CSE 3033 Project 3 Report

Koray Emre ŞENEL : 150117037
Ahmet TURGUT : 150117046
Mehmet Etka UZUN: 150118504

Breafing :

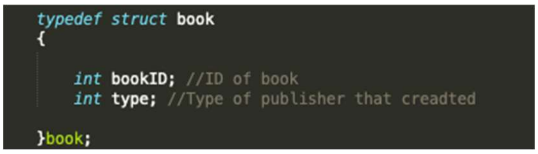
This project purpose is how to use threads and how to avoid critical section by using mutex. This project is about publisher and packager problem. We solve this problem by using two different threads.

- **Packager Threads :** It take book from buffer and put it in its package when package is full it prints books on screen.
- **Publisher Thread:** It publishe the book and put it in buffer.

We have some conditions for these threads. First one is when one publisher publishe book the other publisher is waiting until publisher finish the job. Second one is when packager take book on the buffer, other packager and publisher are waiting until packager finish the job.

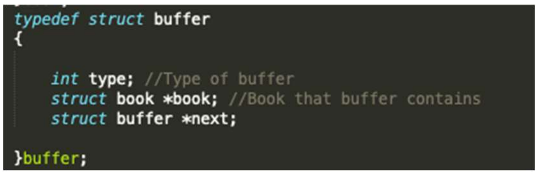
Algorithm:

- **Structs**
  - ➔ **Book:** it holds book id and type of buffer.



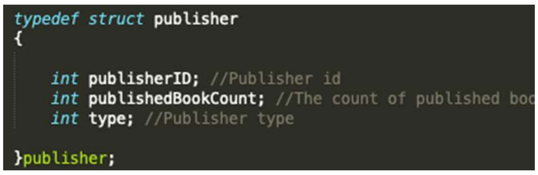
```
typedef struct book
{

    int bookID; //ID of book
    int type; //Type of publisher that creadted

}book;
```

  - ➔ **Buffer:** it holds type of buffer and one book like a shelf.



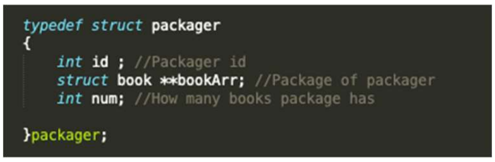
```
typedef struct buffer
{

    int type; //Type of buffer
    struct book *book; //Book that buffer contains
    struct buffer *next;

}buffer;
```

  - ➔ **Publisher:** it holds type of publisher, publisher id and count of published book.



```
typedef struct publisher
{

    int publisherID; //Publisher id
    int publishedBookCount; //The count of published boo
    int type; //Publisher type

}publisher;
```

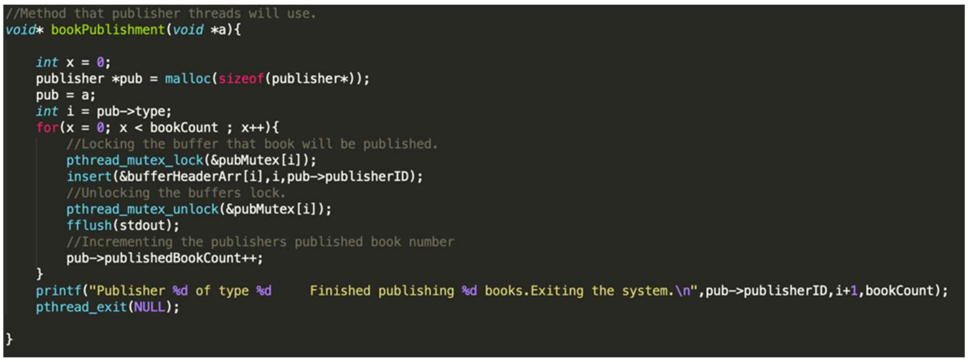
  - ➔ **Packager:** it holds packager id, package of packager and how many books pakage has.



```
typedef struct packager
{
    int id ; //Packager id
    struct book **bookArr; //Package of packager
    int num; //How many books package has

}packager;
```

- **Threads Methods**

bookPublishment: This method take one parameter which is publisher struct because we use mutex in this method according to type of the publisher. Why we are using mutex in this method because this method is also used by other Publisher and they have commen variable, so this cause a critical section.Why mutex is depend on type of publisher because we want to work two different type publisher concurrently. Also if buffer size is not enough for put new book we doubling the size.

