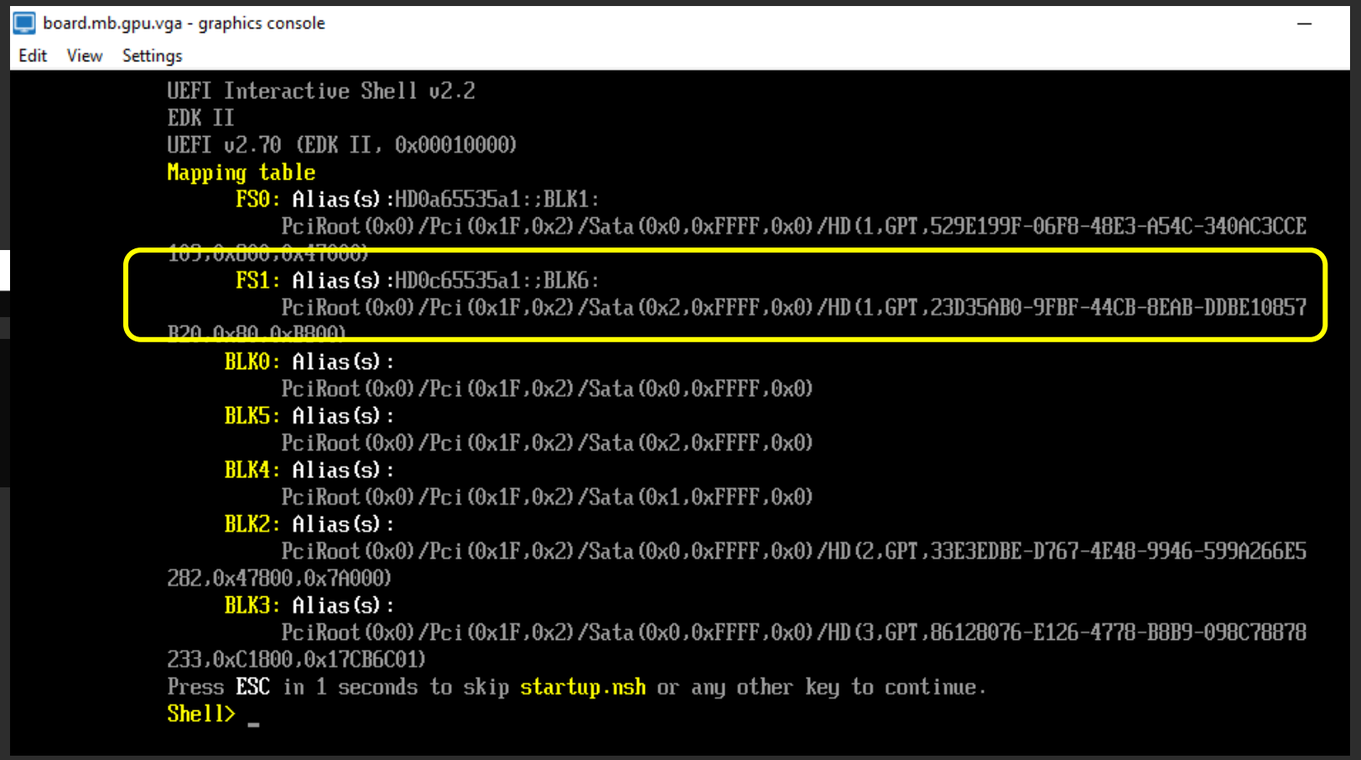
Run Simics QSP Script

Re-run the `qsp-modern-core` script from the Simics Command Prompt:

```
$> ./simics targets/qsp-x86/qsp-modern-core.simics
simics> run
```

(Press “F2” at the logo)





Note: now there is a “FS1” file system

Writing UEFI Shell Scripts

At the shell prompt

```
Shell> fs1:
FS1:\> edit HelloScript.nsh
```

Type : `echo Hello World`

```
UEFI EDIT helloscript.nsh                    UNICODE
echo Hello World
```

