## **Assignment - 01**

Student Name : Priya Gupta

# **Find-VMX-Features**

This assignment is to develop a Linux kernel module that will consult a number of MSRs to determine the virtualization capabilities of your CPU.

At a high level, we need to perform the following:

• Configure a Linux machine, either VM based or on real hardware. You may use any Linux distribution you wish.

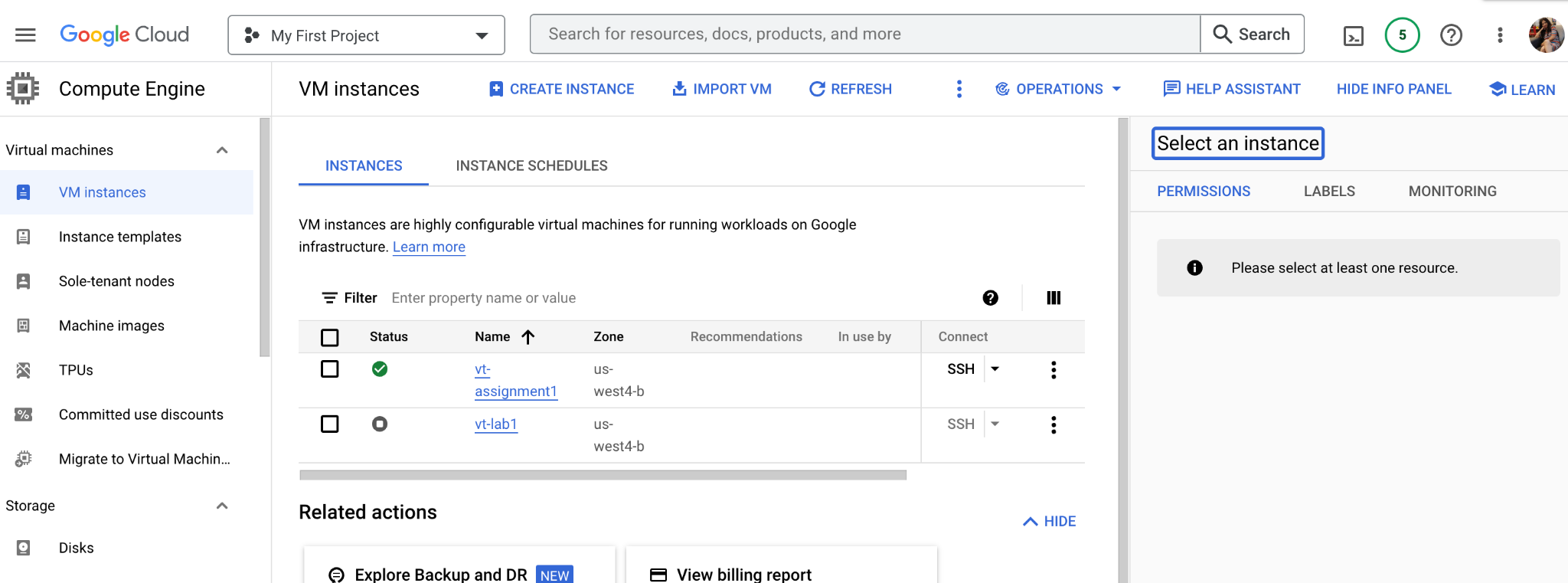
• Download and build the Linux kernel source code

• Create a new kernel module by adding the assignment functionality into it.

• Load (insert) the new module

• Verify proper output in the system message log

Step 1: Create a VM on GCP (Google Cloud Platform) and enable the setting for nested virtualization.

