Operating Systems

Programming Assignment #3

[You will submit a zip file containing your answer to the questions below as pdf as well as the requested source code]

---

1. You are provided with two files: q1.c and assignment.h
Compile with:
gcc -o q1 -std=c99 q1.c -lpthread
The file assignment.h must be in the same directory as q1.c
Read the source code of q1.c then execute the code with ./q1

a) How many threads are created when you execute the code? Justify.
b) If we create four threads a, b, c, and d, in that order, do we need to call join also in the same order as the creation (i.e. join for a, then b, then c, and then d)?
c) There is something wrong with the output. What is it? And what is the correct answer?
d) If we keep executing this code several times, is there a chance that one of the executions produces the correct result? Explain.
e) By looking at the code, where is the bug?
f) How to fix this bug you mentioned in question e) above while keeping the total number of threads fixed?

---

2. Implement the bug fix you described in question 1 above and call the new file q2.c
You will attach this file to your submission.

---

3. You are also provided with file q3.c compile it with:
gcc -o q3 -std=c99 q3.c -lpthread
The file assignment.h must be in the same directory as q3.c
Read the source code and execute the program with ./q3

a) How many threads are created when you execute the program? Justify
b) Is the code correct and the output as expected? Justify your answer.
c) From a performance perspective, the code in q3.c is not very efficient. Why?

---

4. Modify the code in q3.c to make more efficient, using condition variables and attach the code as q4.c
Your solution must keep the number of threads in q3.c the same.

---

5. Suppose we have a single CPU. We run a multithreaded process on that CPU. If two threads from that process want to update a shared variable, do we still have to use locks? At the end, only one thread will be running at a time. Justify your answer.

6. Do condition variables have to be used always with locks? Justify.