Press “F2”
Enter
Press “F3” to exit

Help Menu - Shell

```
Help

Control Key   Function Key   Command
-----------   ------------   ----------------
Ctrl-G        F1             Go To Line
Ctrl-S        F2             Save File
Ctrl-Q        F3             Exit
Ctrl-F        F4             Search
Ctrl-R        F5             Search/Replace
Ctrl-K        F6             Cut Line
Ctrl-U        F7             Paste Line
Ctrl-O        F8             Open File
Ctrl-T        F9             File Type

Use Ctrl-W to exit this help
```

Hello World Script

In the shell, **type** HelloScript for the following result:

```
FS1:\> HelloScript.nsh
FS1:\> echo Hello World
Hello World
FS1:\> _
```

UEFI Shell Script Example

Script1.nsh

```
# Simple UEFI Shell script file
echo -off
script2.nsh
if exist %cwd%Mytime.log then
     type Mytime.log
endif
echo "%HThank you.” “%VByeBye:) %N“
```

Script2.nsh

```
# Show nested scripts
time > Mytime.log
for %a run (3 1 -1)
    echo %a counting down
endfor
```

Run UEFI Shell Scripts

At the Shell prompt Type

```
Shell> fs1:
FS1:\> cd ShellScripts
FS1:\ShellScripts\> Script1
FS1:\ShellScripts\> Edit Script1.nsh
```

```
FS1:\> cd ShellScripts
FS1:\ShellScripts\> Script1.nsh
FS1:\ShellScripts\> script2.nsh
FS1:\ShellScripts\> time > Mytime.log
FS1:\ShellScripts\> for %a run (3 1 -1)
FS1:\ShellScripts\>     echo %a counting down
3 counting down
FS1:\ShellScripts\> endfor
FS1:\ShellScripts\> for %a run (3 1 -1)
FS1:\ShellScripts\>     echo %a counting down
2 counting down
FS1:\ShellScripts\> endfor
FS1:\ShellScripts\> for %a run (3 1 -1)
FS1:\ShellScripts\>     echo %a counting down
1 counting down
FS1:\ShellScripts\> endfor
FS1:\ShellScripts\> for %a run (3 1 -1)
FS1:\ShellScripts\> if exist %cwd%Mytime.log then
FS1:\ShellScripts\> echo " Thank you." " ByeBye" " :) " " Done"
 Thank you.  ByeBye  :)   Done
FS1:\ShellScripts\> _
```

Run UEFI Shell Scripts

Remove the “#” on the first line

```
UEFI EDIT Script1.nsh
echo  -off
script2.nsh
if exist %cwd%Mytime.log then
     type Mytime.log
endif
echo "%HThank you. %VByeBye:) %N"
```

Press “F2”

Enter

Press “F3” to exit

Type

```
FS1:\ShellScripts\> Script1
```

```
FS1:\ShellScripts\> Script1.nsh
FS1:\ShellScripts\> echo  -off
3 counting down
2 counting down
1 counting down
 Thank you.  ByeBye  :)   Done
FS1:\ShellScripts\>
```

