# PROBLEM STATEMENT:

The line for serving Biryani is growing day by day in the Kadamb Mess. Annoyed by waiting for hours in the queue you have decided to automate the entire pipeline of serving Biryani.

**Robot Chef :**

- From now on, there will be M Robot Chefs present in the kitchen preparing vessels of Biryani.
- Each Robot Chef can prepare a random number (greater than or equal to 1) of Biryani vessels at a time.
- Each Biryani vessel has a capacity to serve P Students .
- Each Robot chef takes w seconds (random between 2-5) to prepare r vessels (random between 1-10) of biryani at any particular time. Each vessel has a capacity to feed p students (random between 25-50).
- Once the Robot Chef is done preparing the Biryani Vessels, he invokes the function biryani_ready() and does not return from it until all the biryani vessels he has cooked are loaded into any of the serving containers.
- Once all the biryani vessels are empty, the biryani_ready() function returns and the Robot Chef resumes making another batch of Biryani.

**Serving Tables:**

- There will be N Serving Tables present in the mess.
- Each Serving Table has a Serving Container which loads a Biryani vessel prepared by any of our Robot Chefs. Only one vessel of Biryani can be loaded at a time into the Serving Container. The Serving Container can be refilled only when it is empty.
- The Serving Table incorporates the latest state of the art technologies and has automated the process of serving Biryani through multiple slots available on it.
- As long as a serving table has enough portions of Biryani left in the serving container, it makes a random number of slots available for the students to collect a portion of Biryani from the Serving Table. Note that the number of slots should always be less than or equal to the portions left in the Serving Container. Read down for exact limits.
- Thus there is a possibility that once a Serving Table has x available slots and later has greater than x or less than x slots available. After serving x portions (and consequently x students) the portions available in the Serving Containeris updated.
- Each Serving table's serving container is initially empty. It waits for any of the Robot Chefs to load a vessel of Biryani into the serving container. The serving table cannot go into serving mode as long as its container is empty.
- Once the serving container is full, it enters into the serving mode. It invokes a function ready_to_serve_table(int number_of_slots) in which number_of_slots (chosen randomly

between 1-10) denotes the number of slots available at the particular serving table at that instant. The function must not return until either all the serving slots of the serving table are full or all the waiting students have been assigned a slot. Note that student can be assigned a slot at any of the serving tables. (If there is no waiting student, then the function returns).

**Studenets :**

- K Students  have registered for the Biryani.
- Once a student arrives in the mess, he/she invokes a function wait_for_slot().This function does not return until a Serving table with a free slot is available,that is ready_to_serve_table method is in progress. Once the function returns the students goes to the allocated slot and waits for the slot to serve Biryani.
- Once the student is in the slot, he/she will call the function student_in_slot() to let the Serving Table know that he/she has arrived at the slot.
- Once all the students have been served you are done for the day.

**Instructions :**

- Each Robot Chef, Serving Table and Student are threads.
- Stop the simulation when all the students are served.
- A Serving Table is used multiple times for serving. That means if there are still students left to be served, the serving table should invoke ready_to_serve_table().
- Use appropriate small delays for the simulation.
- The use of semaphores is not allowed. You can use only one mutex lock per Serving Table and Robot Chef.
- Your simulation should allow for multiple students to arrive at a Serving Table simultaneously. It must be possible for several students to have called wait_for_slot() function and the wait_for_slot function returned for each of the student before any of the student calling student_in_slot function
- Your simulation must not result in dead-locks.
- You are allowed to declare more functions if you require.

#### Assumption

It is assumed that the serving phase of a table starts once all the slots of that table are occupied by the students or there are no further students remaining to have biryani.

# SOLUTION EXPLAINED:

1. Robot: Has id of robot, ready (0 when preparing food, 1 waiting for vessels to get empty) and a corresponding mutex,cond. variables , thread id etc.

2. Student: Has id of Student,slot_status (0->not arrived,1->waiting,2->slot alloted),corresponding arrival time)
3. Table: Has idx of Server , arr of students currently eating on slots at one serving phase,total no. of slots etc.

# INITIALIZING FUNCTIONS FOR CHEFS , STUDENTS, TABLES :

- It creates corresponding threads and structs for each of them and joins them accordingly.

# IMPORTANT FUNCTIONS:

## robot_init

- The chef enters this function it preapares the vessels of biryani and changes its ready to 1 and calls biryani_ready function;

## biryani_ready

- The chef after preparing food waits in this function till all the vessels prepared by him are emtied. It then calls the biryani preparing function for again preaparing the biryani.

## table_init

- If the Table container was empty it searches for the chef whose biryani is ready(applies lock to each chef and unlocks it after analysing) and fills its container and decreases the count of the vessels preapared by that chef.
- If the table container was not empty it pours birayni from its container to the slots.
- It further finds the number of slots it can make available at that time and calls the ready_to_serve function.

## ready_to_serve

**It is assumed that the serving phase starts once all the slots are occupied by the students**

- It waits till all of its slots have been assigned to the students or there are no further students and then starts the serving phase by serving biryani to students that came to that table(they were stored in the stack of that table), by poping its stack.
- After serving it calls the waiting to serve again.

## wait_for_slot

- Student comes here and finds the table which has available slots(by applying mutex locks and unlocking them after analysing them).

- On finding a slot it decreases the number of slots availabe on that table(its id is pushed in the stack of that table) and calls the student in slot function.