Operating Systems
Assignment 4

SUBMITTED BY : **ANANYA KANSAL**
ROLL NO : **2019458**

mysemaphore.c

- **Blocking implementation** of the dining philosophers problem.
- The semaphore struct contains ssize_t cnt variable to keep track of semaphores, a mutex variable and some boolean values along with a condition variable.
- For a philosopher to eat it requires 2 forks and 2 sauce bowls.
- Made use of pthread_cond_wait() function to implement wait. When the value of the respective semaphore isn't 0 , it blocks the execution of the current thread until it gets a signal to continue execution.
- When the philosopher is able to lock all the resources like forks and sauce bowls it goes into the eating state , when these resources are freed, the signal function increments the value of the cond_t variable.
- Functions lock() and unlock() have been used which control the critical section.
- This method avoids deadlocks by giving access to critical section of code
  i.e. wait and signal section only to one process at a time using mutex.

non_blocking.c

- **Non Blocking implementation** of the dining philosophers problem.
- Implemented a while loop in the wait function which checks the value of the cnt variable , if it is 0 we wait until a signal increments it by freeing the resources and then assign them to another philosopher.
- Functions trylock() and unlock() have been used for threads instead of lock().