1. The reverse to the pseudocode of Figure 4.10 would be to in the else statement first unlock the mutex.spinlock, and then remove the current thread from the runnable threads. At first glance, this appears to be more efficient since it ends the spinlock faster, which frees up the processor for other threads. The issue, however, is that once you unlock the mutex.spinlock, the thread that is running isn't removed from the waitlist. Thus, this thread can edit the mutex and keep adding itself to the queue because the mutex hasn't removed it from the runnable thread list yet. Mutexes should only be unlocked when the thread is done with it.

2. This would be a bug because the integer "seatsRemaining" and the integer "cashOnHand" would be using different 'snapshots' of the current state. So if someone buys a ticket right in between the two lines, they would output uncoordinated values.

3. JAVA in Github

4. JAVA in Github

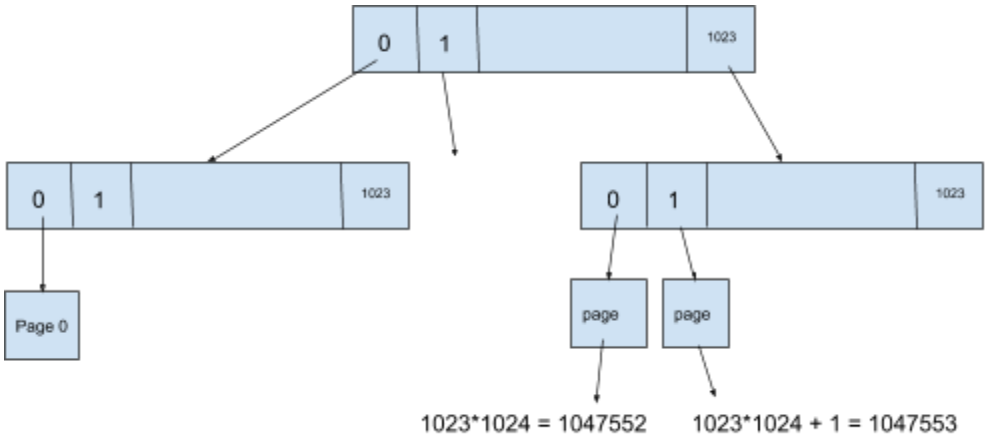
5. Transactions: A transaction is a **unit of computational work** for which **no intermediate state** from the middle of the computation is ever visible. Concurrent transactions are **isolated** from seeing each other's intermediate storage. Additionally, if a **transaction should fail**, the **storage will be left as it was before the transaction started**. Even if the computer system should catastrophically crash in the middle of a transaction's execution, the **storage after rebooting will not reflect the partial transaction**. This **prevents the results of a half-completed transaction** from becoming visible.
    a. Assuming **two-phase locking**, it is not possible for T2 to see the old value of x but the new value of y, since the transactions execute concurrently, meaning it cannot move to the next step before finishing the first one. You cannot write and read at the same time in two-phase locking. With two-phase locking, the system moves from one consistent state to another so a read call will always return the most recent value.
    b. Assuming **read committed isolation level is used and implemented with short read locks**, T2 cannot see the old value of x and new value of y. If there is a write lock nobody can read it. If T1 has a write lock T2 can't read. It is assumed no one can read it or else we would get a dirty read.
    c. Assuming **snapshot isolation**, it is possible for T2 to see the old value of x but the new value of y. Snapshot is not write locked. The two threads are stored in different locations so they both need to update the original storage location of the

value for their not to be inconsistencies. If T2 is reading x whileT1 finishes writing y T2 will read the old value of x and the new value of y.

6. The virtual addresses of the first 4-byte word in page 6 are 24576(first) and 28668(last). They translate to the physical addresses 12288(first) and 16380(last) .

7. Figure 6.13 represents a multi-layered page table, so there is a page directory which maps to the page table which then maps to the pages.

   A page directory is split into page tables. Each page table has 1024 pages within itself. The number 1047552 came from 1023*12024 aka the first page of the 1024 chunk. 1047553 is 1023*1024 + 1 aka the next (second) page of the 1024 chunk.