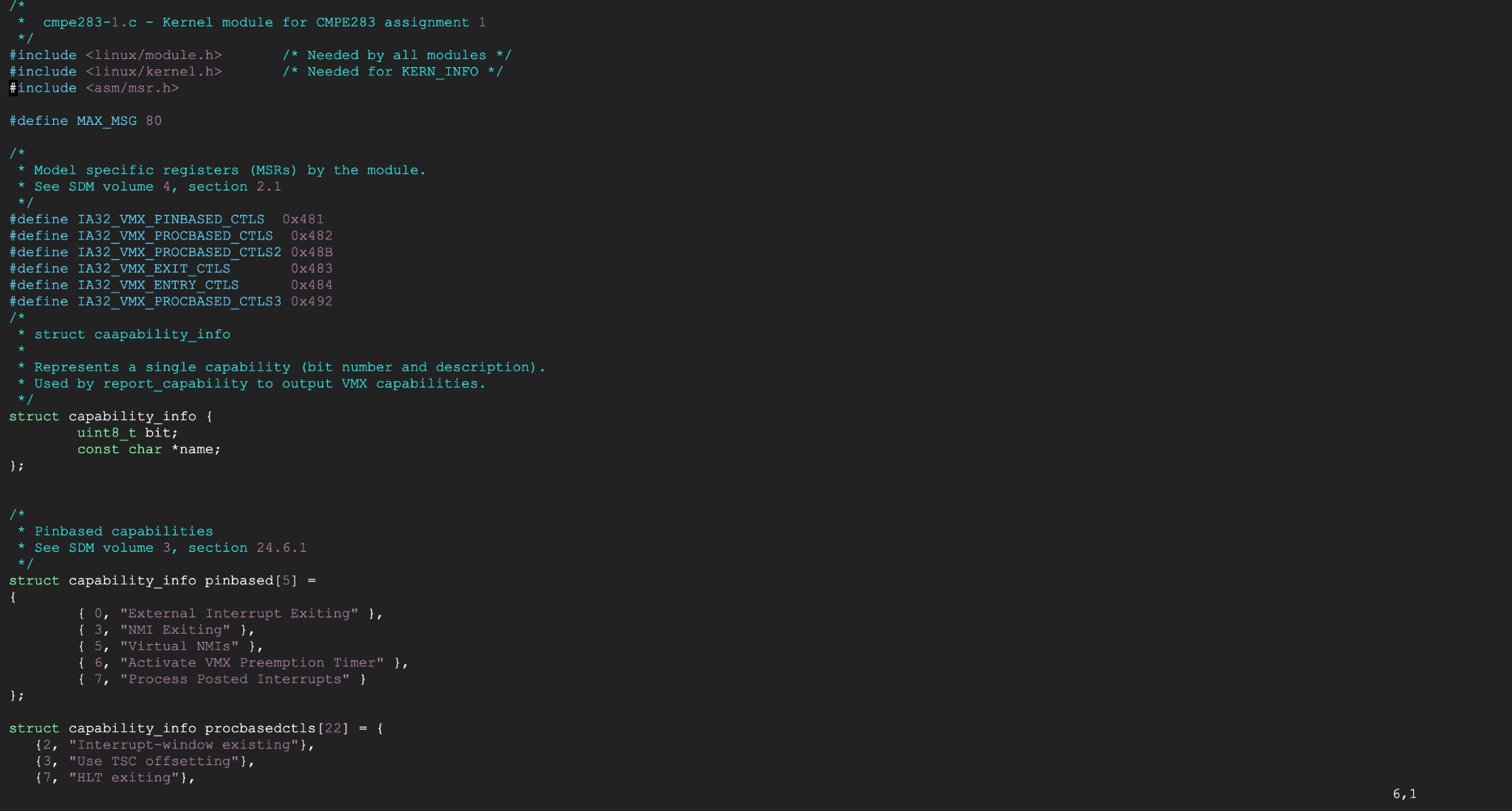


Step 2. Download the cmpe283-1.c source file and Makefile from canvas

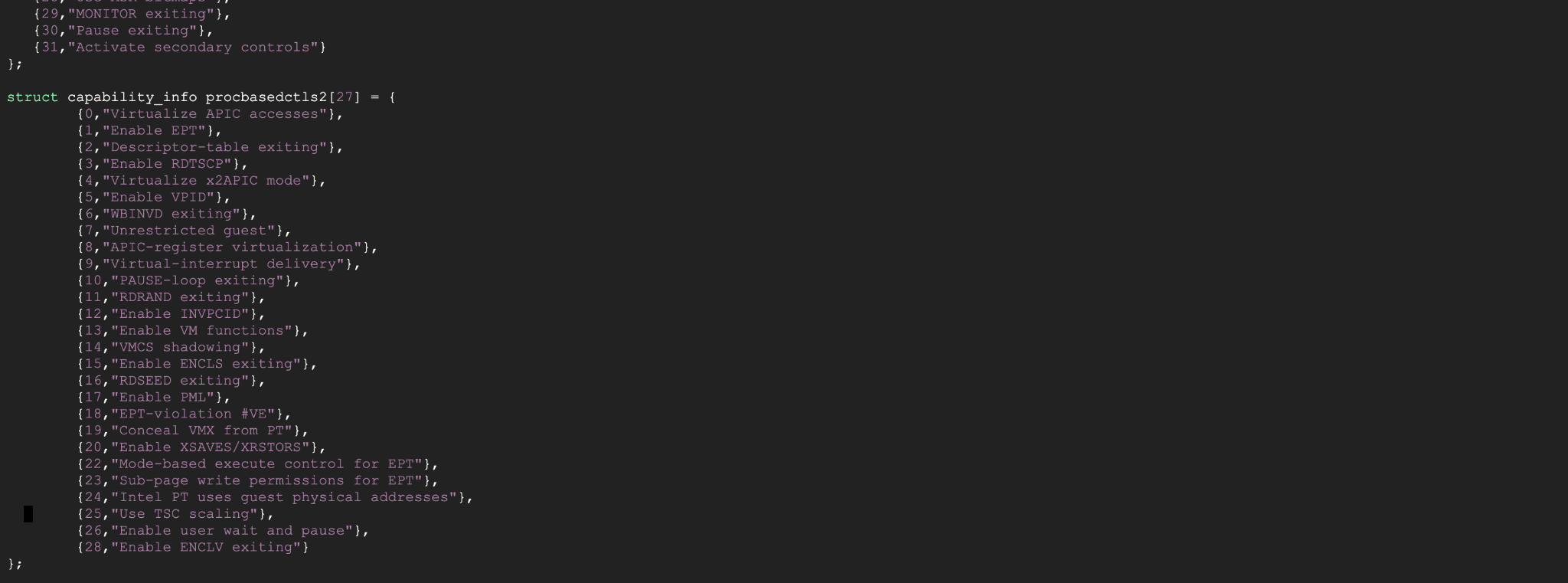


Step 3. The functionality to query all the other MSRs as explained in the assignment description is added to cmpe283-1.c

Taking referral from Intel SDM Volume 4, created structures with name (description) and bit positions for primary procbased, secondary procbased, tertiary procbased, entry and exit controls MSRs.



