CS382 Computer Organization and Architecture Fall 2021

Lab 5 · Emulator Setup and First Assembly Program

Lecturer: Philippos Mordohai, Shudong Hao *Date:*

1 Objectives

- Configure QEMU
- Program translation

2 QEMU

We will install an emulator for ARM8 on the Virtual Machine. **Note that this is an ARM not a LEG emulator** and some differences exist.

Install cross compiler to assemble and compile programs. Specifically, install aarch64 cross compiler for the 64-bit ARM architecture. See https://packages.debian.org/stretch/gcc-aarch64-linux-gnu

```
sudo apt-get install gcc-aarch64-linux-gnu
```

Install QEMU to run cross-compiled programs.

```
sudo apt-get install qemu-user
```

Install multiarch-GDB for debugging:

```
sudo apt-get install gdb-multiarch
```

3 Program Translation

Apply the following steps

```
aarch64-linux-gnu-gcc-9 -E test.c > test.i
cat test.i
```

and **submit a screenshot showing the terminal with your username in the prompt and the print out from** `cat`.

```
aarch64-linux-gnu-gcc-9 test.c -S
cat test.s
```

and **submit a screenshot showing the terminal with your username in the prompt and the print out from** `cat`.

To assemble a program on the terminal:

```
aarch64-linux-gnu-as demo.s -o demo.o
```

Then link the object files:

```
aarch64-linux-gnu-ld demo.o
```

If the assembly code uses the standard C library, use the following command to link it:

```
aarch64-linux-gnu-ld demo.s -o demo.o -lc
```

The output binary name by default is a.out, but can be renamed. To run it:

```
qemu-aarch64 a.out
```

If again the assembly code uses C library, it needs to dynamically link it as well:

```
qemu-aarch64 -L /usr/aarch64-linux-gnu/ a.out
```

Assemble and link demo.s (provided on Canvas) and **submit a screenshot showing the terminal with your username in the prompt, the above steps and the contents of the directory afterwards.** You can use `ls` to list the contents of the directory.

What to Submit

A pdf with the three screenshots specified above.