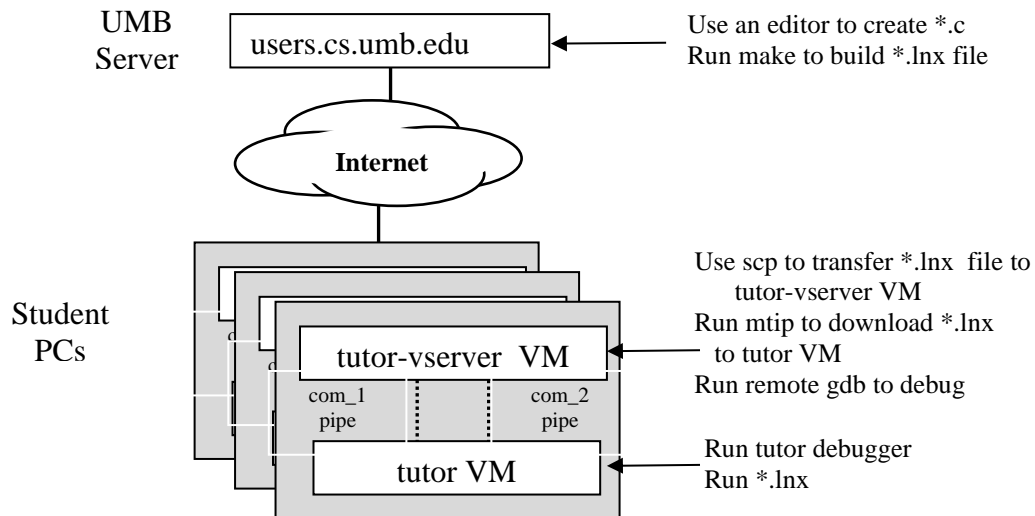


How to Run a Virtualized SAPC on a PC using VMWare Workstation

This document is modified from Professor Ronald Cheung's instruction for CS341-Fall-21. It provides updated instructions on a software development environment that is based on a virtualized Stand-alone PC (SAPC). The SAPC VM is controlled using a small Linux OS VM (XINU) developed by Perdue University. In our environment, the virtualized SAPC is called the tutor VM and the XINU VM is called the tutor-vserver VM.

1. UMB CS341 development environment

The development environment is shown below. The tools were originally developed in UNIX. In 2020, the build environment has been moved to LINUX. Students first create the *.c and *.lnx programs using an editor and the make utility on the umb Linux server. Then the *.lnx file is transferred to the tutor-vserver VM running on a student's PC. Then the code in the lnx file is subsequently downloaded onto the tutor VM using the mtip and ~d command. The VMs can be reset by clicking on the Power Off /On menu item.



2. Downloading and Installing VMWare application

- i) If you have registered for the course, you should receive an email from brightspace.com to set up an account to redeem VMWare software. Follow the directions in [Creating and Registering your VMWare account](#) to set up an account. After setting up your account, follow the instructions in [Downloading VMWare](#) to download licenses and applications. For PCs, please download VMWare Workstation 17 Pro.
- ii) Install the application by double clicking the .exe file.
- iii) Enter the license key received from brightspace.com.

3. Downloading and Installing the VMs

Download the tutor and tutor-vserver directories from [/nobackup/faculty/hefeiqiu/](#) to your PC using a Windows file transfer client. You should have tutor.ovf and tutor-vserver.ovf in the corresponding directories. Install and configure the tutor and tutor-vserver VMs as follows:

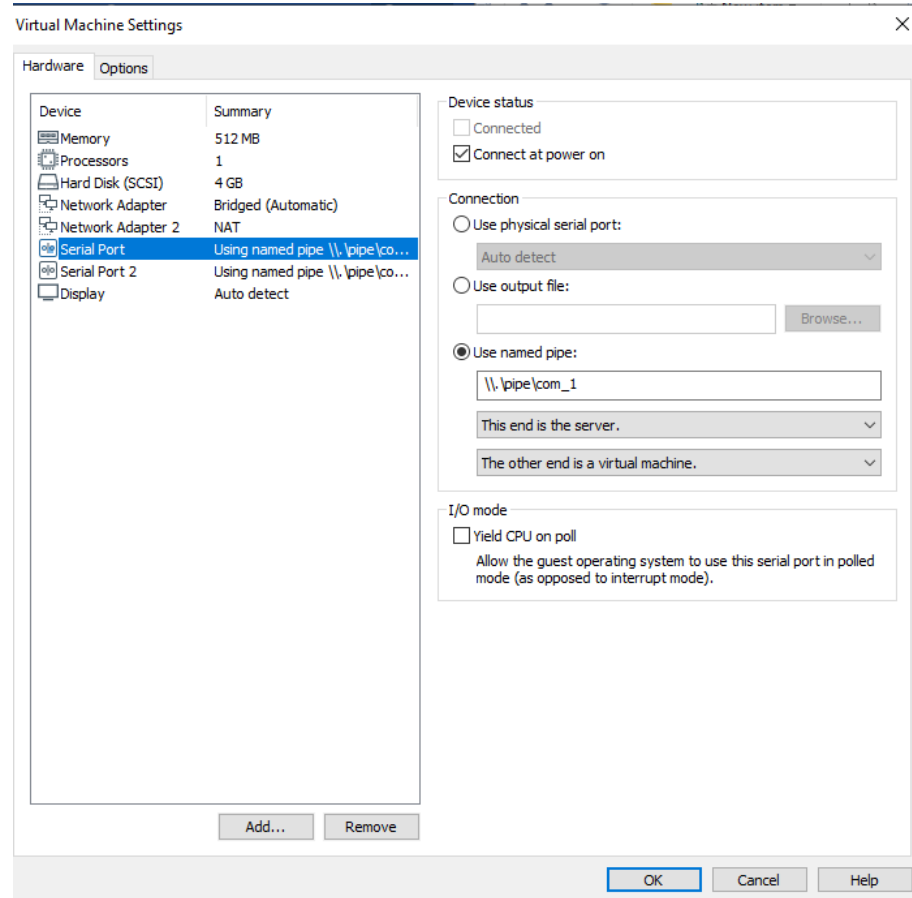
- i) To install the tutor-vserver VM, import the tutor-vserver.ovf (Open Virtualization Format) file as follows:
 - a) Open VMWare Workstation and select **File > Open**.
 - b) In the drop-down to the right of the File Name: box, Choose to Open Virtual Machine Format Images (*.ovf, .ova), browse to **tutor-vserver.ovf**, and click **Open**.
 - c) Click **Import** at the next screen.
 - d) the tutor-vserver VM should be installed. Please note the VM files are stored in directory C:\Users\user_name\Documents\Virtual Machines\tutor-vserver\.
- ii) To install the tutor VM on a separate window. Select File > New Window and then follow the same procedures in i) to import **tutor.ovf**. After installation, the VM files are stored in the directory C:\Users\user_name\Documents\Virtual Machines\tutor\. After installation, you can see the 2 VM windows side by side.

4. Configure the VMs

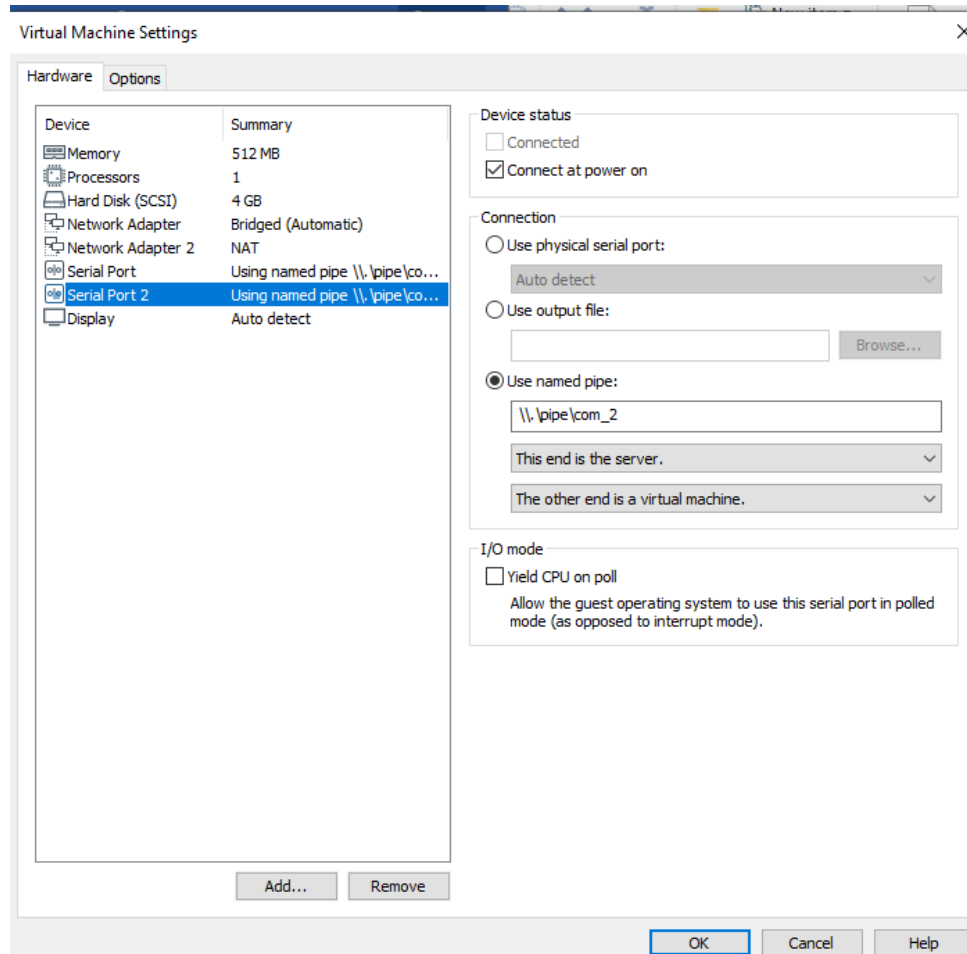
In a real hardware environment, servers (tutor-vserver and tutor) are connected using serial port COM2 hardware for monitoring and serial port COM1 hardware for remote gdb. In the VM environment, you connect the 2 VMs using simulated software pipes. The VMs as provided do not have serial ports configured. You need to add them first and configure their connectivity using software pipes.