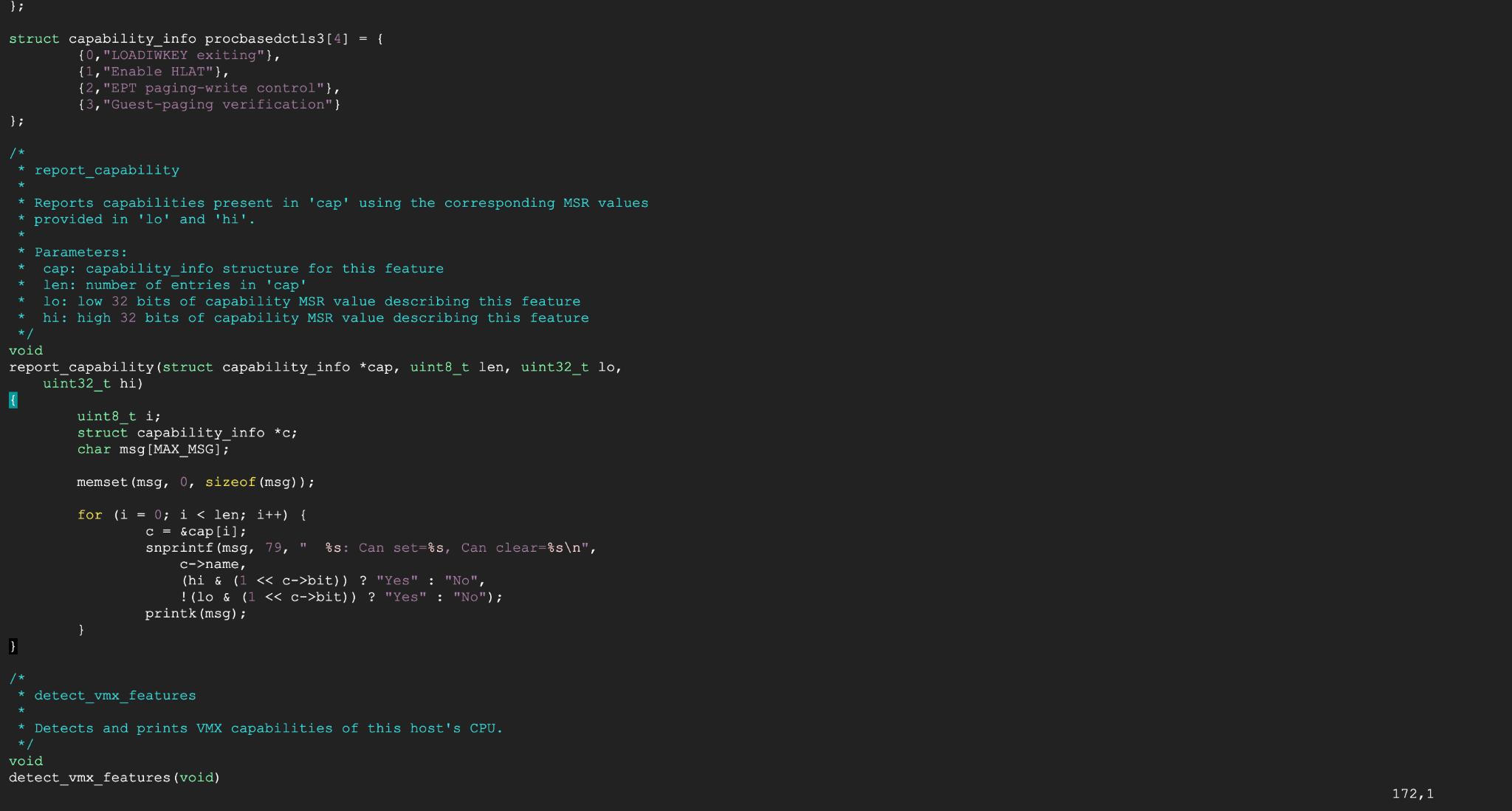




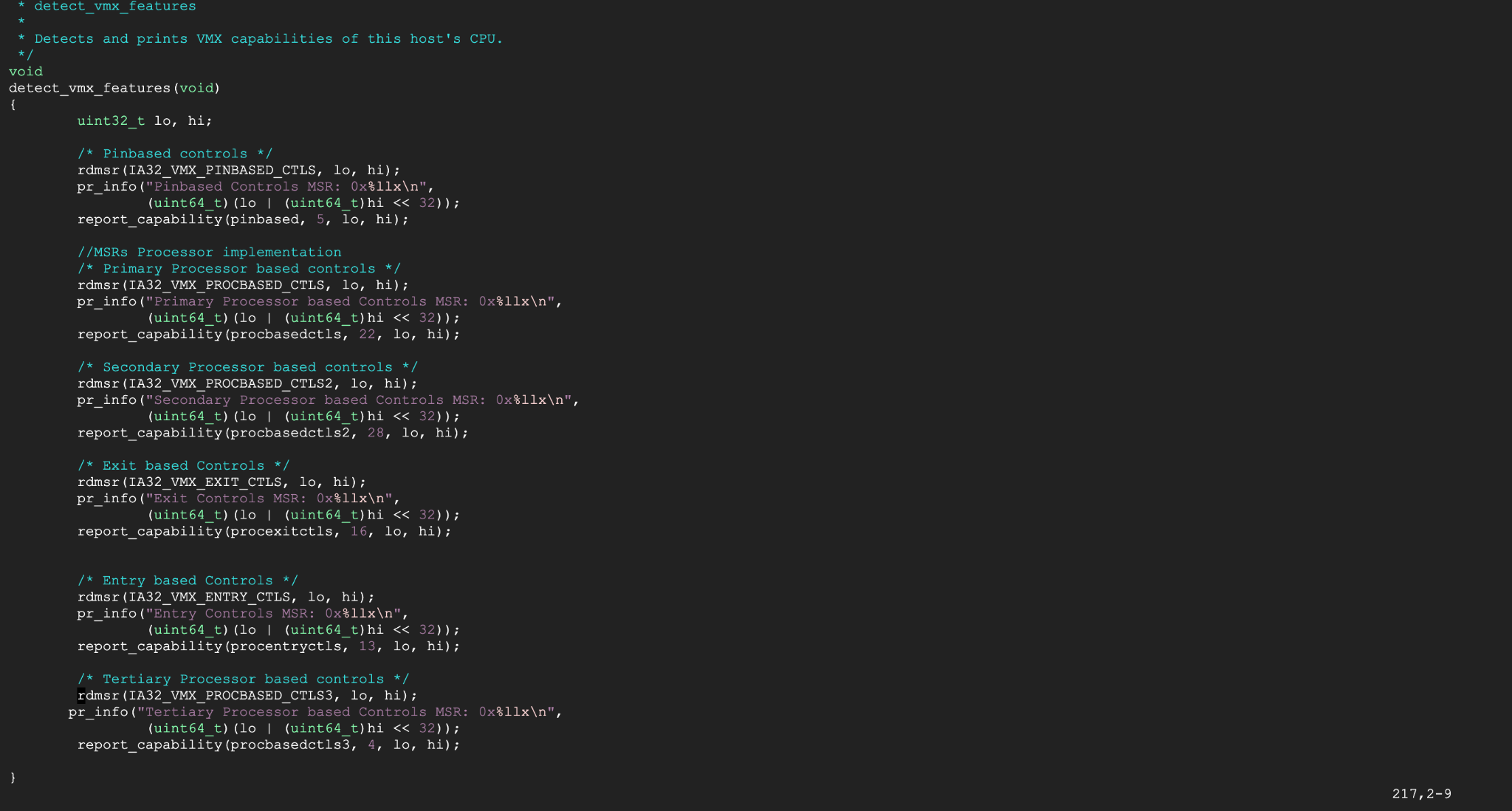


.

In order to identify and print the VMX capabilities of the CPU, the function report capability () is called with the required parameters, printing pin-based, procbased, entry, and exit controls.



To check whether true controls are accessible, a new function called detect vmx has been created. It looks at the VMX MSR's bit. True controls are available if this bit is set, and a different function will be called to print the associated true VMX capabilities. Similar to this, a new function is created for each MSR's controls, and for real VMX capability, the message is printed.



