OPERATING SYSTEM

ASSIGNMENT-4 (WriteUp)

Rahul Khatoliya 2019265

Functionality and Technique Used in the Program :

Blocking wait() and signal() :

We had implemented a semaphore structure called **my_semaphore ,** using the pthread library , in which we had used **Conditional Variable(pthread_cond_t)** , and **Mutex(pthread_mutex_t)** provided by the pthread Library . For our **wait()** and **signal()** implementation , to give mutual exclusion to the several threads running concurrently , we had simply used the **lock** and **unlock** mechanism of **pthread_mutex_lock(mutex)** and **pthread_mutex_unlock(mutex)** along with the **pthread_cond_wait()** and **pthread_cond_broadcast()** , which will help to **block** the other **threads** , and **wake up** all the blocked **threads** running respectively. Also, the prime objective for **wait()** ,would be to decrement semaphore value by 1 , and for **signal()** to increment value by 1 , in both the case mutual exclusion has been provided.

Non-Blocking wait() and signal() :

For Non-blocking variant , we have not provided any **blocking** or **waking** call , but simply the **Mutual exclusion** mechanism , and in the case of **wait()** after decrementing the semaphore value by 1 , if the called semaphore is already locked , then we will just return the specified error number i.e **EAGAIN** using the header **<errno.h>** , otherwise successfully return 0 , but here without blocking other threads . Moreover for **signal()** , we will increment value of semaphore value by 1 as usual and if the value was negative (which should not happen ideally) , we would Account with the respective error number i.e **EINVAL** , otherwise terminate successfully .

Deadlock Free Dining Philosopher Solution :

In our program we had created two arrays of **semaphores** , one for **Forks** and other for the **bowls** .In case of **blocking** variant , every philosopher will wait first for **left fork,** then **right fork** except the last philosopher , so that if all are waiting for the right , it will help to avoid deadlock scenario. Moreover , Every Philosopher have to wait for **bowl 1** and **bowl 2** , then only they will eat .Whereas , in case of **non-blocking** variant , due to **no blocking** there are no chances for **deadlock** , and hence every philosopher will wait for the left fork , then right and eventually both the bowls , then will eat successfully. Also after eating in blocking case , **signa**l has been provided to **wake up** other philosophers , where no such waking in non-blocking case .