Submission Details:
Name: Smitha Venkatesh
Student ID: 011825177

Git Commit ID:
commit: 076264ada9a307300be7a4581165f9c9deed5d1b

2) Describe in detail the steps you used to complete the assignment.
Answer:
  Steps:
  1) $uname -r
    4.14.0-rc3+

  2) $git log

    commit afdecff61bf96ceaa77a13025828c8200272354b
    Author: Smitha <smithav17@gmail.com>
    Date: Thu Oct 12 16:04:00 2017 -0700
    Modified cmpe283-1.c according to assignment1 functionalities.

    commit 355dde33b61d67b19af1deb827fdcd4c562be67e
    Author: Smitha <smithav17@gmail.com>
    Date: Fri Oct 6 07:21:55 2017 -0700
    Added 2 files Makefile and cmpe283-1.c

    commit 076264ada9a307300be7a4581165f9c9deed5d1b
    Merge: 0f38071 41dcf19
    Author: Linus Torvalds <torvalds@linux-foundation.org>
    Date: Thu Oct 5 15:17:40 2017 -0700

  3) $git reset --hard 076264ada9a307300be7a4581165f9c9deed5d1b
    HEAD is now at 076264a Merge tag 'for-4.14/dm-fixes' of git://
git.kernel.org/pub/scm/linux/kernel/git/device-mapper/linux-dm

  4) $git log
    commit 076264ada9a307300be7a4581165f9c9deed5d1b
    Merge: 0f38071 41dcf19
    Author: Linus Torvalds <torvalds@linux-foundation.org>
    Date: Thu Oct 5 15:17:40 2017 -0700

  5) Install virtual manager: $sudo apt-get install virt-manager

  6) Launch virtual manager :
    [Have error
    Unable to connect to libvirt.

    Verify that:
    - The 'libvirt-bin' package is installed
    - The 'libvirtd' daemon has been started

```
    - You are member of the 'libvirtd' group

    Libvirt URI is: qemu:///system

    Traceback (most recent call last):
    File "/usr/share/virt-manager/virtManager/connection.py", line
903, in _do_open
    self._backend.open(self._do_creds_password)
    File "/usr/share/virt-manager/virtinst/connection.py", line 148,
in open
    open_flags)
    File "/usr/lib/python2.7/dist-packages/libvirt.py", line 105, in
openAuth
    if ret is None:raise libvirtError('virConnectOpenAuth() failed')
    libvirtError: Failed to connect socket to '/var/run/libvirt/
libvirt-sock': No such file or directory]


    run this:-
    $ /etc/init.d/apparmor stop
    [ ok ] Stopping apparmor (via systemctl): apparmor.service.

    $sudo update-rc.d apparmor remove

    $sudo apt remove libvirtd
    Reading package lists... Done
    Building dependency tree
    Reading state information... Done
    E: Unable to locate package libvirtd

    $sudo apt remove libvirt
    Reading package lists... Done
    Building dependency tree
    Reading state information... Done
    E: Unable to locate package libvirt

    $sudo apt remove virt-manager
    $sudo apt remove libvirt-bin
    $sudo apt-get install virt-manager
    $reboot
    Launch virtual manager, Install the VM

  7) Edit the code in linux/arch/x86/kvm/cpuid.c to perform the
functionality of assignment

  8) $sudo make && sudo make modules && sudo make modules_install &&
sudo make install

  9) $uname -r
    4.14.0-rc3+
```

10) Reboot

11) $uname -r
4.14.0-rc3+

12) $sudo make clean

13) $sudo make all

14) $lsmod | grep kvm

15) Remove the leaf modules first and later the dependent module.
$sudo rmmod kvm_intel

16) $sudo rmmod kvm

17) $ sudo insmod arch/x86/kvm/kvm.ko

18) $ sudo insmod arch/x86/kvm/kvm-intel.ko

19) Build the kernel again usingbelow commands.
$sudo make && sudo make modules && sudo make modules_install && sudo make install

20) Test cpuid in the guest VM using user code.

21) $git commit -a

22) $git log

commit 23b1478b288c68e9829865573b51029de03c9868
Author: Smitha <smithav17@gmail.com>
Date: Mon Nov 6 09:42:58 2017 -0800
Edited CPUID.c file to implement the assignment 2 functionality.

23)$git diff HEAD~1 > cmpe283-2.diff

3) With the assignment functionality enabled, boot a second linux VM (this can just be a plain linux VM
or a copy of your test VM).

◦ What happens during boot? (Hint: check dmesg output).
Answer:
During boot in the dmesg output we get a CPU Vendor ID unknown warning, and also says the system is unstable.
[Vendor ID "CMPE_283CMPE unknown,using generic init. System might be unstable]

◦ Does the system behave differently?

Answer:

- Irrespective of functionality being enabled or disabled, the test VM works similar to host VM.
- If the functionality is enabled and the test VM is booted, the dmesg output in the test VM shows a warning as CPU Vendor ID unknown warning,
  and also says the system is unstable.

◦ Does the content of /proc/cpuinfo change when the functionality is enabled vs disabled?

Answer:

The vendor_id in /proc/cpuinfo is the only field that changes.

- It will be set to the system default value when the system is booted, given that the functionality is DISABLED.
- It will be set to the custom value when the system is booted, if the functionality is toggled from DISABLED to ENABLED *before* the reboot.
- NOTE: The effect of toggling of the functionality (DISABLED to ENABLED, or, ENABLED to DISABLED) will take effect only on a system reboot. Once the system has already booted, any toggling of the functionality will not affect the value of the /proc/cpuinfo.

◦ What happens if you disable the functionality and restart the test VM?

Answer:

Disabling the functionality and restarting the test VM has no effect the working of test VM.