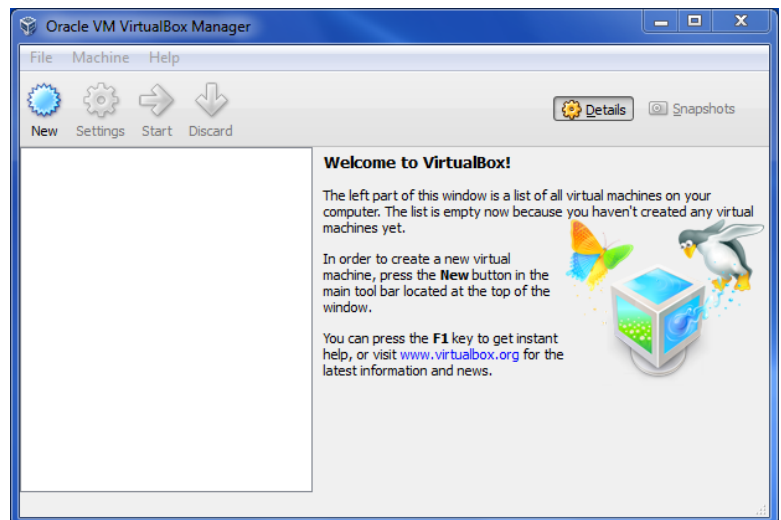


HOWTO configure Xinu under Virtual Box

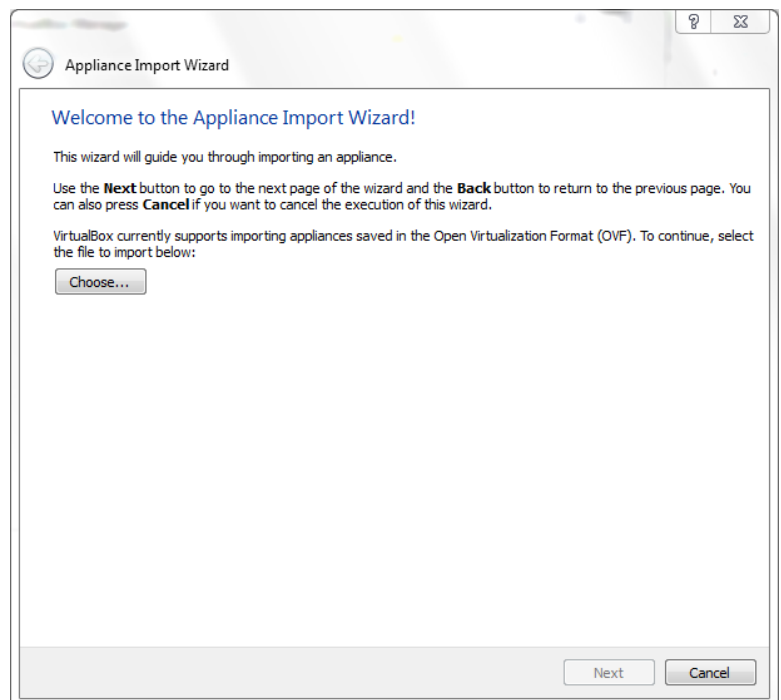
This document describes the steps needed to install and configure Xinu under VirtualBox. When you are finished, you will be able to compile a Xinu image and boot the image in a Virtual Machine. Although it runs as an application on your computer, VirtualBox presents the illusion of a bare machine. Thus, the Xinu image you run is that same as one that can boot on an otherwise-empty PC. In other words, you will be creating an operating system, not an application program.

1. Before running Xinu, you must install VirtualBox on your computer. Versions exist for Windows, Mac, and Linux. You can download the correct version from <https://www.virtualbox.org/wiki/Downloads>
2. Once you have installed VirtualBox, import Open Virtualization Format Virtual Machines from the same tarball that contains this file. There are two virtual machines. One acts as a development platform (develop-end) running Linux on which you can modify and compile Xinu. The other acts as a bare machine (back-end) on which Xinu runs. The two machines will have a virtual serial connection between them that allows you to communicate with the Xinu machine while Xinu runs.

- In **VirtualBox main window**, select **File > Import Appliance**.

