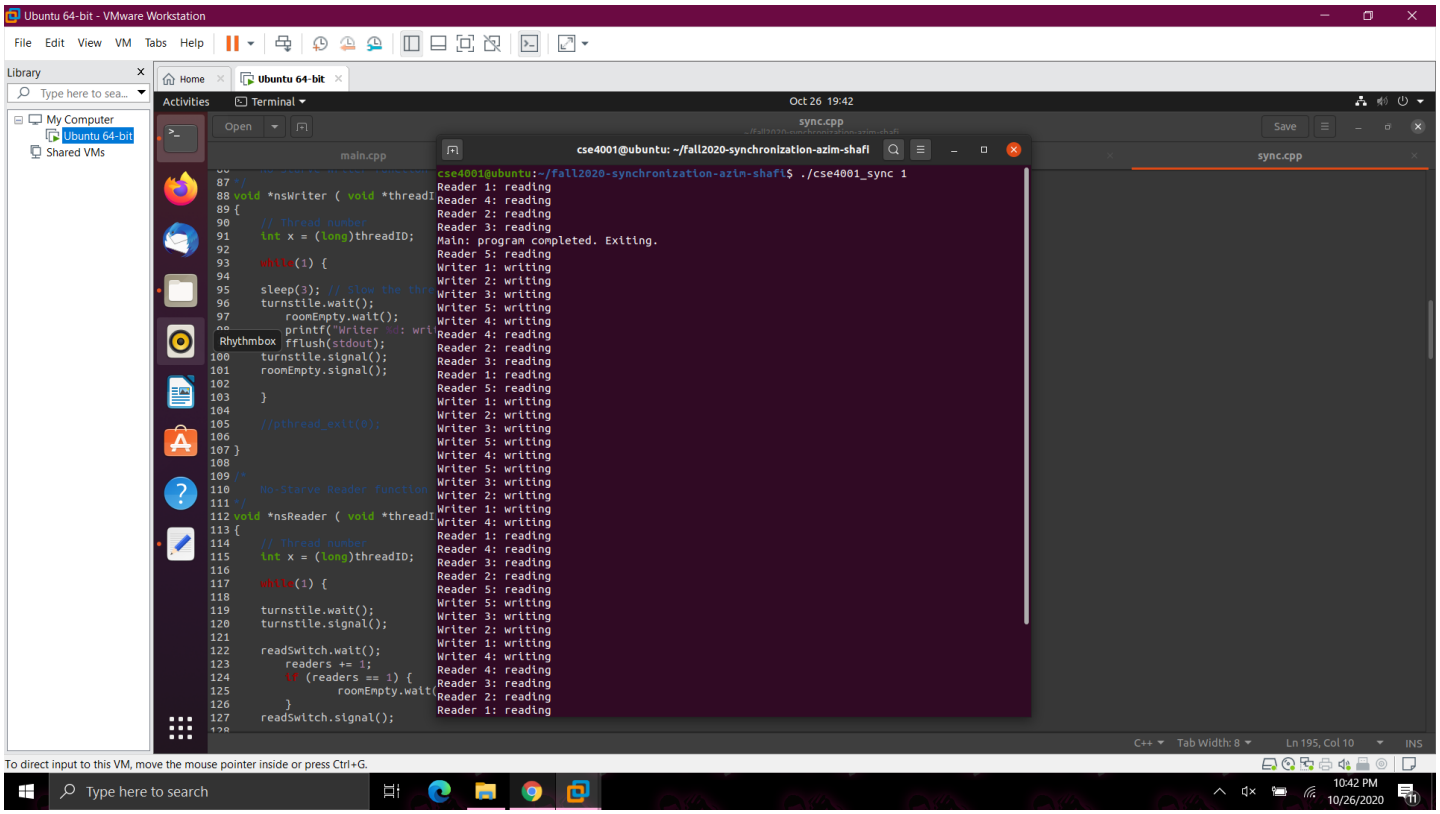
