Automating Biryani Serving

Compilation and Execution

```
make
./restaurant
```

Input

```
number_of_robot_chefs   number_of_serving_tables   number_of_students
```

Process

1. Student arrives and waits to get a table in the restaurant.
2. Table gets free and waits for the robot chef to serve a vessel of biryani.
3. Table starts assigning tables to itself and signals students about it.
4. Biryani is served for all the students sitting on the table.
5. Students, after finishing their food, signals the table.
6. Table, after receiving signals for all the students, repeats the process.

The Flow of the Code

1. Structures: Restaurant, Robot Chef, Serving Tables, Student
2. The program initializes and creates the Restaurant thread
3. The Restaurant thread initializes the Robot Chef threads, the Serving Table threads, and the Student threads.
4. The Student thread waits for a Serving Table to be free. Upon finishing the biryani meal, student thread exits.
5. The Serving Table thread waits for any of the Robot Chef serve biryani vessel and then, assigns students to itself. If all the students are done, it exits.
6. The Robot Chef thread cooks biryani in multiple vessels and waits for the Serving Tables to get them from it.

Implementation Details

1. A mutex lock and two conditional variables are used.
2. Whenever a Serving Table or a Student tries to access a shared data, mutex gets locked. The mutex is used along with two conditional variables.
3. One conditional variable is used to wait for some signal from the students whereas the other from the Serving Tables.