Step 4:

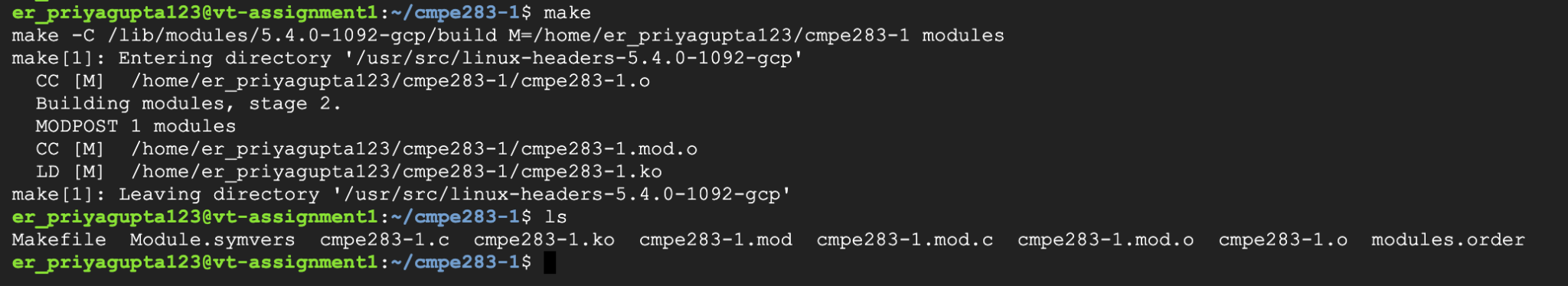
Install packages

1. sudo apt install gcc make
2. er\_priyagupta123@vt-assignment1:~/cmpe283-1$ make

