SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

CMPT 300: Operating Systems I

Assignment 4

Due July 31, 2018

POLICIES:

1. **Coverage**
   Chapters 10-15
2. **Grade**
   10 points, 100% counted into the final grade
3. **Individual or Group**
   Individual based, but group discussion is allowed and encouraged
4. **Academic Honesty**
   Violation of academic honesty may result in a penalty more severe than zero credit for an assignment, a test, and/or an exam.
5. **Submission**
   Electronic copy via CourSys
6. **Late Submission**
   2-point deduction for late submission within one week;
   5-point deduction for late submission over one week;
   Deduction ceases upon zero;
   Late submissions after the sample solution is available will NOT be graded.

QUESTIONS:

1. **1 point**
   Assume the following RAID 3 system uses ODD parity and disk2 fails.

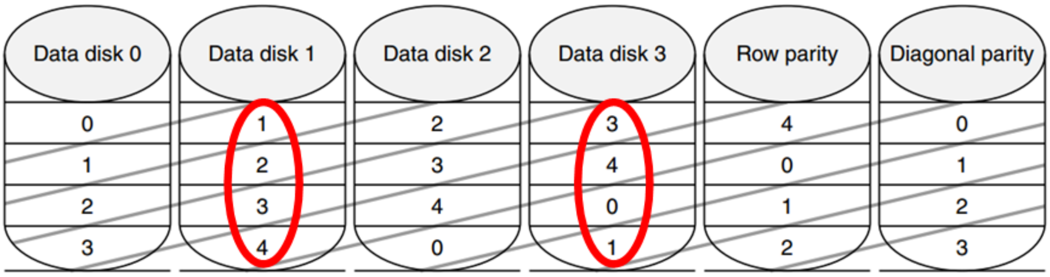
Disk 1	Disk 2 - Failed	Disk 3	Parity
1	0	1	1
0	0	0	1
1	1	1	0
0	1	0	0
0	0	0	1
0	0	0	1
1	1	1	0
1	0	1	1

a. Determine the original data on disk 2 and fill in the form.
b. Use the first two rows as example to explain how the original data is recovered.
[Grading Rubric: 1 point if both subquestions are correctly answered.]

a. As provided in the preceding form.

b. A simple way to derive the original data on disk 2 is counting the number of bit 1's. Since the system follows ODD parity, when all data items are available, the number of bit 1's in each row of the table should be an odd number. That is, in each row, the recovered missing item should make the number of bit 1's in its row to be an odd number.
First Row: The number of bit 1's on remaining disks is three, which is already an odd number. So the item to be recovered on disk 2 is 0, keeping the number of bit 1's on that row as an odd number.
Second Row: The number of bit 1's on remaining disks is one, which is already an odd number. So the item to be recovered on disk 2 is 0, keeping the number of bit 1's on that row as an odd number.
**Notes: It is also acceptable to derive a set of computation rules as those are covered in slides. This way is more complex though.

2. **1 point**
Provide a suitable recovery sequence for the illustrated RAID-DP (or row-diagonal parity) with disks 1 and 3 failed.



[Grading Rubric: 1 point if a correct recovery sequence is provided. 0 point otherwise.]

There might be more than one feasible recovery sequence. To be discussed in class.

3. **2 points**
Suppose that a disk drive has 5,000 cylinders, numbered through 0 to 4,999. The drive is currently serving a request at cylinder 2,150, and the previous request was at cylinder 1,805. The queue of pending requests, in FIFO order, is:
2,069, 1,212, 2,296, 2,800, 544, 1,618, 356, 1,523, 4,956, 3,681

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms?

a. FCFS
b. SSTF
c. SCAN
d. LOOK
e. C-SCAN
f. C-LOOK

[Grading Rubric: 2 points if ALL sixed distances are correctly calculated. 1 point if at least three distances are correctly calculated. 0 point otherwise.]

