

# BLG312E – Computer Operating Systems

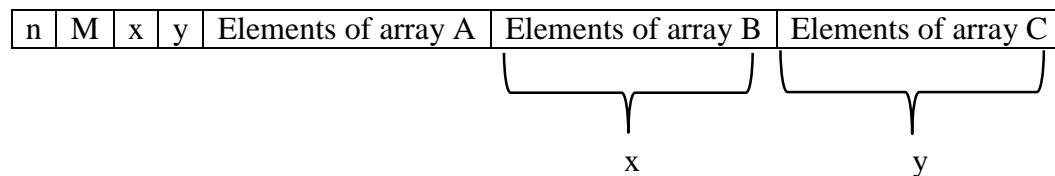
## Assignment 2

**Submission Deadline: 05.05.2021**

You should implement a program that dissects an unordered array  $A[1, \dots, n]$ . The program will first create two sub-arrays (namely B and C), and copy the elements of A into these subarrays based on their values. If the value of an elements is smaller or equal to “M”, it will be copied into B; if not into C, as follows<sup>1</sup>:

$$\begin{aligned} \forall a \in A; \\ (a \leq M \rightarrow a \in B) \wedge (a \in B \rightarrow a \leq M) \\ (a > M \rightarrow a \in C) \wedge (a \in C \rightarrow a > M) \end{aligned}$$

This job will be handled in parallel by two children processes, which are created by parent/main process of your program. The memory space shared among these processes is organized as follows:



where n states the size of array A; x and y state the sizes of arrays B and C respectively.

First, the parent process will read the values n, M and values of array A from the given input file and initiate the required tasks to allocate shared memory space as given above. Then, it (the parent process) will create two children processes. The children processes will be responsible of copying the values from array A into B or C. In this manner, the first child fills array B, whereas the second one fills array C.

As mentioned above, x represents the size of array B and y represents the the size of array C. Child 1 will first go through array A, after fetching values n and M, in order to determine the number of elements for sub-array B (x). Similary, child 2 will do the same in order to determine the size of C (y). Each of the children processes writes the value they determined to the corresponding place in the shared memory.

After value for x is determined, Child 1 will start travesing array A again, this time copying the elements, which are smaller or equal to M from A into array B, which **is directly located after array A** on the shared memory space. Child 2 will follow a similar set of actions in order to fill out array C. Since, array C **is directly located after array B**, thus child 2 should wait until value

---

<sup>1</sup> In addition;  $B \cap C = \emptyset \wedge B \cup C = A \wedge n = x + y$

for  $x$  is known. Your program should ensure necessary **synchronization**. With value of  $x$ , child 2 may start filling out array C, beginning from the address  $\langle \mathbf{A} + \mathbf{n} + \mathbf{x} \rangle$  in the shared space.

After children processes complete their tasks, the parent process will print out value  $M$  and the sizes of arrays A, B and C before terminating itself. Please, implement the program described above by using Unix system calls and semaphores.

Your program must accept two command line arguments. First one is input file and second one is output file. If these arguments are not provided, default arguments are *input.txt* and *output.txt* respectively.

#### **Input File Format:**

$M$

$n$

$A_0 \ A_1 \ \dots \ A_{n-1}$

#### **Output File Format:**

$M$

$n$

$A_0 \ A_1 \ \dots \ A_{n-1}$

$x$

$B_0 \ B_1 \ \dots \ B_{x-1}$

$y$

$C_0 \ C_1 \ \dots \ C_{y-1}$