Topic: PThread programming

Due on 09-Nov-2020, 04:15pm

Late Submission 10-Nov-2020, 04:15pm (Please note that assignments submitted late will not fetch full marks. Hence you are advised to submit your assignments by due date and time)

Mode of submission: MS Teams

Instructions:

- You are required to upload the individual source (.c) files and one output file.
- You are required to follow proper naming convention for the files: ddmmyy118CS****Q#.c.
- Apart from the source files you are also required to upload an output file containing the outputs of execution of all the assignments. The file name of the output file must be in the form: ddmmyy118CS****Output.docx or .pdf
- The submitted programs will be checked for similarity through turnitin, before evaluation. So it is advisable not to borrow code from any source.

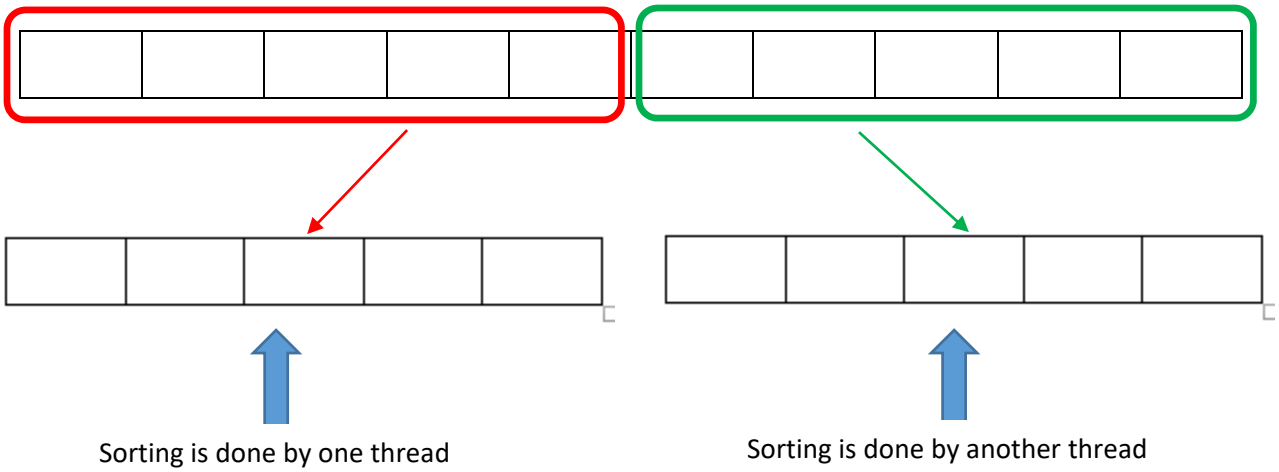
1. **Peterson's 2 process synchronization solution**
   Create two threads and implement Peterson's synchronization solution. Use appropriate delay loop or sleep() to ensure context switch. Verify Mutual Exclusion, Progress and Bounded Waiting.

P_1	P_2
do{	do{

flag[1] = TRUE;
turn = 2;
while(flag[2] && turn==2);
	flag[2] = TRUE;
	turn = 1;
	while(flag[1] && turn==1);

critical section	critical section
flag[1] = FALSE;	flag[2] = FALSE;
reminder section	reminder section
} while(TRUE)	} while(TRUE)

2. **Merge sort using threads**



In every stage, the parent thread divides the array into two halves and creates two threads: One thread sorts the left half and other thread sorts the second half. The parent thread waits for both and then performs the merge operation. Each thread repeats the same until one element in the array.