FI Development Exploration 97 – EDK2 Simulator to Build a Netwo vironment

keep it-> Author: Luo Bing https://blog.csdn.net/luobing4365)

simulator builds network environment

---

tly, many netizens have asked questions related to the Internet. Most of the issues discussed are about setting up the netw nment, and some of the phenomena I have never encountered.

FI Development Exploration 49, I introduced building a UEFI network environment in the Nt32 simulator and a real environn DK2 I used at that time was the UDK2018 version.

e upgrade of EDK2, the original NT32 simulator (compiled with Nt32Pkg) has been canceled and replaced by EmulatorPk rovides more comprehensive functions, including 32-bit and 64-bit simulators, which have been introduced in previous blo

ore, this article is going to use the simulator compiled by EmulatorPkg to rebuild the network test environment. The previou tor was compiled with Nt32Pkg, so I named it Nt32 simulator; the current EmulatorPkg name is too long, so for the conveni ng, I call it EDK2 simulator.

ild EDK2 development environment

tailed construction process, please refer to UEFI Development Exploration 22 and 57. Here is a brief description of the uction process. The steps are as follows:

ɔl Installation

necessary development tools, including Visual Studio, Python, ASL, and Nasm;

wnload the code library

it to download the EDK2 code and StdLib code to your local computer.

a new directory edk202011 on the C drive and use the following command to download:

```
robin@robin-PC MINGW64 /c/edk202011
$ git clone --branch stable/202011 https://gitee.com/luobing4365/edk2.git
$ git clone https://gitee.com/luobing4365/edk2-libc.git
```

note that the download address here is my private repository. I synchronized the edk2 and edk2-libc repositories on githul ository on gitee for your own download.

lownload it according to the command I gave. The original repositories are https://github.com/tianocore/edk2.git and github.com/tianocore/edk2-libc.git. You can use the following command to download:

```
$ git clone --branch stable/202011 https://github.com/tianocore/edk2.git
$ git clone https://github.com/tianocore/edk2-libc.git
```

rse, depending on your internet speed, the download speed will be slow. It is recommended to use the same method to onize the repository on GitHub to gitee, or directly find the same repository on gitee and download it.

anch used in this article is stable/202011, and subsequent experimental results are all based on this version.

date submodules

.gitmodules in edk2 and change the submodule.

can use GitHub to download directly, you don't need to modify .gitmodules, just run the subsequent update commands dire ntent I modified is as follows:

```
[submodule "CryptoPkg/Library/OpensslLib/openssl"]
    path = CryptoPkg/Library/OpensslLib/openssl
    url = https://gitee.com/luobing4365/openssl.git
[submodule "SoftFloat"]
    path = ArmPkg/Library/ArmSoftFloatLib/berkeley-softfloat-3
    url = https://gitee.com/luobing4365/berkeley-softfloat-3.git
[submodule "UnitTestFrameworkPkg/Library/CmockaLib/cmocka"]
    path = UnitTestFrameworkPkg/Library/CmockaLib/cmocka
    url = https://gitee.com/luobing4365/cmocka.git
[submodule "MdeModulePkg/Universal/RegularExpressionDxe/oniguruma"]
    path = MdeModulePkg/Universal/RegularExpressionDxe/oniguruma
```

again, the addresses given for the submodules above are the addresses where I import the submodule repositories on Gith You can refer to my method, but don't use them directly, as these are my private addresses and cannot be used directly.

ecific import methods, please refer to the content of UEFI Development Exploration 57.