8. Report:
   Step 1: Loop a number between 4,096 and 100,000. This will be the range of array sizes that we will test.
   Step 2: Within this loop, create an array of the specified size. Fill the array with random numbers.
   Step 3: Fetch the value for the index 4096
   Step 4: Determine the time that it took to fetch this value
   Step 5: Print the array size, time to access, and value at the index for all the times that the time to access was greater than or equal to .000010 seconds (the majority of indexes have a time to access of .000001 or .000002 so a time of access of .000010 is considerably long). The results are seen below(filtering for times greater than or equal to 10):

   Array Size: 4931 |Time to Access: 0.000014| Value at Index 4096: 75
   Array Size: 5554 |Time to Access: 0.000016| Value at Index 4096: 24

Array Size: 5563 |Time to Access: 0.000010| Value at Index 4096: 75
Array Size: 5573 |Time to Access: 0.000015| Value at Index 4096: 48
Array Size: 5574 |Time to Access: 0.000016| Value at Index 4096: 71
Array Size: 5575 |Time to Access: 0.000017| Value at Index 4096: 81
Array Size: 5578 |Time to Access: 0.000010| Value at Index 4096: 93
Array Size: 5580 |Time to Access: 0.000018| Value at Index 4096: 29
Array Size: 6150 |Time to Access: 0.000019| Value at Index 4096: 11
Array Size: 6392 |Time to Access: 0.000020| Value at Index 4096: 98
Array Size: 6598 |Time to Access: 0.000013| Value at Index 4096: 86
Array Size: 6838 |Time to Access: 0.000017| Value at Index 4096: 94
Array Size: 14085 |Time to Access: 0.000017| Value at Index 4096: 35
Array Size: 27979 |Time to Access: 0.000011| Value at Index 4096: 55
Array Size: 27984 |Time to Access: 0.000015| Value at Index 4096: 60
Array Size: 28044 |Time to Access: 0.000012| Value at Index 4096: 50
Array Size: 28045 |Time to Access: 0.000012| Value at Index 4096: 61
Array Size: 28046 |Time to Access: 0.000011| Value at Index 4096: 54
Array Size: 28047 |Time to Access: 0.000010| Value at Index 4096: 65
Array Size: 28049 |Time to Access: 0.000010| Value at Index 4096: 10
Array Size: 28119 |Time to Access: 0.000010| Value at Index 4096: 43
Array Size: 28122 |Time to Access: 0.000011| Value at Index 4096: 19
Array Size: 28281 |Time to Access: 0.000022| Value at Index 4096: 18
Array Size: 28729 |Time to Access: 0.000014| Value at Index 4096: 5
Array Size: 29360 |Time to Access: 0.000010| Value at Index 4096: 98
Array Size: 29421 |Time to Access: 0.000012| Value at Index 4096: 51
Array Size: 29422 |Time to Access: 0.000010| Value at Index 4096: 12
Array Size: 29549 |Time to Access: 0.000011| Value at Index 4096: 79
Array Size: 29551 |Time to Access: 0.000013| Value at Index 4096: 36
Array Size: 29617 |Time to Access: 0.000013| Value at Index 4096: 45
Array Size: 32644 |Time to Access: 0.000010| Value at Index 4096: 52
Array Size: 32693 |Time to Access: 0.000012| Value at Index 4096: 21
Array Size: 32694 |Time to Access: 0.000011| Value at Index 4096: 47
Array Size: 32695 |Time to Access: 0.000010| Value at Index 4096: 76
Array Size: 32698 |Time to Access: 0.000010| Value at Index 4096: 7
Array Size: 32699 |Time to Access: 0.000010| Value at Index 4096: 30
Array Size: 43029 |Time to Access: 0.000016| Value at Index 4096: 79
Array Size: 43069 |Time to Access: 0.000013| Value at Index 4096: 39
Array Size: 43070 |Time to Access: 0.000012| Value at Index 4096: 93
Array Size: 43109 |Time to Access: 0.000011| Value at Index 4096: 46
Array Size: 43174 |Time to Access: 0.000011| Value at Index 4096: 76
Array Size: 43175 |Time to Access: 0.000010| Value at Index 4096: 14
Array Size: 43176 |Time to Access: 0.000013| Value at Index 4096: 100
Array Size: 43178 |Time to Access: 0.000013| Value at Index 4096: 91
Array Size: 49516 |Time to Access: 0.000033| Value at Index 4096: 95
Array Size: 49551 |Time to Access: 0.000012| Value at Index 4096: 36
Array Size: 49552 |Time to Access: 0.000015| Value at Index 4096: 94
Array Size: 49553 |Time to Access: 0.000012| Value at Index 4096: 36
Array Size: 49554 |Time to Access: 0.000020| Value at Index 4096: 32
Array Size: 49642 |Time to Access: 0.000026| Value at Index 4096: 78

