

Bonus Assignment

- The 3 files are **q1.c**, **q2.c**, & **q3.c** as required by the problem statement.
- All 3 files use semaphores for the 5 forks and 5 threads and unique handler for each of the philosophers.
- **q2.c** and **q3.c** use additional semaphores for the sauces.
- In **q1.c**, in each handler, the **sem_wait()** is used to lock the two forks associated with a philosopher thread, then after 1 second **sem_post()** is used to unlock both forks. The next philosopher thread who had attempted to access the locked fork is now free to access and lock it for its use. **sem_wait()** in each thread is called asymmetrically as well.
- In **q2.c**, in each handler, the **sem_wait()** is used to lock one fork. Then **sem_trywait()** is used to attempt eating sauce 1 through sauce 4 in order, else **sem_wait()** is used to lock sauce 1.
- In **q3.c**, each handler first acquires both forks and only after that attempts to lock onto a sauce in the same manner as **q1.c** and **q2.c**.

Instruction to Run:

1. run **make** in terminal.
2. run **./q1** for 2 forks without sauces.
3. run **./q2** for 1 fork with sauces.
4. run **./q3** for 2 forks with sauces.