CSIE 5310 Assignment 4 (Due on December 18th 14:10)

You have learned about the process VM and emulation in class. In this assignment, you will modify QEMU's MMU emulation (i.e., SoftMMU) to bypass the permission check for a VM's memory access.

0. Late submission policy

- 1 pt deduction for late submissions within a day (before Dec. 19th 14:10)
- 2 pts deduction for late submissions within 2 days (before Dec. 20th 14:10)
- zero points for submissions delayed by more than 2 days.

1. Before you start

- As in Assignment 1, you should prepare a working Ubuntu environment, which you are allowed to install any software packages as you wish. We recommend installing Ubuntu on a VM (on VMWare workstation/fusion or parallel), or on a bare-metal machine (laptop or lab server) that you have full control. The tutorial provided as follows is based on Ubuntu 22.04 LTS. Please make sure to have at least 50GB free storage in your Ubuntu environment.
- Download the attachment `vm_hw4_files.zip` from NTU Cool into your Ubuntu host. There are 3 files used in this assignment.
    - `hijack`
    - `hijack.c`
    - `run-vm.sh`

2. Run a VM

You need to launch a VM first. Please follow the instructions below.

Compile kernel image

> You can skip this step if you still have the kernel image from Assignment 1 on your system.

First, clone the mainline 5.15 Linux kernel source code.

```
# git clone --depth 1 --branch v5.15 https://github.com/torvalds/linux.git
# cd linux
```

Next, compile your KVM host:

```
# make ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- defconfig
# make ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- -j4 (-j is to compile in
parallel)
```

Compile QEMU

> You can skip this step if you still have the QEMU setup from Assignment 1 on your system.

Clone QEMU from the repo and checkout to version `v7.0.0`:

```
# git clone https://gitlab.com/qemu-project/qemu.git
# cd qemu/
# git checkout tags/v7.0.0
```

Then configure and compile qemu from the source. You may run into some errors when you do the `configure` command, this is normally because you have missing packages. Google will be your friend for addressing the errors.

```
# cd qemu
# ./configure --target-list=aarch64-softmmu --disable-werror
# make -j4 (-j is to compile in parallel)
# sudo make install
```

Create virtual disk image

> You can skip this step if you still have the virtual disk image from Assignment 1 on your system.

You need to create a virtual disk image to store its file system.

First, download Ubuntu 20.04's file system binaries here: https://cloud-images.ubuntu.com/releases/focal/release/ubuntu-20.04-server-cloudimg-arm64-root.tar.xz

You can then follow the instructions below to create an Ubuntu 20.04 virtual disk image:

```
# qemu-img create -f raw cloud.img 25g
# mkfs.ext4 cloud.img
# mount cloud.img /mnt
# tar xvf ./ubuntu-20.04-server-cloudimg-arm64-root.tar.xz -C /mnt
# sync
# sudo touch /mnt/etc/cloud/cloud-init.disabled
```

Next, open `/mnt/etc/passwd`, and update the first line to the following to disable root login password.

```
root::0:0:root:/root:/bin/bash
```

Finally, unmount the file system image from your Ubuntu host.

```
# umount /mnt
```

Run a VM

After the compilation of QEMU, let's run a VM. Execute the following command to launch it.

```
# ./run-vm.sh -k $PATH_TO_vm_Image -i $PATH_TO_cloud.img
```

3. Bypass Permission Check

Now, you need to run `hijack` within the VM. `hijack` attempts to overwrite its own code located from virtual address `0x4005e4` to `0x40060b` (the source code is available in `hijack.c`). However, since the page containing the code is marked as read-only, any attempt to overwrite it will trigger a page fault.

You are tasked to modify QEMU's SoftMMU so that `hijack` can successfully overwrite its own code. Once it succeeds, `hijack` will print `"Succeed! You've done it!"`.

You **are only allowed to** modify QEMU.

4. Grading

You need to submit a write-up and a QEMU patch. The requirements are detailed below.

QEMU patch (8pts)

You will get full points if you let `hijack` overwrite its code successfully via modifying QEMU's SoftMMU.

Write-up (2pts)

You need to explain how your implementation works in the write-up.

5. Submission

You should submit the assignment via NTU Cool. You are required to submit the following files. Replace `[Student-ID]` with your student number. For example, if your student number is `r01234567`, then `[Student-ID]` should be `r01234567`.

- `write-up.pdf`: the write-up file.
- `[Student-ID]_qemu.patch`: the QEMU patch.

Please place those files into a folder named `[Student-ID]_hw4`. The folder structure should be the same as follows.

```
[Student-ID]_hw4
|---- write-up.pdf
└---- [Student-ID]_qemu.patch
```

Then, compress the folder into `[Student-ID]_hw4.zip` and submit it to NTU Cool.