Array Size: 49715 |Time to Access: 0.000011| Value at Index 4096: 30
Array Size: 49782 |Time to Access: 0.000020| Value at Index 4096: 69
Array Size: 51481 |Time to Access: 0.000017| Value at Index 4096: 42
Array Size: 51515 |Time to Access: 0.000012| Value at Index 4096: 7
Array Size: 51518 |Time to Access: 0.000014| Value at Index 4096: 68
Array Size: 51519 |Time to Access: 0.000012| Value at Index 4096: 64
Array Size: 51574 |Time to Access: 0.000010| Value at Index 4096: 35
Array Size: 51575 |Time to Access: 0.000011| Value at Index 4096: 4
Array Size: 51577 |Time to Access: 0.000013| Value at Index 4096: 86
Array Size: 51617 |Time to Access: 0.000011| Value at Index 4096: 66
Array Size: 57541 |Time to Access: 0.000014| Value at Index 4096: 50
Array Size: 57579 |Time to Access: 0.000013| Value at Index 4096: 100
Array Size: 57580 |Time to Access: 0.000012| Value at Index 4096: 11
Array Size: 57581 |Time to Access: 0.000011| Value at Index 4096: 8
Array Size: 57626 |Time to Access: 0.000012| Value at Index 4096: 66
Array Size: 57627 |Time to Access: 0.000012| Value at Index 4096: 38
Array Size: 57629 |Time to Access: 0.000010| Value at Index 4096: 62
Array Size: 59905 |Time to Access: 0.000037| Value at Index 4096: 79
Array Size: 59957 |Time to Access: 0.000011| Value at Index 4096: 91
Array Size: 59958 |Time to Access: 0.000011| Value at Index 4096: 14
Array Size: 59959 |Time to Access: 0.000016| Value at Index 4096: 12
Array Size: 60025 |Time to Access: 0.000011| Value at Index 4096: 18
Array Size: 60026 |Time to Access: 0.000014| Value at Index 4096: 66
Array Size: 60027 |Time to Access: 0.000013| Value at Index 4096: 91
Array Size: 60035 |Time to Access: 0.000014| Value at Index 4096: 85
Array Size: 60036 |Time to Access: 0.000015| Value at Index 4096: 43
Array Size: 60038 |Time to Access: 0.000011| Value at Index 4096: 48
Array Size: 60039 |Time to Access: 0.000011| Value at Index 4096: 58
Array Size: 63033 |Time to Access: 0.000014| Value at Index 4096: 80
Array Size: 63060 |Time to Access: 0.000011| Value at Index 4096: 67
Array Size: 63062 |Time to Access: 0.000011| Value at Index 4096: 80
Array Size: 63092 |Time to Access: 0.000010| Value at Index 4096: 13
Array Size: 63124 |Time to Access: 0.000011| Value at Index 4096: 87
Array Size: 74576 |Time to Access: 0.000022| Value at Index 4096: 39
Array Size: 74603 |Time to Access: 0.000015| Value at Index 4096: 71
Array Size: 74651 |Time to Access: 0.000013| Value at Index 4096: 93
Array Size: 74689 |Time to Access: 0.000010| Value at Index 4096: 29
Array Size: 74691 |Time to Access: 0.000014| Value at Index 4096: 49
Array Size: 88537 |Time to Access: 0.000015| Value at Index 4096: 27
Array Size: 88558 |Time to Access: 0.000012| Value at Index 4096: 19
Array Size: 88581 |Time to Access: 0.000011| Value at Index 4096: 80
Array Size: 88599 |Time to Access: 0.000014| Value at Index 4096: 31
Array Size: 88605 |Time to Access: 0.000010| Value at Index 4096: 16
Array Size: 88606 |Time to Access: 0.000013| Value at Index 4096: 33
Array Size: 91507 |Time to Access: 0.000030| Value at Index 4096: 16
Array Size: 91526 |Time to Access: 0.000011| Value at Index 4096: 31
Array Size: 91562 |Time to Access: 0.000011| Value at Index 4096: 53
Array Size: 91593 |Time to Access: 0.000016| Value at Index 4096: 52

Array Size: 99256 |Time to Access: 0.000012| Value at Index 4096: 60
Array Size: 99275 |Time to Access: 0.000010| Value at Index 4096: 80
Array Size: 99277 |Time to Access: 0.000012| Value at Index 4096: 27
Array Size: 99302 |Time to Access: 0.000010| Value at Index 4096: 6
Array Size: 99306 |Time to Access: 0.000013| Value at Index 4096: 2
Array Size: 99326 |Time to Access: 0.000011| Value at Index 4096: 42

9.