

How to Run a Virtual SAPC with VMWare

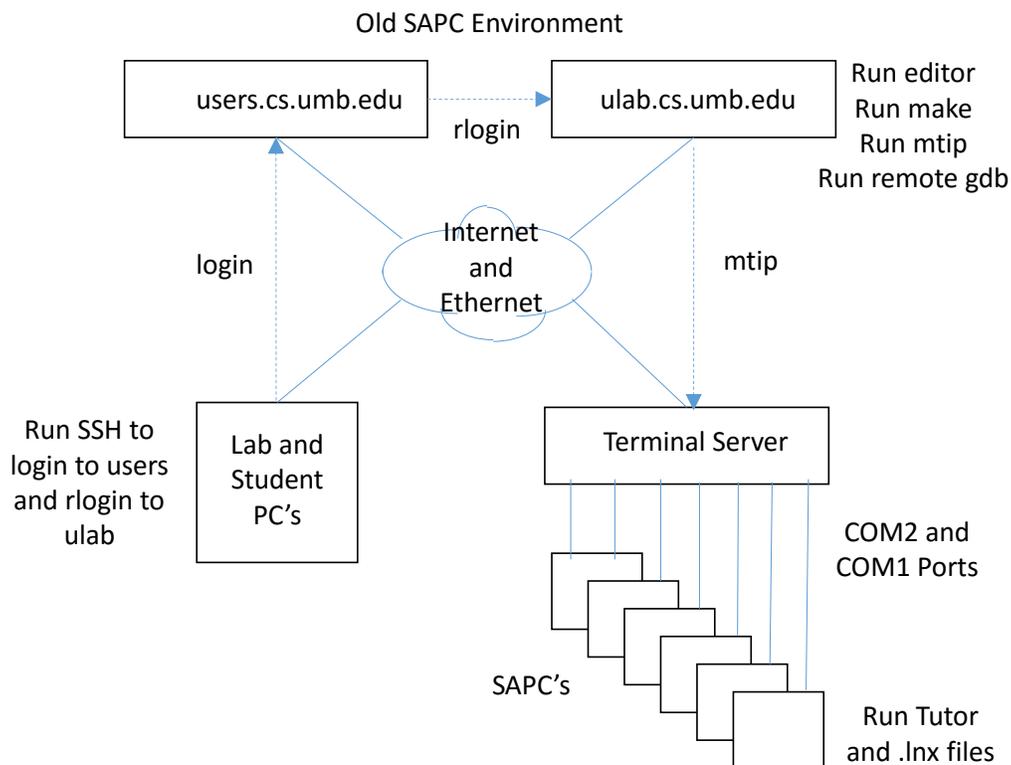
Fall 2016 – Bob Wilson

This document is modified from Professor O’Neil’s CS444-Fall 14 and Professor Ron Cheung’s CS444-Fall 15 materials. It provides instructions for installing the software development environment that creates a virtualized Stand-alone PC’s (SAPCs).

0. Background

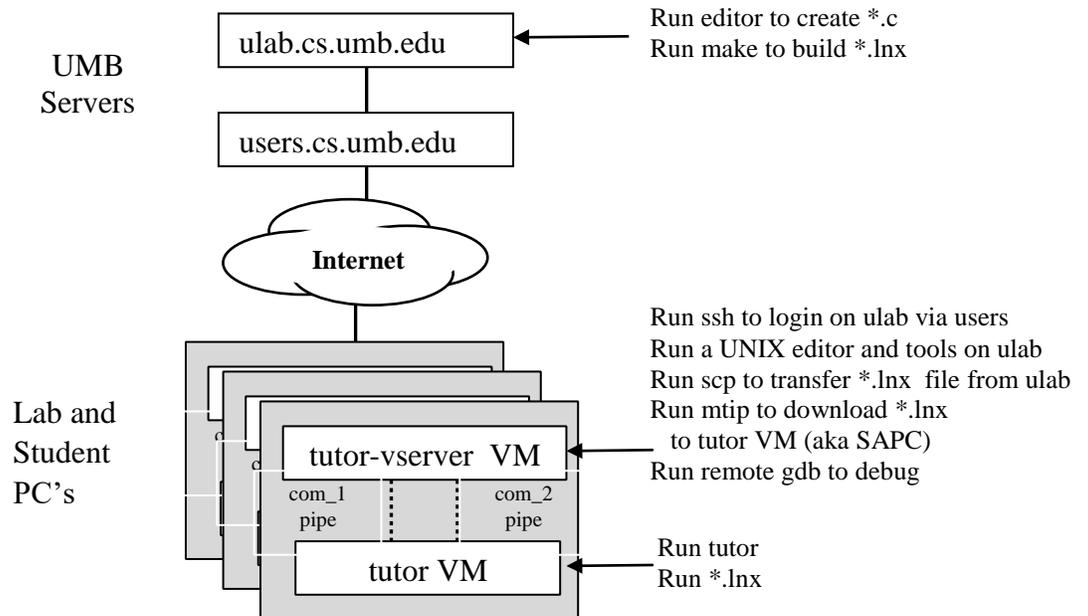
When we first offered this course, we provided a development host (ulab) and a set of embedded systems implemented on Stand-alone PC’s (SAPCs) that were attached to the development host across the internet. Over time the SAPC’s have become old/obsolete and difficult to maintain. Rather than upgrade to a new hardware version of the SAPCs, we developed a virtual embedded system development environment with VMWare.

