UEFI & EDK II TRAINING

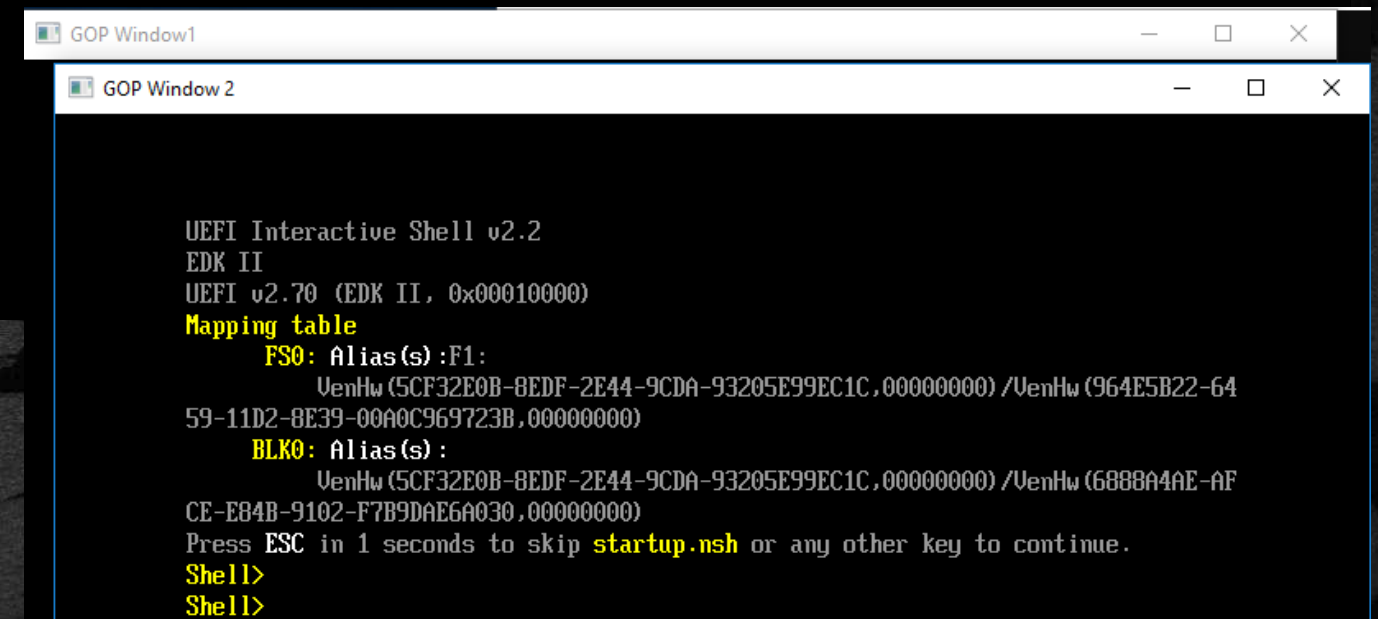## UEFI SHELL LAB w/ WINDOWS EMULATION

tianocore.org

LESSON OBJECTIVE

Run UEFI Shell (Windows Emulation)

Run UEFI Shell Commands

Run UEFI Shell Scripts

UEFI SHELL LAB WITH WIN EMULATOR

tianocore

From the VS command prompt

```
CD C:\FW\edk2-ws
# set up PACKAGES_PATH
$> set WORKSPACE=%CD%
$> set PACKAGES_PATH=%WORKSPACE%\edk2;%WORKSPACE%\edk2-libc
$> cd edk2
$> edksetup Rebuild
$> Build -a X64
$> RunEmulator.bat
```

