

Project Report

Summary

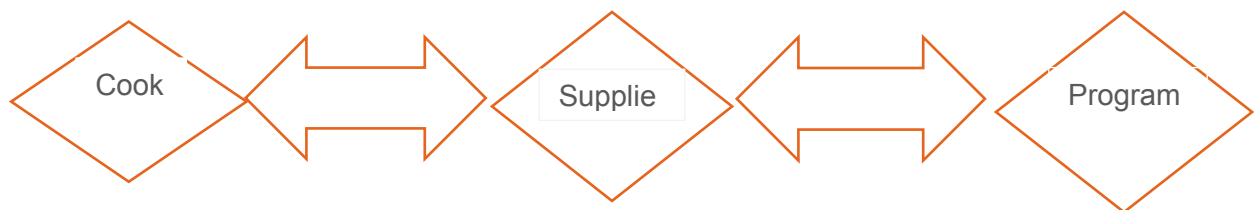
My project contains four files each file represents a character which I need to implement.

1. Supplier (help.c)
2. Cook (cook.c)
3. Student (Graduates + Under-Graduates) (student.c)
4. Program (main.c)

All the characters are working simultaneously. First of all, Program takes related information from user as a command argument. Then program sends all the information to Supplier, Cook and Student. Supplier, Cook and Students are also sharing the information among each other's. For the communication between each other pipes are being used. Each character is working as an individual process.

Supplier(help.c):

In supplier process I have used two pipes one is used to communicate with the main to get the number of plates supplier can send and other is used to communicate with cook for placing the dishes in to kitchen.



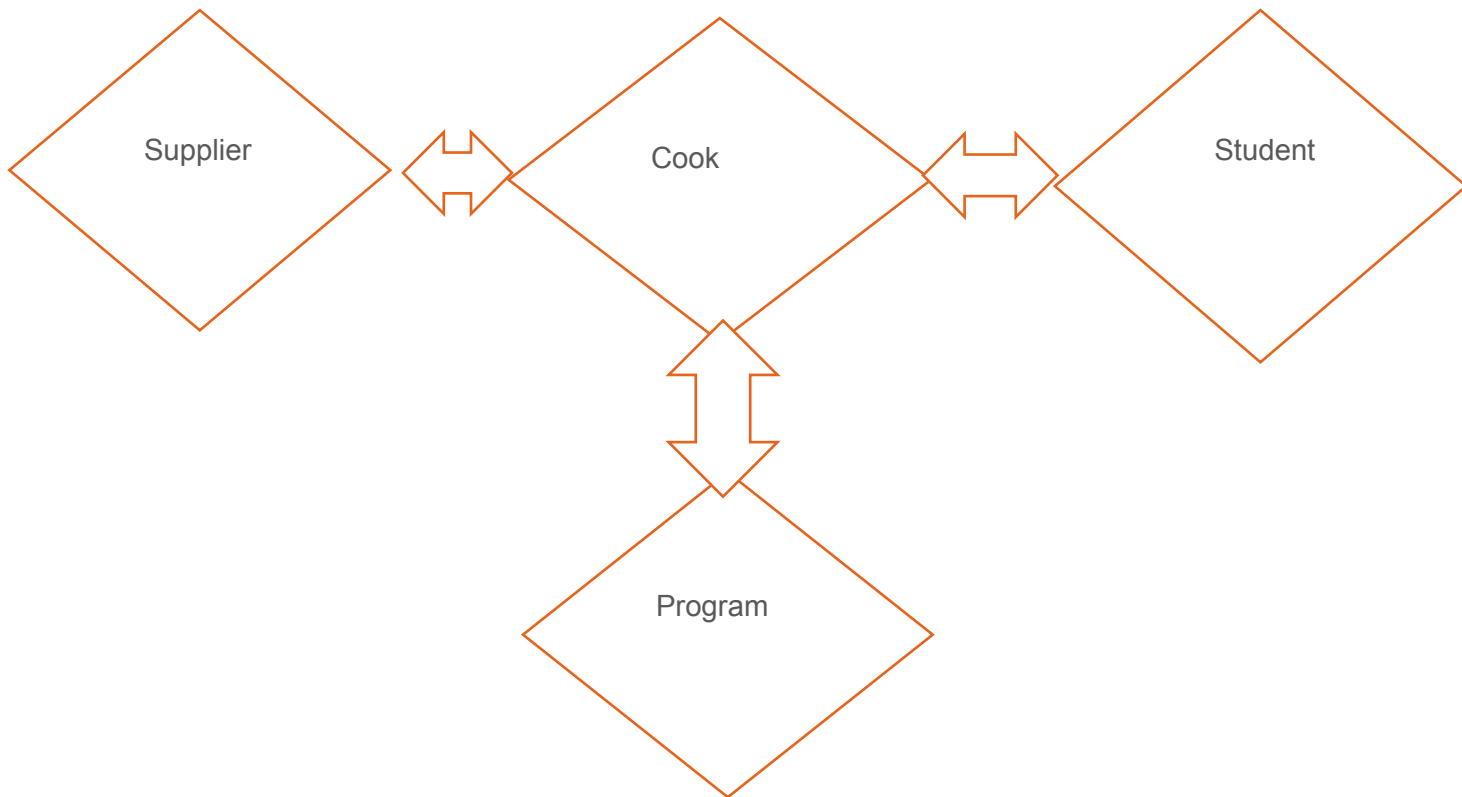
Cook(cook.c):

In cook process I have used three pipes to communicate with supplier, program and student. From program I have taken the number of cooks. Cooks are given ids (0,1,2.....n). Threads are created which are equal to number of cook, For the synchronization of all threads I have used semaphores, I have used the counter capacity if counter capacity is full then cook waits. If cook serve all the item to counter the cook prints a message prints "Good Bye" and terminated.

