

CS307 Term Project Phase 2

Due date: 07/01/2019, Thursday, 23:55

In this phase of the project, you are required to create a program, which is going to build up on the program of the first phase. If you haven't finished the first phase, or your code is not complete you can continue to work on it and finish both phases within this step.

Your program should already be able to execute several processes in a way that is similar to how computer systems work.

For this phase of the project, you will not be given the template code to work on. You are free to modify your code from the first phase. However, this will not affect the grade from phase 1.

If you haven't submitted phase 1, you are free to start from scratch and do both phases in this part of the project. The grading will be different for those, who will only submit this phase of the project. You can find the grading details at the end of the document.

Outline

With the second phase of the project you are going to extend your code to read multiple input files and read each content of the input files (.asm files) as **memory mapped files**. You will have to search for the project local directory for all files that have the type txt and store them in a new queue. For each of them you will open them and go over it as in phase 1. In addition to that you will have to transform every assembly file into a memory mapped file and read them accordingly.

Mapping files to memory will decrease the I/O bound of the processes running if the files can fit the memory. Hence mapping each txt file to memory will decrease the I/O operations.

In addition to the queues you already implemented, you will have to create an additional queue for the new input files. This new queue will be a Folder Input Queue, in which you will store the names of the txt files. Due to this new queue you will need to implement two new threads to produce and consume this new queue.

Implementation detail and Explanation of Threads

In total, you will have 7 threads. Here are the two new threads you need to implement:

Producer Thread for Folder Input

This thread will search the project directory for any input files of type txt. There might be other folders and files inside the folder, which you should avoid. After finding a file of type txt your thread should put it into the Folder Input Queue and sleep for 1 second. The thread should stop receiving file paths if the Folder Input Queue is full and it should sleep until there is an empty place in the queue.

The reason this thread is called as a Producer is actually because it produces input for the Folder Input Queue. As the Folder Input Queue is a bounded buffer, we want you to implement the producer implementation of producer-consumer problem to make the **sleep** if the queue is full and **wake up** if there is an empty space.

Consumer Thread for Folder Input

This is the consumer thread for Folder Input Queue. This thread will remove an item from the Folder Input Queue and then will wake up the Producer Thread for File Input that you have already implemented in phase 1.

As this thread is the consumer part of the Folder Input Queue, the thread should **sleep** if there are no elements in the queue and **wake up** if there is an element. This should be done using the consumer implementation of the Producer-Consumer problem.

Memory Mapped Reading

For this part of your project, you will need to implement the Producer and Consumer threads for Folder Input without using any Buffered readers. You will need to memory map each text file and read them accordingly.

You may find examples in lecture slides (lec6) and also in recitation material under SUCourse (Recitation 13). Remember to also check the Storage example that might help you organize your code and make the project easier to implement.

Grading and Demo

1. The first phase of the project is 7%
2. The second phase is 3%
3. If you haven't submitted phase 1 but do submit phase 2 with a working phase 1 you will get up to 6% in total.
 - a. Phase 1: 3%
 - b. Phase 2: 3%
4. Following the deadline, you will have a demo with the TA's showing your work.
 - a. The demos will be held between 8-10 January 2019.
 - b. We will submit a signup sheet for you, so you get to choose your demo day and time.
5. There will be 1 day of late submission with 10% penalty.
 - a. Late submission will cause you to only have demo dates on 9th and 10th of January.
 - b. If you have finals at those dates, try to finish your project on time.

Submission

1. Submission of the homework will be through SUCourse under CS307 → Assignments → Term Project Phase 2.
2. Project → Clean your project before archiving it.
 - a. This way your project document will shrink in size
 - b. Upload will be faster
3. Archive all your class files in ***su-username_termproject_2.zip***
4. Submit this file.