CMPE283 - Assignment 2 & 3
Sangwon Song

Q1. For each member in your team, provide 1 paragraph detailing what parts of the lab that member implemented / researched. (You may skip this question if you are doing the lab by yourself).

This part is skipped. I did the assginment by myself

Q2. Describe in detail the steps you used to complete the assignment. Consider your reader to be someone skilled in software development but otherwise unfamiliar with the assignment. Good answers to this question will be recipes that someone can follow to reproduce your development steps.

Note: I may decide to follow these instructions for random assignments, so you should make sure they are accurate.
I have updated assignment 2 part as well as assignment 3 part. Please make sure you have the latest commit.

- Pre-condition: We need to use the Assignment 1 environment. I have the information about Assignment 1 in my linux github repository.
- Check the version of the linux kernel by `uname -r` command. It should be the same as Assignment 1.
  - In my case, it was `5.4.0-rc+1`.
- Modify the code in `cpuid.c` and `vmx.c` in the linux repository.
  - Detailed modification is described in the git diff file in my github.
- Remake the module by following commands
  - `make modules && make moduels_install && make install`
- Reboot by `reboot` command
- Run virt-manager
- In the guest VM, install cpuid package by the command:
  - `sudo apt install cpuid`
- Run cpuid command with leaf in the guest VM
  - `cpuid -l 0x4fffffff`
  - `cpuid -l 0x4ffffffe`
- Check the outputs in the guest VM
  - `cpuid -l 0x4fffffff` output

```
song-virt@songvirt-Standard-PC-i440FX-PIIX-1996:~$ cpuid -l 0x4fffffff
CPU 0:
   0x4fffffff 0x00: eax=0x0008eb5b ebx=0x00000000 ecx=0x00000000 edx=0x00000000
CPU 1:
   0x4fffffff 0x00: eax=0x0008eb64 ebx=0x00000000 ecx=0x00000000 edx=0x00000000
```

    - **Note:** 0x4FFFFFFF leaf contains total number of exits in %eax.
  - `cpuid -l 0x4ffffffe` output

```
song-virt@songvirt-Standard-PC-i440FX-PIIX-1996:~$ cpuid -l 0x4ffffffe
CPU 0:
   0x4ffffffe 0x00: eax=0x00000000 ebx=0x00000000 ecx=0x54478515 edx=0x00000000
CPU 1:
   0x4ffffffe 0x00: eax=0x00000000 ebx=0x00000000 ecx=0x544806db edx=0x00000000
```

    - **Note:** 0x4FFFFFFE leaf contains the high 32 bits of the total time spent processing all exits in %ebx and the low 32 bits in %ecx. In addition, ebx contains some numerical values when ecx overflows.
  - `cpuid -l 0x4ffffffd` output. It requires a sub-leaf by `-s exit-reason-number` command

```
song-virt@songvirt-Standard-PC-i440FX-PIIX-1996:~$ cpuid -l 0x4fffffd -s 0
CPU 0:
   0x4fffffd 0x00: eax=0x0000274a ebx=0x00000000 ecx=0x00000000 edx=0x00000000
CPU 1:
   0x4fffffd 0x00: eax=0x0000274a ebx=0x00000000 ecx=0x00000000 edx=0x00000000
```

- **Note:** Above screenshot shows that eax register contains the number of exits with exit reason 0.

```
song-virt@songvirt-Standard-PC-i440FX-PIIX-1996:~$ cpuid -l 0x4fffffd -s 11
CPU 0:
   0x4fffffd 0x0b: eax=0x00000000 ebx=0x00000000 ecx=0x00000000 edx=0x00000000
CPU 1:
   0x4fffffd 0x0b: eax=0x00000000 ebx=0x00000000 ecx=0x00000000 edx=0x00000000
```

- **Note:** Above screenshot shows that all 4 registers contain 0 values because exit reason 11 is not defined in KVM.

```
song-virt@songvirt-Standard-PC-i440FX-PIIX-1996:~$ cpuid -l 0x4fffffd -s 35
CPU 0:
   0x4fffffd 0x23: eax=0x00000000 ebx=0x00000000 ecx=0x00000000 edx=0xffffffff
CPU 1:
   0x4fffffd 0x23: eax=0x00000000 ebx=0x00000000 ecx=0x00000000 edx=0xffffffff
```

- **Note:** Above screenshot shows that edx contains 0xFFFFFFF and other 3 registers contain 0 values because exit reason 35 is not defined in SDM. In addition, it is not defined in KVM as well.

