

Group 1: Phase 6

Adam Mertzenich

November 17, 2021

Contents

1	Repository Information	1
2	Progress Report	1
2.1	Work Done	2
2.2	Problems Encountered	2
2.3	Changes to Plan	3
3	Work Log	3
3.1	Michael Musa	3
3.2	Adam Mertzenich	4
3.3	Kritib Bhattarai	4

1 Repository Information

Below you will find our repository URL, hosted on GitHub, as well as the revision number and date of our final commit for this phase of the project.

URL	https://github.com/mertad01/vm252-project
Revision	0102039d77abd33b916a29a15c0c425f167c46d5
Date	2021-11-17T15:00:20-06:00

2 Progress Report

We have made some large changes moving towards adding functionality. Much of this work was done behind the scenes, but we have also cleaned up our visual design quite as well.

2.1 Work Done

We have changed a few elements of the design. First, we moved the help button from the left panel into a dedicated menu on the top bar. It contains a couple different components which give either an overview of what can be done or show you how to access help via the tool tips. This seemed to be a more elegant way of handling help than just having a single button on the side which took up a lot of space and was not pleasant looking.

Loading object files into memory has also been added to the file menu. We made it so that you can use the file -> open dialog to select any object file from your system. The file name will then be displayed to the right, as indicated by the “File Name” label. This is a better solution than having to manually enter in the location of a file. Opening a file currently simply overwrites the memory of the VirtualMachine252 object, though we may consider having it create a whole new object in the future if that helps our design. Also note that while you can load a file the debugger currently, for testing purposes, automatically loads the greatest non-identical divisor program into memory preventing the need to load a file every time you want to test functionality.

Clicking the “N” next instruction button on the left panel will now increase the program counter, accumulator, and function properly. On top of that we have consolidated “run” and “pause” under the “Run” menu on the bar. Using run it will by default play through the entire program, or you can enter a value in the delay field. We have also enabled pausing so you can manually inspect the PC and ACC in the middle of running, and then resume by clicking run again.

2.2 Problems Encountered

Input and output gave us the largest issue, which resulted in a small change to our plans (see “Changes to Plan” below). Taking input and supplying output in one box proved to be more difficult than originally thought. To deal with this problem we pivoted to using pop-up boxes for input and output. This isn’t ideal but due to a variety of constraints this was necessary at this stage of the project.

Having the program counter and accumulator update when instructions are executed was not easy at first, but by using the observer-observable design pattern it was solved. We implemented the Observer and Observable classes from class so that any time the machine mutator methods are ran, the values will update. A problem we can foresee is that since every observer

gets notified of changes, and in turn runs their update() method body, it may create performance issues if we do complex operations.

Finally, some changes to specifics of the model need to be changed to allow more freedom in the debugger and prevent issues. Since we opted to base our model on the code provided in the example solution to phase 2 it introduces some issues. The lastInstructionCausedHalt method is used for running the entire program but also stops us from executing next instructions manually. To solve this we will need to make changes to the method that executes next instructions and change how we can reinitialize the program state fully.

2.3 Changes to Plan

Our plan has been executed for the most part without too much change. We did make some changes to how input and output is taken. We originally planned on having a box at the bottom that would take input and also serve the purpose of supplying output but due to time and issues with functionality we have abandoned that plan for the time being. We might go back to it at some point but for now using the a pop-up box to take input and provide output is satisfactory. Another change we have made is to consolidate more menu items. We did not see the need to have everything seperate, but instead have opted to move certain items together to create a more cohesive visual design.

3 Work Log

3.1 Michael Musa

Date	Hours Worked
11/12/21	03:00pm - 05:00pm
11/13/21	06:00pm - 09:00pm
11/14/21	01:00pm - 02:00pm
11/15/21	11:00am - 12:00pm
Total:	~7

3.2 Adam Mertzenich

Date	Hours Worked
11/11/21	11:30pm-12:30pm
11/11/21	03:00pm-05:00pm
11/11/21	06:00pm-08:00pm
11/17/21	04:30am-05:30am
11/17/21	11:00am-01:30pm
Total:	~8.5

3.3 Kritib Bhattarai

Date	Hours Worked
11/14/21	04:00pm-06:00pm
11/15/21	02:00pm-04:00pm
11/16/21	02:00pm-07:00pm
11/16/21	10:00pm-02:00am
11/17/21	09:00am-11:00am
11/17/21	01:15pm-02:45pm
Total:	~17