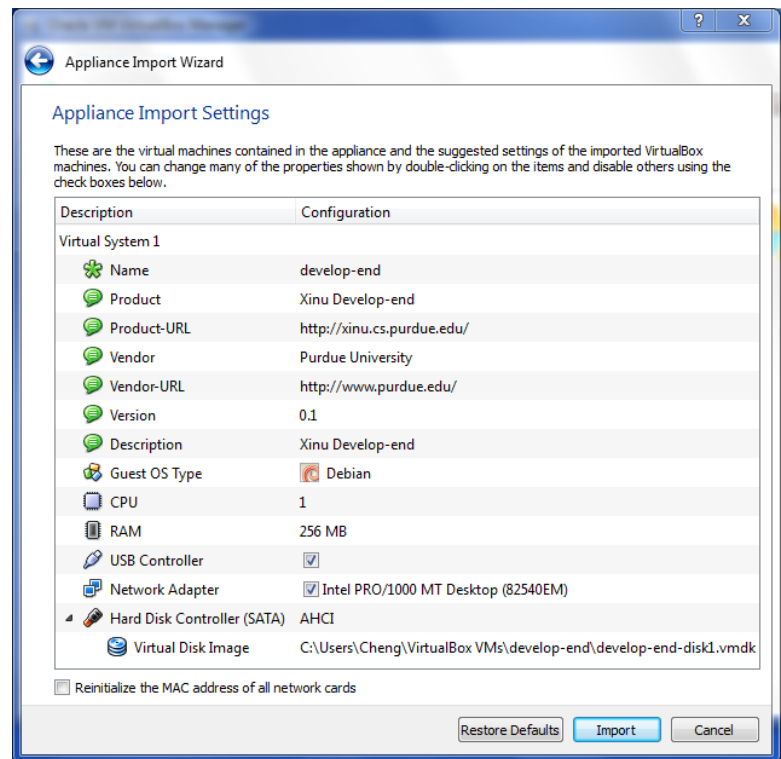


- In **Appliance Import Wizard**, select **Choose**, browse to *develop-end.ova* and click **Open**.

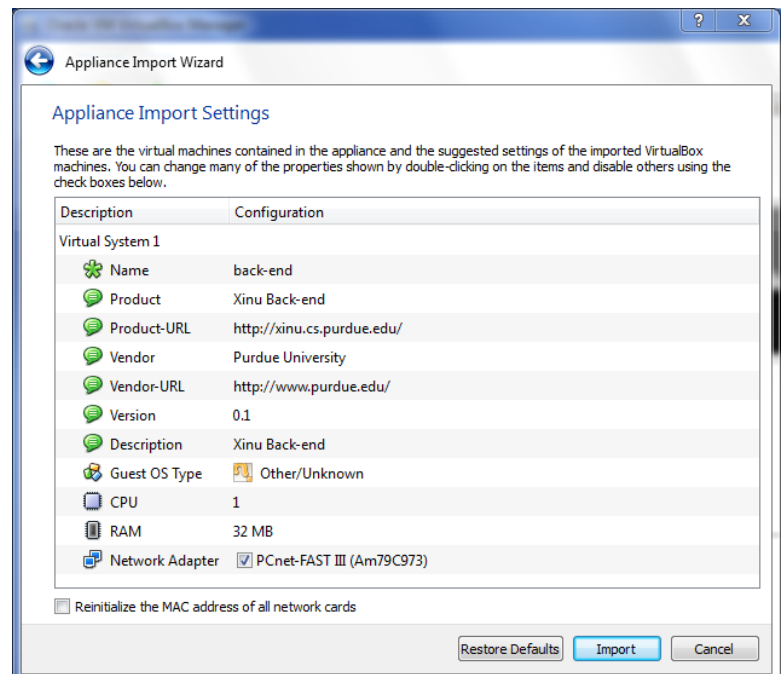


HOWTO configure Xinu under Virtual Box

- After verifying the Import Settings, click **Import** to deploy the appliance.
- Note: ***DO NOT*** click **checkbox of Reinitialize** the address of all network cards.



