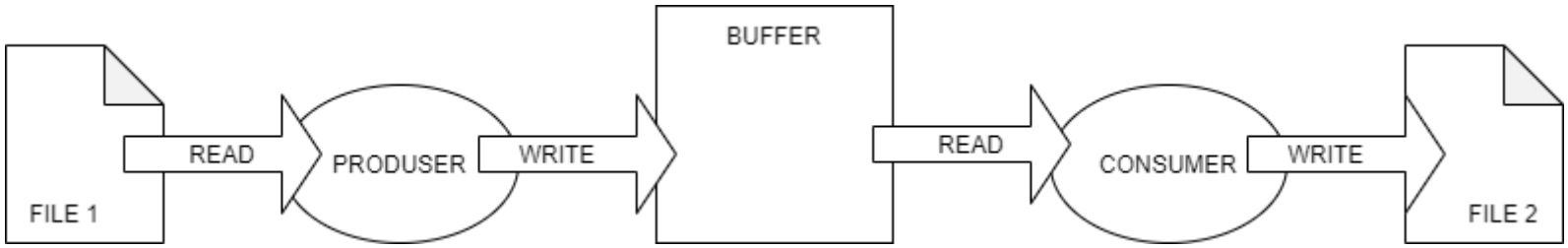
Homework 4

Team 10
Apostolopoulou Ioanna
Toloudis Panagiotis

Assignment 1



Assignment 1

In Main:

Create the Threads and start them.

In producer:

If remains_size > buffer_size

Read for the file.

Else

Realloc the buffer

Read for the file.

Put to buffer and switch coroutine.
If the end this file, make flag equal 1

In consumer:

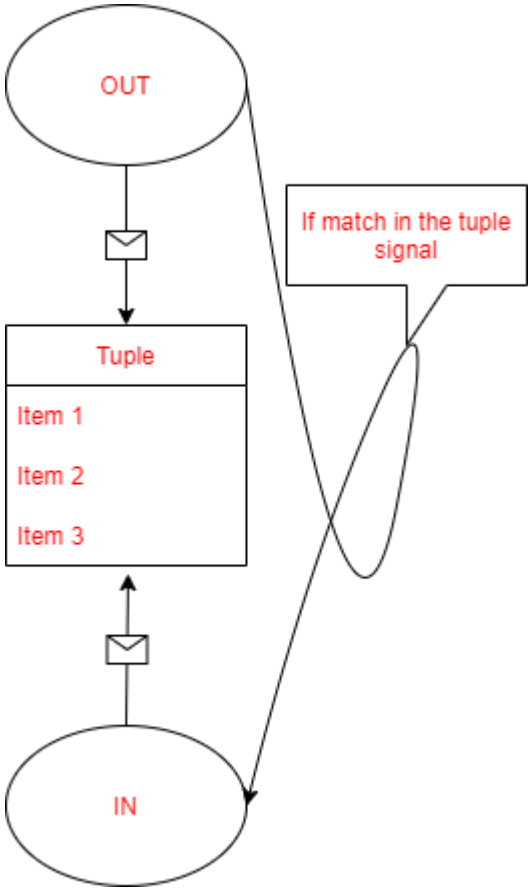
Read form the buffer, write to the file and switch coroutine.

If flag not equal to 1, switch to producer.

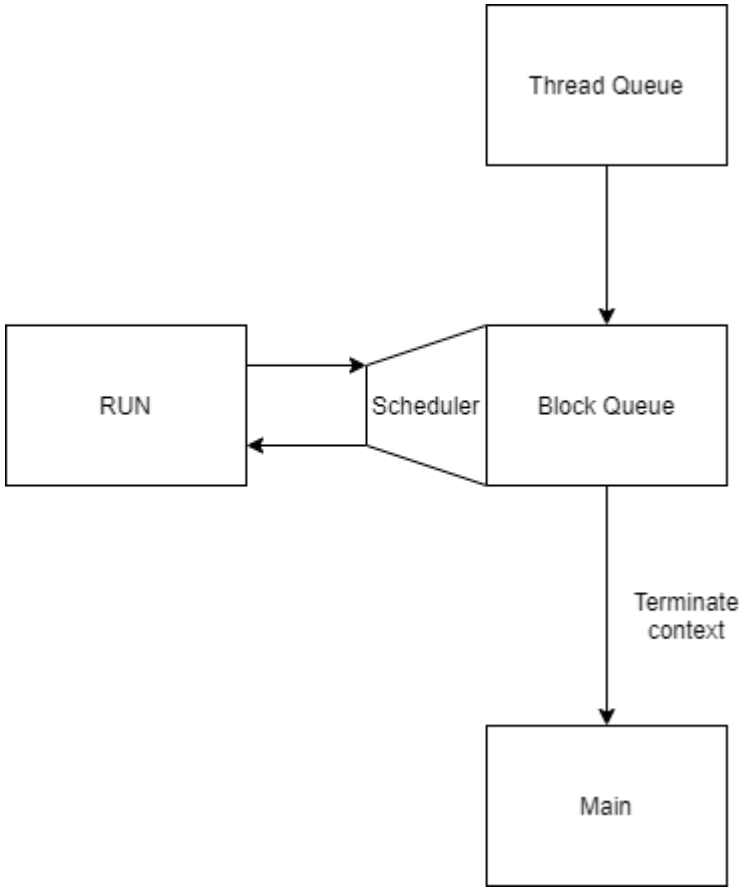
Else switch to main.