As I have already mentioned above the pipe communicate with supplier and get items (main course, desert and soup).

Pipe communicate with student tells the capacity of counter to student act like shared the present situation of counter but student don't need to wait for that as busy waiting is not allowed so only student waits and the take his/her meal if counter have all the three items.

Pipe to communicate with program is used to tells the capacity of counter and total number of cooks.

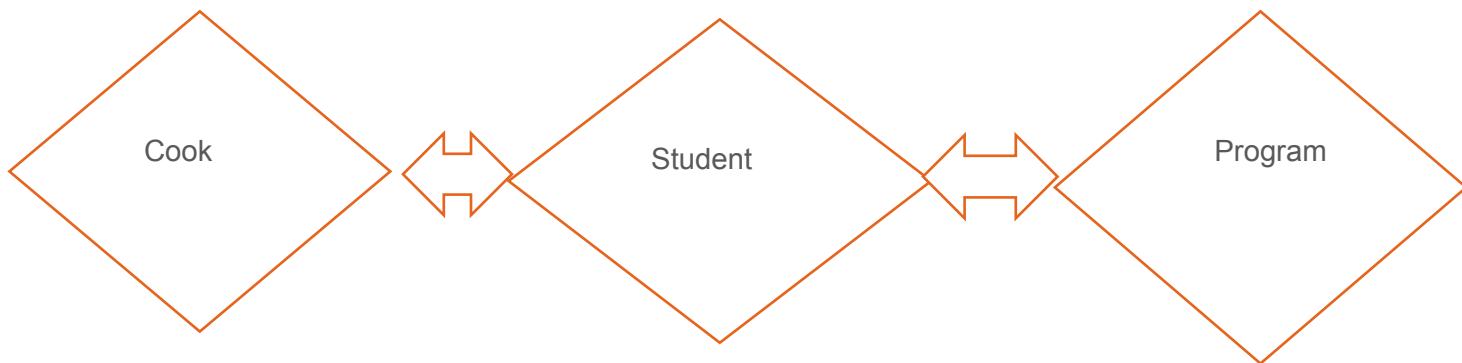


Student(student.c):

In student process I have used two pipes to communicate with supplier and program. From program I have taken the number of students and rounds. Students are given ids (0,1,2.....n). Threads are created which are equal to number of students, For the synchronization of all threads I have used semaphores, I have used the counter if counter does not have all items then students waits. If student complete all its rounds, then prints a message prints "Good Bye" and terminated.

Pipe communicate with cook tells the counter to student act like shared the present situation of counter but student don't need to wait for that as busy waiting is not allowed so only student waits if counter does not have all the situation and the take his/her meal if counter have all the three items.

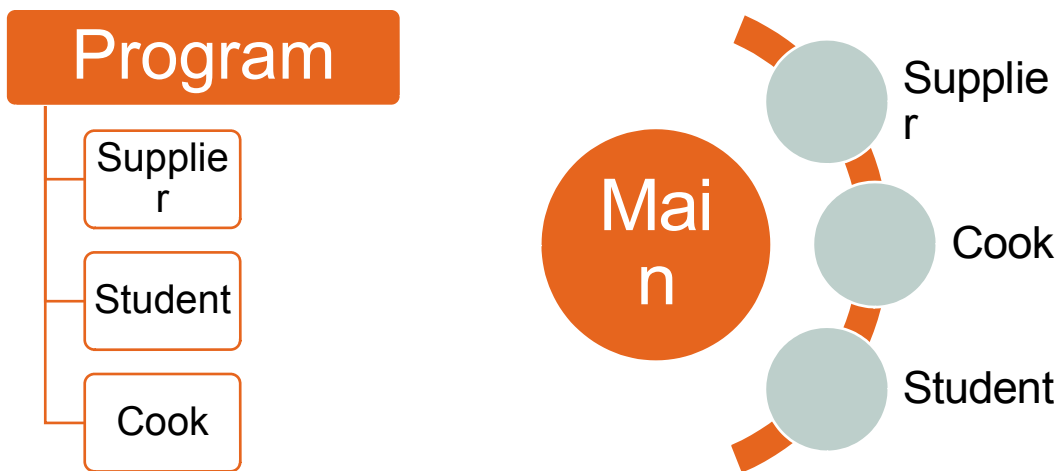
Pipe to communicate with program is used to tells total number of students and rounds.



Program(main.c)

Program communicate with all the process and also run then as independent process does fork in such a way that below situation occurs.

It has four thread function All the functionality has discussed above but the fourth thread function run to share data with other process or initialize other necessary requirements.



How to Run?

From make file

`make -f Makefile`

`./main.o` (terminal arguments)

Manually Compilation and running

`gcc main.c -o main.o`

`gcc supplier.c -o supplier.o`

`gcc student.c -o student.o`

`gcc help.c -o help.o`

`./main.o -N 2 -M 10 -T 5 -S 4 -L 13`

Milestones

I have faced many challenges in this project that are mentioned below:

1. Synchronization issue
2. Communication between individual process
3. Not to use Busy waiting