Virtual Machine HW1

2019

Objective

- Understanding the state of the art emulator, QEMU
- Learning DBT (Dynamic Binary Translation) based instrumentation
- Getting familiar with the basics of ARMv8 ISA.

Start up

- Prepare a PC platform with x86-64 and ubuntu16.04
- Download QEMU2.9.1
  - https://www.qemu.org/download/#source
  - https://download.qemu.org
- Build QEMU
  - The guest of this homework is ARMv8 (aarch64) so you only need to install *aarch64-linux-user*.
  - ../qemu-2.9.1/configure --prefix=*where-you-want-to-install* --target-list=aarch64-linux-user --enable-debug-tcg --enable-debug

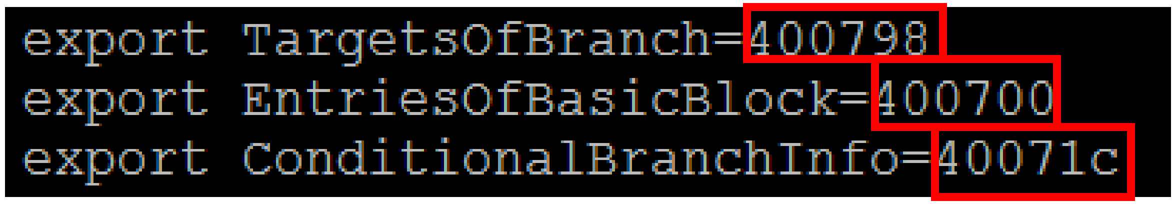
For debugging, this is optional.

Assignment 1

- In this assignment, you need to output:
  - All targets of a given branch, and the number of times the target is jumped to
  - All branches lead to a given basic block, and the respective frequency
  - The number of times a given conditional branch that is taken or not-taken
- The given branch instruction may be an indirect branch.

Example of Assignment 1

Input:



Hex number

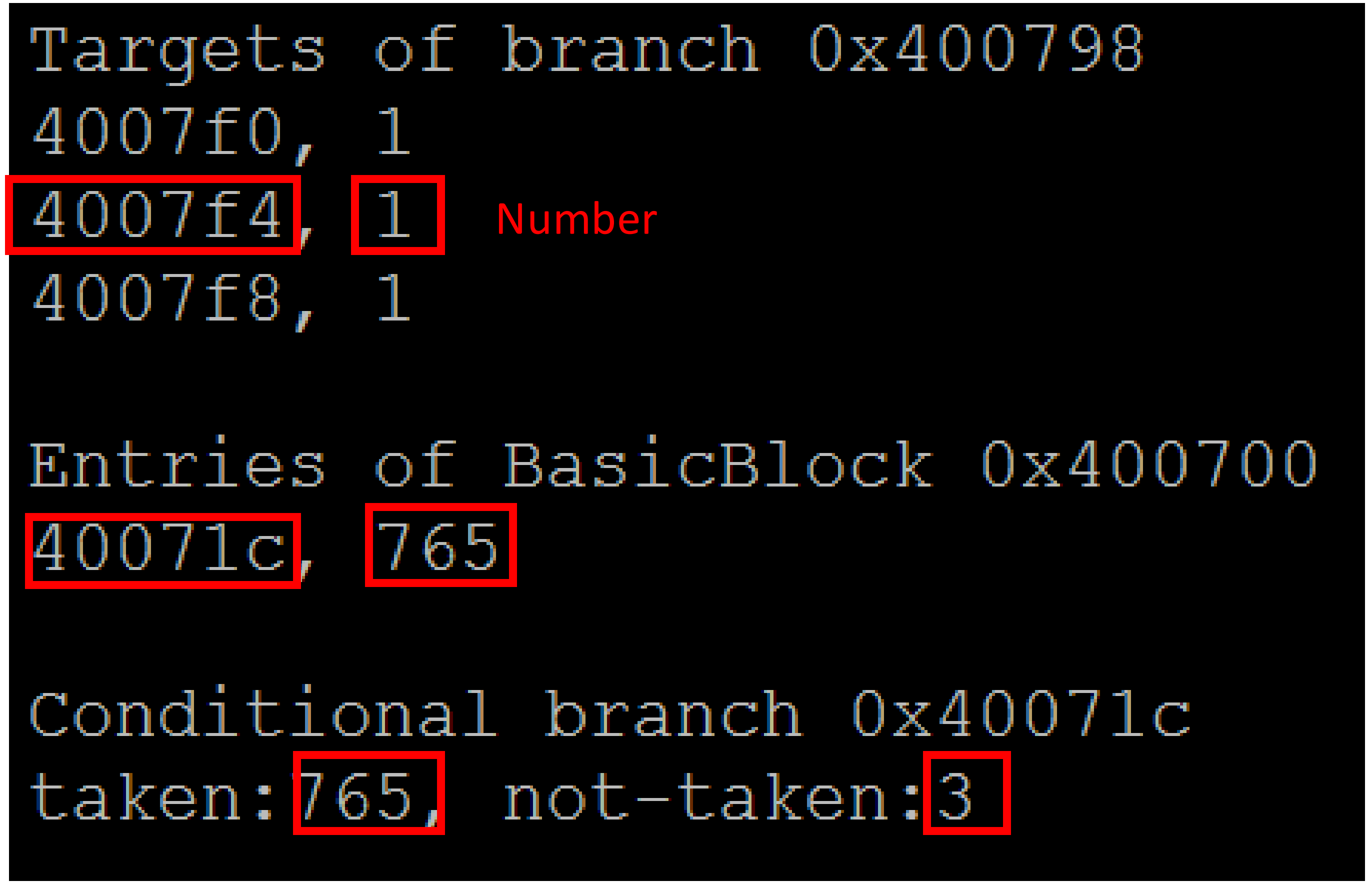
```
export TargetsOfBranch=400798
export EntriesOfBasicBlock=400700
export ConditionalBranchInfo=40071c
```