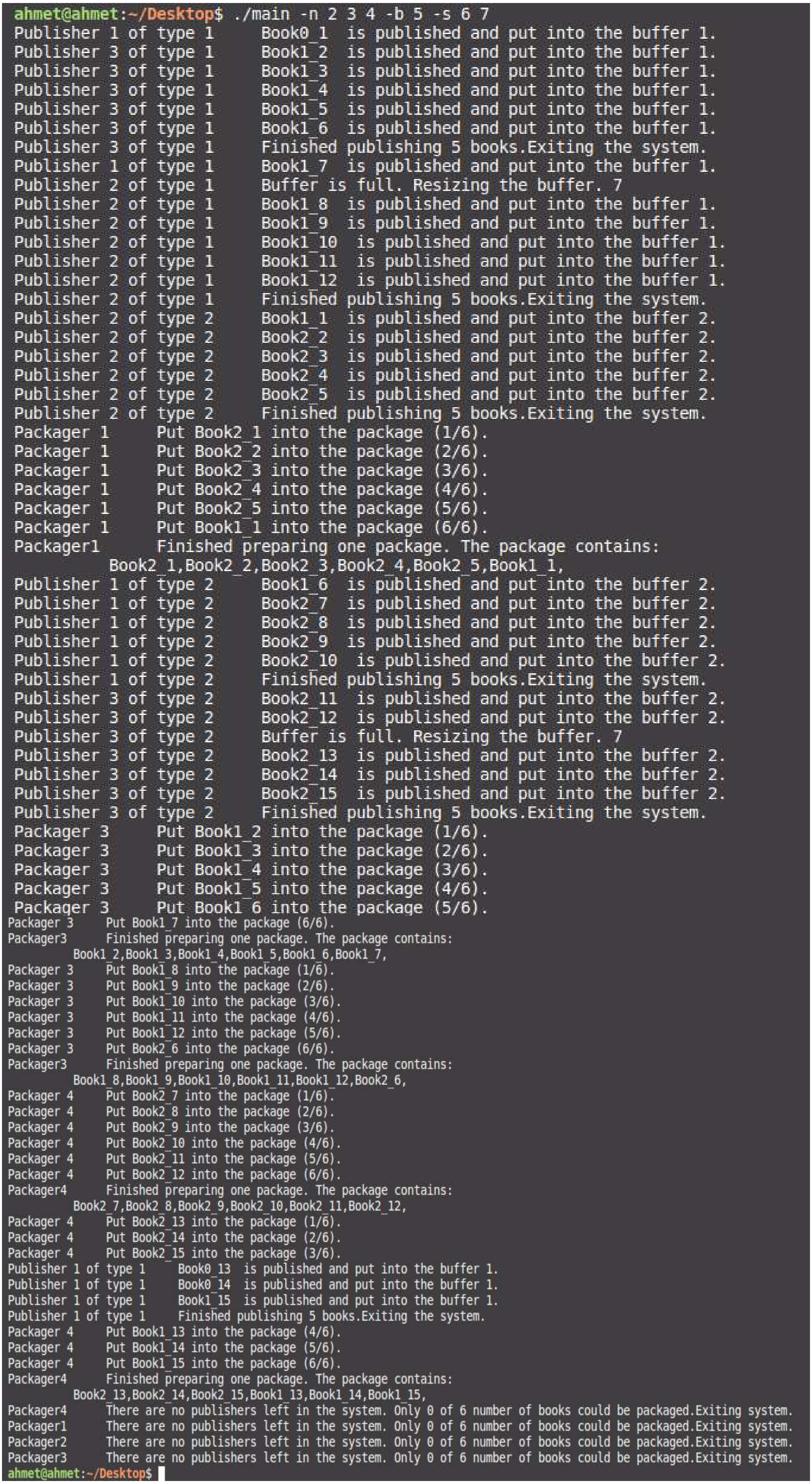


```
//Method that publisher threads will use.
void* bookPublishment(void *a){

    int x = 0;
    publisher *pub = malloc(sizeof(publisher*));
    pub = a;
    int i = pub->type;
    for(x = 0; x < bookCount ; x++){
        //Locking the buffer that book will be published.
        pthread_mutex_lock(&pubMutex[i]);
        insert(&bufferHeaderArr[i],i,pub->publisherID);
        //Unlocking the buffers lock.
        pthread_mutex_unlock(&pubMutex[i]);
        fflush(stdout);
        //Incrementing the publishers published book number
        pub->publishedBookCount++;
    }
    printf("Publisher %d of type %d     Finished publishing %d books.Exiting the system.\n",pub->publisherID,i+1,bookCount);
    pthread_exit(NULL);

}
```

