

Homework Assignment 5

ECE 151 / CMPSC 171 - Spring 2014

Due Monday, May 12

1. In the examples below, the operations are performed by the same process on two different local copies (L1 and L2) of the same data store.

(a) Does the following example satisfy **monotonic reads consistency**? Explain why or why not.

(b) Does the following example satisfy **monotonic writes consistency**? Explain why or why not.

L1	W(x ₁)			WS(x ₁ ;x ₃)
L2		WS(x ₁ ;x ₂)	R(x ₂)	R(x ₁)

(c) Does the following example satisfy **read-your-writes consistency**? Explain why or why not.

(d) Does the following example satisfy **writes-follow-reads consistency**? Explain why or why not.

L1		WS(x ₁ ;x ₂)	R(x ₂)
L2	W(x ₁)		WS(x ₂ ;x ₃)

2. (a) Explain the difference between a **primary-backup remote-write** and a **primary-backup local-write** protocol in terms of whether the data or the operations are sent from one replica to another and when that is done.

(b) Which of the two protocols requires the backups to perform the write operations before the client initiating the write operation is informed that the writes are completed? How does this affect the response time seen by the initiating client and what the client can do?

3. Consider a k **fault-tolerant group** containing seven process replicas.
- (a) If clocks are **synchronized**, how many faults can be tolerated if the replicas are subject to **crash** faults? Justify your answer.
 - (b) If clocks are **synchronized**, how many faults can be tolerated if the replicas are subject to **arbitrary (Byzantine)** faults? Justify your answer.
 - (c) If clocks are **not synchronized**, how many faults can be tolerated if the replicas are subject to **crash** faults? Justify your answer.
 - (d) If clocks are **not synchronized**, how many faults can be tolerated if the replicas are subject to **arbitrary (Byzantine)** faults? Justify your answer.
4. To protect **Remote Procedure Call (RPC)** against server crashes, you need to decide whether to support at-most-once semantics or at-least-once semantics for the application.
- (a) Give an example application in which you would choose **at-most-once** semantics and explain why.
 - (b) To protect RPC against server crashes, what additional mechanism(s) would you use to provide **at-most-once** semantics?
 - (c) Give an example application in which you would choose **at-least-once** semantics and explain why.
 - (d) To protect RPC against server crashes, what additional mechanism(s) would you use to provide **at-least-once** semantics?