

## Answer 1

VMX (Virtual Machine Extensions) provide support for Processor level (hardware-assisted) virtualization. VMX has VMX root operations and VMX non-root operations. VMM runs in VMX root operation and guest OS will run in VMX non-root operation. GuestOS sends a command to VMM and VMM handles it using the instructions specified in its design. Hence, sensitive but not privileged instructions are caught by the VMM and handled by itself.

The VM Execution control fields govern the VMX non-root operations. There are 5 types of execution controls i.e. Pin-based, Processor-based, Secondary Processor-based, Entry-based and Exit-based.

I implemented the logic for Entry-based and Exit-based controls. Entry-based controls report on the allowed settings of VM during the entry of VM in the VMM and Exit-based controls report the allowed settings of VM while exiting the VMM.

## Answer 2 – Steps performed to complete the assignment.

1. Forked github repository torvalds/linux.
2. Clone github repository
3. Added Sarthak Jain as collaborator in Github repository.
4. Copy .config file from usr/src/linux-5.8.40-generic to /home/linux folder.
5. apt-get install flex
6. apt-get install build-essentials
7. make oldconfig
8. Press Enter for all inputs
9. make modules
10. install openssl : sudo apt-get update && sudo apt-get install libssl-dev
11. kernel build failed
12. sudo apt-get install libncurses5-dev gcc make git exuberant-ctags bc libssl-dev
13. cp /boot/config-`uname -r`\*.config
14. make oldconfig
15. make
16. sudo make modules\_install install
17. sudo update-grub2
18. Restarted the machine and could see new kernel in grub menu in Advanced Options for Ubuntu.
19. Team member Sarthak created cmpe283 folder and added his code file.
20. Added IA32\_VMX\_ENTRY\_CTL and IA32\_VMX\_EXIT\_CTL hexcode from assignment information.
21. Created struct(arrays) for Entry and Exit MSR controls of VMX with Bit Positions and Name of control.
22. Added "report\_capability" for Entry and Exit Controls in "detect\_vmx\_features".

23. Compiled cmpe283-1.c with "make" command and created cmpe283-1.ko kernel output file.
24. Added module with command "insmod cmpe283-1.ko"
25. View VMX properties in bash with "dmesg" command.