For the **tutor-vserver VM**:

- i) Select the virtual machine **tutor-vserver** and select **VM > Settings**.
- ii) On the **Hardware** tab, select **ADD**. On the next screen, select **serial port** and **Finish**
- iii) On the next screen, select **Use name pipes** and enter data into the screen as follows:



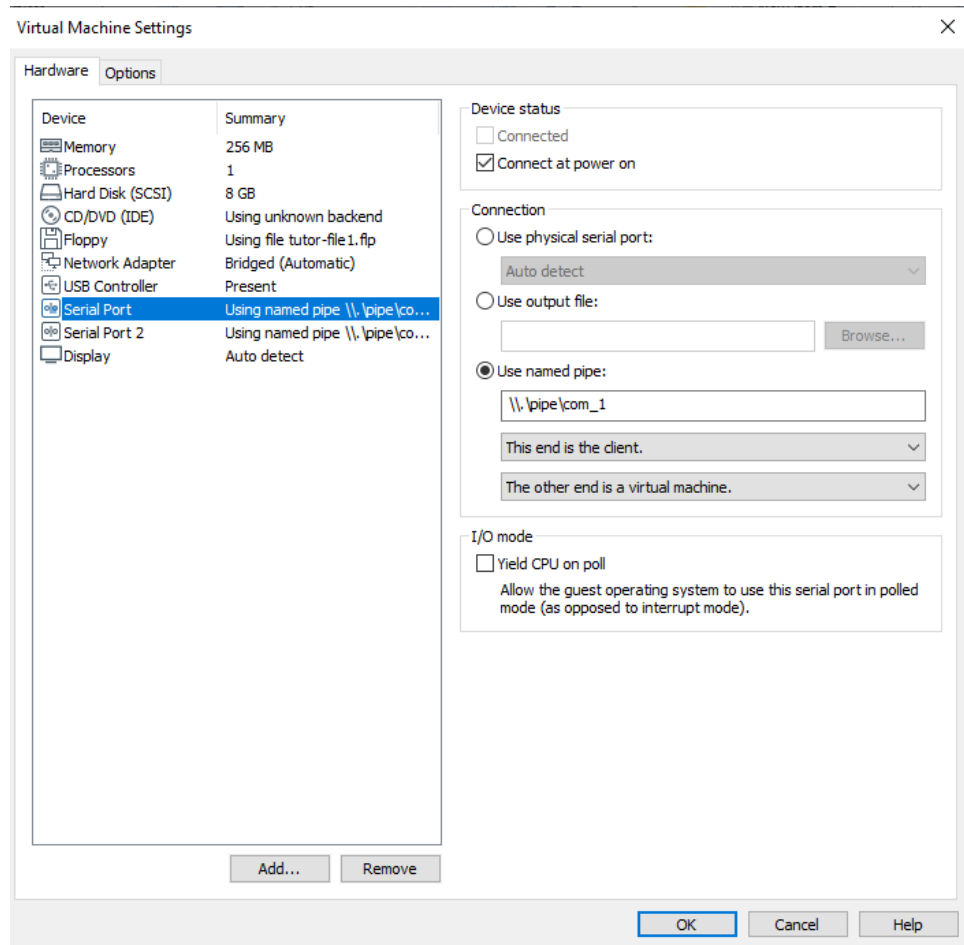
- iv) Select **OK** when done.
- v) Repeat the same to add and configure serial port 2. The name pipe is `\\.\pipe\com_2`. Then select **OK** as follows:



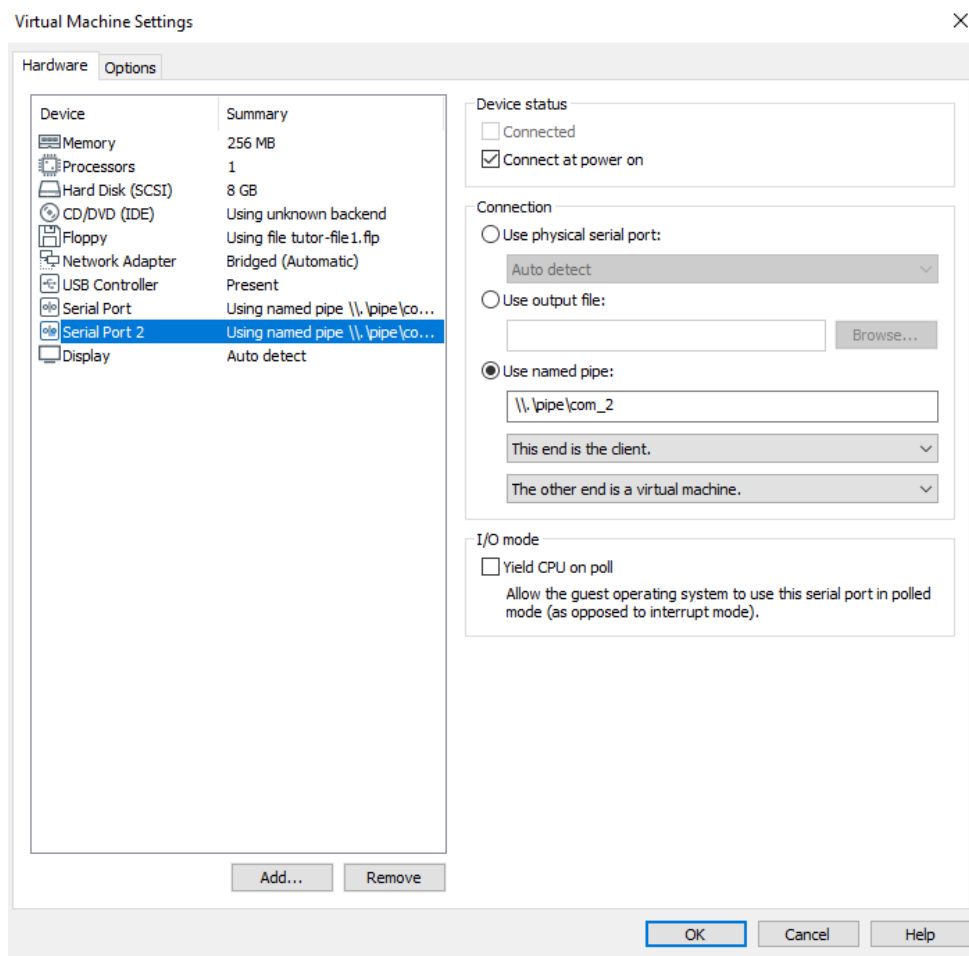
- vi) Go back to **VM > Setting**, on the **Hardware** tab, select **Network Adapter 2**.
- vii) Select **Connected at Power on** and **NAT**. Then select **OK**.
- viii) Now switch to configure the tutor VM.

For the **tutor VM**:

- i) Add the serial ports using the same procedures as above and then configure the 2 serial ports as follows:



ii) For the name pipe selection on both ports, please select “**This end is the client**”. There is no need to configure the network adapter for this VM since it only connects to tutor-vserver using the serial port pipes.