tianocore

UEFI SHELL GLOBAL VARIABLES

Use BCFG and DmpStore

Show the UEFI Boot Variables

At the Shell Prompt:

```
Shell> FS1:

FS1:> BCFG Boot Dump
```

```
Option: 02. Variable: Boot0002
  Desc    - UEFI Simics Turbo Harddrive VT00002
  DevPath - PciRoot(0x0)/Pci(0x1F,0x2)/Sata(0x0,0xFFFF,0x0)
  Optional- Y
Option: 03. Variable: Boot0003
  Desc    - UEFI Simics Turbo Harddrive VT00003
  DevPath - PciRoot(0x0)/Pci(0x1F,0x2)/Sata(0x2,0xFFFF,0x0)
  Optional- Y
Option: 04. Variable: Boot0004
  Desc    - UEFI PXEv4 (MAC:0017A0000000)
  DevPath - PciRoot(0x0)/Pci(0x19,0x0)/MAC(0017A0000000,0x0)/IPv4(0.0.0.0)
  Optional- Y
Option: 05. Variable: Boot0005
  Desc    - UEFI PXEv6 (MAC:0017A0000000)
  DevPath - PciRoot(0x0)/Pci(0x19,0x0)/MAC(0017A0000000,0x0)/IPv6(0000:0000:0000:0000:000
:0000)
  Optional- Y
Option: 06. Variable: Boot0006
  Desc    - UEFI HTTPv4 (MAC:0017A0000000)
  DevPath - PciRoot(0x0)/Pci(0x19,0x0)/MAC(0017A0000000,0x0)/IPv4(0.0.0.0)/Uri()
  Optional- Y
Option: 07. Variable: Boot0007
  Desc    - UEFI HTTPv6 (MAC:0017A0000000)
  DevPath - PciRoot(0x0)/Pci(0x19,0x0)/MAC(0017A0000000,0x0)/IPv6(0000:0000:0000:0000:000
:0000)/Uri()
  Optional- Y
Option: 08. Variable: Boot0009
  Desc    - EFI Internal Shell
  DevPath - Fv(EACAB9EA-C3C6-4438-8FD7-2270826DC0BB)/FvFile(7C04A583-9E3E-4F1C-AD65-E0526
  Optional- N
FS1:\> _
```

Use the Dmpstore to Show the Boot Order

At the Shell Prompt:

```
FS1:> Dmpstore BootOrder
```

```
FS1:\> Dmpstore BootOrder
Variable NV+RT+BS 'EFIGlobalVariable:BootOrder' DataSize = 0x12
  00000000: 00 00 01 00 02 00 03 00-04 00 05 00 06 00 07 00  *................*
  00000010: 09 00                                          *..*
FS1:\> _
```

Use the BCFG to Move a boot item

Use BCFG to Move the 8th boot item too 1st location (location 0).

Then verify using the “dmpstore”

(Hint: use BCFG -? –b for help menu)

The dmpstore output should look like the screen shot

Result

```
FS1:\> Dmpstore BootOrder
Variable NV+RT+BS 'EFIGlobalVariable:BootOrder' DataSize = 0x12
  00000000: 09 00 00 00 01 00 02 00-03 00 04 00 05 00 06 00  *.......
  00000010: 07 00                                           *..*
FS1:\> _
```

Use the BCFG to Add a boot item

Use the file from on FS1 `/OldShell/Shell_FullX64.efi` and use BCFG to Add a 08 entry for a new boot option with `Shell_FullX64.efi`

(hint: check bcfg –h for adding a boot entry)

Then verify using the "`BCFG Boot Dump`"

```
FS1:\OldShell\> bcfg boot add 08 Shell_FullX64.efi "Old EFI Shell 1.0"
Target = 0008.
bcfg: Add Boot0008 as 8
```

After the `bcfg` add, The output should look like

Result

```
Option: 08. Variable: Boot0008
  Desc    - Old EFI Shell 1.0
  DevPath - PciRoot(0x0)/Pci(0x1F,0x2)/Sata(0x2,0xFFFF,0x0)/HD(1,GPT,23D35AB0-9FBF-44CB-8EAB-DDBE108
57B20,0x80,0xB800)/\OldShell\Shell_FullX64.efi
  Optional- N
Option: 09. Variable: Boot0007
  Desc    - UEFI HTTPv6 (MAC:0017A0000000)
  DevPath - PciRoot(0x0)/Pci(0x19,0x0)/MAC(0017A0000000,0x0)/IPv6(0000:0000:0000:0000:0000:0000:0000
:0000)/Uri()
  Optional- Y
FS1:\OldShell\> _
```

