University of Connecticut
CSE 4302: Computer Architecture
Fall 2022

Programming Assignment 0: Getting Started with *riscy-uconn*

Due September 8, 2022 (Thursday) @ 11:59 PM on HuskyCT

Summary

The programming assignments (PAs) in this course will make use of *riscy-uconn*, a MIPS-like CPU simulator developed by UConn's *Computer Architecture Group (CAG)*. Each PA will provide incomplete simulator code and a detailed description of the functionality that must be implemented, as well as the expected deliverables to be submitted through HuskyCT. **For each PA (except this one), students must also schedule a one-on-one code review with the TA to receive credit.**

The objective of this introductory PA is to guide the student through setting up the development environment that will be used for the remainder of the course. The student will also gain familiarity with obtaining PA materials, using *riscy-uconn*, and submitting course deliverables through HuskyCT.

1 Environment Setup

The PA materials for this course will be provided through a git repository hosted on UConn's GitHub server. The PA themselves will be written in C and require an appropriate toolchain (compiler, build system, etc.). Follow the instructions below to setup the required development environment for your platform.

NOTE: For terminal commands, the "$" character denotes the shell prompt and should not be typed literally.

Windows 10/11

Windows 10 users will leverage the Windows Subsystem for Linux (WSL). WSL is essentially a lightweight Linux Virtual Machine (VM) that runs as a native Windows application.

The following instructions describe how to install WSL, Ubuntu Linux, and the required developer tools:

1. Ensure that your machine is completely up to date.
2. Follow the **Manual Installation Steps** on the following webpage to install WSL:

   `https://docs.microsoft.com/en-us/windows/wsl/install`

   and install **Ubuntu 20.04.1** LTS from the Microsoft Store.

   NOTE: It is recommended to install Windows Terminal as well. Instructions to do so are on the previously linked webpage.
3. After installation, launch the Ubuntu instance by using the new Ubuntu shortcut in the start menu.

   If you opted to install Windows Terminal, an Ubuntu instance can be launched from within the application by using the new tab drop-down menu in the title bar.
4. Enter the following commands in the Ubuntu shell to install GNU Make, Git, the GNU Compiler Collection (GCC), and the GNU Project Debugger (GDB):

```
$ sudo apt update
```

```
$ sudo apt install make git gcc gdb
```

Mac OS

Mac OS users will install the Xcode command line tools, which includes GNU Make, Git, the Clang C/C++ compiler, and the Clang debugger. Instructions to do so are as follows:

1. Launch Terminal from Finder → Applications → Utilites → Terminal.
2. Install the Xcode command line tools with the following command:

   ```
   $ xcode-select --install
   ```

   Follow the Finder dialog prompts to finish installing the tools.

2 (Optional) Install Visual Studio Code

Visual Studio Code (VS Code) is a cross-platform Integrated Development Environment (IDE) with powerful built-in developer tooling. VS Code provides first-party support for several popular programming languages (C, C++, Python, etc.) and developer tools (git, gcc, gdb, etc.).

Those interested in trying out VS Code can install it from the following webpage:

`https://code.visualstudio.com/download`

Instructions for using VS Code with C/C++ can be found on the following webpage:

`https://code.visualstudio.com/docs/languages/cpp`

Windows users can learn how to use VS Code with WSL on the following webpage:

`https://code.visualstudio.com/docs/remote/wsl-tutorial`

3 Getting *riscy-uconn*

All PA materials will be available at the following GitHub repository:

`https://github.uconn.edu/omk12001/cse4302`

You can clone this repository by executing the following command in your platform's terminal:

```
$ git clone https://github.uconn.edu/omk12001/cse4302.git
```

The above command will create a new directory named `cse4302` in the current working directory (the directory the command was executed in).

You can always pull the latest repository contents by executing the following command within the `cse4302` directory:

```
$ git pull
```

You will never push anything to this repository.

NOTE: Git is a popular Source Control Management (SCM) tool that is used extensively in software development. Those interested in learning more about Git can do so on the following website:

`https://git-scm.com/book/en/v2`

4 Getting Started with *riscy-uconn*

Once the repository is cloned, you will find two subdirectories underneath `cse4302`: `assembler` and `PA0`.

The `assembler` directory contains the assembler for the MIPS-like ISA that will be used for this course. Follow the instructions in the `README.md` file to build the assembler in this directory. The assembler will not be modified for the remainder of this course.

The `PA0` directory contains the skeleton simulator that you will extend for the next PA. For now, just build the simulator using the instructions in the `README.md` file.

After building the simulator using the `make` command, invoke the assembler on the `nop.asm` file using the following command from within the `PA0` directory:

```
$ ../assembler/assembler nop.asm nop.out
```

A `nop.out` file should now be created inside the `PA0` directory. Verify that the contents in this file match the following:

```
00000000000000000000000000000000
11111111111111111111111111111111
00000000000000000000000000001010
00000000000000000000000000100000
```

You are now ready to execute the simulator using the following command:

```
$ ./simulator nop.out
```

The simulator output will be printed to the terminal. Verify that the output includes the following line:

```
TOTAL INSTRUCTIONS EXECUTED: 1
```

If both verifications pass, you are now ready to proceed with future PAs.

To get credit for this assignment, submit screenshots of the two verifications in your terminal window through HuskyCT.