**REPORT**

First of all, our aim in this project is work with multithreaded program. We use publisher and packager as threads. Our program first take input from user as Publisher type, Publisher thread count, packager thread count, number of book each Publisher, package and initial buffer size.

As struct we have book and that’s attributes are published id, Publisher thread id, book name.

As struct we have Publisher and that’s attributes are Publisher id, Publisher id string(this is just string type of Publisher id), mutex, book pointer, last published book number, buffer size, finished thread count, Publisher threads pointer

As struct we have Publisher meta data and that’s attributes are thread id, is thread finished, publisher pointer.

As struct we have package and that’s attributes are buffer index, book pointer.

As struct we have packager and that’s attributes are package id, package index, package pointer.

In main first for loop, we create our Publisher threads. We first assign its id after make it string and load it to list using strcpy function and in rest 5 row we load values which we get from user and mutex.In next for loop we assign values into metadata to send pthread\_create function.

In second for loop, we created packager and load values after creat packager thread at every iteration.

In last 2 for loop we wait for our threads with pthread\_join function.

We have speacial function for our Publisher threads called as publisherRunnerFunction.It first take input args as parameter and load it to metadata.In for loop first we assign name of each book to its related place book.bookName. In if case that is located at 98th row we enter that parti f buffer is full.In that if block we double size of buffer, hold last values in buffer after assing it to new buffer and to save space free old buffer. After in line 123 we started to load book into buffer and increment index, print related statment into screen after all of there we unlock the mutex to let other thread to load or store boks into shared buffer.

We have speacial function for our packager threads called as packagerRunnerFunction. It first take input args as parameter and load it to packager after it created random number Publisher type count to collect book. Make packager buffer index 0 and allocate memory for package buffer.In line 160 we have if statement and it work if package is empty.In line 162 we have if statement which check if there is any thread in system related to that Publisher if yes line 165 make no book as true and in next for loop we try make it 0 to indicate there are book left in other Publisher lists. In line 175 if we have book in other ppublisher types we create random number and go for that type.If we have no book in other types we print packager’s book after write exiting system. If packager is not empty w ego into line 212. In 214 we assign current package after in line 215 check if packager size if full then print book that package contains after free old buffer to save memory and at the end assign new package to current pointed package point.In line 246 we look book and in line 252 load into package after we makes related increment and decrement opertions unlock the mutex and try for other packager with created random number in next cycle of for loop.

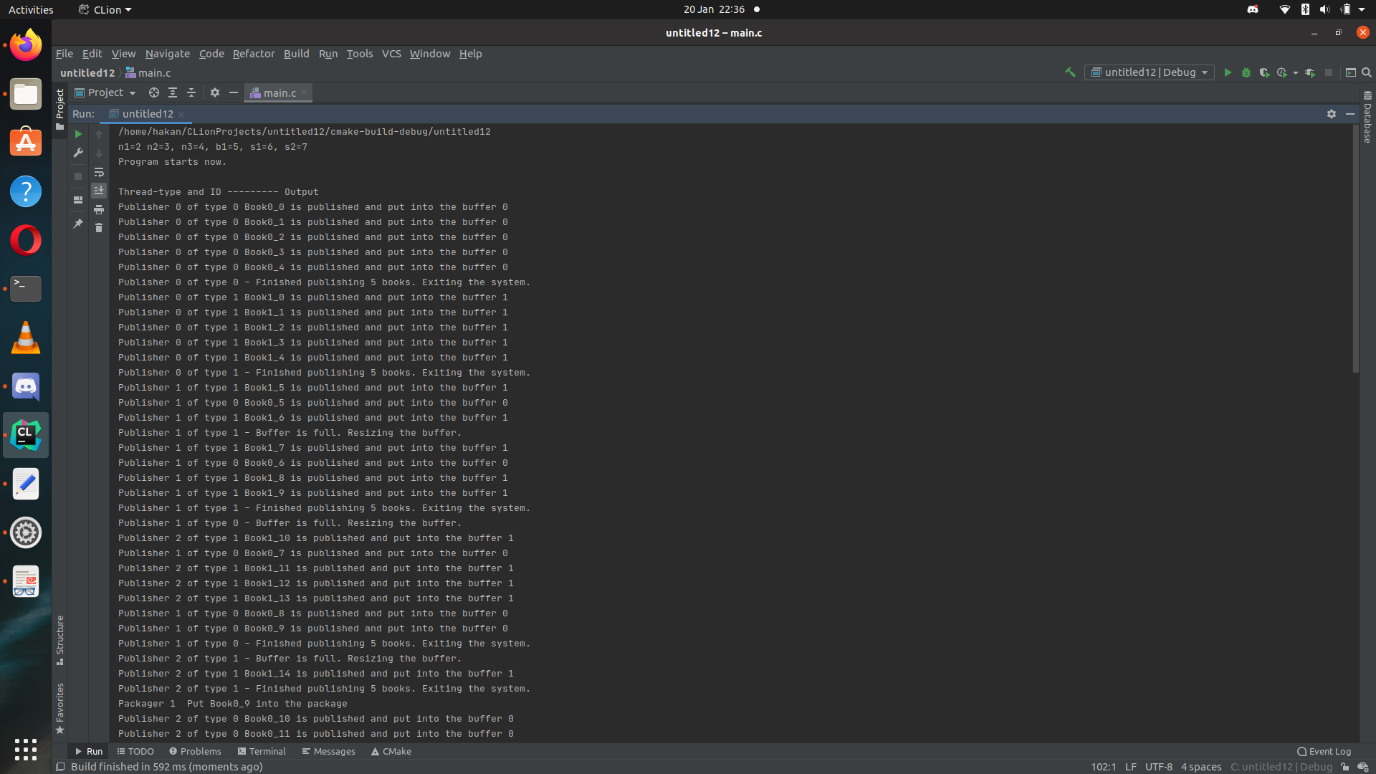
While doing this Project we have learned that how to handle multiple threads while using same buffer and hwo to synchronize them reaching memory one by one if it is eampty other wait for it. The most hardest part for us while implementing these processes, managing buffer informations and package need to wait if there is no book left while there are threads producing is a bit more complex compare to other parts.