Verify Results from BCFG Commands

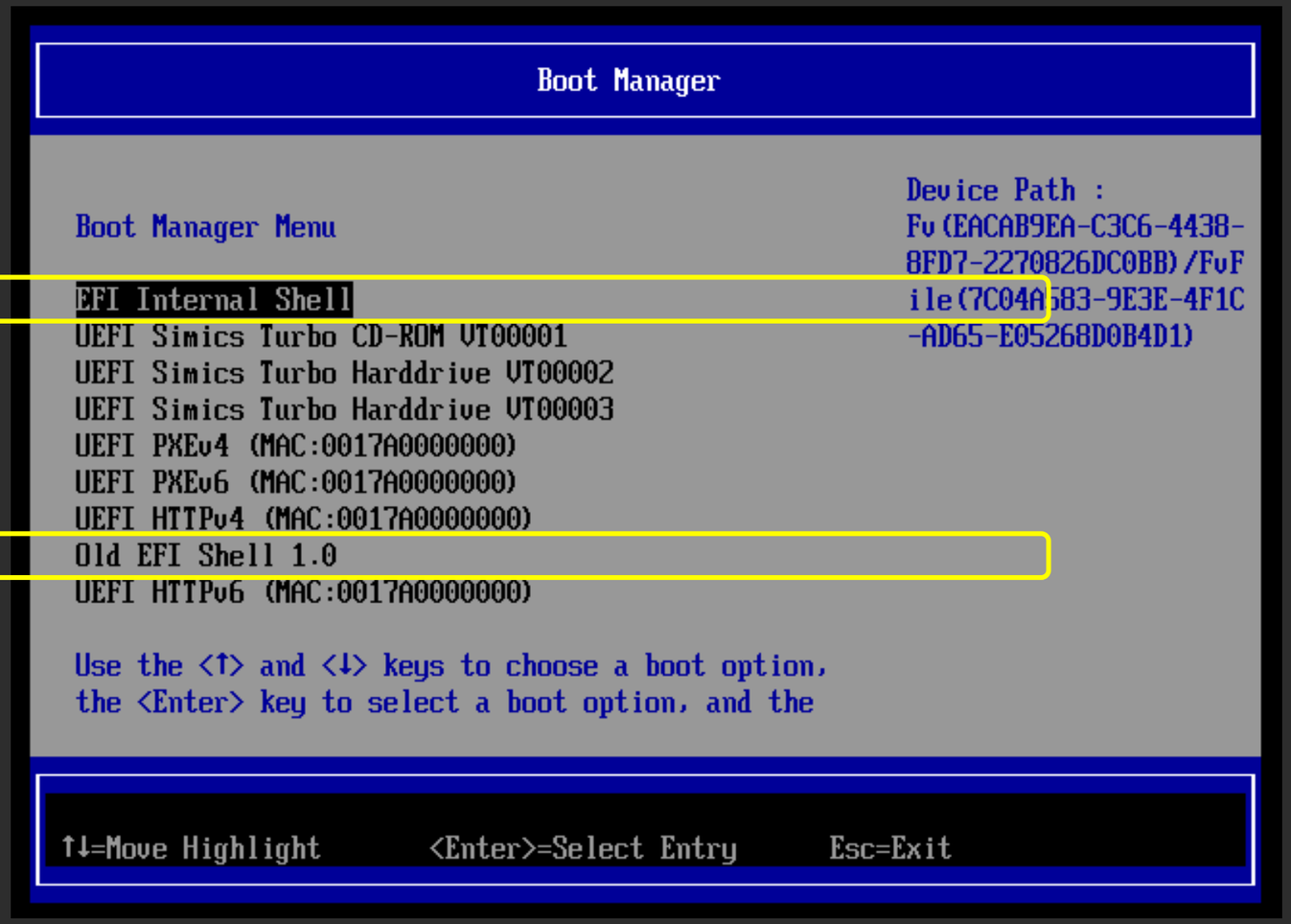
At the Shell Prompt Type “exit” to get back to the BIOS Setup

Press escape

Select “Boot Manager”