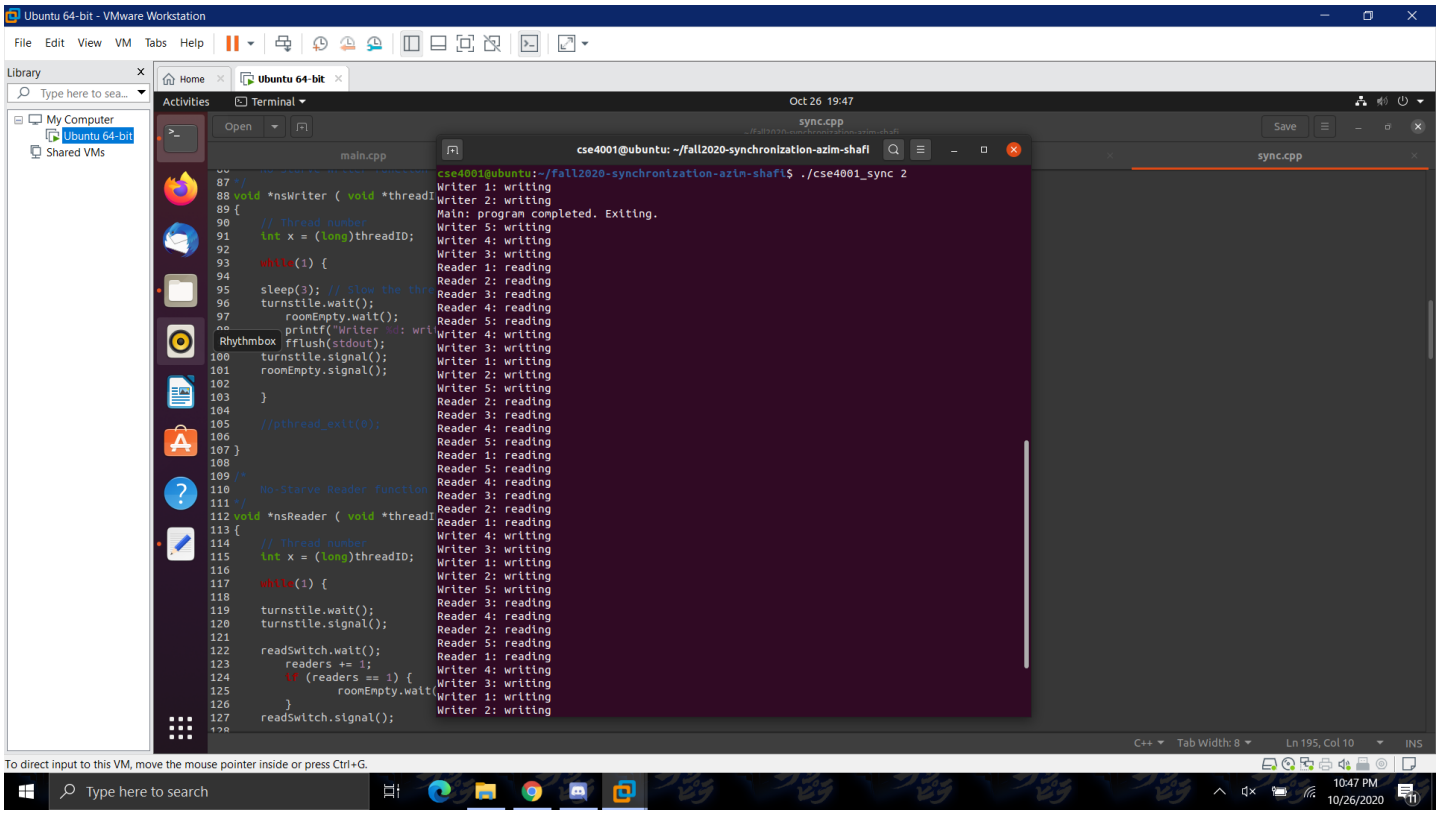
Solution #1



