CMPE 283: Virtual Technologies

Assignment 2 & 3: Instrumentation via hypercall

Dharahasini Gangalapudi (015961361)

Environment Setup:

- Fork the Linux repository from Torvalds Github
  - https://github.com/torvalds/linux.git

- Clone the forked Linux repository from your Github.
  - git clone https://github.com/hasinireddy23/linux.git

- Enter sudo mode with sudo bash.

- Get all the build-essentials required for compilation by running the following command.
  - apt-get install build-essential kennel-package fakeroot libncurses5-dev libssl-dev ccache bison flex libelf-dev

- Check the current kernel version using uname -a.

- Copy that version config file and create a config file.
  - cp /boot/config-5.11.0-40-generic ./.config

- Now, run make oldconfig cmd and hold the enter button to take the default values.

- Now run the following command.
  - sudo make && sudo make modules && sudo make install && sudo make modules-install.

- Now reboot the system and give the command uname -a to know the latest version of the Linux kernel.

Modifying Kernel code:

- Modify cpuid.c and vmx.c files accordingly to implement the given functionalities :
    - To calculate the total number of exits
    - To return the low 32bits and high 32 bits of the total time spent processing all exits
    - To calculate the number of exits based on the exit provided as input.
    - To return the low 32bits and high 32 bits of the total time spent processing for the exit provided as input.

Note: file locations : cpuid.c - linux/arch/x86/kvm/cpuid.c
vmx.c - linux/arch/x86/kvm/vmx/vmx.c

- Rebuild the kernel using the following command.
    - sudo make -j 2 modules M=arch/x86/kvm (use command nproc to know no. of CPUs)

- Now, perform loading and unloading of kvm kernel module (kvm.ko) and kvm-intel-module (kvm-intel.ko) using the following commands:
    - sudo rmmod arch/x86/kvm/kvm-intel.ko
    - sudo rmmod arch/x86/kvm/kvm.ko
    - sudo insmod arch/x86/kvm/kvm.ko
    - sudo insmod arch/x86/kvm/kvm-intel.ko

Perform testing in the inner virtual machine:

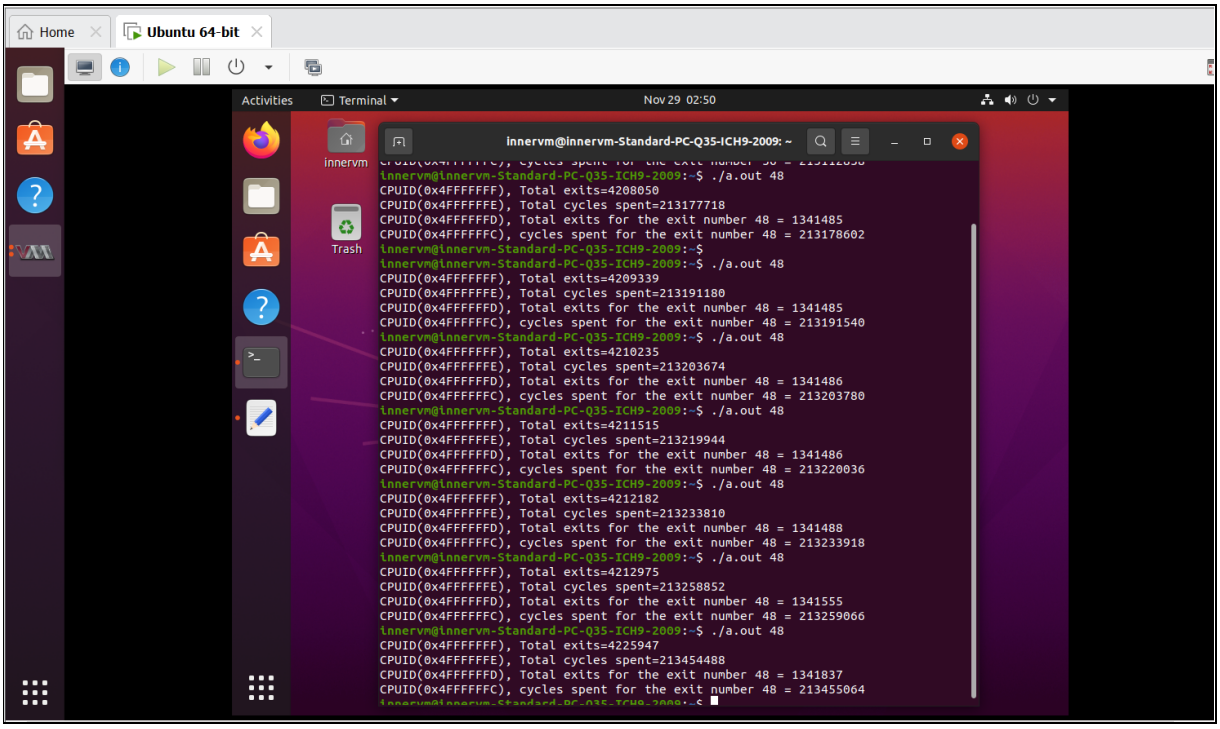
- Now you have to install kvm and other supporting packages along with virt-manager by using the following commands to install an inner vm
    - sudo apt-get update

    - sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virt-manager
- After the kvm installation , verify if there are any VMs by using the below command. You should see none.
    - virsh -c qemu:///system list
- Download the ubuntu 64 bit iso desktop file
- Now, open the virtual machine manager and install ubuntu
- Now, install cpuid in the inner vm
    - sudo apt-get update
    - sudo apt-get install cpuid
- Create test codes for all the functionalities in the inner with file name test_assignment2_3.c
- Now install gcc and compile the code using gcc test_Assignment2_3.c
- Now run the test file with ./a.out

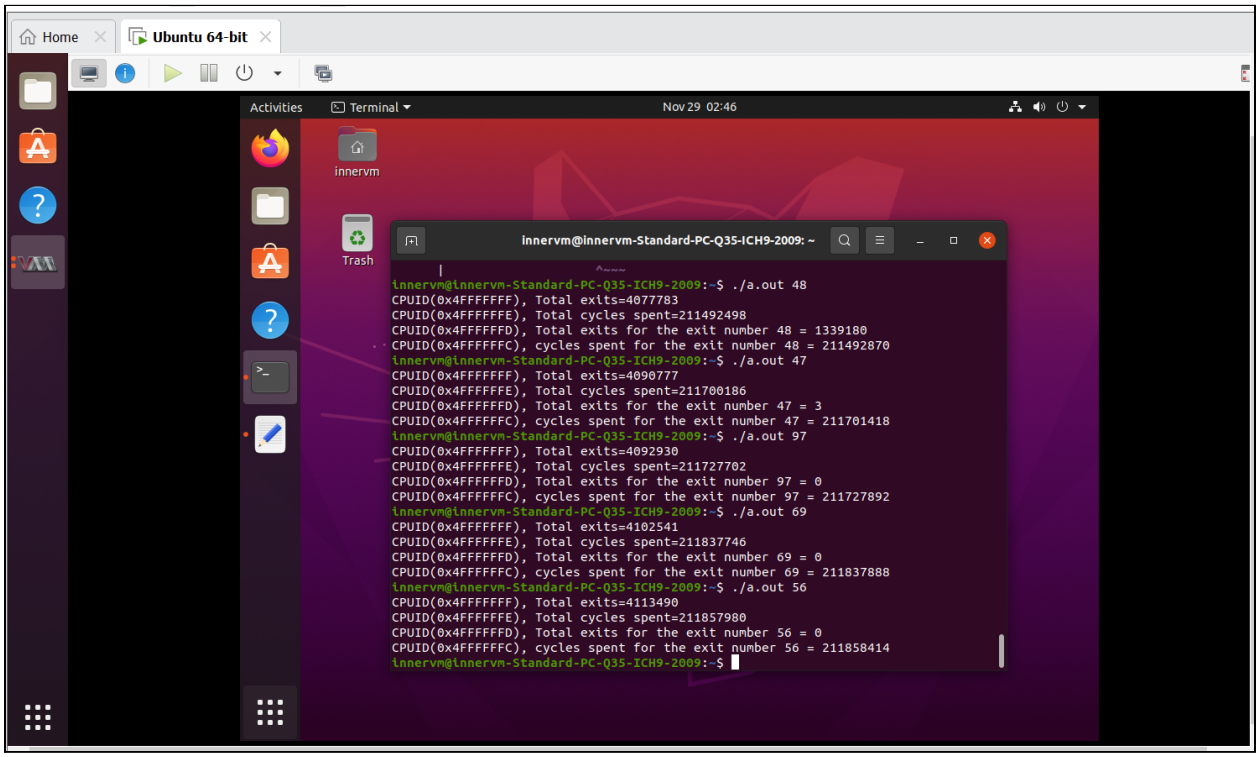
<u>Observations:</u>

Comment on the frequency of exits – does the number of exits increase at a stable rate? Or are there more exits performed during certain VM operations? Approximately how many exits does a full VM boot entail?

- As we can see in the below screenshot, number of exits are increasing at a stable rate with 1000 increments in total exits, when tested 5 times.But, after that i could see stability in the increment of number of exits.

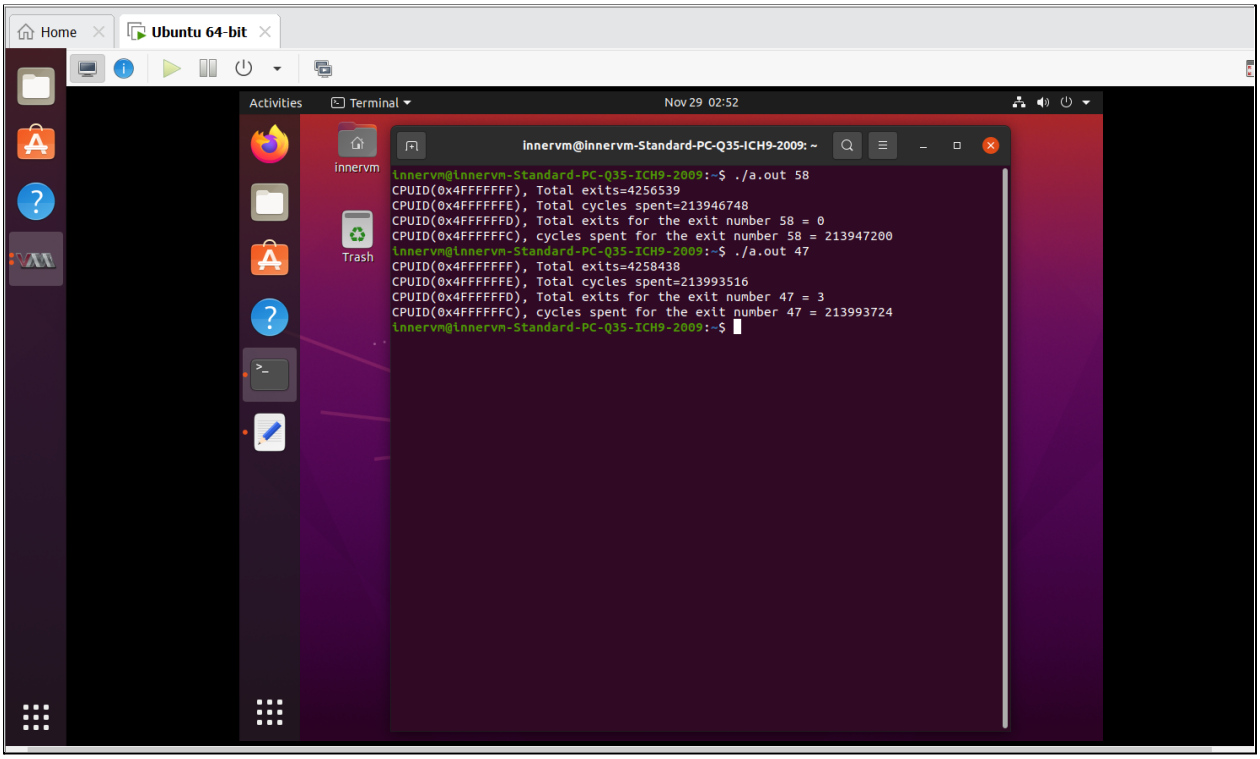


- Yes, as you can see in the below screenshot, there are more exits performed during certain vm operations