ıse the following command to update the submodule:

```
$ git submodule update --init
```

mpilation tools

ıll three folders in ekd2-libc: AppPkg, StdLib and StdLibPrivateInternalFiles to the edk2 directory to facilitate subsequent ation. (Of course, you don't have to do this. The method has been introduced in the previous blog, please check it out your

he Visual Studio Native command line (such as "VS2015 x86 Native Tools Command Prompt") and use the following comr pile BaseTools and other tools:

```
C:\edk202011\edk2> edksetup.bat Rebuild
```

ɔmpilation is complete, BaseTools and other tools will be automatically generated.

n use the following command to test whether the development environment is successfully built:

```
C:\edk202011\edk2> build -p AppPkg\AppPkg.dsc -a IA32
```

ll find an error message. This is mainly because there is a sentence on line 31 of AppPkg (I am using the latest StdLib libra 321):

```
!include MdePkg/MdeLibs.dsc.inc
```

e is not found in the source code of edk2 . Add a "#" symbol before the statement to comment it out and compile AppPk You can see that the compilation is successful, indicating that the development environment has been set up.

ild a network test environment

twork test environment built this time is only for the 32-bit simulation environment of EDK2. This is mainly because the 2Io file provided by Intel only supports 32-bit. Although the EDK2 simulation environment (EmulatorPkg compilation) suppo and 64-bit, it is better to test 64-bit network programs on the actual machine.

nstruction steps are as follows:

tall Winpcap

ap is a professional software for network packet capture . It is a free and public network access system. It can provide w ıtions with the ability to access the underlying network. In the simulator, it is equivalent to the driver of the network card. ɔad address:

www.winpcap.org/default.htm.

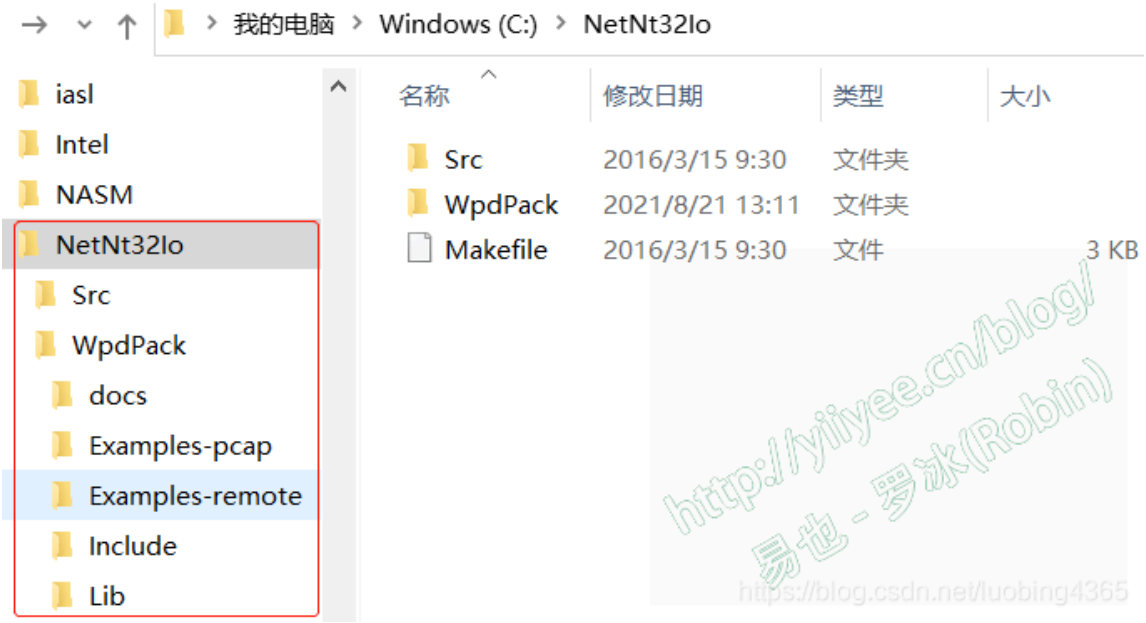
ol must be installed. The network packets sent by the simulator will be captured and forwarded by Winpcap.

mpile SnpNt32Io

ɔad the source code from github to your local computer: https://github.com/tianocore/edk2-NetNt32Io.

a folder NetNt32Io in drive C and copy the source code into it.