We are in the middle of a transition and are still somewhat dependent on ulab. The SAPC’s are still attached to ulab, but we won’t use them. We support a virtual SAPC environment in our labs and provide you the tools to implement it on your own PC’s.



1. The VM development environment

We'll use the setup as shown below. The *.c and *.lnx programs are developed on the UMB ulab server using an editor and the UNIX make utility via your PC with SSH. Then *.lnx files are transferred to the xinu-vserver VM and they are subsequently loaded onto the tutor VM using the tutor ~d command. The tutor ~r function for resetting the SAPC is replaced by selecting the cntl-alt-del button on the tutor VM.



2. Download and Install the VMs

Download the vmtutor directory from our UNIX server from ~bobw/cs341 and store it on your home machine. This directory will require about 2 Gigabytes of disk space. It includes the VMWare10 executable code and tutor and tutor-vserver packages. Among others, you should have tutor.ovf and tutor-vserver.ovf in sub-directories.

Install VMWare Workstation 10 as follows:

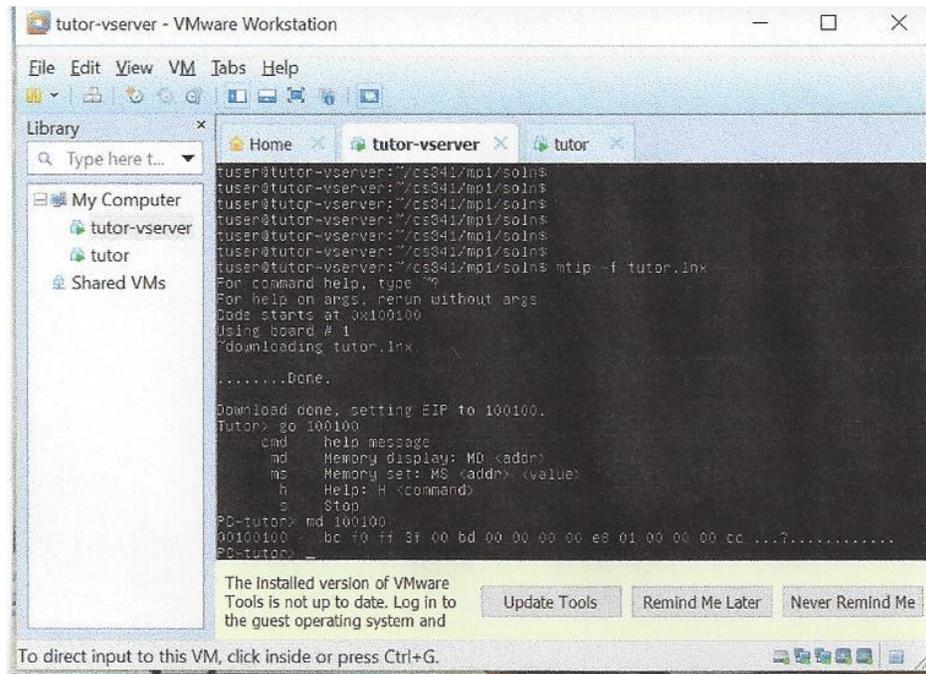
- i) In the vmtutor directory, execute the file:
VMWare-workstation-full-10.0.3-1895310.exe
- ii) Follow the installation program prompts.
- iii) Enter the license key from the file:
VMWare 10 PC Key.txt
- iv) Finish the installation

Configure VMWare Workstation 10 with the tutor and tutor-vserver VMs as follows:

- i) Open the VMware Workstation that you just installed
- ii) Install the tutor-vserver VM by importing the tutor-vserver.ovf (Open Virtualization Format) file into the VMWare Workstation

- a) In VMWare Workstation, 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 **tutorvserver.ovf**, and click **Open**.
 - c) Click **Import** at the next screen.
 - d) the tutor-vserver VM should be installed.
- iii) Install the tutor VM in a separate tab.
Follow the same procedures as in ii) a-d to import **tutor.ovf**.