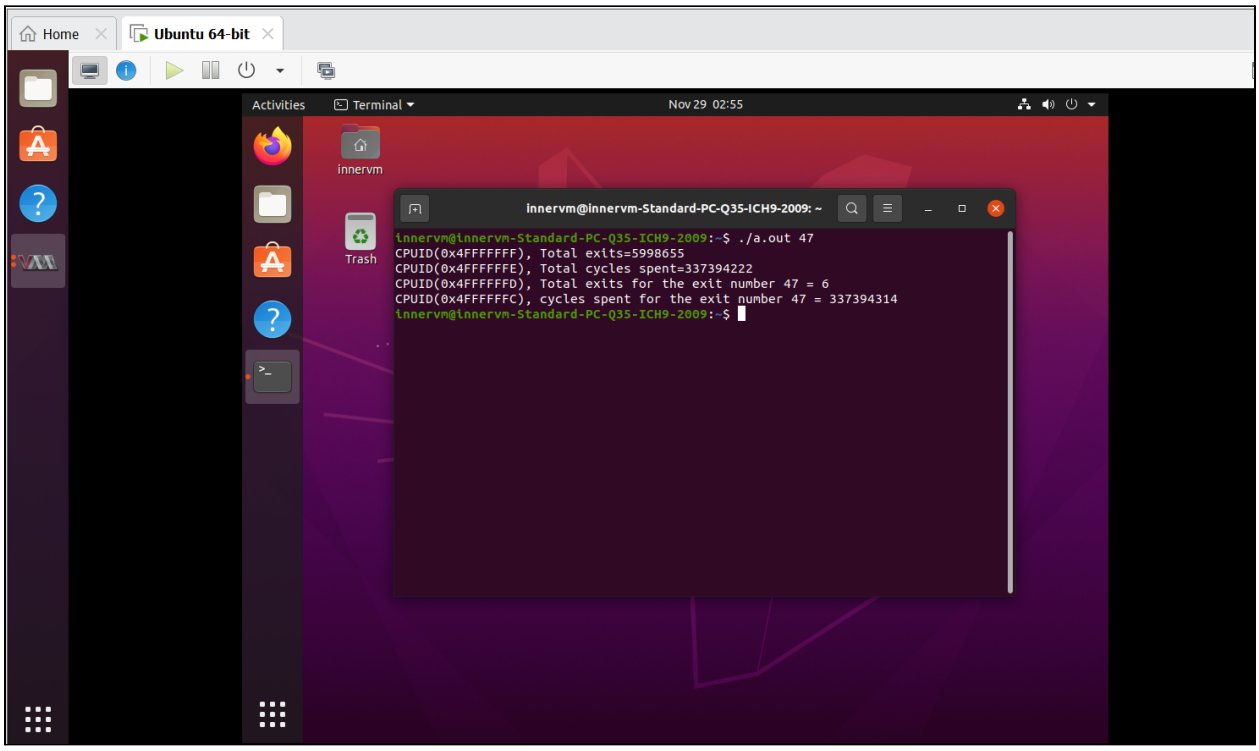


- Total number of exits are approximately around 4300000 before reboot, after reboot number of exits are nearly 6000000.

  <u>Before Reboot</u>



After Reboot



Of the exit types defined in SDM, which are the most frequent? Least?

- The most frequent exits I observed is for the exit number 48 , that is EPT violation and the least is for the exit number 54, WBINVD