lownload the Winpcap development package WpdPack, the download address is: https://www.winpcap.org/devel.htm. Unz ack and copy it to the folder NetNt32Io just created. The created directory is as follows:



1 SnpNt32Io compilation directory structure

he Visual Studio command line (the same command line as used to compile the UEFI code), enter the source code directo ter the following command:

```
C:\NetNt32Io> nmake TARGET=RELEASE
```

ectory Release_IA32 will be automatically generated in the NetNt32Io folder, which contains the SnpNt32Io.dll we need.

mpile EDK2 simulator

le the 32-bit simulator as follows:

```
C:\edk202011\edk2> edksetup.bat
C:\edk202011\edk2> build -p EmulatorPkg\ EmulatorPkg.dsc -a IA32
```

mpiled simulator is located in the folder C:\edk202011\edk2\Build\EmulatorIA32\DEBUG_VS2015x86\IA32.

nfigure the simulator network environment

he previously compiled SnpNt32Io.dll to the directory where the simulator is located, and create a batch file loadnetwork.ns ectory with the following content:

```
load MnpDxe.efi ArpDxe.efi Ip4Dxe.efi VlanConfigDxe.efi Udp4Dxe.efi Dhcp4Dxe.efi Mtftp4Dxe.efi TcpDxe
```

EFI drivers in the batch file can be found in the simulator directory and are compiled together when the simulator is compile

-click the executable file WinHost.exe in the simulator directory to start the simulator and enter the UEFI Shell environmen ch file to load network-related drivers:

```
Shell> fs0:
FS0:\> loadnetwork.nsh
```

address of my machine is 192.168.181.132, and the gateway is 192.168.181.2. Set the network address in the simulator t 8.181.135 with the following command:

```
FS0:\> ifconfig -s eth0 static 192.168.181.135 255.255.255.0 192.168.181.2
```

ıe configuration is complete, you can use the "ifconfig -l eth0" and ping commands to check whether the configuration is sful.

```
FS0:\> ifconfig -l eth0
name       : eth0
Media State  : Media state unknown
policy       : static
mac addr    : 00:0C:29:D5:97:D7
ipv4 address  : 192.168.181.135
subnet mask  : 255.255.255.0
default gateway: 192.168.181.2

 Routes(2 entries):
   Entry[0]
Subnet  : 192.168.181.0
Netmask : 255.255.255.0
Gateway: 0.0.0.0
   Entry[1]
Subnet  : 0.0.0.0
Netmask : 0.0.0.0
Gateway: 192.168.181.2
DNS server   :

FS0:\> ping 192.168.1.42
ping 192.168.1.42 16data bytes.
16 bytes from 192.168.1.42 : icmp_seq=1 ttl=0 time0~26ms
…
```

FI development machine is a VMware virtual machine with Win10 operating system, on which the UEFI development ıment is built. The host machine's operating system is also Win10, and a bunch of virtual machines are running on the hos ıe for development of various needs.

ıst machine is equivalent to another machine in the local area network, and its IP address is 192.168.1.42.

t, the network environment has been set up, and the network program test is now carried out.

work Program Testing

usly, in order to test the UEFI network, a server program on the Windows side and a client program on the UEFI side we ped, namely EchoServerTCP4.exe and EchoTcp4.efi.

:hoServerTcp4.exe program has been compiled before, and only the client code needs to be compiled.