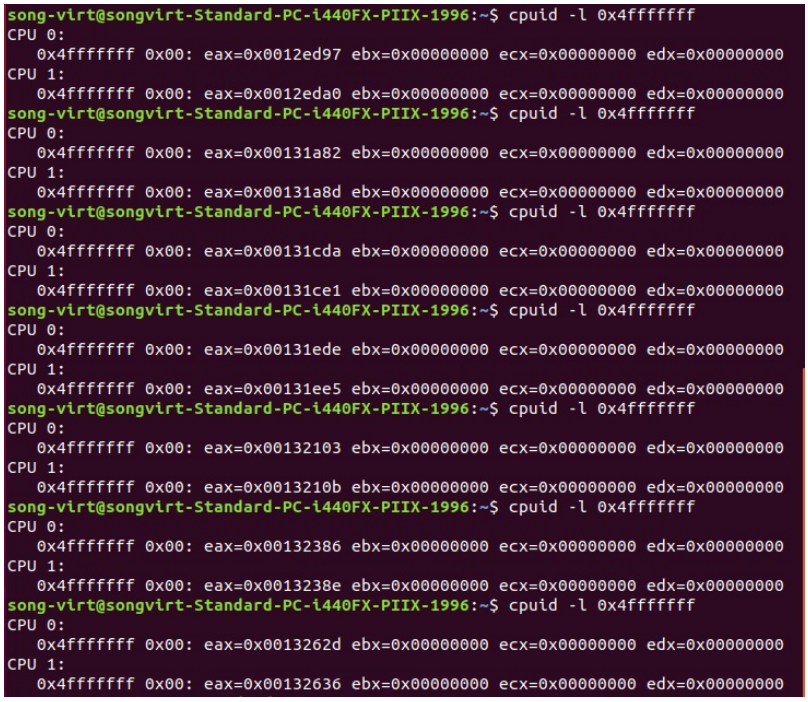
- `cpuid -l 0x4ffffffc` output. It requires a sub-leaf by `-s exit-reason-number` command

```
song-virt@songvirt-Standard-PC-i440FX-PIIX-1996:~$ cpuid -l 0x4ffffffc -s 0
CPU 0:
   0x4ffffffc 0x00: eax=0x00000000 ebx=0x00000000 ecx=0x038aa0d5 edx=0x00000000
CPU 1:
   0x4ffffffc 0x00: eax=0x00000000 ebx=0x00000000 ecx=0x038aa0d5 edx=0x00000000
song-virt@songvirt-Standard-PC-i440FX-PIIX-1996:~$ cpuid -l 0x4ffffffc -s 11
CPU 0:
   0x4ffffffc 0x0b: eax=0x00000000 ebx=0x00000000 ecx=0x00000000 edx=0x00000000
CPU 1:
   0x4ffffffc 0x0b: eax=0x00000000 ebx=0x00000000 ecx=0x00000000 edx=0x00000000
song-virt@songvirt-Standard-PC-i440FX-PIIX-1996:~$ cpuid -l 0x4ffffffc -s 35
CPU 0:
   0x4ffffffc 0x23: eax=0x00000000 ebx=0x00000000 ecx=0x00000000 edx=0xffffffff
CPU 1:
   0x4ffffffc 0x23: eax=0x00000000 ebx=0x00000000 ecx=0x00000000 edx=0xffffffff
```

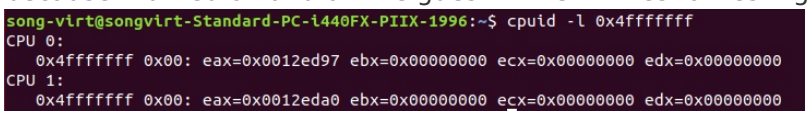
- **Note:** Above screenshot describes the same cases as `0x4fffffd`. Exit number 0 shows the correct total time spent in ecx. Ebx will have value once ecx overflows. Exit 11 is not defined in KVM, thus all 4 registers have 0 values. Exit 35 is not defined in SDM, thus edx contains `0xffffffff` and other 3 registers have 0 values.

Q3. 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?

When I look at the eax values in the guest VM with `0x4fffffff` leaf, the number of exits increases, but not in a stable rate as in figures below. Some VM operations generates more VM exits, such as turning on and off the guest VM generates more exits than other operations.



As soon as I booted the guest VM, I had approximately 1240000 exits. It was more than I expected because I turned on and off the guest VM few times for testing the module.



Q4. Of the exit types defined in the SDM, which are the most frequent? Least?
In the Intel SDM, exit numbers from 0 to 68 except for 35, 38, 42, and 65 are defined as valid exit reasons.
By comparing each exit reason with `cpuid -l 0x4ffffffd` and `eax` register value, exit reason 32, WRMSR, has the most frequent value as 0x46788. Interestingly, exit reason 12, HLT, and exit reason 30, I/O instruction also have high frequent values. In contrast, exit reason 29, MOV DR, has the least frequent value as 0x02. There are many 0-count exit reasons in this setting such as Tirple fault or RDTSC, and none-defined exits in KVM and SDM. Therefore, I focused only meaningful numerical values.