RDG Installation and User Guide

Author: Arseniy Kouzmenkov

Date: 30/09/2020

Section 1: Installation Guide on UAlberta Lab Machines

1.1: Installation Script and Assumptions

In order to set up RDG to run on the lab machines, I have prepared a very simple script that will install everything successfully on your machine. There is one assumption that I made when making the install script: you are using `bash` as your main terminal shell, since it is unlikely that you changed your basic shell. In case you are using `zsh` or other shell environments, I assume that you will be able to figure out how to install RDG successfully by inspecting the below command and modifying the install script after downloading it - Appendix A addresses this in more detail.

1.2: Installation Guide

The steps to install RDG on the lab machine are as follows:

1. Log into `ohaton` with your preferred remote access tool (`ssh`, `xming`, `PuTTy`, etc.)

2. Copy and paste the following command into the terminal, after making sure that you are connected to `ohaton`:

```
touch $HOME/.bashrc && curl -so rdg_setup.sh
https://cmput229.github.io/229-labs-RISCV/RISC-V-Examples_Publ
ic/set_up_rdg_on_lab_machine.sh && chmod +x rdg_setup.sh &&
./rdg_setup.sh
```

3. Once you get your prompt back, you are done, feel free to move on to the next section to learn how to use RDG. If you wish to verify that your installation ran correctly, type in the word `rdg` in your terminal and verify that the output is not an error message from `bash`.

1.3: Explaining the Installation Command

Since it is a VERY bad practice to run random commands that a stranger on the internet told you to run, let me quickly explain what this command does:

1. The necessary file `~/.bashrc` is verified to exist in your file system. If it does not exist, then it creates it.
2. The install script is downloaded from Dr. Amaral's CMPUT 229 website directly. Feel free to inspect the URL, and even visit the site to verify that the source is legitimate.
3. The setup script is made executable by running `chmod +x <script name>`.
4. The setup script is executed.

I hope that by inspecting the command and the source of the file, you will be satisfied that your lab machine is staying secure.

The install script itself is written in the same way that I ran the necessary commands to get RDG set up on my lab machine with a couple modifications to ensure that it works in your environment.

A quick thank you is in order to Onn Qazilbash for aiding me with testing the script to make sure it works in your lab machine environment.

Section 2: User Guide

As part of the installation script, I add a function to your `.bashrc` file that simplifies the use of RDG to be as convenient as possible for you. Here are a couple basic workflows that you can follow to use RDG.

2.1: Simplest Workflow with `rdgpdf`

Arguably the simplest way to generate a PDF file with a call graph for your RISC-V assembly program is to simply run the following command:

```
rdgpdf <path to your .s file> <name of the output file to generate>
```

Please note one thing when running this command, you do not need to supply the .pdf extension to the name of the output file, as that gets appended during runtime. This will create a PDF file that you can view later with your call graph.

Additionally, there will be another file generated in the same directory, with the same name but without the .pdf extension. This file contains the generated Dot graph in the syntax specified by GraphViz. You can inspect it to learn how to describe a graph using Dot yourself :)

2.2 Step by Step Workflow with `rdg`

Let's say that you would like to do each step yourself instead of having my function generate your PDF for you. You can do that by using RDG directly as follows:

```
rdg <path to your .s file> <name of the output file to generate>
```

This will create the Dot graph file in the syntax specified by GraphViz. In order to generate the PDF of your graph, you will need to run the follow command:

```
dot -Tpdf <name of output file from previous step> -o <name of pdf>
```

In this case, make sure that you do append the .pdf suffix to the file specified after `-o` argument, as otherwise pdf viewers may not recognize the file.

2.3 Viewing Your PDF

In this section, I would like to provide you with a couple ways of how to view your generated call graph pdf after generating it on the lab machines.

2.3.1 Copy it over to your host machine

This may be the simplest way to view the PDF. Just copy it over with `scp` or other tools that your host machine supports. I will not be providing instructions on how to do that in this section, Google is your best friend.

2.3.2 View it on the lab machine by enabling X forwarding

You might want to view the generated PDF on the lab machine so as not to clutter your own machine. If you want to do so, then you will want to run the program that you used to connect to the lab machine with X forwarding enabled. If you are using `ssh`, then that is done by passing the `-X` flag to `ssh`. Since I do not use Windows personally, I do not entirely know how to use `xming` or `PuTTy` to do this properly, but I am sure you will be able to find loads of information online.

Once you have connected to the lab machine with X forwarding enabled, you can use programs such as `xpdf` and `firefox` to view the PDF.

2.3.3 Upload the PDF to a personal private GitHub repository

The last option is to upload the PDF to a private GitHub repository, as that will allow you to view the PDF online within the repository itself. I personally use this as it is slightly faster than downloading the file every time.

Section 3: RDG Source Code

If you wish to view the source code for the RDG program, you can do so by going to the `~/tools/build/RDG/` folder and inspecting the code. The only files that you really need to look at are `main.cpp` and `riscdot.g4`. If you find any serious bugs in the program, then please let me know by emailing me on my UAlberta account. The repository for RDG is located here, in case you wish to fork the repository and make your own additions :)

Appendix A - Non-standard Lab Machine Installation Procedure

In the case that you, like me, prefer to use a different shell environment, then you will have to potentially modify the installation script slightly to make it work for your situation. The only modification I can think of is if you are using `zsh`, as nothing else should break my install script. Here is the set of steps to make it work.

1. First, download the script by running the following command:
   ```
   curl -so rdg_setup.sh
   https://cmput229.github.io/229-labs-RISCV/RISC-V-Examples_Publ
   ic/set_up_rdg_on_lab_machine.sh
   ```
2. Open the file `rdg_setup.sh` with your text editor of choice.
3. Replace all occurrences of the word `.bashrc` with `.zshrc` or the equivalent for your shell environment.
4. Run the following command to make the script executable in case it is not:
   ```
   chmod +x rdg_setup.sh
   ```
5. Run the script:
   ```
   ./rdg_setup.sh
   ```
6. RDG is now installed on your instance of the lab machine!