After running make command, couple of files .o/.ko files will be generated

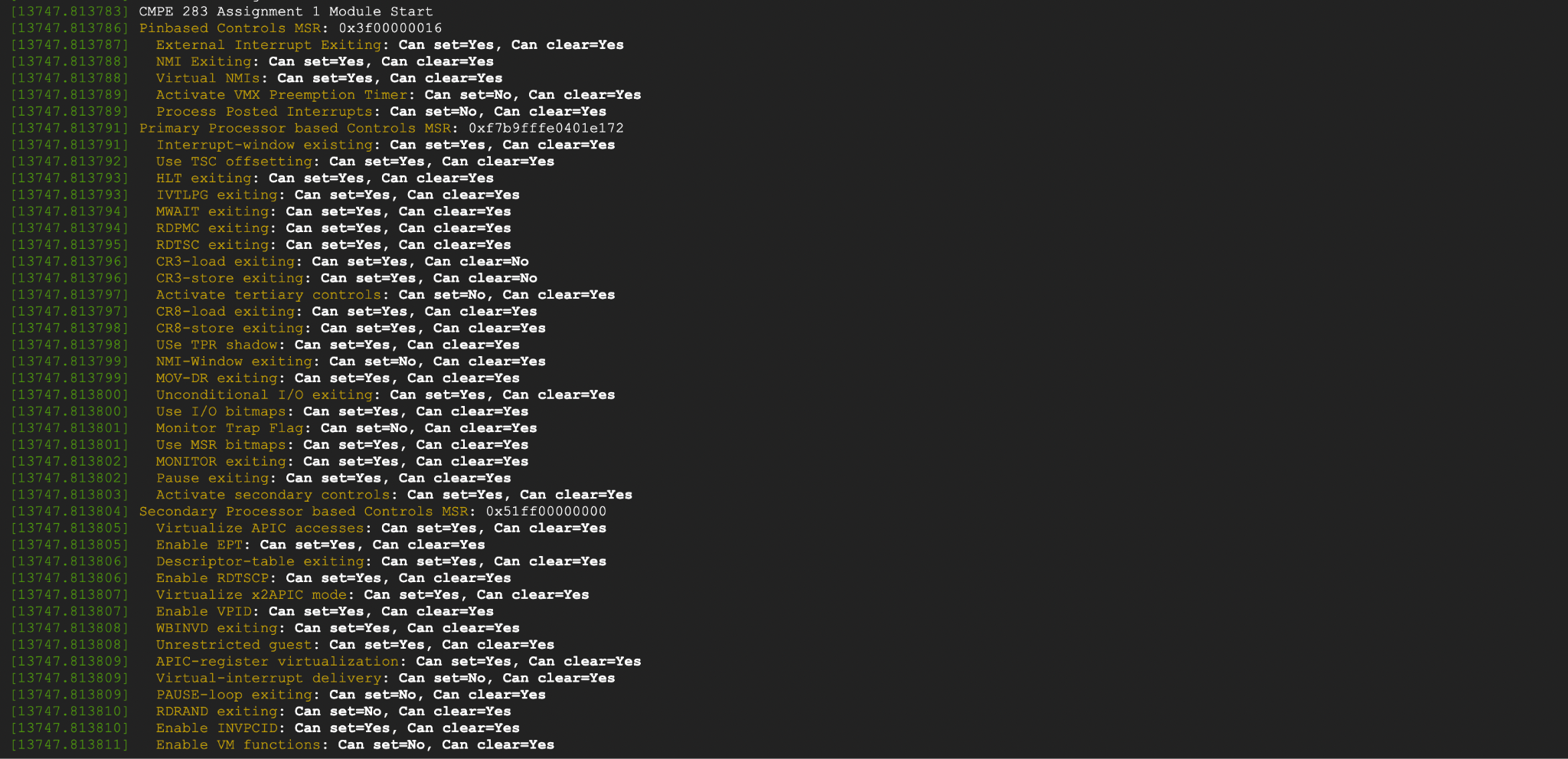
To insert module use below command

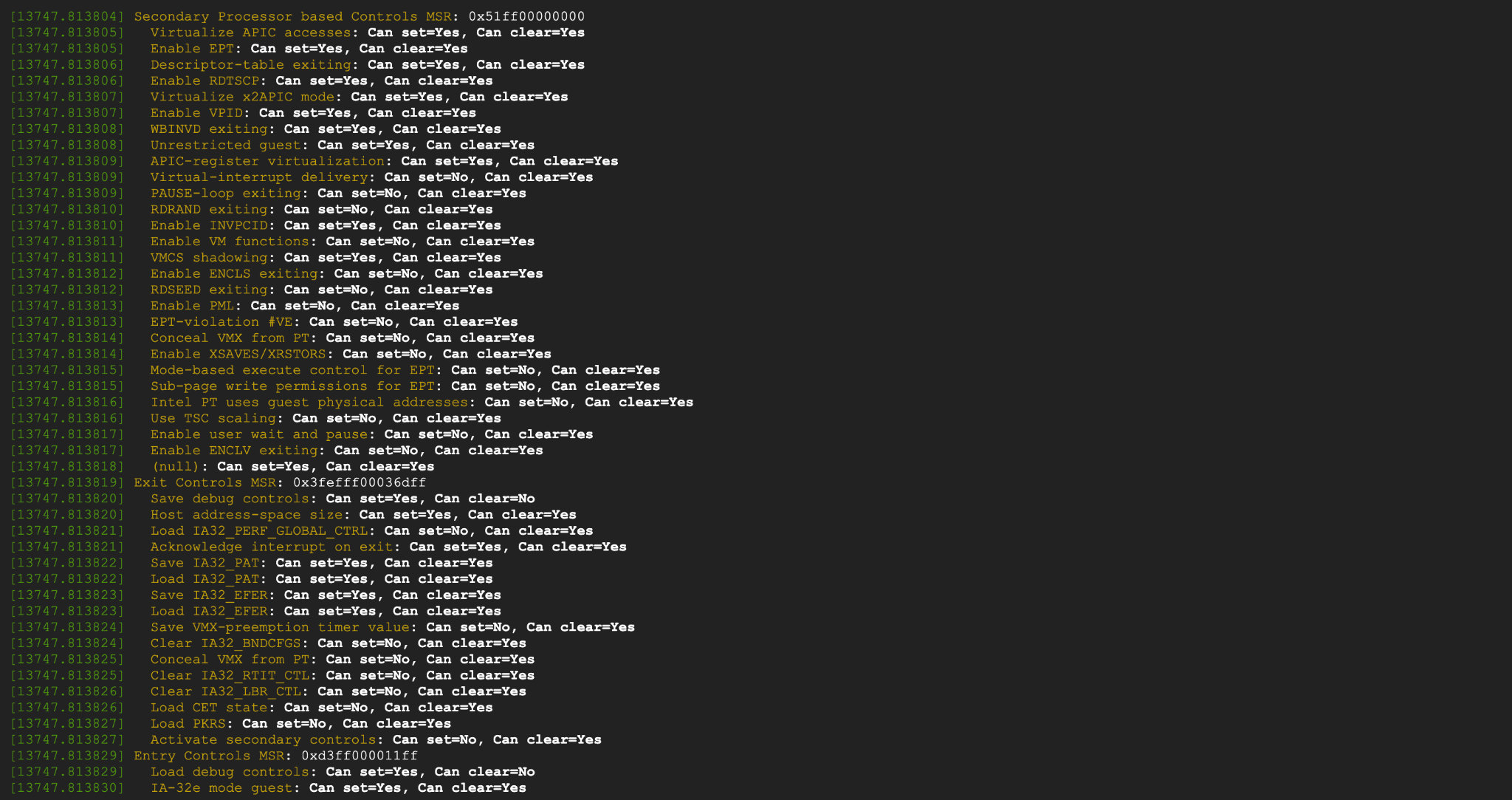
1. er\_priyagupta123@vt-assignment1:~/cmpe283-1$ sudo insmod ./cmpe283-1.ko
2. er\_priyagupta123@vt-assignment1:~/cmpe283-1$ sudo dmesg