Run:

```
qemu-aarch64    vadd-vm
```

Example of Assignment 1

Output:



```
Targets of branch 0x400798
4007f0, 1
4007f4, 1
4007f8, 1

Entries of BasicBlock 0x400700
40071c, 765

Conditional branch 0x40071c
taken:765, not-taken:3
```

Target address (hex)

Number

The address of branch instruction to this basic block(hex)

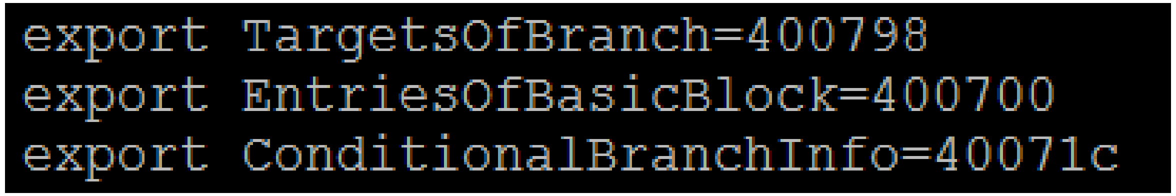
Number

Number

Input and output format

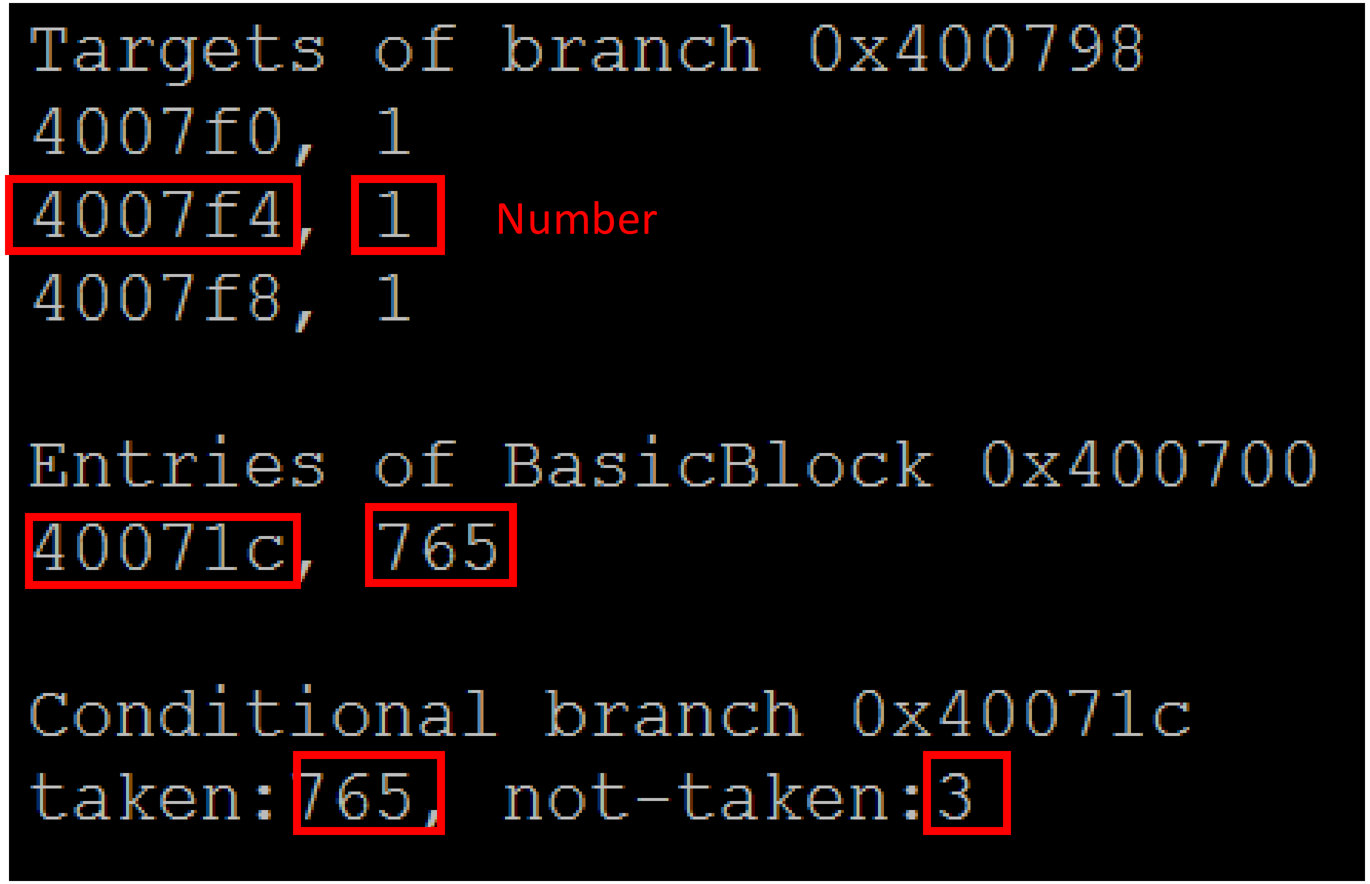
- Inputs are passed by environment valuables. You can use Linux command *export* and C API *getenv().*

EX:



```
export TargetsOfBranch=400798
export EntriesOfBasicBlock=400700
export ConditionalBranchInfo=40071c
```

Output format should be:



```
Targets of branch 0x400798
4007f0, 1
4007f4, 1
4007f8, 1

Entries of BasicBlock 0x400700
40071c, 765

Conditional branch 0x40071c
taken:765, not-taken:3
```

Target address (hex)

Number

The address of branch instruction to this basic block(hex)

Number

Number

Hints for printing the output

- You can print the output in

File: *linux-user/syscall.c*

Function: *do_syscall*()