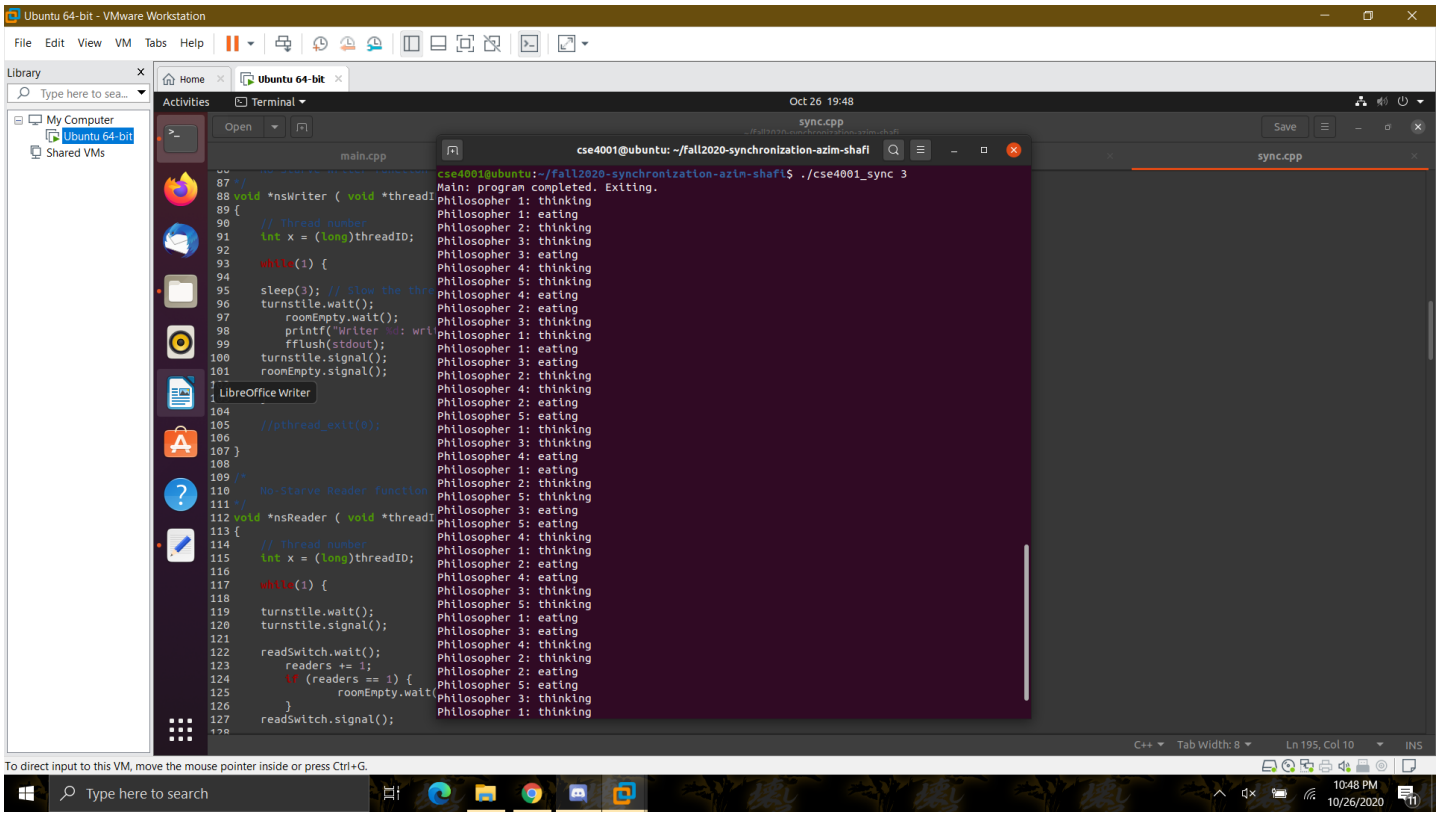
> No-Starve Reader-Writer: This solution works by allowing the readers to lock the room using a lightswitch, but aren't allow to stay indefineitely since the writes control a turnstile semaphore that limits how many readers can enter. Locks were implemented using semaphores and counters rather than creating a class.

