Assignment 05 - Threads

Question 1 - Multi-threaded file conversion - Unlimited number of threads

Download the program `upper.c` which converts to uppercase the entire content of a text file whose name is passed as an argument. Write a program `multi-upper.c` that adapts `upper.c` so that it accepts a list of several file names as argument. The program must create a thread for each filename, and each thread concurrently performs the conversion on the corresponding text file.

For example, the following command:

```
$ bin/multi-upper fname1 fname2 fname3 fname4
```

shall generate 4 files, each with the suffix `.UPPER.txt`, and each containing the content of their respective text source in capital letters.

Question 2 - Multi-threaded file conversion - Fixed pool of threads

Now consider that the main thread only creates a fixed number `NB_THREAD` of secondary threads at startup. `NB_THREAD` can be smaller than the number of files to process. Immediately after converting a file, a thread must check whether there still are files to process. If such is the case, the thread takes on the conversion of one of the remaining files; otherwise it ends.

Question 3 - Bounded buffer with threads

Write a program `bounded-buffer` that creates `NP` producers and `NC` consumers.

All threads exchange data via an array with a maximum capacity of `MAX_CAP` integer values. Access to the array is LIFO: producers push random values `0 < v ≤ 100` on top of the previous values, while consumers pop values from the top and display them. The program should stop when a total of `NVAL` values have been consumed.