CMPE283-Virtualization Techniques Nihanjali Mallavarapu

Spring 2020 014492933

Assignment-1 https://github.com/nihanjali/linux

Steps for setting up the assignment environment-

1. Fork the Linux source code from github.com/torvalds/linux into your personal GitHub repository(github.com/Nihanjali/linux)
2. For a MacOS- download **VMWare Fusion 11 Pro** and for windows 10 download **VMWare Workstation 15.5 Pro**
3. Download **Ubuntu 18.04 LTS** iso file from the Ubuntu website. Verify your download with the checksum on the site.
4. After installing VMWare, run through the installation steps, which include selecting the drive which will store the VMs being created. In this case, it was C: Drive.
5. Once the installation procedure is finished, select the create a new virtual machine option from the homescreen.
6. The VM wizard will open and run through a series of prompts- opt for the Easy Install option with all the default settings and vary the memory from the 20GB default option to a greater size(I went for **100GB and a 4GB RAM)**
7. On selecting the create option, a new virtual machine is created which is displayed within a tab on the VMWare homescreen.
8. Set the name, password for the virtual machine.
9. The OS begins its installation process, which is then preceded by a series of Open VM tools installation.
10. Once the OS is installed, a login prompt is shown which when filled in opens up to the Ubuntu Desktop.
11. Using Ctrl+Alt+T, we open the terminal and first check the current file system status of the OS using the ***ls*** command

12. The installation of several key libraries is to be done before attempting to build the Linux kernel.
13. We use the sudo command to enter root and install the libraries using the following command-***sudo apt-get install git build-essential kernel-package fakeroot libncurses5-dev libssl-dev ccache bison flex*** . The command prompts the user to enter the password which then allows admin access to the system.
14. The libraries once installed are followed up with a git clone command which takes a few minutes to clone the forked repository with the linux source code onto the local system. The command is ***git clone https://github.com/nihanjali/linux.git***
15. On cloning the repository, we change the current directory to linux using the command- ***cd linux***
16. Copy the kernel config onto the existing system by using this command- ***cp /boot/config-`uname -r` .config***
17. Bring the old config file up to date by using the command - ***make oldconfig*** This command will prompt a multitude of y/n questions and it's best to stick with enter as a default unless there is a particular config you need to change.
18. Then clean the kernel source directory using the command- ***make clean***
19. Build the linux debian header files and image using this command- ***make -j `getconf _NPROCESSORS_ONLN` deb-pkg LOCALVERSION=-custom*** . Preferably done overnight, as the process takes a long time.
20. Change the directory to a higher level by using the command cd .. That is the level at which the linux header files and the linux image files will be placed.
21. Use this command prefixed with sudo to gain admin privileges- ***sudo dpkg -i linux-image-5.6.0-rc2-custom_5.6.0-rc2-custom-1_amd64.deb***
22. Follow it up with this command- ***sudo dpkg -i linux-headers-5.6.0-rc2-custom_5.6.0-rc2-custom-1_amd64.deb*** This command generally is accompanied with a lot of warnings which may be ignored.

23. Now boot the new kernel which has been built.
24. The command for booting the new kernel would be- ***sudo reboot***
25. On rebooting make sure to select the new kernel which has been built.

Steps for the assignment-

1.Create a directory inside the linux folder, using ***mkdir asgnmt1***

2.Open a text editor like vi, gedit and create the module being inserted. ***gedit pcbctls.c***

3. Save the module and exit the editor.

4.Copy the makefile provided to us and paste it in the assignment folder, check if done properly by using the ls command.

5.Compile the file by using make, trying a C compile with cc will throw lots of errors.

6.If facing errors with make command directly, please check if config operations were done properly when inserting the module.

7.If make executes without errors, a kernel object file with extension .ko will be created.

8.Insert this .ko file into the kernel modules by using sudo insmod filename.ko

9.We can check if the module is inserted by listing all modules using the lsmod command- lsmod | grep filename