Assignment 2

Tuple Space



Scheduler

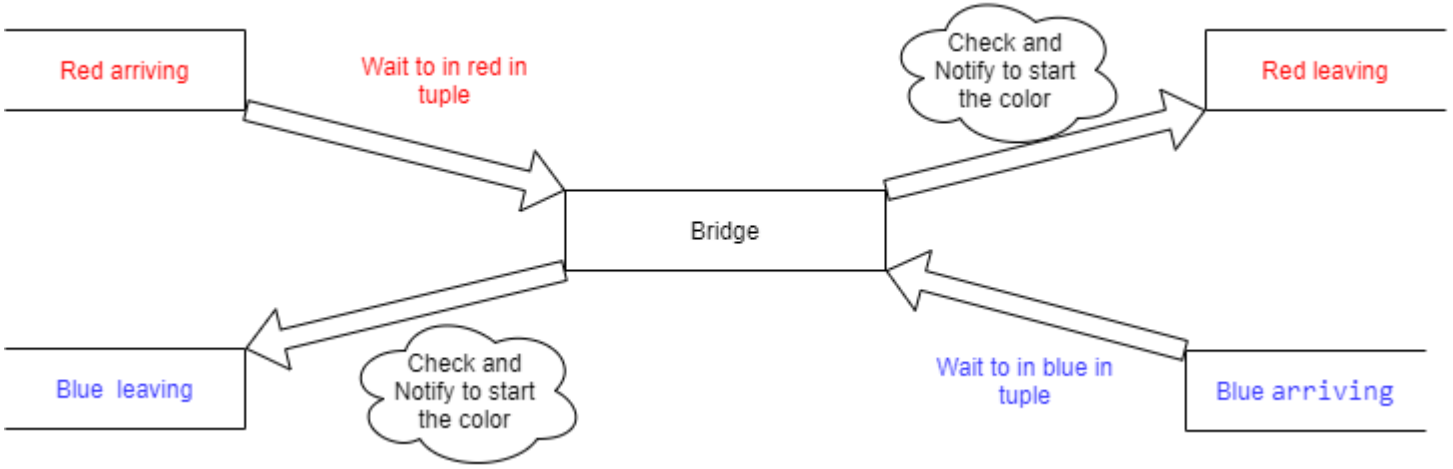


Assignment 2

- In init function initialize the hash table.
- In destroy function destroy the not using node.
- In match function check the node about the type of value return 1 for success
- and 0 to not.
- In bind val function copy the value to a one note to the anther node.
- The out function put the values in to the tuple.
- The in function comes out the values into the tuple, using the “%” to the standard value and “?” to the unknown value.

- Create the Cpu clock.
- Add the coroutines in the queue.
- Receives signals and blocks or unblocks them.
- If terminate a coroutines and removes them from queue.

Assignment 3



Assignment 3

- Arriving_Cars()
- Checks the color of the Car that Arrived at the edge of the Bridge
- Checks if Bridge reached Full Capacity
- If specific color has priority
- And if there are not Cars of other Color Waiting
- If all above are true Car should Cross the Bridge

- Leaving_Cars()
- Checks if the Car_On_Bridge is the last on Bridge
- Checks the color of Cars on the Bridge
- Checks if there is only one Car on the Bridge
- Prioritizes the turn of Color of Cars
- Make out in the tuple

- Main()
- Create and Initialize tuple space and then start main coroutine.
- Reads from file the amount of Cars, the Color and the time they need to generate new ones
- If the amount of Cars given is negative program Exits
- Wait to all thread make join and finally destroy and free the memory