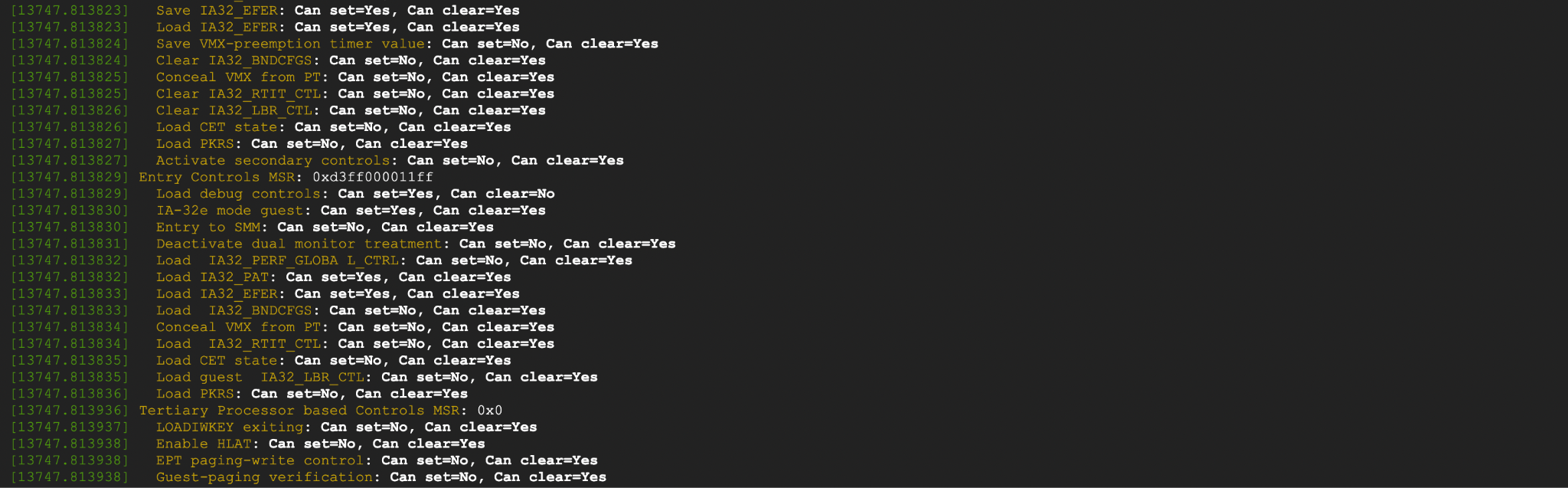




The output for the file is shown in the below screenshots







**Primary Processor based Controls MSR**: 0xf7b9fffe0401e172

[80238.802864] Interrupt-window existing: Can set=Yes, Can clear=Yes

[80238.802864] Use TSC offsetting: Can set=Yes, Can clear=Yes

[80238.802865] HLT exiting: Can set=Yes, Can clear=Yes

[80238.802865] IVTLPG exiting: Can set=Yes, Can clear=Yes

[80238.802866] MWAIT exiting: Can set=Yes, Can clear=Yes

[80238.802867] RDPMC exiting: Can set=Yes, Can clear=Yes

[80238.802867] RDTSC exiting: Can set=Yes, Can clear=Yes

[80238.802868] CR3-load exiting: Can set=Yes, Can clear=No

[80238.802868] CR3-store exiting: Can set=Yes, Can clear=No

[80238.802869] Activate tertiary controls: Can set=No, Can clear=Yes

[80238.802869] CR8-load exiting: Can set=Yes, Can clear=Yes

[80238.802870] CR8-store exiting: Can set=Yes, Can clear=Yes

[80238.802870] USe TPR shadow: Can set=Yes, Can clear=Yes

[80238.802871] NMI-Window exiting: Can set=No, Can clear=Yes

[80238.802871] MOV-DR exiting: Can set=Yes, Can clear=Yes

[80238.802872] Unconditional I/O exiting: Can set=Yes, Can clear=Yes

[80238.802872] Use I/O bitmaps: Can set=Yes, Can clear=Yes

[80238.802872] Monitor Trap Flag: Can set=No, Can clear=Yes

[80238.802873] Use MSR bitmaps: Can set=Yes, Can clear=Yes

[80238.802873] MONITOR exiting: Can set=Yes, Can clear=Yes

[80238.802874] Pause exiting: Can set=Yes, Can clear=Yes

[80238.802874] Activate secondary controls: Can set=Yes, Can clear=Yes

**Secondary Processor based Controls MSR**: 0x51ff00000000

[80238.802876] Virtualize APIC accesses: Can set=Yes, Can clear=Yes

[80238.802877] Enable EPT: Can set=Yes, Can clear=Yes

[80238.802877] Descriptor-table exiting: Can set=Yes, Can clear=Yes

[80238.802878] Enable RDTSCP: Can set=Yes, Can clear=Yes

[80238.802878] Virtualize x2APIC mode: Can set=Yes, Can clear=Yes

[80238.802879] Enable VPID: Can set=Yes, Can clear=Yes

[80238.802879] WBINVD exiting: Can set=Yes, Can clear=Yes

[80238.802880] Unrestricted guest: Can set=Yes, Can clear=Yes

[80238.802880] APIC-register virtualization: Can set=Yes, Can clear=Yes

[80238.802881] Virtual-interrupt delivery: Can set=No, Can clear=Yes

[80238.802881] PAUSE-loop exiting: Can set=No, Can clear=Yes

[80238.802882] RDRAND exiting: Can set=No, Can clear=Yes

[80238.802882] Enable INVPCID: Can set=Yes, Can clear=Yes

[80238.802882] Enable VM functions: Can set=No, Can clear=Yes

[80238.802883] VMCS shadowing: Can set=Yes, Can clear=Yes

[80238.802883] Enable ENCLS exiting: Can set=No, Can clear=Yes

[80238.802884] RDSEED exiting: Can set=No, Can clear=Yes

[80238.802884] Enable PML: Can set=No, Can clear=Yes

[80238.802885] EPT-violation #VE: Can set=No, Can clear=Yes