10.For checking the information of the module, we use dmesg command which when run gives an output like this.

```
[ 2391.206166] CMPE 283 Assignment 1 Module Start
[ 2391.206174] Pinbased Controls MSR: 0x3f00000016
[ 2391.206176]   External Interrupt Exiting: Can set=Yes, Can clear=Yes
[ 2391.206177]   NMI Exiting: Can set=Yes, Can clear=Yes
[ 2391.206178]   Virtual NMIs: Can set=Yes, Can clear=Yes
[ 2391.206179]   Activate VMX Preemption Timer: Can set=No, Can clear=Yes
[ 2391.206180]   Process Posted Interrupts: Can set=No, Can clear=Yes

[ 2391.206183] Procbased Controls MSR: 0xfff9fffe04006172
[ 2391.206185]   Interrupt-window exiting: Can set=Yes, Can clear=Yes
[ 2391.206186]   Use TSC Offsetting: Can set=Yes, Can clear=Yes
[ 2391.206186]   HLT Exiting: Can set=Yes, Can clear=Yes
[ 2391.206187]   INVLPG Exiting: Can set=Yes, Can clear=Yes
[ 2391.206188]   MWAIT Exiting: Can set=Yes, Can clear=Yes
[ 2391.206189]   RDPMC Exiting: Can set=Yes, Can clear=Yes
[ 2391.206190]   RDTSC Exiting: Can set=Yes, Can clear=Yes
[ 2391.206191]   CR3-Load Exiting: Can set=Yes, Can clear=Yes
[ 2391.206192]   CR3-Store Exiting: Can set=Yes, Can clear=Yes
[ 2391.206193]   CR8-Load Exiting: Can set=Yes, Can clear=Yes
[ 2391.206193]   CR8-Store Exiting: Can set=Yes, Can clear=Yes
[ 2391.206194]   Use TPR Shadow: Can set=Yes, Can clear=Yes
[ 2391.206195]   NMI-Window Exiting: Can set=Yes, Can clear=Yes
[ 2391.206196]   MOV-DR Exiting: Can set=Yes, Can clear=Yes
[ 2391.206197]   Unconditional I/O Exiting: Can set=Yes, Can clear=Yes
[ 2391.206198]   Use I/O Bitmaps: Can set=Yes, Can clear=Yes
[ 2391.206199]   Monitor Trap Flag: Can set=Yes, Can clear=Yes
[ 2391.206200]   Use MSR Bitmaps: Can set=Yes, Can clear=Yes
[ 2391.206201]   MONITOR Exiting: Can set=Yes, Can clear=Yes
[ 2391.206202]   PAUSE Exiting: Can set=Yes, Can clear=Yes

[ 2391.206203]   Activate Secondary Controls: Can set=Yes, Can clear=Yes
[ 2391.206206] Secondary Procbased Controls MSR: 0x553cfe00000000
[ 2391.206207]   Virtualize APIC Accesses: Can set=No, Can clear=Yes
[ 2391.206208]   Enable EPT: Can set=Yes, Can clear=Yes
[ 2391.206209]   Descriptor-table Exiting: Can set=Yes, Can clear=Yes
[ 2391.206210]   Enable RDTSCP: Can set=Yes, Can clear=Yes
[ 2391.206210]   Virtualize x2APIC Mode: Can set=Yes, Can clear=Yes
[ 2391.206211]   Enable VPID: Can set=Yes, Can clear=Yes
[ 2391.206212]   WBINVD Exiting: Can set=Yes, Can clear=Yes
[ 2391.206213]   Unrestricted Guest: Can set=Yes, Can clear=Yes
[ 2391.206214]   APIC-register Virtualization: Can set=No, Can clear=Yes
[ 2391.206215]   Virtual-interrupt Delivery: Can set=No, Can clear=Yes
[ 2391.206216]   PAUSE-loop Exiting: Can set=Yes, Can clear=Yes
[ 2391.206217]   RDRAND Exiting: Can set=Yes, Can clear=Yes
[ 2391.206217]   Enable INVPCID: Can set=Yes, Can clear=Yes
[ 2391.206218]   Enable VM Functions: Can set=Yes, Can clear=Yes
[ 2391.206219]   VMCS Shadowing: Can set=No, Can clear=Yes
[ 2391.206220]   Enable ENCLS Exiting: Can set=No, Can clear=Yes
[ 2391.206221]   RDSEED Exiting: Can set=Yes, Can clear=Yes
[ 2391.206222]   Enable PML: Can set=No, Can clear=Yes
[ 2391.206223]   EPT-violation #VE: Can set=Yes, Can clear=Yes
[ 2391.206224]   Conceal VMX From PT: Can set=No, Can clear=Yes
[ 2391.206225]   Enable XSAVES/XRSTORS: Can set=Yes, Can clear=Yes

[ 2391.206228] Exit Controls MSR: 0x3fffff00036dfb
[ 2391.206229]   Save Debug Controls: Can set=Yes, Can clear=Yes
[ 2391.206230]   Host address-space size: Can set=Yes, Can clear=Yes
[ 2391.206231]   Load IA32_PERF_GLOBAL_CTRL: Can set=Yes, Can clear=Yes
[ 2391.206232]   Acknowledge interrupt: Can set=Yes, Can clear=Yes
[ 2391.206233]   Save IA32_PAT: Can set=Yes, Can clear=Yes
[ 2391.206234]   Load IA32_PAT: Can set=Yes, Can clear=Yes
[ 2391.206234]   Save IA32_EFER: Can set=Yes, Can clear=Yes
[ 2391.206235]   Load IA32_EFER: Can set=Yes, Can clear=Yes
[ 2391.206236]   Save VMX Preemption Timer Value: Can set=No, Can clear=Yes
[ 2391.206237]   Clear IA32_BNDCFGS: Can set=No, Can clear=Yes
[ 2391.206238]   Conceal VMX from PT: Can set=No, Can clear=Yes

[ 2391.206241] Entry Controls MSR: 0xf3ff000011fb
[ 2391.206242]   Load Debug Controls: Can set=Yes, Can clear=Yes
[ 2391.206243]   IA-32e mode guest: Can set=Yes, Can clear=Yes
[ 2391.206244]   Entry to SMM: Can set=No, Can clear=Yes
[ 2391.206245]   Deactivate dual-monitor treatment: Can set=No, Can clear=Yes
[ 2391.206246]   Load IA32_PERF_GLOBAL_CTRL: Can set=Yes, Can clear=Yes
[ 2391.206247]   Load IA32_PAT: Can set=Yes, Can clear=Yes
[ 2391.206248]   Load IA32_EFER: Can set=Yes, Can clear=Yes
[ 2391.206248]   Clear IA32_BNDCFGS: Can set=No, Can clear=Yes
[ 2391.206249]   Conceal VMX from PT: Can set=No, Can clear=Yes
[ 2391.206250]   CMPE 283 Assignment 1 Module end
```

On the first attempt all the MSR values were set to 0x0, but after shutting down the system, changes were made to enable the vt-x mode in the VM settings menu.