

Assignment 1 CSEP 545

```
• /* Start new transaction */
int start()
Returns: -1, if there is an active transaction
tId > 0 (a new Transaction Identifier) otherwise
• /* If possible, read block at blockAddr on behalf of transaction tId
from disk into main memory and save a pointer to it in block */
int read(int blockAddr, int tId, int *block)
Returns: 0 if successful, -1 otherwise
• /* Tell B1 that transaction tId has updated a block at blockAddr */
int write(int blockAddr, int tId)
Returns: 0 if there is no cached entry for blockAddr
1 otherwise
• /* Try to commit transaction tId */
int commit(int tId)
Returns: 0 if successful, -1 otherwise
• /* Abort transaction tId */
int abort(int tId)
Returns: 0
```

1/11/05

Start

```
lastTrans = 1;
...
int start()
{
    if (lastTrans < 0) return -1
    else
    {
        lastTrans = -(++lastTrans);
        return lastTrans;
    }
}
```

1/11/05

Read

```
int read(int blockAddr, int tId, int *block)
{
    if (there is such a cache element e)
    {
        Cache(e).tId = tId;
        block = &Cache(e).newBlock;
        return 0;
    }
    else
    {
        /* pick a cache entry e, where Cache(e).tId = 0.
        If there is no such entry, then return -1 */
        diskRead(blockAddr, &Cache(e).oldBlock);
        Cache(e).newBlock = Cache(e).oldBlock;
        block = &Cache(e).newBlock;
        Cache(e).tId = tId;
        return 0;
    }
}
}
```

1/11/05

Write

```
/* A transaction should call
write(&Cache(e).newBlock, tId)
after it updates Cache(e).newBlock. */
int write(int blockAddr, int tId)
{
    /* find the cache entry e for block blockAddr */
    if (there is no such entry) return 0
    else
    {
        Cache(e).tId = tId;
        return 1;
    }
}
```

1/11/05

Commit

```
int commit(int tId)
{
    for (each cache entry e where Cache(e).tId == tId)
    {
        status = diskWrite(Cache(e).blockAddr, &Cache(e).newBlock);
        Cache(e).tId = -tId;
        if (status == -1)
        {
            Abort(tId);
            return -1;
        }
    }
    for (all cache entries e) set Cache(e).tId = 0;
    lastTrans = -lastTrans;
}
return 0;
}
```

1/11/05

Abort

```
int abort(int tId)
{
    for (all cache entries e, where Cache(e).tId == -tId)
    {
        repeat
        {
            status = diskWrite(Cache(e).blockAddr, &Cache(e).oldBlock)
        } until (status == 0)
    }
    for (all cache entries e)
    {
        Cache(e).tId = 0
    }
    lastTrans = -lastTrans;
    return 0;
}
```

1/11/05

Questions

For each of the scenarios below, analyze whether transactions that use B1 satisfy the four ACID properties.

- The OS and B1 never crash. Transactions run serially.
- The operating system can crash, in which case main memory is lost. Transactions run serially.
- The operating system and B1 never crash. Transactions run serially. But if `disk_write(b, add)` returns 0 then it might have computed block b on disk. Variation of (c): If `disk_write(b, add)` returns -1 then it might have computed block b on disk.
- The operating system and B1 never crash. Two transactions are allowed to run concurrently.

1/11/05