GOP Window1

GOP Window 2

```
UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Mapping table
        FS0: Alias(s):F1:
              VenHw(5CF32E0B-8EDF-2E44-9CDA-93205E99EC1C,00000000)/VenHw(964E5B22-64
59-11D2-8E39-00A0C969723B,00000000)
        BLK0: Alias(s):
              VenHw(5CF32E0B-8EDF-2E44-9CDA-93205E99EC1C,00000000)/VenHw(6888A4AE-AF
CE-E84B-9102-F7B9DAE6A030,00000000)
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.
Shell>
Shell>
```
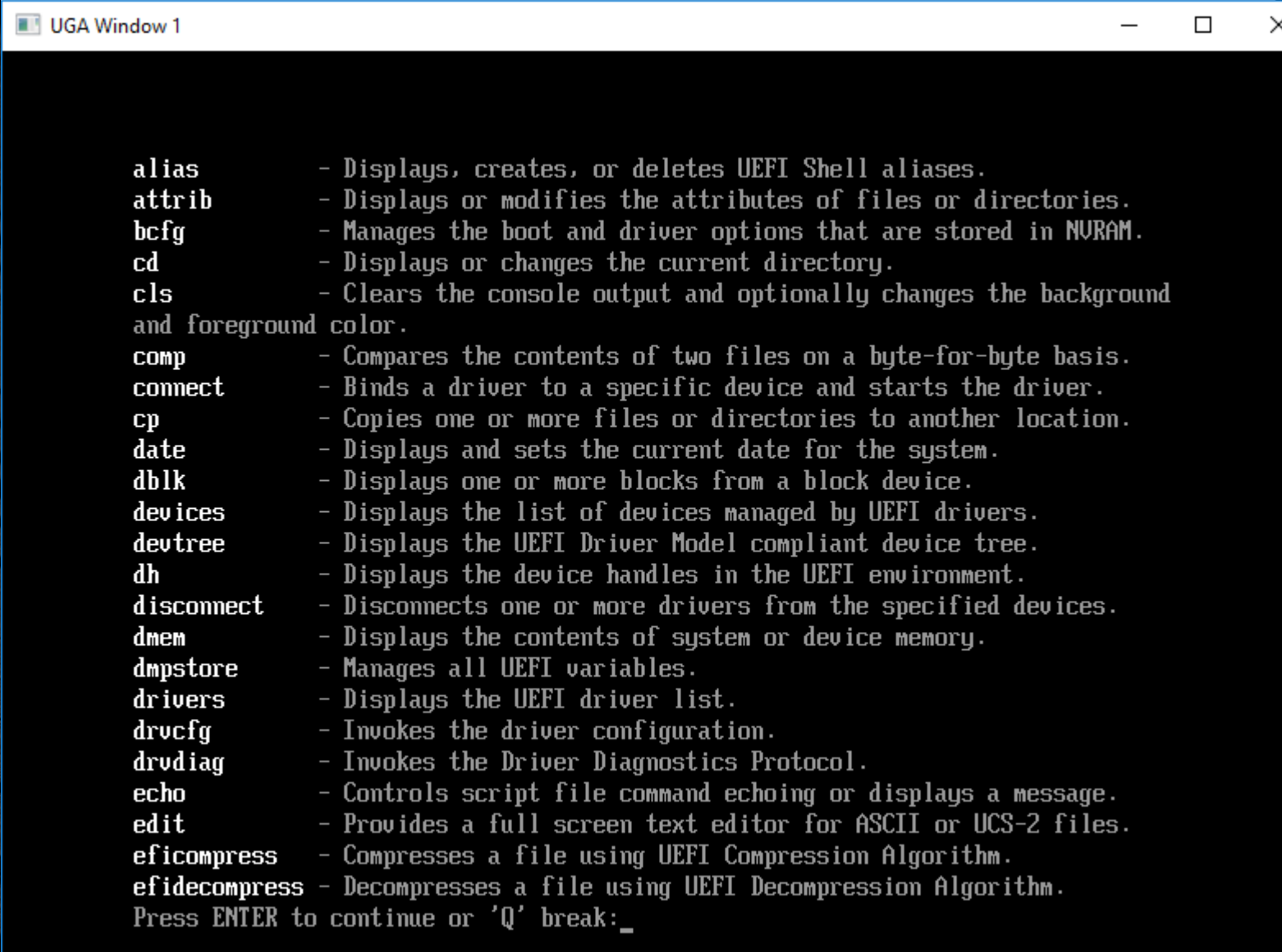
UEFI SHELL COMMANDS