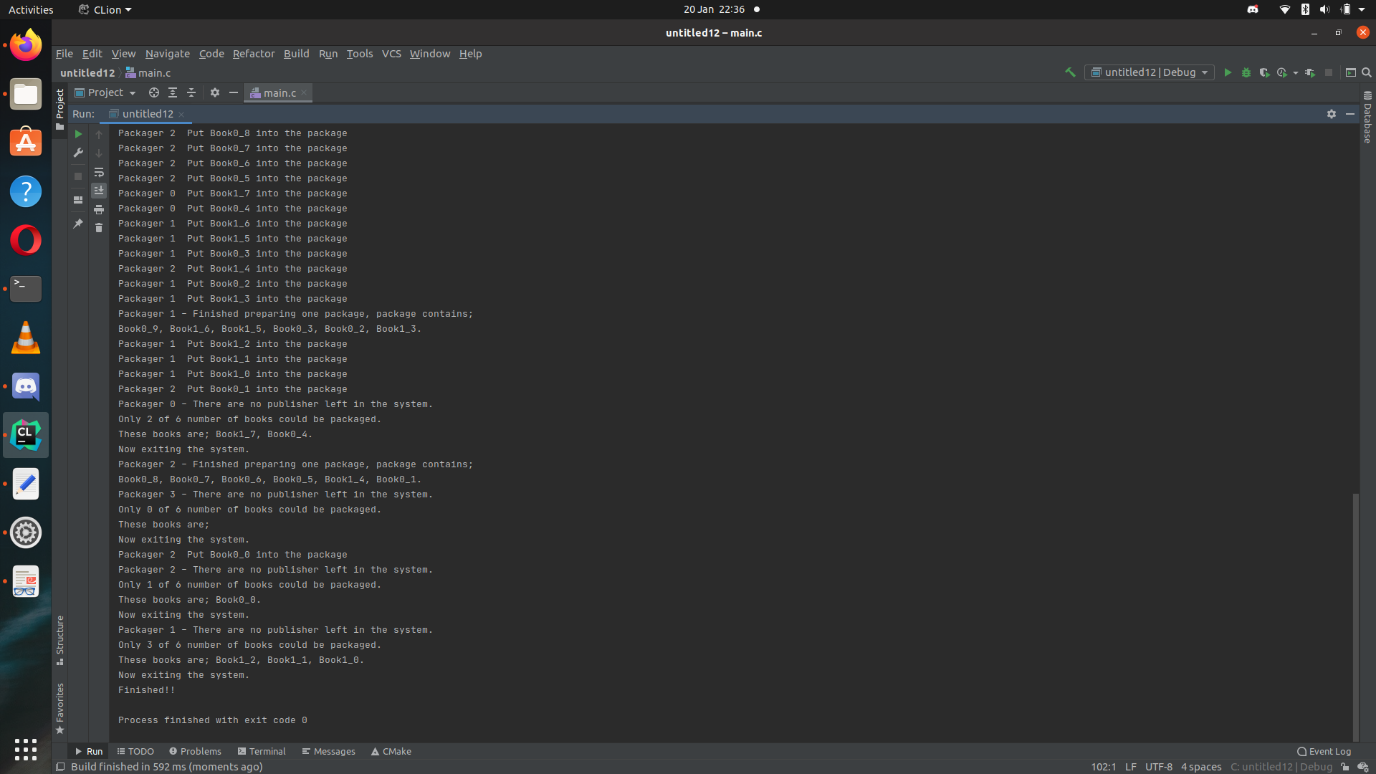
Below you can see example of execution

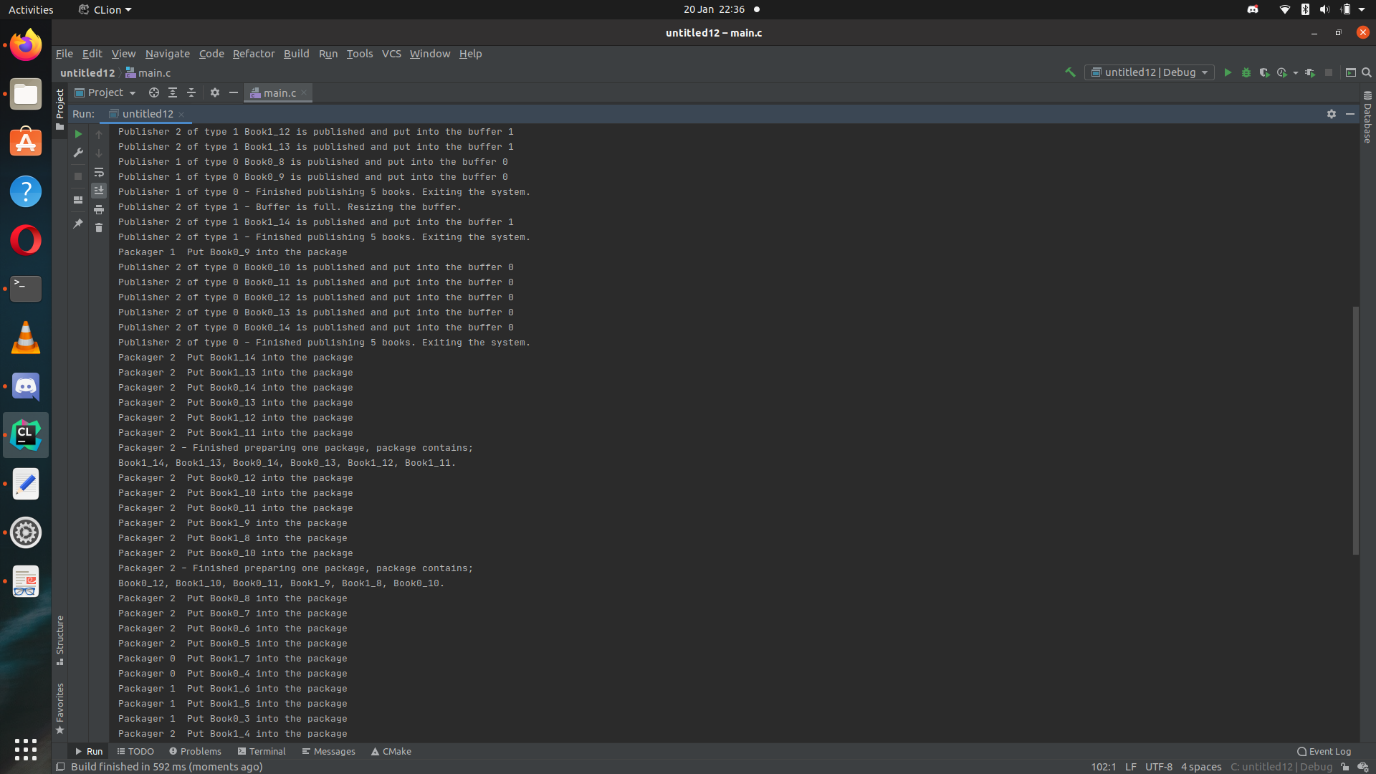
If we write input as in pdf file

-n 2 3 4 -b 5 -s 6 7

Output is as seen like below







If input is

-n 2 2 2 -b 2 -s 2 2

Then result is like below