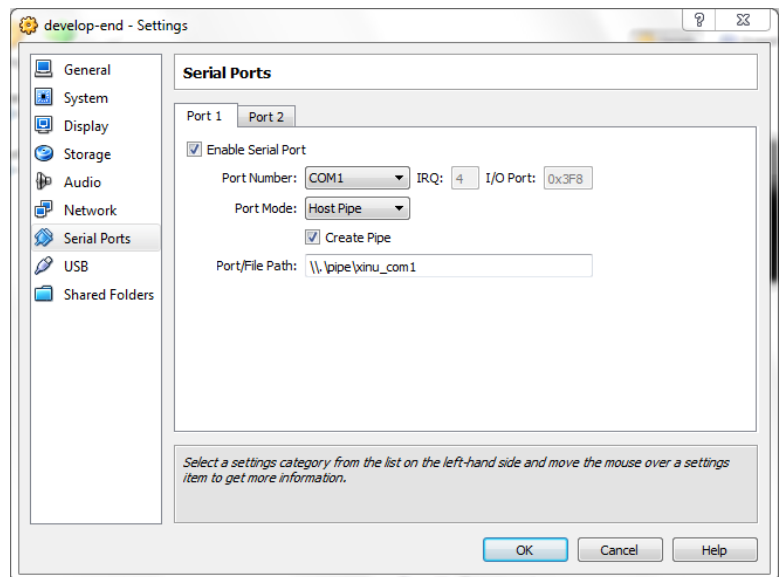
- Use the same procedure to import **back-end.ova**.



3. When Xinu starts, it communicates over a console serial port. To see the output and communicate with Xinu, you must set up a connection between the serial port on the development machine and the serial port on the back-end.
 - In the **VirtualBox main window**, choose *develop-end* from right side, then click **Settings**.
 - In **develop-end – Settings**, choose **Serial Ports** from right side.
 - Check **Enable Serial Port** checkbox.
 - Select **COM1** from the **Port Number** drop-down menu.

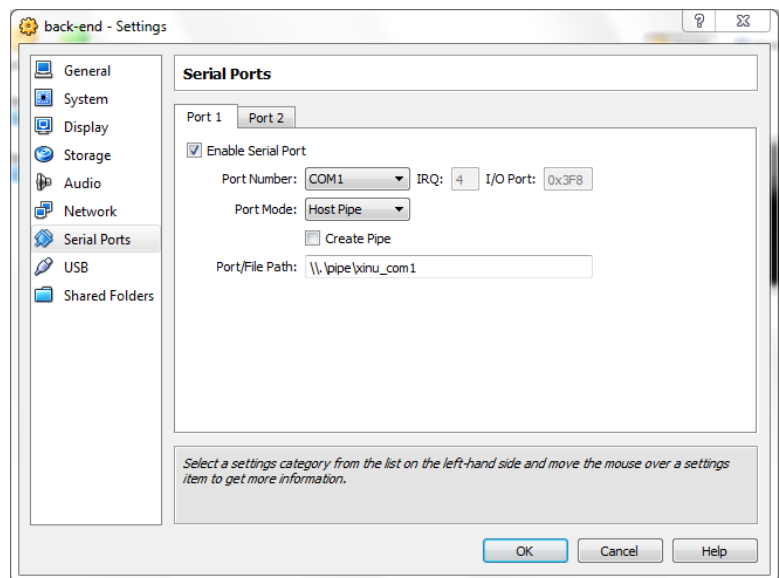
HOWTO configure Xinu under Virtual Box

- Select **Host Pipe** from the **Port Mode** drop-down menu.
- Check **Create Pipe** checkbox.
- For a Windows host, the **Port/Port Path** must begin with `\\.\pipe\` and must be the same on both the develop-end and backend. (e.g., `\\.\pipe\xinu_com1`).
- For a Linux host, the **Port/Port Path** must be a file begin with an absolute path and must be the same on both develop-end and back-end. (e.g., `/tmp/xinu_com1`).



Apply the same procedure to the back-end machine.