Solution #2



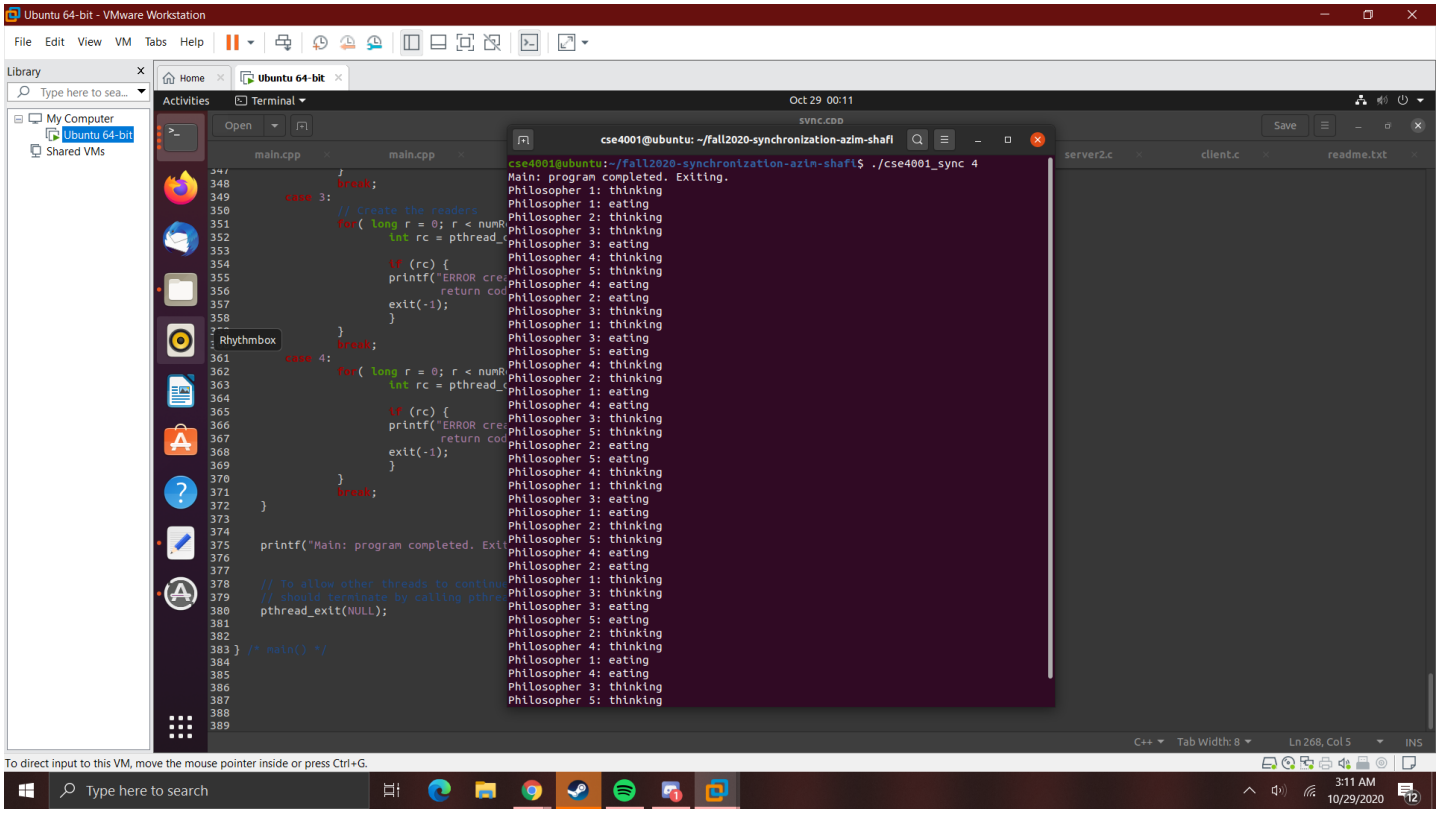
> Writer Priortiy Reader-Writer: This solution works by allowing both readers and writers to lock the room, but writers are given priority since writers must enter and leave one-by-one, but readers enter one-by-one then exit enmasse. The readswitch implementation used in the first solution was mirrored to create the writeswitch here.

Solution #3



> Dining Philosophers #1: This solution solves deadlock by ensuring that only four can be seated at the table to ensure a situation in which all aren't waiting for their second fork. An array of semaphores and modulo statments were used as suggested by the book, but due to trouble implementing the provided semaphore class as an array, five semaphores and a switch statement were used orignally, but a discussion in class cleared up that issue.

Solution #4



> Dining Philosophers #2: This solution solves deadlock by ensuring at least one leftie and rightie are at the table. This was done by assigning odd philosphers to grab the left fork first, while even philospohers grab the right fork first.