Commands from the Command Line Interface

tianocore

```
help
mm
mem
memmap
drivers
devices
devtree
dh
Load
dmpstore
stall
```

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

tianocore

`Shell> help -b`

UGA Window 1

```
alias         - Displays, creates, or deletes UEFI Shell aliases.
attrib        - Displays or modifies the attributes of files or directories.
bcfg          - Manages the boot and driver options that are stored in NVRAM.
cd            - Displays or changes the current directory.
cls           - Clears the console output and optionally changes the background
and foreground color.
comp          - Compares the contents of two files on a byte-for-byte basis.
connect       - Binds a driver to a specific device and starts the driver.
cp            - Copies one or more files or directories to another location.
date          - Displays and sets the current date for the system.
dblk          - Displays one or more blocks from a block device.
devices       - Displays the list of devices managed by UEFI drivers.
devtree       - Displays the UEFI Driver Model compliant device tree.
dh            - Displays the device handles in the UEFI environment.
disconnect    - Disconnects one or more drivers from the specified devices.
dmem          - Displays the contents of system or device memory.
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 displays a message.
edit          - Provides a full screen text editor for ASCII or UCS-2 files.
eficompress   - Compresses a file using UEFI Compression Algorithm.
efidecompress - Decompresses a file using UEFI Decompression Algorithm.
Press ENTER to continue or 'Q' break:_
```

7

tianocore

```
Shell> memmap
```

Displays the memory map maintained by the UEFI environment

```
RT_Data      000002B208574000-000002B208574FFF 0000000000000001 800000000000000F
BS_Data      000002B208575000-000002B20859CFFF 0000000000000028 000000000000000F
RT_Data      000002B20859D000-000002B20859FFFF 0000000000000003 800000000000000F
MMIO         000002B200580000-000002B20058BFFF 000000000000000C 8000000000000001

    Reserved  :               0 Pages (0 Bytes)
    LoaderCode:             307 Pages (1,257,472 Bytes)
    LoaderData:               0 Pages (0 Bytes)
    BS_Code   :           1,239 Pages (5,074,944 Bytes)
    BS_Data   :           5,936 Pages (24,313,856 Bytes)
    RT_Code   :              97 Pages (397,312 Bytes)
    RT_Data   :             193 Pages (790,528 Bytes)
    ACPI_Recl :               0 Pages (0 Bytes)
    ACPI_NVS  :               0 Pages (0 Bytes)
    MMIO      :              12 Pages (49,152 Bytes)
    MMIO_Port :               0 Pages (0 Bytes)
    PalCode   :               0 Pages (0 Bytes)
    Available :          24,996 Pages (102,383,616 Bytes)
    Persistent:               0 Pages (0 Bytes)
                     --------------
Total Memory:             128 MB (134,217,728 Bytes)
Shell> _
```

tianocore

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

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

GOP Window1   —

```
Displays or modifies MEM/MMIO/IO/PCI/PCIE address space.

MM Address [Value] [-w 1|2|4|8] [-MEM | -MMIO | -IO | -PCI | -PCIE] [-n]

    Address  -  Starting address in hexadecimal format.
    Value    -  The value to write in hexadecimal format.
    -MEM     -  Memory Address type
    -MMIO    -  Memory Mapped IO Address type
    -IO      -  IO Address type
    -PCI     -  PCI Configuration Space Address type:
                Address format: ssssbbddffrr

                  ssss - Segment
                  bb   - Bus
                  dd   - Device
                  ff   - Function
                  rr   - Register
    -PCIE    -  PCIE Configuration Space Address type:
                Address format: ssssbbddffrrr

                  ssss - Segment
                  bb   - Bus
                  dd   - Device
                  ff   - Function
                  rrr  - Register
    -w       -  Unit size accessed in bytes:
Press ENTER to continue or 'Q' break:_
```

```
Shell> mm **
```

```
Shell> mm 2b208575000
MEM   0x000002B208575000 : 0x70 >

MEM   0x000002B208575001 : 0x68 >

MEM   0x000002B208575002 : 0x64 >

MEM   0x000002B208575003 : 0x30 >

MEM   0x000002B208575004 : 0x01 >

MEM   0x000002B208575005 : 0x00 >

MEM   0x000002B208575006 : 0x00 > q

Shell> _
```

