Project 2: CS61Classify

Part A Deadline: Tuesday, February 20, 11:59:59 PM PT

Part B Deadline: Tuesday, March 5, 11:59:59 PM PT

In this project, you will write RISC-V assembly code to classify handwritten digits with a simple machine learning algorithm.

The goal of this project is to familiarize you with RISC-V, specifically calling convention, calling functions, using the heap, interacting with files, and writing some tests.

Office Hour Policy

Before coming to office hours, please make sure you have done the following:

1. Add descriptions for each register in your functions to indicate their purpose ("this holds i" is too vague, please be as specific as possible so it is easier for us to read and comprehend your code.)
2. Read the calling convention guide.
3. Use calling convention checker and resolve all calling convention errors caught by the checker (unless your question is specifically about how to fix these errors).
4. Use memcheck and resolve all memory access errors (unless your question is specifically about how to fix these errors).
5. Watch the debugging videos (linked below).

Make sure to fill out the template on the OH Queue with as much description as possible. Questions that do not use the template or have very low-effort descriptions (such as "it doesn't work") will be skipped/deleted.

Debugging Videos

Check out this playlist for videos demonstrating how to debug memory and calling convention issues! Please make sure you've watched these videos before coming to office hours or posting a question on Ed.

Setup

> You must complete this lab on your local machine. See Lab 0 if you need to set up your local machine again.

Project 2-specific errors are listed at the bottom of the common errors page.

Setup: Git

This assignment can be done alone or with a partner.

Warning: Once you create a group on Gradar, you will *not* be able to change (add, remove, or swap) partners for this project (both Project 2A and 2B), so please be sure of your partner before starting the project. You must add your partner on both Gradar and to every Gradescope submission.

If there are extenuating circumstances that require a partner switch (e.g. your partner drops the class, your partner is unresponsive), please reach out to us privately.

1. **Visit** Gradar. **Log in** and **register** your Project 2 group (and **add** your partner, if you have one), then **create** a GitHub repo for you or your group. If you have a partner, one partner should create a group and invite the other partner to that repo. The other partner should accept the invite without creating their own group.
2. **Clone** the repository on your **local machine** (you don't need the hive machine at all for this project). Windows users should clone in Git Bash.

```
$ git clone git@github.com:61c-student/sp24-proj2-USERNAME.git 61c-proj2
```

(**replace** `sp24-proj2-USERNAME` with the name of your GitHub repo)

3. **Navigate** to your repository:

```
$ cd 61c-proj2
```

4. **Add** the starter repository as a remote:

```
$ git remote add starter https://github.com/61c-teach/sp24-proj2-starter.git
```

5. **Pull** from the starter repo:

```
$ git pull starter main
```

If you run into `git` issues, please check out the common errors page.

Setup: Java and Python

Make sure that you followed the local computer setup directions in Exercise 2 of Lab 0.

Setup: Venus

We will use the Venus web interface for debugging. The procedure is very similar to the Venus setup in Lab 3.

1. In the `61c-proj2` directory, run `bash test.sh download_tools` to download Venus for this project. (You only need to run this once.)
2. **On your local computer** (not the hive), **navigate** to your `61c-proj2` directory and **run** this command. Windows users should run outside WSL (Git Bash is recommended).

   ```
   $ java -jar tools/venus.jar . -dm
   ```
3. **In your web browser, open** https://venus.cs61c.org ↗ (Chrome or Firefox is recommended). **In the Venus web terminal, run** `mount local vmfs`. In your browser, you may see a prompt saying `Key has been shown in the Venus mount server! Please copy and paste it into here`. You should be able to see a key (string of letters and numbers) in the most recent line of your local terminal output; **copy and paste** the key into the dialog.
4. Now the project files from your local computer are loaded into the Venus web interface. **In Venus**, you can **open** the Files tab or **run** `ls vmfs` in the Venus terminal to see all your files in your browser.

If you can see your files in Venus, you can skip the rest of this section. If the above steps didn't work, you can follow the guide below to manually upload files.

▶ Manually Uploading Files

Restoring Starter Files

To restore starter files, please check out the common errors page.