bookPackagment: This method take one parameter which is packager struct because we want to know whic packager works in this method. We use mutex in this method too since we have same problem in this method like other one. This method select randomly buffer type by using rand(). Also this method have some conditions like checking buffer if it is empty or not and checking if the publishers published all the books and they are packaged. Main aim of this method is take the book in buffer and put it own package and if package is full it prints the books on screen.

- **Other methods**

Insert: This method is different version of normal linked list insert. We increase count every next operation in linked list when count equals the buffer size we doubling the buffer size. The buffer size is stored in global array.



```
ahmet@ahmet:~/Desktop$ ./main -n 2 3 4 -b 5 -s 6 7
Publisher 1 of type 1     Book0_1  is published and put into the buffer 1.
Publisher 3 of type 1     Book1_2  is published and put into the buffer 1.
Publisher 3 of type 1     Book1_3  is published and put into the buffer 1.
Publisher 3 of type 1     Book1_4  is published and put into the buffer 1.
Publisher 3 of type 1     Book1_5  is published and put into the buffer 1.
Publisher 3 of type 1     Book1_6  is published and put into the buffer 1.
Publisher 3 of type 1     Finished publishing 5 books.Exiting the system.
Publisher 1 of type 1     Book1_7  is published and put into the buffer 1.
Publisher 2 of type 1     Buffer is full. Resizing the buffer. 7
Publisher 2 of type 1     Book1_8  is published and put into the buffer 1.
Publisher 2 of type 1     Book1_9  is published and put into the buffer 1.
Publisher 2 of type 1     Book1_10  is published and put into the buffer 1.
Publisher 2 of type 1     Book1_11  is published and put into the buffer 1.
Publisher 2 of type 1     Book1_12  is published and put into the buffer 1.
Publisher 2 of type 1     Finished publishing 5 books.Exiting the system.
Publisher 2 of type 2     Book1_1  is published and put into the buffer 2.
Publisher 2 of type 2     Book2_2  is published and put into the buffer 2.
Publisher 2 of type 2     Book2_3  is published and put into the buffer 2.
Publisher 2 of type 2     Book2_4  is published and put into the buffer 2.
Publisher 2 of type 2     Book2_5  is published and put into the buffer 2.
Publisher 2 of type 2     Finished publishing 5 books.Exiting the system.
Packager 1     Put Book2_1 into the package (1/6).
Packager 1     Put Book2_2 into the package (2/6).
Packager 1     Put Book2_3 into the package (3/6).
Packager 1     Put Book2_4 into the package (4/6).
Packager 1     Put Book2_5 into the package (5/6).
Packager 1     Put Book1_1 into the package (6/6).
Packager1      Finished preparing one package. The package contains:
          Book2_1,Book2_2,Book2_3,Book2_4,Book2_5,Book1_1,
Publisher 1 of type 2     Book1_6  is published and put into the buffer 2.
Publisher 1 of type 2     Book2_7  is published and put into the buffer 2.
Publisher 1 of type 2     Book2_8  is published and put into the buffer 2.
Publisher 1 of type 2     Book2_9  is published and put into the buffer 2.
