Assignment 1

Reading – Read Chapter 1 of the textbook and review the slides that were presented in the Jan. 3 lecture.

Problem 1
This problem is designed to help you think about implications of the ACID properties on the internal structure and behavior of data management software. It will also help you think about some design considerations that may affect your project. It involves the implementation of a database system, called *Array*, on top of a sequential file system, called *SFS*.

In what follows, we use SFS[k] to denote the word at offset k in SFS and MainMem[k] to denote the word at offset k in main memory. SFS supports two operations:

- SFSread(Offset, Length, MemoryAddr) – Read SFS[Offset] through SFS[Offset+Length] into MainMem[MemoryAddr] through MainMem[MemoryAddr+Length].
- SFSwrite(Offset, Length, MemoryAddr) – Write MainMem[MemoryAddr] through MainMem[MemoryAddr+Length] into SFS[Offset] through SFS[Offset+Length] and return a “success” or “failure” status. If SFSwrite returns a success status, then you can be sure the content of SFS contains the data that was written. If it returns “failure”, then the content of SFS[Offset] through SFS[Offset+Length] is undefined (i.e., it could be anything).

The Array database system offers access to a database consisting of an array of N words. It supports the following operations:

- Read(d, m) – Read word d from the database into main memory location m.
- Write(d, m) – Write the value of main memory location m into word d of Array.
- Start(t) - Start a new transaction and return a transaction identifier t > 0.
- Commit(t) - commit transaction t
- Abort(t) - abort transaction t

For a database of size N, the implementation of Array uses 2N+2 storage locations, as follows:

- SFS[0] and SFS[N+1] each contain a transaction id.
- SFS[1] to SFS[N] contains a copy of the Array database. SFS[N+2] to SFS[2N+1] contains a second copy of the database.

It also uses the following global variables: Active, NextT, NextStore.

Array uses the following start-up procedure.

```
1. Active = 0
2. Read SFS[0] and SFS[N+1]
3. If SFS[0] > SFS[N+1], then
           NextT = SFS[0]+1
           NextStore = N+1
           read SFS[1] to SFS[N] into main memory using SFSread
   Otherwise,
           NextT = SFS[N+1]+1
           NextStore = 0
           read SFS[N+2] to SFS[2N+1] into main memory using SFSread
```

Array implements Read and Write by reading and writing the main memory copy of the Array database.

Array implements Start(t) as follows:

```
1. If Active =1, then return an exception
2. Active =1
3. Return NextT
```

Array implements Commit(t) as follows:

```
1. Use SFSwrite to write the content of the database into SFS[NextStore+1] to SFS[NextStore+N].
2. If SFSwrite returns a success status, then
       Use SFSwrite to set SFS[NextStore] = NextT
       If NextStore = 0 then NextStore=N+1 else NextStore = 0
       NextT = NextT + 1
       Active = 0,
       Return “success.”
   Otherwise (SFS returns a failure status) invoke abort and return “failure.”
```

Array implements Abort(t) by running the start-up procedure.

a. Explain whether the implementation of Array satisfies each of the ACID properties.

b. Suppose we use a low-priced version of SFS whose implementation of SFSwrite can return success even if some of the data to be written didn’t get stored into SFS. How does this affect your answer to (a)? Suggest how to improve the implementation of Array to fix any problems that arise.

c. How does the following implementation of Commit affect your answer to (a)?

```
1. Use SFSwrite to set SFS[NextStore] = NextT
2. Use SFSwrite to write the content of the database into SFS[NextStore+1] to SFS[NextStore+N]
3. If SFSwrite returns a success status, then
       If NextStore = 0 then NextStore=N+1 else NextStore = 0
       NextT = NextT + 1
       Active = 0,
       Return “success.”
   Otherwise (SFS returns a failure status) invoke abort and return “failure.”
```

d. Suppose we modify the implementation of Start to allow two transactions to be active at the same time. How does this affect your answer to (a)?

Problem 2

In this problem, we will explore the different styles of system that may be relevant for a business-to-consumer e-commerce application. Refer to Sections 1.2 and 1.6 of the lecture slides and Section 1.7 of the textbook. For each of the following actions, is it important that it run as an ACID transaction? Explain why or why not.

a. Look up a catalog entry for a given product, to get the part number and price.

b. Accept an order for a given product, provided that it’s in stock.

c. Display an advertising banner on the customer’s screen.

d. Get a list of the most popular models of a given type of product.

e. Display a summary of the customer’s purchases, by month, for the past two years.

f. Print the packing slip for a given order.

g. When a shipment arrives from a supplier, process all the back orders for products in that shipment.