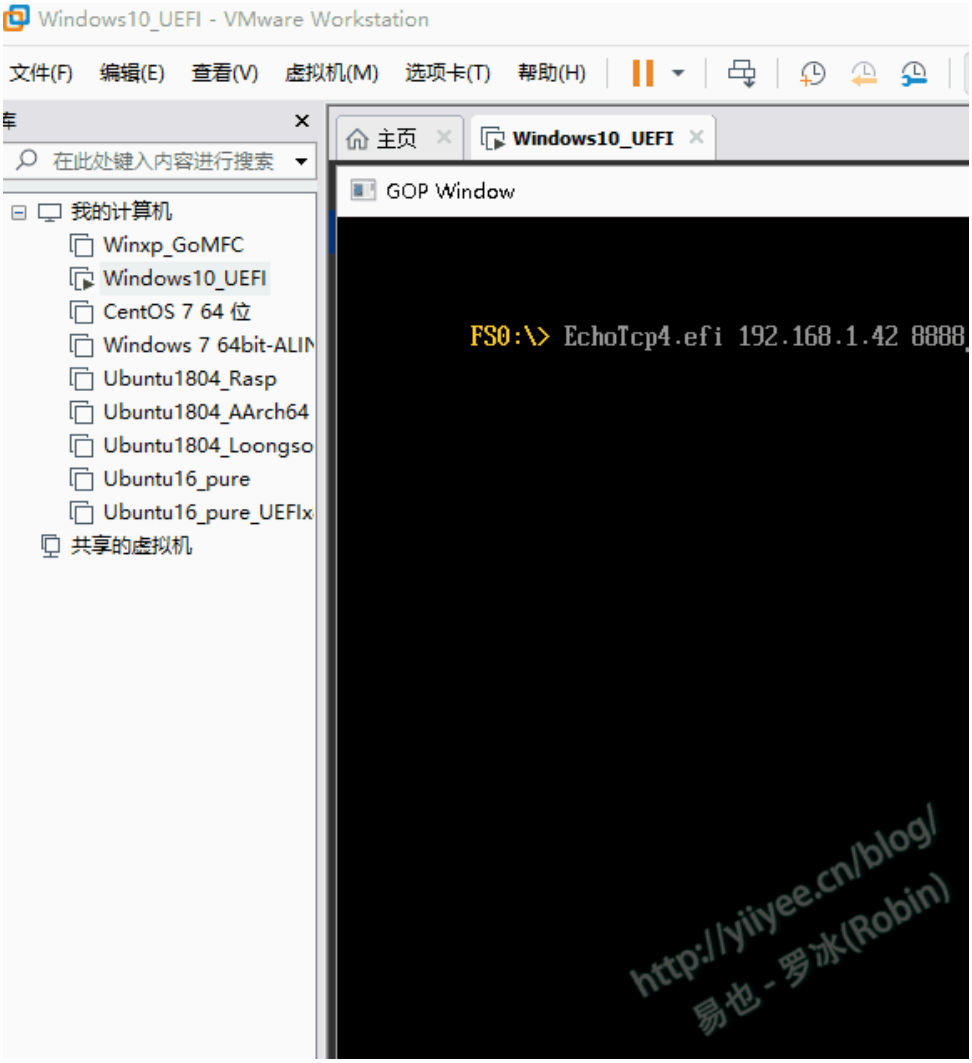
RobinPkg from the repository of the UEFI Development Exploration Blog to the currently built UEFI development environme mpile EchoTcp4.efi:

```
C:\edk202011\edk2>build -p RobinPkg\RobinPkg.dsc -m RobinPkg\Applications\EchoTcp4\EchoTcp4.inf -a IA
```

ıtly reworked EchoTcp4 and updated the code. In fact, the same is true for projects using stdEchoTcp4.)