Publisher 1 of type 2     Book2_10  is published and put into the buffer 2.
Publisher 1 of type 2     Finished publishing 5 books.Exiting the system.
Publisher 3 of type 2     Book2_11  is published and put into the buffer 2.
Publisher 3 of type 2     Book2_12  is published and put into the buffer 2.
Publisher 3 of type 2     Buffer is full. Resizing the buffer. 7
Publisher 3 of type 2     Book2_13  is published and put into the buffer 2.
Publisher 3 of type 2     Book2_14  is published and put into the buffer 2.
Publisher 3 of type 2     Book2_15  is published and put into the buffer 2.
Publisher 3 of type 2     Finished publishing 5 books.Exiting the system.
Packager 3     Put Book1_2 into the package (1/6).
Packager 3     Put Book1_3 into the package (2/6).
Packager 3     Put Book1_4 into the package (3/6).
Packager 3     Put Book1_5 into the package (4/6).
Packager 3     Put Book1_6 into the package (5/6).
Packager 3     Put Book1_7 into the package (6/6).
Packager3      Finished preparing one package. The package contains:
          Book1_2,Book1_3,Book1_4,Book1_5,Book1_6,Book1_7,
Packager 3     Put Book1_8 into the package (1/6).
Packager 3     Put Book1_9 into the package (2/6).
Packager 3     Put Book1_10 into the package (3/6).
Packager 3     Put Book1_11 into the package (4/6).
Packager 3     Put Book1_12 into the package (5/6).
Packager 3     Put Book2_6 into the package (6/6).
Packager3      Finished preparing one package. The package contains:
          Book1_8,Book1_9,Book1_10,Book1_11,Book1_12,Book2_6,
Packager 4     Put Book2_7 into the package (1/6).
Packager 4     Put Book2_8 into the package (2/6).
Packager 4     Put Book2_9 into the package (3/6).
Packager 4     Put Book2_10 into the package (4/6).
Packager 4     Put Book2_11 into the package (5/6).
Packager 4     Put Book2_12 into the package (6/6).
Packager4      Finished preparing one package. The package contains:
          Book2_7,Book2_8,Book2_9,Book2_10,Book2_11,Book2_12,
Packager 4     Put Book2_13 into the package (1/6).
Packager 4     Put Book2_14 into the package (2/6).
Packager 4     Put Book2_15 into the package (3/6).
Publisher 1 of type 1     Book0_13  is published and put into the buffer 1.
Publisher 1 of type 1     Book0_14  is published and put into the buffer 1.
Publisher 1 of type 1     Book1_15  is published and put into the buffer 1.
Publisher 1 of type 1     Finished publishing 5 books.Exiting the system.
Packager 4     Put Book1_13 into the package (4/6).
Packager 4     Put Book1_14 into the package (5/6).
Packager 4     Put Book1_15 into the package (6/6).
Packager4      Finished preparing one package. The package contains:
          Book2_13,Book2_14,Book2_15,Book1_13,Book1_14,Book1_15,
Packager4      There are no publishers left in the system. Only 0 of 6 number of books could be packaged.Exiting system.
Packager1      There are no publishers left in the system. Only 0 of 6 number of books could be packaged.Exiting system.
Packager2      There are no publishers left in the system. Only 0 of 6 number of books could be packaged.Exiting system.
Packager3      There are no publishers left in the system. Only 0 of 6 number of books could be packaged.Exiting system.
ahmet@ahmet:~/Desktop$
```