

## Project 2: Inter-process communication (IPC) and synchronization

The purpose of this project is to learn solving inter-process communication problems during concurrent execution of processes.

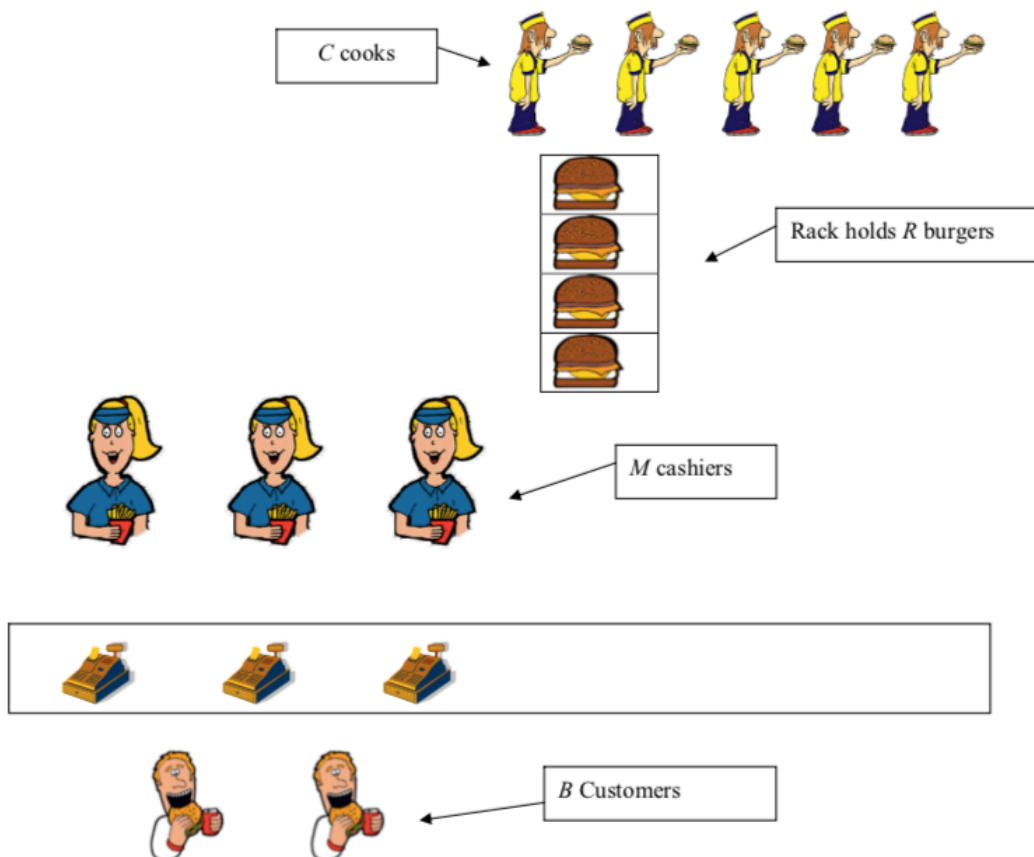
### Project explanation

1. BurgerBuddies.c : Design, implement and test a solution for the **IPC problem** specified below. Suppose we have the following scenario:

Operation of this scenario is as follows:

- > Cooks, Cashiers, and Customers are each modeled as a thread
- > Cashiers sleep until a customer is present
- > A Customer approaching a cashier can start the order process
- > A Customer can not order until the cashier is ready
- > Once the order is placed, a cashier has to get a burger from the rack
- > If a burger is not available, a cashier must wait until one is made
- > The cook will always make burgers and place them on the rack
- > The cook will wait if the rack is full
- > There are NO synchronization constraints for a cashier presenting food to the customer.

Implement a (concurrent multi-threaded) solution to solve the problem.



2. Submit a README file that lists the files you have submitted along with a one sentence explanation. Call it Prj2README.txt.

3. Copy your programs' outputs into Output.txt.

4. Submission Notes:

Use the given names for files as given in Project explanation above and as listed below, and put files into Prj2\_YourID folder. Compress the Prj2\_YourID folder as Prj2\_YourID.tar.gz with the command below. Submit through platindys.

**tar -czvf Prj2\_YourID.tar.gz Prj2\_YourID**

Prj2\_YourID.tar.gz should include the files below.

- Prj2README.txt
- Prj2header.h
- BurgerBuddies.c
- Output.txt

5. Due date: 01.05.2020, submit before midnight.

Late policy

For every day the assignment is late after the due date, you will lose 4 points from your assignment score. Assignments will not be accepted after they are five days late.