

CS 471: Software Engineering

Spring 2018

Homework 2 – Implementation of a simple maintenance task in an unfamiliar Open Source Software

Due date: Wednesday, February 21, 2017 (before the beginning of the class)

1. Brief Description

This assignment requires you to practice **compiling** and **modifying** an unfamiliar software system. As a student, most class projects are small enough that you can understand the entire program. In industry, it is generally impossible for one person to understand the entire product. Being able to navigate a large and unfamiliar codebase is a fundamental and necessary skill.

A software engineer is expected to compile, install, test, deploy, debug and troubleshoot complex software written in heterogeneous programming languages and environments. Writing code is only a small portion of the job. In other words, a software engineer is expected to “make things work” and solve any software related issue.

2. Software System

You will be compiling and modifying the muCommander open-source software system. The purpose is to give you an exposure to a large, and unfamiliar codebase. Version 0.9.2 of muCommander has ~1,100 Java files containing ~83KLOC (without comments) and ~67KLOC of comments.

The official website for muCommander (<http://www.mucommander.com/>) is geared towards end users. As a developer, you will want to refer to their GitHub page (<https://github.com/mucommander/mucommander>) for instructions on how to build the project.

3. Compiling and Running the Software

You have the full freedom in this assignment to use whatever tools you deem appropriate for compiling and making the change. However, it is recommended to follow these steps:

- You can access the source code of muCommander by cloning its git repository (<https://github.com/mucommander/mucommander>).
- Use the following command to run muCommander, as described in the README.md file (<https://github.com/mucommander/mucommander>)
 - o `./gradlew run`
 - o gradlew is a wrapper to Gradle (<https://gradle.org/>) build environment that allows the **compilation** and **run** of the muCommander from the **command line**
- you can import muCommander in any IDE (e.g., Eclipse, IntelliJ IDEA, etc.) in order to **navigate**, **search** and **implement** the change request in muCommander
 - o HINT: To import muCommander in Eclipse
 - click “File->Import...->Gradle->Existing Gradle Project”
 - In the “Import Gradle Project” window, under “Point root directory” select the folder of the cloned muCommander repository
 - In the “Import Options” window, under “Gradle distribution” select “Gradle wrapper”
 - Importing the project could take several seconds/minutes
- OPTIONAL: If you imported correctly muCommander in the IDE, you should be able to **compile** and **run** it within the IDE.
 - o Remember: You can still **compile** muCommander from command line, even if you cannot **compile** it from the IDE

4. Modifying the Software

Once you have successfully **compiled** and **ran** the software, you are ready to start the implementation of the following change (feature) request:

Add your full name to the middle-top of the initial splash window of the application, and make sure the splash window is visible for at least 5 seconds (see below an image illustrating the location of your name).

Notes:

- Adding this software “feature” would require writing around 5-15 LOC.
- However, do not underestimate the complexity of compiling/running this software, in addition to the complexity of identifying the location where you need to make the change and the actual implementation.
- Your name should be added to the splash window programmatically (i.e., writing source code) as opposed to “photoshop-ing” the image using “paint”-like applications
- You can commit your changes locally, in the master branch, but do not push your committed changes to the origin repository



5. Homework submission

Submit via [Blackboard](#) (see HW2OSSMaintenanceTask assignment) a single pdf file named CS471_S18_HW2_[LastName].pdf, based on the [CS471_S18_HW2_Report_Template](#) template found under “Homework Assignments” in <https://piazza.com/boisestate/spring2018/cs471/resources>. **The template contains ample description regarding the information you need to submit.**

Your submitted report will contain screenshots of your successful software compilation and a link to a short video capturing the implemented feature (the video can be taken either with your phone camera or through screen capture software such as [OBS Studio](#)). **Therefore, you do not need to submit your source code for this HW.**

6. Grading Rubric

The maximum points for this homework representing 5% of the final grade is 100, and the points are distributed as follows:

Item (see CS471_S18_HW2_Report_Template file)	Points
Software system and version	2
(Detailed) Environment Description	3
Compiling or Running Issues Description	35
Identifying the Relevant Code Location	15
Successful Implementation Video	25
Highlighted Source Code of the Implementation in readable format	15
Time Required for Completing this Assignment	5