……

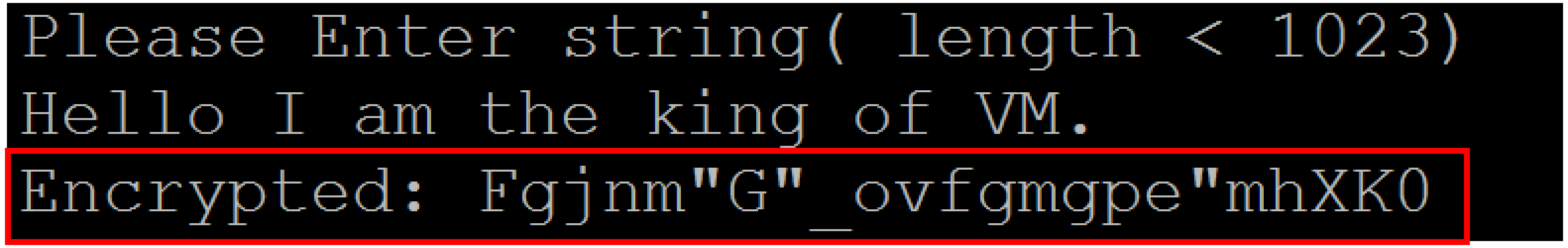
case TARGET_NR_exit_group:

print the output here

Assignment 2

- In this assignment, you will get an executable, *encr-vm*.

When executing: *Original-qemu ./encr-vm*



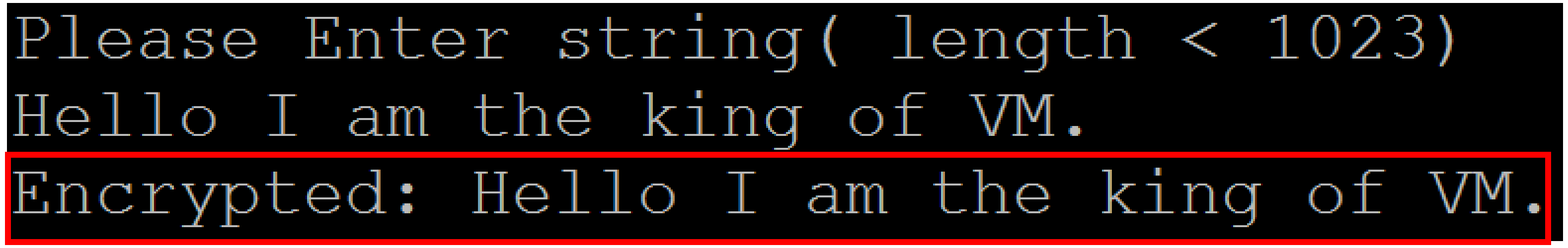
```
Please Enter string( length < 1023)
Hello I am the king of VM.
Encrypted: Fgjnm"G"_ovfgmgpe"mhXK0
```

Assignment 2

- Your mission is to recover the encrypted string by modifying QEMU.

  Note: You cannot modify the *encr-vm*.

Your-qemu ./encr-vm



```
Please Enter string( length < 1023)
Hello I am the king of VM.
Encrypted: Hello I am the king of VM.
```

Note:

- For assignment 1:

Benchmarks for grading will be different from *vadd-vm.*

- For assignment 2:

encr-vm is the grading benchmark.

Before submitting your homework, please verify the correctness. Here are some recommended benchmarks.

- MiBench:
  - http://vhosts.eecs.umich.edu/mibench//source.html
- PolyBench
  - http://web.cse.ohio-state.edu/~pouchet.2/software/polybench/download.html

Team rule and what to submit

- You may do this assignment in a team of two members, however, you are welcome to do it by yourself.

- What to Submit?
    - Your source code: wrap up all your codes in a tar file, and upload to CEIBA. File name format: vm_hw_ID1_ID2，ex: vm_hw_d12345678_ d12345679
    - A report (PDF file): describe your design and implementation.