

# Operating Systems - II

## Theory Assignment - 1

### Pseudo Algorithm :

```
semaphore mutex = 1;           // Binary semaphore
semaphore update_values = 1;   // Binary semaphore
semaphore grp_lock = 0;        // Binary semaphore
bool grp_formed = false;       // Boolean variable to indicate if a grp is formed or not
int number_of_ppl_dining = 0;  // Number of people that are currently dining

// Entering restaurant
wait(mutex);                   // This is to ensure that one person is processed at a time

wait(update_values);           // Lock to update grp_formed boolean if needed
if(number_of_ppl_dining == X) // Checking if the group is formed or not
{
    grp_formed = true; // If group is formed we update this variable to true
}
signal(update_values);

if(grp_formed)                 // If grp is formed wait on grp_lock
{
    wait(grp_lock);            // Waiting
}

wait(update_values); // Here we are updating number of people dining as they ready to enter
number_of_ppl_dining++;
signal(update_values);

signal(mutex);                // Releasing the mutex lock as the person is processed

// Simulate eating

// Exiting restaurant
wait(update_values); // Using update_values lock to decrement count as customer is exiting
number_of_ppl_dining--;
signal(update_values);
```

```

if(number_of_ppl_dining==0 && grp_formed==true)
{
    wait(update_values); // Lock to update grp_formed boolean if needed
    grp_formed = false;
    signal(update_values);

    signal(grp_lock); // Signalling the grp_lock as the group left people can occupy the seats
}

```

---

## Explanation :

- The variable `number_of_ppl_dining` is to keep track of the number of people who are inside the restaurant and are currently eating.
- Boolean variable `grp_formed` is used to indicate if a group is formed or not.
- Here I used 3 locks to make sure that there are no race conditions and mutual exclusion is maintained.
- The locks are used as follows
  - **mutex** - This lock makes sure that one person enters the restaurant at a time.
  - **update\_values** - If we were to update the variable `number_of_ppl_dining` or `grp_formed` we use this lock, so that there will not be a race condition.
  - **grp\_lock** - If a group is formed in the restaurant and new customers come to the restaurant they see that a group is already formed and they wait on this `grp_lock`
- **Entering the restaurant :**
  - We start by locking the mutex semaphore, we use it to deal person by person basis.
  - As soon as a thread(customer) enters we first check if the group is already formed or not. If it is formed we update the corresponding boolean value that is keeping track of it. We use a lock while doing this checking, updating to ensure there are no race conditions.
  - If we find that a group is formed we wait on `grp_lock` semaphore.
  - Here there are two possibilities a person can enter and see that a group is already present or a group is not present. In the second case he can take the seat immediately hence we increment the value of `number_of_ppl_dining` using `update_values` lock to ensure mutual exclusion. However for the former case, let's assume that at this point he is waiting on `grp_lock` as the group is formed, but once the group is done eating and left, signal on `grp_lock` is invoked after invoking that, this

person can enter the restaurant. Even then we should increment the `number_of_ppl_dining` variable.

- After this the person can safely enter the restaurant, hence we signal the mutex semaphore.
- For example if  $X = 5$  and there is a group formed. Let's assume 2 new members come. First one member is sent, he sees that group is formed and he is waiting on `grp_lock`, this person is also holding a mutex lock. Hence the next person cannot enter until the group leaves and this person enters the restaurant. This decision is made so that mutual exclusion is maintained.
- After a person is found safe to get a seat, he starts eating. After he finishes eating he exits.
- **Exiting the restaurant :**
  - As the person is exiting the restaurant we use `update_values` lock to decrement the value of `number_of_ppl_dining`.
  - If we find that a group is formed and all the members of the group have left the restaurant we signal the `grp_lock` and update the value of the `grp_formed` variable.