In the VMWare Workstation frame, you should see the two tabbed VM windows side by side:

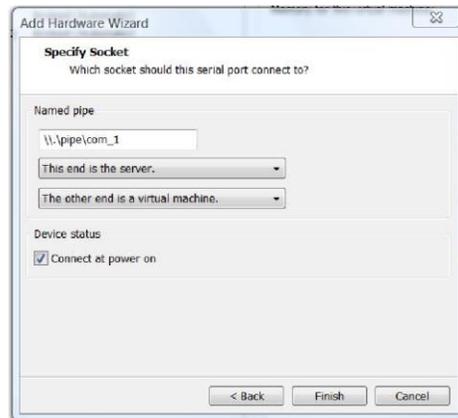


3. Configure the VMs

On a real SAPC, server ulab.cs.umb.edu connects to the SAPC using serial port COM2 for monitoring and serial port COM1 for remote gdb. In the VM environment, we connect the serial ports of the 2 VMs using name pipes. The VMs as provided do not have serial ports configured. We need to add them first and configure their connectivity using name pipes.

In 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 **next**
- iii) Select **Output to named pipes** and **next** and enter data into the next screen as follows:



iv) Select **Finish** when done. Repeat the same to add and configure serial port 2. The named pipe is [\\.\pipe\com_2](#). Then select **OK**.

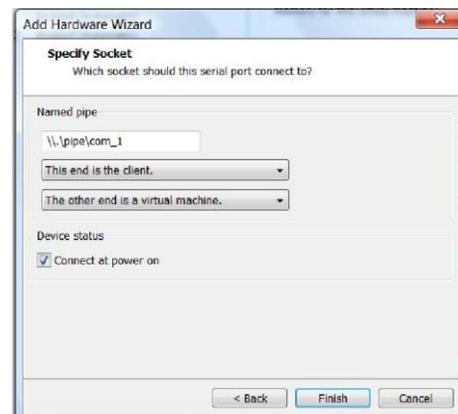
v) Go back to **VM > Setting**, on the **Hardware** tab, select **Network Adapter 2**.

vi) Select **Connected**, **Connected at Power on** and **NAT**. Then select **OK**.

vii) Now switch to configure the tutor VM.

In the **tutor VM**:

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



ii) There is no need to configure the network adapter for this VM since it only connects to tutor-vserver via the serial ports.

3. Running the VMs

i) Power on **tutor-vserver** and logon to its xinu installation 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. Under the home directory, there are several useful directories for building SAPC programs. For now, use ulab for building your program. xinu is a Linux distribution and you can use regular Linux commands to

set up directories, ssh 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) Power on tutor's VM (leave the tutor-vserver VM by Cntl+Alt first). Note that it offers you the chance to enter a CR and make this screen the Tutor console. You can do the CR here (after clicking on that window) and play with Tutor there, but the really useful way to use Tutor is by talking to it on COM2 as we do on ulab.cs.umb.edu. If you want to reset the tutor VM, select the "Send Cntl+Alt+Delete to this VM" button on the VM's menu bar.

iii) Go back to the Linux VM "tutor-vserver" by clicking on its window. Now you should be able to talk to Tutor using mtip and the COM2-COM2 connection between the two systems. This connects using COM2 on the SAPC end, and COM2 known as /dev/ttyS1 on the Linux end. Specifically, just type "mtip" at the shell prompt. Try out "dd" and "h" to see that it works the same as on ulab. Type ~q or two control-Cs to exit back to the Linux prompt.

iv) To test the downloading command, change directory to testelf and you'll see a makefile, test.c and test.lnx, just as in the examples directory on ulab. Type "mtip -f test.lnx" and "~d" to download the test.lnx file to the tutor VM. To run test.lnx, issue "go 100100" just the same as on ulab.

v) After building your programs on ulab, transfer your .lnx file in ~/cs341/mp1/ to the tutor-vserver VM by typing at the tutor-vserver VM:

```
scp username@users.cs.umb.edu:cs341/mp1/*.lnx .
```

4. Running Remote GDB

See \$pcex/gdb.script on how to run remote gdb on the real SAPC. To use remote gdb in the VM environment, build your program (e.g. test.c) on ulab 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.

In vm-vserver VM shell:

run /sbin/ifconfig to find the ip address on your local net: the one for LAN adapter eth1 (e.g. 192.168.1.131).

```
mtip -f test.lnx
```

After clicking "**sending cntl-al-del to VM**" at the tutor VM, and clicking "**CR**" at the vm-server VM, you should see the Tutor prompt.

Tutor> ~d to download the test.lnx executable file to the tutor VM

Tutor> gdb to tutor to accept remote gdb commands.

----get out of the VM with alt+ctrl---

In your host PC:

- i) Run putty or other host tool to ssh to the tutor-vserver VM 192.168.1.131
- ii) Login the VM as tuser with password cs444
- iii) For remote gdb, 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
...
```

OR

- iii) For the user port in mp5:
mtip -l /dev/ttyS0