Schematic for Genre Map Reducer

Naturally the project is composed of 3 main namespaces

1. Map Reducer : in charge of orchestrating the processes before full reduction
2. Reducer : an image that can be executed multiple times which represents an orchestration and communication with respective mappers
3. Mapper : an image that is executed once for each file and is in charge of mapping the data into a KV form and communicating it over pipes to the reduction process

We dispatch a genre reduceer for each genre , and a file mapper for each file Their communication is as follows.

Because of the nature of the named pipe:

because of the nature of the named pipe communication had to be done asynchronously between reducing and mapping endpoints.

Also both sides should keep the pipe open , or else the pipe will be unavailable .

The communication of main process and the reducing and mapping processes is done through unnamed pipes , the communication of reducing processes and mapping processes is done asynchrously through named pipes.

A naming convention is used to relate the processes and removes a liability to communicate pipe names to both these processes

I have used named pipe names as keywords such as this

```
// Each  named pipe's  name looks like this
/tmp/processed<file_no><genre>
```

Finally each genre reducer prints a final result that represents the count of books in this directory with a specific genre

run with :

```
make
./map_reducer lib
```

make sure to clean :

```
make clean
```