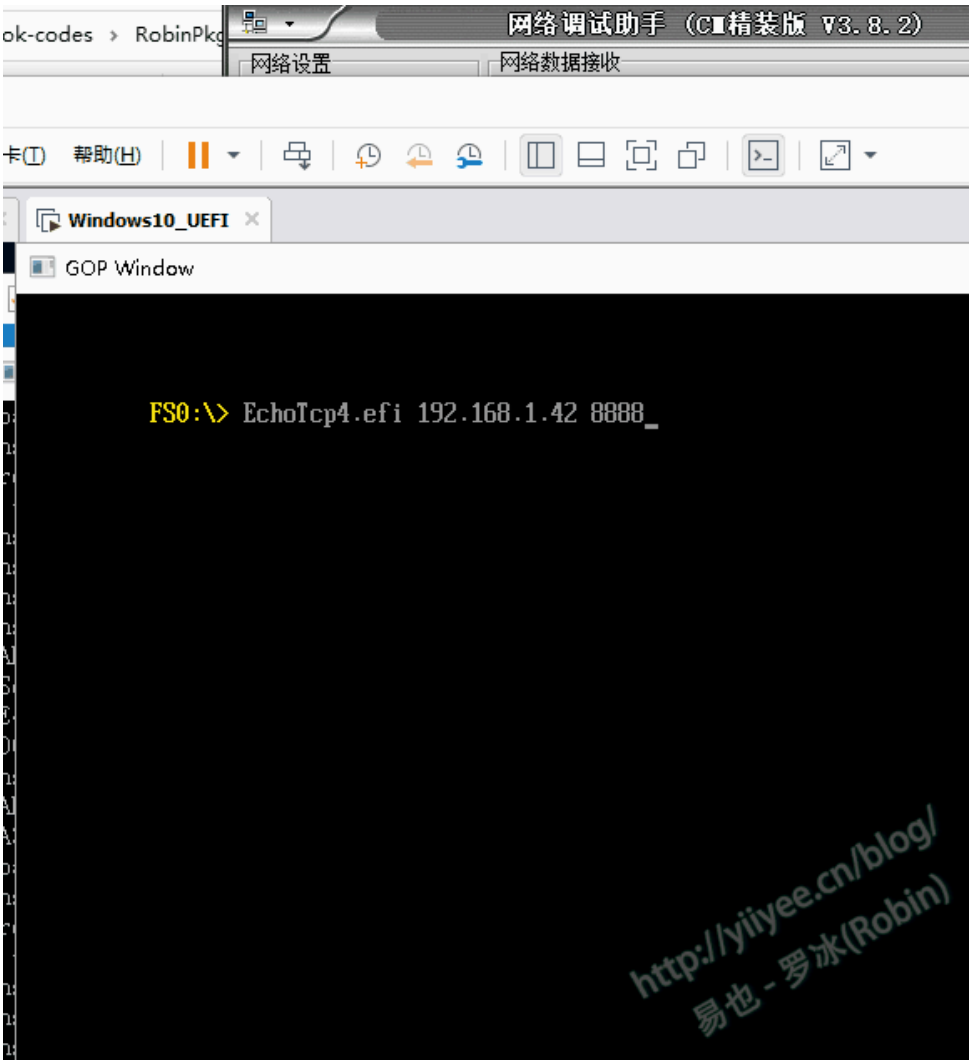
he compiled EchoTcp4.efi to the directory where the simulator is located.

e server program on the host machine and the client program on the EDK2 simulator. The test results are as follows:



2 Test network communication

rse, you can also use the commonly used network debugging assistant to test. The test results are as follows:



3 Testing with the network debugging assistant

lly I rarely need to debug UEFI network programs, so I always use the environment built by the method in this article for te:

the experiment, I also found several problems:
firewall will affect the sending of packets, so it is best to turn off all firewalls. Maybe Winpcap's packet capture ability is affe
never used Winpcap to write programs. Netizens who understand the principle can help me take a look;

en setting the EDK2 simulator URL, when setting it to DHCP, sometimes it succeeds, sometimes it fails, I suspect it is relate
ter. The specific reason is unknown, maybe this problem can be explained after deepening the simulator principle.

he operating system running the EDK2 simulator, it is unsuccessful to use the server software to communicate with the cli
re on the simulator. In other words, the server software needs to be run on other machines in the same LAN. For example,
machine to run the server software, a virtual machine Win10 to run the EDK2 simulator, and the client software on the
tor. runs normally.

questions are for your reference. I am not sure about the specific reasons. You are welcome to leave a message for discu:

litate the experiment, I put the source code of Winpcap, WdpPack (Winpcap development kit) and NetNt32Io required for tl
nent in the warehouse:

address: https://gitee.com/luobing4365/uefi-explorer
files are located under: /97 Network-Emulator