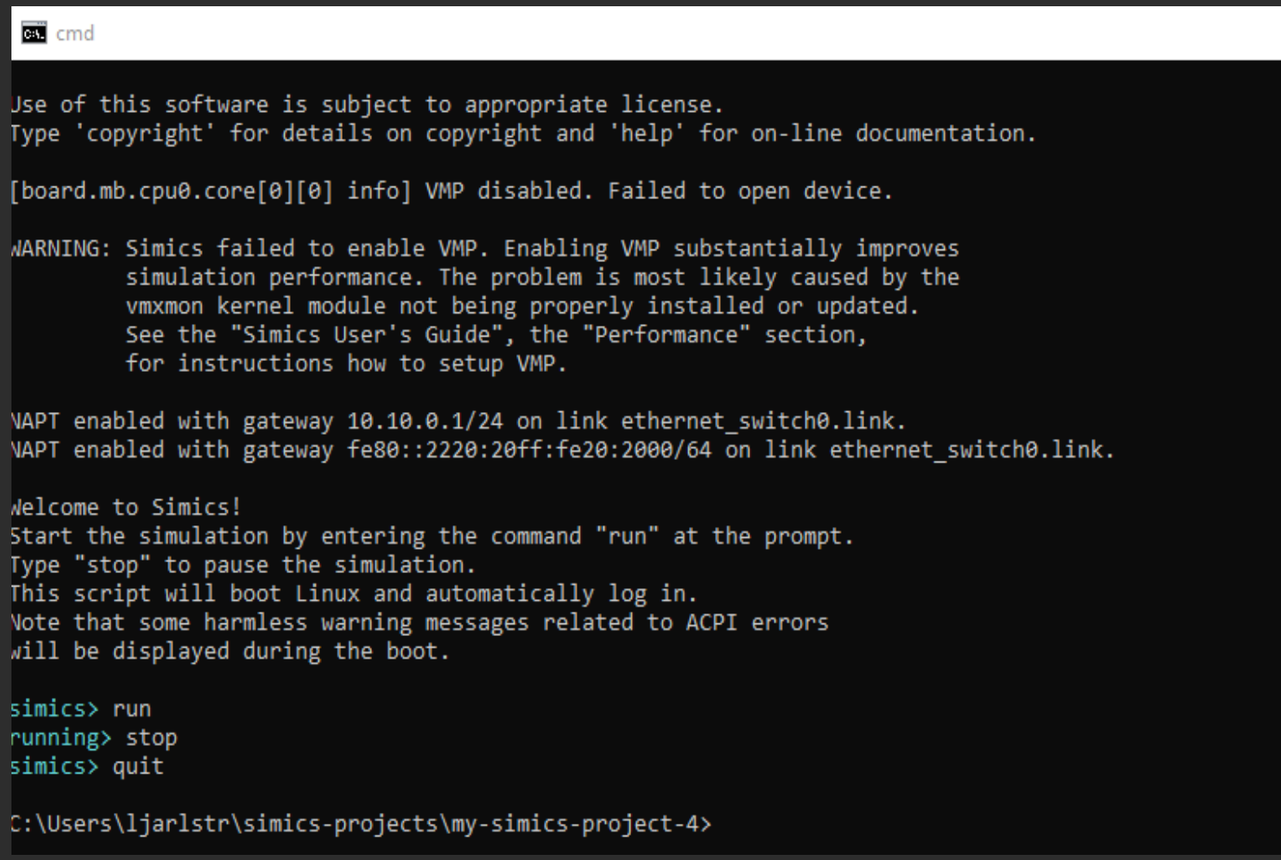
Verify:

- EFI Internal Shell – item 1
- Old EFI Shell 1.0 – item 8



Exit QSP UEFI Shell & Simics

- To Stop the QSP Simulation, from the Simics Command Line Prompt Window, Type: “`stop`”
  - This will stop the Simics simulation of the QSP board
  - To continue, type: “`run`”
- To Exit this Simulation, type: “`quit`”
  - This will remove all other Simics windows



Summary

- Run UEFI Shell (Simcs QSP Emulation)
- Run UEFI Shell Commands
- Run UEFI Shell Scripts

tianocore

Questions?

Return to Main Training Page

Return to Training Table of contents for next presentation link

tianocore

ACKNOWLEDGEMENTS