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

```
BS_Data    000002B208575000-000002B20859CFFF 0000000000000028 00000
```

MM in can display / modify any location

Do **not** try in Win Emulator

```
Shell> mm 0000
```

"q" to quit

```
Shell> mem
```

Displays the contents of the system or device memory without arguments, displays the system memory configuration.

```
061EC170: AF AF AF AF AF AF AF AF-AF AF AF AF AF AF AF AF  *................*
061EC180: AF AF AF AF AF AF AF AF-AF AF AF AF AF AF AF AF  *................*

Valid EFI Header at Address 00000000061EBF90

--------------------------------------------------------

System: Table Structure size 00000048 revision 0002001F
ConIn (000000000A3271F4) ConOut (0000000005373114) StdErr (000000000A3273A4)
Runtime Services 00000000061EBF10
Boot Services    0000000000415C40
SAL System Table 0000000000000000
ACPI Table       0000000000000000
ACPI 2.0 Table   0000000000000000
MPS Table        0000000000000000
SMBIOS Table     00000000622F000
Shell> _
```

UEFI System Table Pointer

```
Shell> drivers -b
```

```
              T  D
              Y  C  I
D             P  F  A
R
V   VERSION   E  G  G  #D #C DRIVER NAME                           IMAGE NAME
==  ========  =  =  =  == == ===================================== ==========
47  0000000A  D  -  -  2   - Platform Console Management Driver    ConPlatformDxe
48  0000000A  D  -  -  2   - Platform Console Management Driver    ConPlatformDxe
49  0000000A  B  -  -  2   2 Console Splitter Driver               ConSplitterDxe
4A  0000000A  B  -  -  2   2 Console Splitter Driver               ConSplitterDxe
4B  0000000A  ?  -  -  -   - Console Splitter Driver               ConSplitterDxe
4C  0000000A  B  -  -  2   2 Console Splitter Driver               ConSplitterDxe
4D  0000000A  ?  -  -  -   - Console Splitter Driver               ConSplitterDxe
51  0000000A  D  -  -  2   - Graphics Console Driver               GraphicsConsoleDxe
52  0000000A  B  -  -  1   1 Serial Terminal Driver                TerminalDxe
53  0000000A  D  -  -  1   - Generic Disk I/O Driver               DiskIoDxe
54  0000000B  ?  -  -  -   - Partition Driver(MBR/GPT/El Torito)   PartitionDxe
57  0000000A  ?  -  -  -   - PCI Bus Driver                        PciBusDxe
59  0000000A  ?  -  -  -   - SCSI Bus Driver                       ScsiBus
5A  0000000A  ?  -  -  -   - Scsi Disk Driver                      ScsiDisk
5B  0000000A  B  -  -  1   4 Emu Bus Driver                        EmuBusDriver
5C  0000000A  D  -  -  2   - Emulator GOP Driver                   EmuGopDxe
5D  0000000A  D  -  -  1   - Emu Simple File System Driver         EmuSimpleFileSystem
5E  0000000A  D  -  X  1   - Emu Block I/O Driver                  EmuBlockIo
Press ENTER to continue or 'Q' break:_
```

```
Shell> devices -b
```

Displays a list of devices that UEFI drivers manage.

```
Shell> devices -b
     T  D
     Y C I
     P F A
CTRL E G G #P #D #C  Device Name
==== = = = == == === ==================================================
  1C R - - 0  1    4 VenHw(5CF32E0B-8EDF-2E44-9CDA-93205E99EC1C,00000000)
  20 R - - 0  1    1 VenHw(D3987D4B-971A-435F-8CAF-4967EB627241)/Uart(115200,8,N
,1)
  4E D - - 2  0    0 Primary Console Input Device
  4F D - - 2  0    0 Primary Console Output Device
  6F B - - 1  7    2 GOP Window1
  70 B - - 1  7    2 GOP Window 2
  71 D - - 1  1    0 .
  72 D - X 1  2    0 disk.dmg:FW
  73 D - - 1  0    0 VT-100 Serial Console
Shell> _
```