[80238.802885] Conceal VMX from PT: Can set=No, Can clear=Yes

[80238.802886] Enable XSAVES/XRSTORS: Can set=No, Can clear=Yes

[80238.802886] Mode-based execute control for EPT: Can set=No, Can clear=Yes

[80238.802887] Sub-page write permissions for EPT: Can set=No, Can clear=Yes

[80238.802887] Intel PT uses guest physical addresses: Can set=No, Can clear=Yes

[80238.802888] Use TSC scaling: Can set=No, Can clear=Yes

[80238.802888] Enable user wait and pause: Can set=No, Can clear=Yes

[80238.802889] Enable ENCLV exiting: Can set=No, Can clear=Yes

**Exit Controls MSR**: 0x3fefff00036dff

[80238.802891] Save debug controls: Can set=Yes, Can clear=No

[80238.802892] Host address-space size: Can set=Yes, Can clear=Yes

[80238.802892] Load IA32\_PERF\_GLOBAL\_CTRL: Can set=No, Can clear=Yes

[80238.802893] Acknowledge interrupt on exit: Can set=Yes, Can clear=Yes

[80238.802893] Save IA32\_PAT: Can set=Yes, Can clear=Yes

[80238.802894] Load IA32\_PAT: Can set=Yes, Can clear=Yes

[80238.802894] Save IA32\_EFER: Can set=Yes, Can clear=Yes

[80238.802895] Load IA32\_EFER: Can set=Yes, Can clear=Yes

[80238.802895] Save VMX-preemption timer value: Can set=No, Can clear=Yes

[80238.802896] Clear IA32\_BNDCFGS: Can set=No, Can clear=Yes

[80238.802896] Conceal VMX from PT: Can set=No, Can clear=Yes

[80238.802896] Clear IA32\_RTIT\_CTL: Can set=No, Can clear=Yes

[80238.802897] Clear IA32\_LBR\_CTL: Can set=No, Can clear=Yes

[80238.802897] Load CET state: Can set=No, Can clear=Yes

[80238.802898] Load PKRS: Can set=No, Can clear=Yes

[80238.802898] Activate secondary controls: Can set=No, Can clear=Yes

**Entry Controls MSR**: 0xd3ff000011ff

[80238.802901] Load debug controls: Can set=Yes, Can clear=No

[80238.802901] IA-32e mode guest: Can set=Yes, Can clear=Yes

[80238.802902] Entry to SMM: Can set=No, Can clear=Yes

[80238.802902] Deactivate dual monitor treatment: Can set=No, Can clear=Yes

[80238.802903] Load IA32\_PERF\_GLOBA L\_CTRL: Can set=No, Can clear=Yes

[80238.802903] Load IA32\_PAT: Can set=Yes, Can clear=Yes

[80238.802904] Load IA32\_EFER: Can set=Yes, Can clear=Yes

[80238.802904] Load IA32\_BNDCFGS: Can set=No, Can clear=Yes

[80238.802905] Conceal VMX from PT: Can set=No, Can clear=Yes

[80238.802905] Load IA32\_RTIT\_CTL: Can set=No, Can clear=Yes

[80238.802906] Load CET state: Can set=No, Can clear=Yes

[80238.802906] Load guest IA32\_LBR\_CTL: Can set=No, Can clear=Yes

[80238.802907] Load PKRS: Can set=No, Can clear=Yes

**Tertiary Processor based Controls MSR**: 0x0

[80238.803005] LOADIWKEY exiting: Can set=No, Can clear=Yes

[80238.803006] Enable HLAT: Can set=No, Can clear=Yes

[80238.803006] EPT paging-write control: Can set=No, Can clear=Yes

[80238.803006] Guest-paging verification: Can set=No, Can clear=Yes