Discovering VMX Features

Submitted by:

Akash Gupta <akash.gupta@sjsu.edu>

Nikhilesh Chaudhary < nikhilesh.chaudhary@sjsu.edu>

Question 1: For each member in your team, provide 1 paragraph detailing what part of the lab that member implemented / researched.

Answer :
Work done by Akash Gupta:

- Performed environment setup.
- Built and compiled Kernel modules using various make commands also ensuring correct compilation.
- Wrote new kernel module for MSRs.
- Prepared report for the final submission.

Work done by Nikhilesh Chaudhary:

- Performed environment setup.
- Built and compiled Kernel modules using various make commands also ensuring correct compilation.
- Fixed the issues encountered while building the linux kernel modules by researching about the issues on the internet.
- Wrote new kernel module for MSRs.
- Generated diff file and the output logs.

Question 2: Describe in detail the steps used to complete the assignment.

Answer:

1. Install git using following commands:
   - sudo apt-get update
   - sudo apt-get upgrade
   - sudo apt --fix-broken install (We had to run this command to fix a problem which was occurring during installation of git)
   - sudo apt-get install git

2. Clone linux kernel tree from github using following command
   - git clone https://github.com/torvalds/linux.git

3. Change directory to linux
   - cd linux

4. Enter following command to see git all commit and save latest git commit id
   - git log
   - commit id: 8f5fd927c3a7576d57248a2d7a0861c3f2795973

5. Before trying to build the Linux Kernel enter the following commands:

- sudo apt-get install libncurses-dev
- sudo apt-get install libssl-dev
- sudo apt-get install bison
- sudo apt-get install flex
- sudo apt install libelf-dev

6. Run command ‘make menuconfig’ and save the configuration file.
7. In order to build the linux kernel module, Run the following commands in super user mode
    - sudo make
    - sudo make modules
    - sudo make modules_install
    - sudo make install

8. Create a new directory cmpe283 in linux directory and go into that directory
    - mkdir cmpe283
    - cd cmpe283
    - Create makefile and cmpe283-1.c and run following command
    - sudo make all

9. Insert kernel module into kernel and view the message buffer using following commands:
    - sudo insmod ./cmpe283-1.ko

10. View the message buffer / output from the kernel
    - dmesg

git log:

```
commit ac4ca0183181bda0a08f55b1a1809a35e63b3265
Author: NikhileshChaudhary2903 <nikhilchaudhary2903@gmail.com>
Date:   Tue Mar 20 21:46:51 2018 -0700

    Inserted New Module for detecting VMX Features for Assignment 1

commit 8f5fd927c3a7576d57248a2d7a0861c3f2795973
Merge: 8757ae2 093e037
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Fri Mar 16 13:37:42 2018 -0700

    Merge tag 'for-4.16-rc5-tag' of git://git.kernel.org/pub/scm/linux/kernel/git/kdave/linux
```