tianocore

```
Shell> devtree -b
```

Displays s tree of devices currently managed by UEFI drivers.

```
Shell> devtree -b
 Ctrl[03] Fv(6D99E806-3D38-42C2-A095-5F4300BFD7DC)
 Ctrl[04] MemoryMapped(0xB,0x1D5A0F40000,0x1D5A0F5FFFF)
 Ctrl[13] MemoryMapped(0xB,0x1D5A09C0000,0x1D5A0F3FFFF)
 Ctrl[1C] VenHw(5CF32E0B-8EDF-2E44-9CDA-93205E99EC1C,00000000)
   Ctrl[6F] GOP Window1
     Ctrl[4E] Primary Console Input Device
     Ctrl[4F] Primary Console Output Device
   Ctrl[70] GOP Window 2
     Ctrl[4E] Primary Console Input Device
     Ctrl[4F] Primary Console Output Device
   Ctrl[71] .
   Ctrl[72] disk.dmg:FW
 Ctrl[20] VenHw(D3987D4B-971A-435F-8CAF-4967EB627241)/Uart(115200,8,N,1)
   Ctrl[73] VT-100 Serial Console
 Ctrl[2A] Fv(6D99E806-3D38-42C2-A095-5F4300BFD7DC)/FvFile(462CAA21-7614-4503-836
E-8AB6F4662331)/Enter Setup
 Ctrl[2B] Fv(6D99E806-3D38-42C2-A095-5F4300BFD7DC)/FvFile(EEC25BDC-67F2-4D95-B1D
5-F81B2039D11D)/BootManagerMenuApp
 Ctrl[2C] Fv(6D99E806-3D38-42C2-A095-5F4300BFD7DC)/FvFile(7C04A583-9E3E-4F1C-AD6
5-E05268D0B4D1)/Shell
 Ctrl[6D] VenHw(A04A27F4-DF00-4D42-B552-39511302113D)
 Ctrl[6E] VenHw(B3F56470-6141-4621-8F19-704E577AA9E8)
Shell> _
```

SHELL HANDLE DATABASE - "DH"

```
Shell> dh -b
```

Dump Handle - Displays the device handles associated with UEFI drivers

```
Handle dump
01: 4C8A2451-C207-405B-9694-99EA13251341 LoadedImage(DxeCore)
02: Decompress
03: FirmwareVolume2 DevicePath(..3D38-42C2-A095-5F4300BFD7DC)) FirmwareVolumeBlo
ck
04: DevicePath(..ped(0xB,0x2980000,0x299FFFF)) FirmwareVolumeBlock
05: A31280AD-481E-41B6-95E8-127F4C984779 FC1BCDB0-7D31-49AA-936A-A4600D9DD083
06: 4C8A2451-C207-405B-9694-99EA13251341 ImageDevicePath(..AD6B-4F3A-B60B-F59899
003443)) LoadedImage(DevicePathDxe)
07: DevicePathFromText DevicePathToText DevicePathUtilities
08: 4C8A2451-C207-405B-9694-99EA13251341 ImageDevicePath(..87AB-47F9-A3FE-D50B76
D89541)) LoadedImage(PcdDxe)
09: GetPcdInfo GetPcdInfoProtocol Pcd Pcd
0A: 4C8A2451-C207-405B-9694-99EA13251341 ImageDevicePath(..52B5-46CD-99C3-4368AB
BACFFD)) LoadedImage(Metronome)
0B: MetronomeArch
0C: 4C8A2451-C207-405B-9694-99EA13251341 ImageDevicePath(..A7EB-4730-8C8E-CC466A
9ECC3C)) LoadedImage(ReportStatusCodeRouterRuntimeDxe)
0D: SmartCardReader RscHandler
0E: 4C8A2451-C207-405B-9694-99EA13251341 ImageDevicePath(..4E0E-4BE4-B14C-340EB4
AA5891)) LoadedImage(StatusCodeHandlerRuntimeDxe)
0F: 4C8A2451-C207-405B-9694-99EA13251341 ImageDevicePath(..847E-4E5D-AD3F-21CABF
E5E23C)) LoadedImage(WinNtOemHookStatusCodeHandlerDxe)
10: 4C8A2451-C207-405B-9694-99EA13251341 ImageDevicePath(..1644-4EF4-8944-48C4FP
ress ENTER to continue or 'Q' break:_
```

```
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:
```

```
Shell> dmpstore -all -b
```