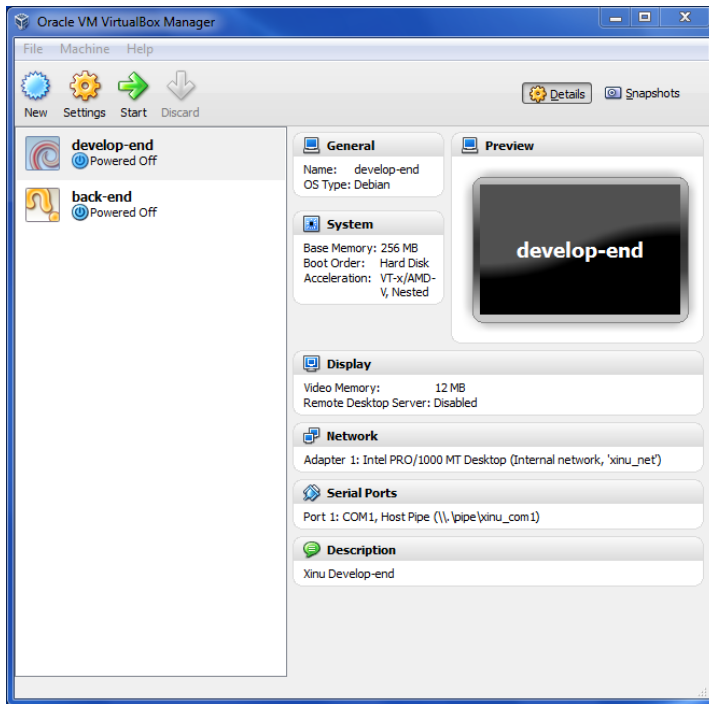
- In **VirtualBox main window**, choose *develop-end* from the right side, then click **Settings**.
- In **develop-end – Settings**, choose **Serial Ports** from the right side.
- Check **Enable Serial Port**.
- Select **COM1** from the **Port Number** drop-down menu.
- Select **Host Pipe** from the **Port Mode** drop-down menu.
- For a Windows host, the **Port/Port Path** must begin with `\\.\pipe\` and must be the same on both the develop-end and backend. (e.g., `\\.\pipe\xinu_com1`).
- For a Linux host, the **Port/Port Path** must be an absolute file name (i.e., begins with `/`), and must be the same on both develop-end and back-end. (e.g., `/tmp/xinu_com1`).
- Note: for the back-end, DO NOT check **Create Pipe**.



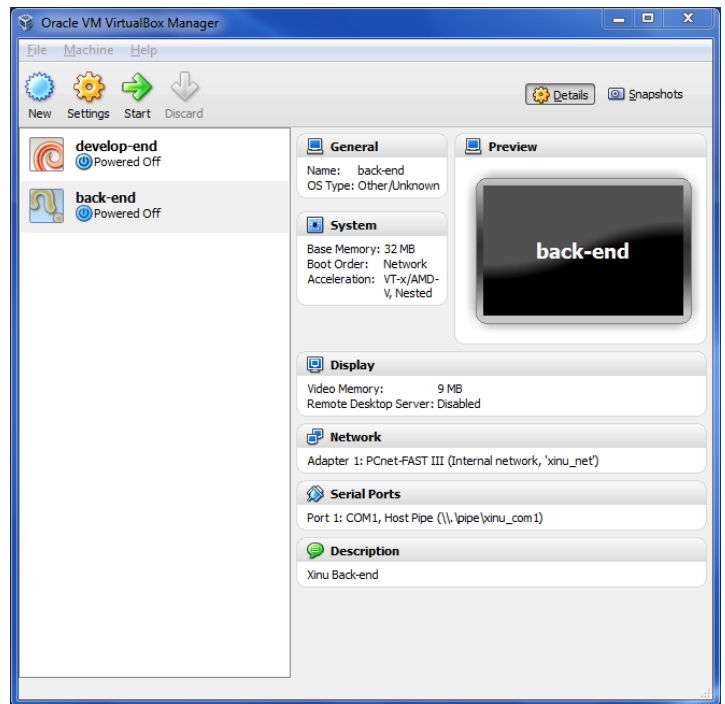
4. After configuration, the two virtual machines are ready to use. They should appear as follows:

HOWTO configure Xinu under Virtual Box

develop-end:

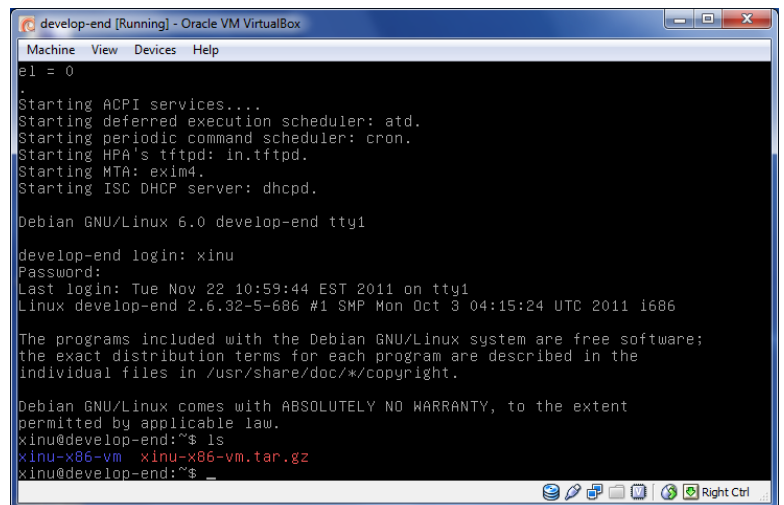


back-end:



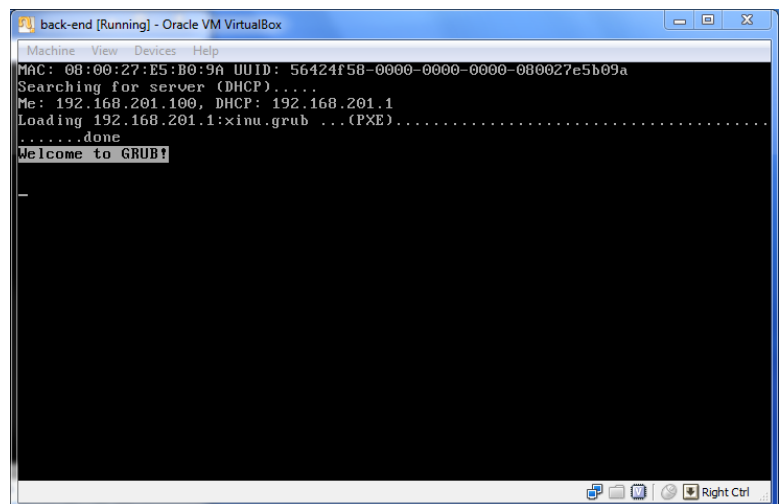
5. Start the develop-end virtual machine, which will boot Linux.

- Power on the develop-end virtual machine. The default user name is **xinu** and the default password is **xinuocks**.
- Under the home directory, there is a tarball called **xinu-x86-vm.tar.gz**, which includes a basic version of Xinu. Untar the tarball
- Go to directory **xinu-x86-vm.tar.gz**, issue command **make clean;make;./upload.sh** to compile and upload the newly compiled Xinu kernel to the TFTP directory.
- Before booting Xinu on the back-end VM, you must run a terminal emulator on the development machine. Issue **sudo minicom** to open a minicom terminal.



6. Start the back-end virtual machine.

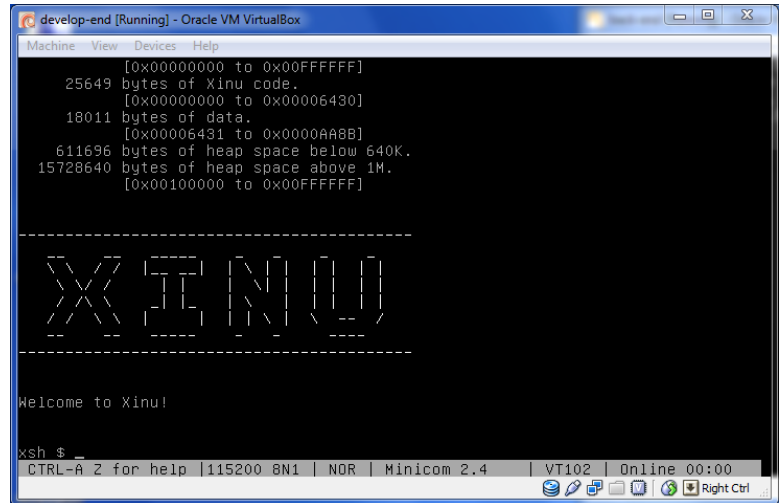
- Power on the end-end virtual machine. The back-end is configured to use PXE boot, which will access a TFTP server, download and image, and run it. The download occurs over an emulated Ethernet and the development machine runs the TFTP server. You will see a GRUB message during the bootstrap, and Xinu will begin.



HOWTO configure Xinu under Virtual Box

7. The minicom terminal will display the information that Xinu emits when it boots, and then you will see the block-letter “XINU” displayed when the shell starts. Type question mark at the prompt to see the available shell commands.

Enjoy yourself!



The screenshot shows a VirtualBox window titled "develop-end [Running] - Oracle VM VirtualBox". Inside, a Minicom terminal window is open, displaying the following text:

```
[0x00000000 to 0x00FFFFFF]
25649 bytes of Xinu code.
[0x00000000 to 0x00006430]
18011 bytes of data.
[0x00006431 to 0x0000AA8B]
611696 bytes of heap space below 640K.
15728640 bytes of heap space above 1M.
[0x00100000 to 0x00FFFFFF]

-----
X I N U
-----

Welcome to Xinu!

xsh $ _
```

The status bar at the bottom of the Minicom window shows: "CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.4 | VT102 | Online 00:00".