Fall 2015 Final Exam RTS

(答案卷)

Name:________________ Score:__________

ID:__________________

Read each question over carefully several times. Answer all questions in the space provided. The exam is two hours long. Total score = 103.

(1). Please define the following terminologies (15pts):

a. Wear Leveling (of Flash Memory)

③ *Erases should be evenly distributed over the blocks of flash memory because each block has a limited number of erasing counts.*

② ③ correct answer!

b. DC Erasing of a Programmed Cell (Hint: Program Disturb)

③ *Electrons might be tunneled from the floating gate to the control gate through interpoly oxide in all the programmed cells when another cell of the same word line is programmed.*

c. Atomocity in ACID

③ *All of nothing.*

d. Broadcast Commit

③ *When a transaction commits, it tells all the transactions that it conflicts with so that they abort.*

e. Backward Validation (in Optimistic Concurrency Control)

③ *The validation of a transaction is performed against concurrently executing transactions.*

(2) Please answer the following questions in disk scheduling. You might provide explanation to receive any credits. (17pts)

(a) Please explain the major components of the access time of a disk read request, i.e., the time between the time that a read request is issued and the time the data is received by the disk controller. (8pts)

(b) When the disk workload is very very light, which one of the SCAN and FIFO has the better response time? (3pts)

(c) Which one of the EDF, SCAN, and Shortest Seek Time Fist (STTF) might have starvation? (6pts)

② ② ② ②

Ans: (a) access time = queuing time + seek time + latency delay + transfer time; (b) FIFO because it does need to go across the disk surface. (c) EDF and STTF

∵ SCAN needs ③ ① ① 答錯 −2

+ \+

explanation ④

(3) For "a software automation strategy," there are three stages: (1) Capture the computational requirements of the application domain in terms of an appropriate model, (2) Translate requirements specifications into an instance of the domain-specific model for resource allocation analysis, (3) Solve the well-defined optimization problems to minimize chosen cost/risk criteria. Please answer the following three questions: (18pts).

(a) UML (Unified Modeling Language), mentioned in the class, is belonging to which part? (3pts)

(b) Please compare "Decomposition by Critical Timing Constraints" and "Decomposition by Centralizing Concurrency Control" in terms of Ease of Understanding, Ease of Modification, and Processor Speed Requirement. You must have explanation to receive any credits. (12pts)

(c) What does it mean "an execution trace F have a latency of K time units with respect to a timing constraint C=(c, p, d)"? (3pts)

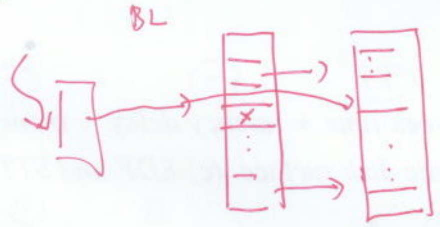
Ans: (A) The first ③ stage. (B) The "Ease of Understanding ④" and "Ease ④ of Modification" of "Decomposition by Centralizing Concurrency Control" are poor, but it needs less Processor Speed ④ Requirement. (C) F contains an execution of C in any time interval of length ≥ K. ③

(4) BL adopts a block-level address translation mechanism. For every write with a given LBA, it is written to the corresponding page of a corresponding physical block of the LBA. If the page is already used, then a new physical block is allocated so that all valid pages of the corresponding physical block are copied to the corresponding pages of the new physical block. Give me two problems in using BL? (8pts)

Ans: Low space utilization ④ and significant write ④ overeheads (or a lots of copyings and erasings)

如果只寫 → ⑥

extra



(5) Phase Change Memory (PCM) is bit-alterable non-volatile memory. Each cell of PCM can only survive over a limited number of writes. Suppose that PCM is used as the main memory, and Paging is used for memory management. When a page is requested for a page of the data segment of a process, what would be your choice in this page allocation? (6pts)

Ans: Give the page of the smallest (worst-case/average) number of writes to the data segment because the data segment might have writes in the future. ③

③ +

(6) Please explain why transaction executions with semaphore locking to protect every reading or writing of data can not guarantee (View/Conflict) Serializability (i.e., having the locking and unlocking of the corresponding semaphore before and after the data: Lock(X); Read(X); Unlock(X))? (8pts)

Ans: You can provide an example in executing T1: X=X-100, Y=Y+100) T2: R(X); R(Y)

That is: R1(X)W1(X) R2(X) R2(Y) R1(Y) W1(Y)

[illegible] serializable schedule. ⑧

(7) Consider real-time databases. What are the other two new consistency requirements then the Internal Consistency? Please also define them. (8pts)

② + ②

Ans: (1) Absolute/External Consistency: Data reflect the changings of the external environment; (2) Relative/Temporal Consistency: The ages of any two data read by a transaction are within a tolerable length of time ② + ②

(8) Please answer the following questions for real-time concurrency control. You must have explanation to receive any credits. (23pts)

(a) When many transactions try to lock the same data objects, shall we choose lock-based concurrency control or optimistic concurrency control? Why? (5pts)

(b) Please explain how RWPCP extends PCP in incorporating read and write locks. Why there is no deadlock? (10pts)

(c) Consider BAP presented in the class, where BAP is directly extended from PCP in which a high-priority transaction T_H aborts a low-priority transaction T_L when the low-priority transaction T_L blocks the high-priority transaction T_H in semaphore locking. Is it possible that two low-priority transactions T_{L1} and T_{L2} both blocks T_H under BAP and both get aborted by T_H? (8pts)

2 + 3

ANS: (a) Lock-based concurrency control is preferred because there would be too many restarts if optimistic concurrency control is adopted. (b) RWPCP uses 3 *absolute and write priority ceilings to implement read and write locks so that* 3 *readers still set a write priority ceiling to restrict subsequent low-priority readers from blocking writers, and absolute ceiling guarantees exclusive locks.* 4 *Deadlock-freeness can be proved by no transitive blocking and no wait for cycle of two transactions. (c) It is not possible with a proof by a contradiction: If both* T_{L1} and T_{L2} *blocks* T_H, *then each of* T_{L1} and T_{L2} must l*ock some semaphore that* 8 *can block* T_H *when* T_H *arrives. The problem is that which one of* T_{L1} and T_{L2} *locks such a semaphore successfully will block the other one from doing so.*