Display the contents of the NVRAM variables

```
Shell> dmpstore -all -b
Variable NV+BS '4C19049F-4137-4DD3-9C10-8B97A83FFDFA:MemoryTypeInformation' Data
Size = 0x40
  00000000: 0A 00 00 00 2A 00 00 00-09 00 00 00 08 00 00 00  *....*..........*
  00000010: 00 00 00 00 29 00 00 00-06 00 00 00 F2 00 00 00  *....)..........*
  00000020: 05 00 00 00 50 00 00 00-03 00 00 00 57 03 00 00  *....P.......W...*
  00000030: 04 00 00 00 AC 14 00 00-0F 00 00 00 00 00 00 00  *...............*
Variable NV+RT+BS 'EFIGlobalVariable:ErrOut' DataSize = 0x49
  00000000: 02 01 0C 00 D0 41 03 0A-00 00 00 00 01 01 06 00  *.....A.........*
  00000010: 00 01 02 01 0C 00 D0 41-01 05 00 00 00 00 03 0E  *.......A.......*
  00000020: 13 00 00 00 00 00 C2-01 00 00 00 00 00 08 01     *..............*
  00000030: 01 03 0A 14 00 53 47 C1-E0 BE F9 D2 11 9A 0C 00  *.....SG.........*
  00000040: 90 27 3F C1 4D 7F FF 04-00                       *.'?.M....*
Variable NV+RT+BS 'EFIGlobalVariable:ConIn' DataSize = 0x7A
  00000000: 02 01 0C 00 D0 41 03 0A-00 00 00 00 01 01 06 00  *.....A.........*
  00000010: 00 01 02 01 0C 00 D0 41-03 03 00 00 00 00 7F 01  *.......A........*P
ress ENTER to continue or 'Q' break:_
```

```
Shell> stall 10000000
```

Stalls the operation for a specified number of microseconds

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

UEFI SHELL SCRIPTS

Use Scripting with UEFI Shell

tianocore

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).

WRITING UEFI SHELL SCRIPTS

At the shell prompt

```
Shell> fs0:
FS0:\> edit HelloScript.nsh
```

Type: echo Hello World

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
```

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

Press "F2"
Enter
Press "F3" to exit

21

In the shell, **type** HelloScript for the following result:
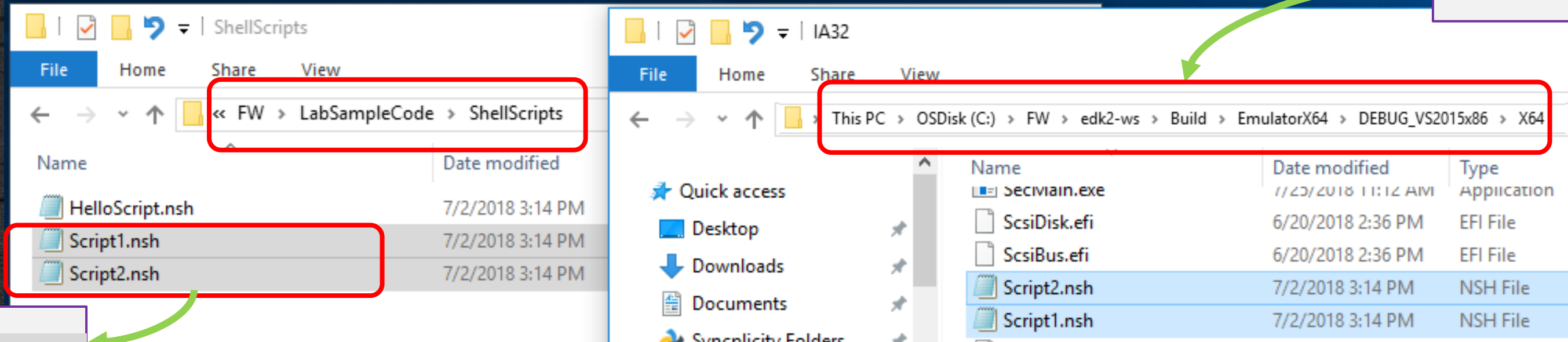
```
FS0:\> helloscript.nsh
FS0:\> echo Hello World
Hello World
FS0:\> _
```

Close the Win emulation, type: "reset"

