CS3160 (Aug 2024) Lab Setup Guide

Sandeep Chandran

August 23, 2024

This document prescribes the steps to follow when using the machines kept in the lab A01-201.

1 RISC-V Toolchain Installation

NOTE: Skip this step if you already have setup and tested `riscv-none-elf-gcc`.

1. Login to the lab machine with your LDAP username and password. (Rest of the steps assume that the environment variable `$HOME` points to your home directory `/home/<ROLLNO>`)

2. Open a browser and point it to `http://10.129.5.7/data`

3. Download `tools.zip` file
   (You should see this file in your `$HOME/Downloads` folder)

4. Right-click and extract the zip file
   (You will see a new folder named tools appear under `$HOME/Downloads`)

5. Rename this folder to `riscv-tools`
   (You will now see a folder named `riscv-tools` under `$HOME/Downloads`)

6. Move this folder to your home directory.
   (You should now see `riscv-tools` folder under `$HOME`)

7. Add the binaries (executables) under this folder to the `$PATH` variable. Open a terminal and type the following command.

   ```
   $ export PATH=$HOME/riscv-tools/bin:$PATH
   ```

8. In the same terminal, type the following. (You should see a message that says GCC 13.1.0, and NOT get the "Command Not Found" error).

   ```
   $ riscv-none-elf-gcc --version
   ```

2 Spike Installation

Spike is a RISC-V Instruction Set Simulator. We will use this to execute (or "run") the RISC-V assembly programs that we will write in this course.

1. Open a browser and point it to `http://10.129.5.7/data`

2. Download the `spike.zip` file
   (You should see this file in `$HOME/Downloads`)

3. Right-click and extract the zip file
   (You should see a new folder named `spike` in `$HOME/Downloads`)

4. Move this spike folder to the home directory.
   (You should now see the spike folder under the `$HOME` directory)

5. Add the binaries (executables) under the spike to the `$PATH` variable. Open a terminal and type the following command.

   ```
   $ export PATH=$HOME/spike/bin:$PATH
   ```

6. In the same terminal, type the following. (You should see a message that says "1.1.1-dev 628ba126", and NOT get the "Command Not Found" error).

   ```
   $ spike --version
   ```

3 Bitbucket and Git

3.1 Setup SSH Keys

1. Open a terminal and type the following command.

   ```
   $ ssh-keygen
   ```

   Keep hitting enter as and when the prompt appears.

2. Login to your bitbucket account (`https://bitbucket.org/`).

3. Click the gear icon (top-right corner), and go to "Personal Bitbucket Settings".

4. On the left panel, click on "SSH Keys".

5. Click on "Add key" button.

6. Go back to the terminal where you typed the earlier command, and type the following command.

   ```
   $ cat ~/.ssh/id_rsa.pub
   ```

7. Copy the output of the above command, paste it into the "Key" field (in the Browser window/Bitbucket page) and then press the "Add key" button.

3.2 Clone a Repo

1. On the bitbucket page, click on the menu item "Repositories" (at the top of the page).

2. Select the repo with your <*ROLLNO*> (under the project `CA_2024`). The url in your address bar will be similar to
   https://bitbucket.org/sandeepchandran28/<ROLLNO>/src/main/

3. Click on the "Clone" button (top right)

4. Copy the link (SSH is selected in the drop-down box). The link will be similar to
   "*git clone git@bitbucket.org:sandeepchandran28/<ROLLNO>.git*"

5. Paste this copied text/link in a terminal, write `cs3160` after it, and then hit Enter.

   ```
   $ git clone git@bitbucket.org:sandeepchandran28/<ROLLNO>.git cs3160
   ```

(a) This will clone the contents of the repo into a folder called `cs3160`.

(b) If you forgot to mention `cs3160` in the git clone, the contents will be cloned into a folder named <*ROLLNO*>. You can rename this folder to `cs3160`.

3.3 Committing and submitting changes

1. Open a terminal and go to (`cd` into) the `cs3160` folder (git repo that was cloned)

2. The lab assignments will require you to work inside this folder. Make the additions/modifications necessary to solve the assignment.

3. Commit your changes at regular intervals by following the commands given below.

   (a) Check which files have been modified/changed.

   ```
   $ git status
   ```

   (b) Add the changes to commit. The command below will add all the changes. If you do not want to add all the changes, replace `.` with the list of files that have changed.

   ```
   $ git add .
   ```

   (c) Commit the changes.

   ```
   $ git commit -m "<YOUR MSG HERE>"
   ```

   (d) Pull any changes that may have happened in the remote repo in the meantime

   ```
   $ git pull --rebase
   ```

   (e) If the above command is successful, push the changes to the remote repo.

   ```
   $ git push
   ```