5. Running the VMs

i) Power on **tutor-vserver** and logon to it using user name **tuser** (for Tutor user) and the password **cs444**. You may need to move the cursor to the window and select it first. At any point, you can do <Cntl> + <Alt> to move the cursor out of the VM window. **tutor-vserver** is a Linux distribution and you can use regular Linux commands such as `ls`, `mkdir`, `ssh`, `script`, `scp` etc. The `mtip` and `gdb` command is also available. Run `mtip` and you should see the normal print out. There is no need to select a board since it always talks to board 1.

ii) You can transfer the test `.lnx` file from the LINUX server `/nobackup/faculty/hefeiqu/examples/test_dir/` directory to the **tutor-vserver** VM by entering the following command at the **tutor-vserver** VM:

```
scp username@users.cs.umb.edu:
/nobackup/faculty/hefeiqiu/examples/test_dir/test.lnx .
```

Please note there is a “.” at the end of the line. Issue the ls command to verify the presence of the transferred file.

iii) Leave the tutor-vserver VM by pressing <Ctrl> + <Alt> keys together. Select the tutor VM. Power on tutor VM . Note that it offers you the chance to enter a <CR> (the same as the <Enter> key) and make this screen the tutor console. You can do the <CR> here (after clicking on that window) and play with tutor, but the really useful way to use tutor is by talking to it on COM2 port using tutor-vserver VM. In this case, do the <CR> at the tutor-vserver VM. To do this, click on <Ctrl> + <Alt> keys to leave the tutor VM.

iv) Go to the tutor-vserver VM by clicking on its window. Power on the VM if it is not. After logging in, enter the command mtip at the Linux prompt. It should return the following:

```
....
Using board #1
```

Now you should be able to check whether the tutor-vserver VM is connected to the tutor VM via the COM2 connection by power off and power on the tutor VM. If they are connected, you should see the following message on the tutor VM as well as the tutor-vserver VM.

Please type <CR> to confirm console setting:

At the tutor-vserver VM, click the <CR> (the same as the <Enter>) key. It should print out a long welcome message and end with a prompt:

```
Tutor>
```

Try out the tutor commands "dd", "md" and "h" to see how they work. Type “~q” or two control-Cs to exit back to the Linux prompt.

v) To test the downloading command, type "mtip -f test.lnx" and “~d” to download the test.lnx file to the tutor VM. To run test.lnx, issue “go 100100”.

6. Running Remote GDB

See remgdb-testio_mod.script at the class webpage for an example on how to run remote gdb on the VM. To use remote gdb in the VM environment, build your executable (e.g. test.lnx) using the gcc -g option. Transfer the executable (e.g. test.lnx) and c file (e.g.test.c) to the tutor-vserver VM using the scp command. Then the .lnx and .c files should be in the home directory of tuser. We can debug the program using remote gdb as follows:

i) Find out what is tutor-vserver VM's IP address:

At the tutor-vserver VM, enter command “ /sbin/ifconfig” to show all IP addresses of the VM. IP address for Network Adapter 2 is listed under eth1 (e.g. 192.168.1.131).

ii) Set up tutor VM for gdb

Follow the steps in Section 5 and download an executable lnx (e.g. test.lnx) program to the tutor VM using mtip. When the tutor prompt returns after downloading, enter command “gdb”. You will see:

```
Setting gdb dev to COM 1, starting gdb...
```

iii) Start gdb using a separate window on your PC

On your computer, bring up a separate ssh window (using SSH client such as putty on PCs or command ssh in terminal window in Macs) and make connections to tutor-vserver VM (e.g. ssh tuser@192.168.1.131). Log into tutor-vserver VM using the tuser credential.

Go to the directory that has the lnx and c file. Enter the following to run remote gdb:

```
gdb test.lnx
tar rem /dev/ttyS0
set $eip = 0x100100
b main
c
    <stops at main>
c
...
    < enter more gdb commands >
...
```

7. Setting a Larger Display Size on the Vserver VM Window

The Vserver VM is a Linux distribution and the resolution of the console window is defaulted to be 640x480. It only displays 25 rows and each row has 80 characters. In order to display more rows, you need to change the resolution to a higher one. This [blog](#) gives detailed instructions on how to change the resolution to 1024x768x16. Make sure you edit the grub file as a super user (su), save the new grub setting using the command update-grub, and then reboot.

8. Setting a Larger Font Size on the Vserver VM Window

For laptops with smaller monitors, you can set the font size to be bigger by editing the `/etc/default/console-setup` file and then reconfigure the terminal. You have to be a super user to edit the file as follows:

```
sudo su
  (enter your tuser password)
vi /etc/default/console-setup
  (edit the line FONTFACE= "Terminus" :
  edit the line FONTSIZE= "28x14"
  for my 12.3" Microsoft Surface Pro
  save the file)
sudo service console-setup restart
```

Power down and up your VM.