```
FS0:\> reset
```

Copy the Scripts from the `/FW/LabSampleCode/ShellScripts` to the runtime directory `C:/FW/edk2-ws/Build/EmulatorX64/DEBUG_VS201nx86/X64`

Paste

Copy

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"
```
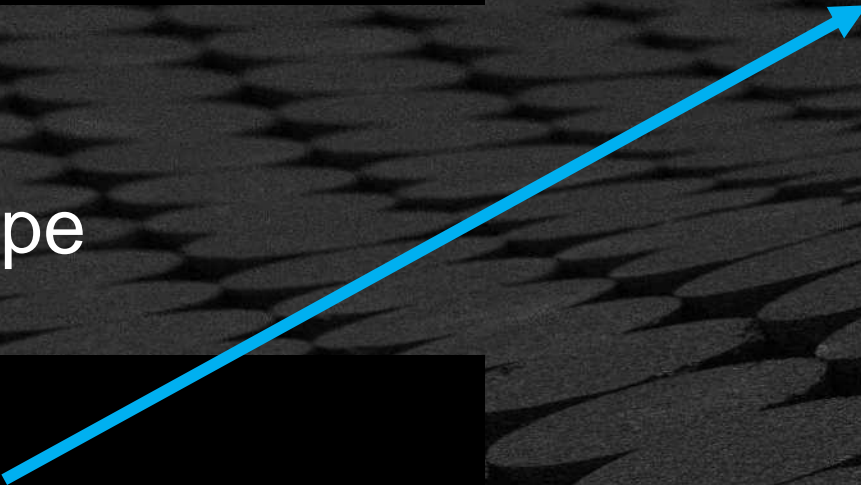
Script2.nsh

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

From the VS command Prompt

```
C:\FW\edk2> RunEmulator.bat
```

At the Shell prompt Type

```
Shell> fs0:
FS0:\> Script1

FS0:\> Edit Script1.nsh
```

```
                0 Dir(s)
FS0:\> Script1
FS0:\> script2.nsh
FS0:\> time > Mytime.log
FS0:\> for %a run (3 1 -1)
FS0:\>      echo %a counting down
3 counting down
FS0:\> endfor
FS0:\> for %a run (3 1 -1)
FS0:\>      echo %a counting down
2 counting down
FS0:\> endfor
FS0:\> for %a run (3 1 -1)
FS0:\>      echo %a counting down
1 counting down
FS0:\> endfor
FS0:\> for %a run (3 1 -1)
FS0:\> if exist %cwd%Mytime.log then
FS0:\>      type Mytime.log
20:08:54 (UTC 00:00)

FS0:\> endif
FS0:\> echo "Thank you. ByeBye:) "
Thank you. ByeBye:)
FS0:\> _
```

Remove the "**#**" on the first line

Press "F2"

Enter

Press "F3" to exit

Type

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

```
FS0:\> Script1
```

```
FS0:\> Script1
FS0:\> echo  -off
3 counting down
2 counting down
1 counting down
20:19:52 (UTC 00:00)

Thank you. ByeBye:)
FS0:\> _
```

SUMMARY

★ Run UEFI Shell (Nt32 Emulation)

★ Run UEFI Shell Commands

★ Run UEFI Shell Scripts

tianocore

Questions?

www.tianocore.org