Output :

```
[  701.868847] CMPE 283 Assignment 1 Module Start!
[  701.868848] Reading TRUE Based MSRs since 55th bit is set to 1
[  701.868849] Pinbased Controls MSR: 0x7f00000016
[  701.868852]  External-interrupt exiting: Can set=Yes, Can clear=Yes

[  701.868855]  NMI exiting: Can set=Yes, Can clear=Yes

[  701.868858]  Virtual NMIs: Can set=Yes, Can clear=Yes

[  701.868860]  Activate VMX-preemption timer: Can set=Yes, Can clear=Yes

[  701.868863]  Process posted interrupts: Can set=No, Can clear=Yes


[  701.868866] True Procbased Controls MSR: 0xfff9fffe04006172
[  701.868868]  Interrupt-window: Can set=Yes, Can clear=Yes

[  701.868870]  Use TSC offsetting: Can set=Yes, Can clear=Yes

[  701.868873]  HLT exiting: Can set=Yes, Can clear=Yes

[  701.868875]  INVLPG exiting: Can set=Yes, Can clear=Yes

[  701.868877]  MWAIT exiting: Can set=Yes, Can clear=Yes

[  701.868880]  RDPMC exiting: Can set=Yes, Can clear=Yes

[  701.868882]  RDTSC exiting: Can set=Yes, Can clear=Yes

[  701.868885]  CR3-load exiting: Can set=Yes, Can clear=Yes

[  701.868887]  CR3-store exiting: Can set=Yes, Can clear=Yes

[  701.868890]  CR8-load exiting: Can set=Yes, Can clear=Yes

[  701.868893]  CR8-store exiting: Can set=Yes, Can clear=Yes

[  701.868895]  Use TPR shadow: Can set=Yes, Can clear=Yes

[  701.868898]  NMI-window exiting: Can set=Yes, Can clear=Yes

[  701.868900]  MOV-DR exiting: Can set=Yes, Can clear=Yes
[  701.868903]  Unconditional I/O: Can set=Yes, Can clear=Yes

[  701.868905]  Use I/O bitmaps: Can set=Yes, Can clear=Yes

[  701.868907]  Monitor trap flag: Can set=Yes, Can clear=Yes

[  701.868910]  Use MSR Bitmaps: Can set=Yes, Can clear=Yes

[  701.868912]  MONITOR exiting: Can set=Yes, Can clear=Yes

[  701.868915]  PAUSE exiting: Can set=Yes, Can clear=Yes

[  701.868917]  Activate secondary controls: Can set=Yes, Can clear=Yes


[  701.868920] Reading Secondary procbased MSR since 63rd bit of Procbased CTLS is set to 1
[  701.868936] Secondary procbased Controls MSR: 0x5fbcff00000000
[  701.868938]  Virtualize APIC accesses: Can set=Yes, Can clear=Yes

[  701.868945]  Enable EPT: Can set=Yes, Can clear=Yes

[  701.868956]  Descriptor-table exiting: Can set=Yes, Can clear=Yes

[  701.868971]  Enable RDTSCP: Can set=Yes, Can clear=Yes

[  701.868985]  Virtualize x2APIC mode: Can set=Yes, Can clear=Yes

[  701.868998]  Enable VPID: Can set=Yes, Can clear=Yes

[  701.869010]  WBINVD exiting: Can set=Yes, Can clear=Yes

[  701.869021]  Unrestricted guest: Can set=Yes, Can clear=Yes

[  701.869034]  APIC-register virtualization: Can set=No, Can clear=Yes

[  701.869048]  Virtual-interrupt delivery: Can set=No, Can clear=Yes

[  701.869060]  PAUSE-loop exiting: Can set=Yes, Can clear=Yes

[  701.869072]  RDRAND exiting: Can set=Yes, Can clear=Yes

[  701.869083]  Enable INVPCID: Can set=Yes, Can clear=Yes

[  701.869097]  Enable VM functions: Can set=Yes, Can clear=Yes
[ 701.869114]  VMCS shadowing: Can set=No, Can clear=Yes

[ 701.869129]  Enable ENCLS exiting: Can set=Yes, Can clear=Yes

[ 701.869140]  RDSEED exiting: Can set=Yes, Can clear=Yes

[ 701.869154]  Enable PML: Can set=Yes, Can clear=Yes

[ 701.869164]  EPT-violation: Can set=Yes, Can clear=Yes

[ 701.869172]  Conceal VMX nonroot operation from Intel PT: Can set=Yes, Can clear=Yes

[ 701.869186]  Enable XSAVES/XRSTORS: Can set=Yes, Can clear=Yes

[ 701.869188]  Mode-based execute control for EPT: Can set=Yes, Can clear=Yes

[ 701.869191]  Use TSC scaling: Can set=No, Can clear=Yes


[ 701.869194] True Entry Controls MSR: 0x3ffff000011fb
[ 701.869198]  Load debug controls: Can set=Yes, Can clear=Yes

[ 701.869210]  IA-32e mode guest: Can set=Yes, Can clear=Yes

[ 701.869227]  Entry to SMM: Can set=Yes, Can clear=Yes

[ 701.869241]  Deactivate dual-monitor treatment: Can set=Yes, Can clear=Yes

[ 701.869255]  Load IA32_PERF_GLOBAL_CTRL: Can set=Yes, Can clear=Yes

[ 701.869272]  Load IA32_PAT: Can set=Yes, Can clear=Yes

[ 701.869283]  Load IA32_EFER: Can set=Yes, Can clear=Yes

[ 701.869299]  Load IA32_BNDCFGS: Can set=Yes, Can clear=Yes


[ 701.869302] True Exit Controls MSR: 0x1ffffff00036dfb
[ 701.869304]  Save debug controls: Can set=Yes, Can clear=Yes

[ 701.869307]  Host addressspace size: Can set=Yes, Can clear=Yes

[ 701.869310]  Load IA32_PERF_GLOB AL_CTRL: Can set=Yes, Can clear=Yes
[  701.869314]  Acknowledge interrupt on exit: Can set=Yes, Can clear=Yes
[  701.869317]  Save IA32_PAT: Can set=Yes, Can clear=Yes
[  701.869320]  Load IA32_PAT: Can set=Yes, Can clear=Yes
[  701.869322]  Save IA32_EEFR: Can set=Yes, Can clear=Yes
[  701.869325]  Load IA32_EFER: Can set=Yes, Can clear=Yes
[  701.869328]  Save VMXpreemption timer value: Can set=Yes, Can clear=Yes
[  701.869330]  Clear IA32_BNDCFGS: Can set=Yes, Can clear=Yes
[  701.869333]  Conceal VM exits from Intel PT: Can set=Yes, Can clear=Yes
```