a. FCFS: 12993
Hint: service order
2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4956, 3681
Sum of movements per step, starting from 2150:
81 + 857 + 1084 + 504 + 2256 + 1074 + 1262 + 1167 + 3433 + 1275

b. SSTF: 7568
Hint: service order
2069, 2296, 2800, 3681, 4956, 1618, 1523, 1212, 544, 356
Sum of movements per step, starting from 2150:
81 + 227 + 504 + 881 + 1275 + 3338 + 95 + 311 + 668 + 188

c. SCAN: 7492
Hint: service order
2296, 2800, 3681, 4956, (4999), 2069, 1618, 1523, 1212, 544, 356
Sum of movements per step, starting from 2150:
146 + 504 + 881 + 1275 + 43 + 2930 + 451 + 95 + 311 + 668 + 188

d. LOOK: 7406
Hint: service order
2296, 2800, 3681, 4956, 2069, 1618, 1523, 1212, 544, 356
Sum of movements per step, starting from 2150:
146 + 504 + 881 + 1275 + 2887 + 451 + 95 + 311 + 668 + 188

e. C-SCAN: 9917
Hint: service order
2296, 2800, 3681, 4956, (4999), (0), 356, 544, 1212, 1523, 1618, 2069
Sum of movements per step, starting from 2150:
146 + 504 + 881 + 1275 + 43 + 4999 + 356 + 188 + 668 + 311 + 95 + 451

f. C-LOOK: 9119
Hint: service order

2296, 2800, 3681, 4956, 356, 544, 1212, 1523, 1618, 2069
Sum of movements per step, starting from 2150:
146 + 504 + 881 + 1275 + 4600 + 188 + 668 + 311 + 95 + 451

4. **1 point**
How does DMA increase system concurrency?
[Grading Rubric: Derivation of the correct result should be provided.]

For a device that does large transfers, such as a disk drive, it seems wasteful to use an expensive general-purpose processor to watch status bits and to feed data into a controller register one byte at a time. Many computers avoid burdening the main CPU with such I/O operations by offloading some of them to a special-purpose processor called a direct-memory-access (DMA) controller.

To initiate a DMA transfer, the host writes a DMA command block into memory. This block contains a pointer to the source of a transfer, a pointer to the destination of the transfer, and a count of the number of bytes to be transferred. The CPU writes the addresses of this command block to the DMA controller, then goes on with other work. The DMA controller proceeds to operate the memory bus directly, placing addresses on the bus to perform transfers without the help of the main CPU.
**Notes: The key point to be answered is the ability of DMA handling transfers without occupying CPU resources. Meanwhile, CPU can concurrently perform other tasks.

5. **1 point**
What is the purpose of using a "salt" along with the user-provided password?
[Grading Rubric: Both the limitation of password designs without salts and the benefit of using salts should be discussed.]

Key points:

The limitation of password designs without salts:
Without using salts, the system simply store the hash results of user passwords. It is possible that two different passwords generate the same hash result. This induces the probability of a forged password passing authentication.

The benefit of using salts:
The system can assign a random salt to each user. The salt value will be added to the password to ensure that if two plaintext passwords are the same, they still result in different hash values.

6. **2 points**
a. What is the key difference between symmetric encryption and asymmetric

encryption?
b. What is the key difference between a message-authentication code and a digital signature?
[Grading Rubric: 1 point per subquestion.]

a. Using symmetric encryption, two communicating entities need to use the same key that they previously agreed upon.
Using asymmetric encryption, each entity has a pair of keys, that is, a public key and a secret key. The public key is known to whichever entity to be communicated with. The secret key should be kept secret locally. When an entity sends a message to another entity, the sender encrypts the message using the secret key and then receiver decrypts the message using the sender's public key.

b. In a message-authentication code (MAC), a cryptographic checksum is generated from the message using a secret key. A MAC provides a way to securely authenticates short values. Both the sender and the receiver of a certain message should follow the same computation to generate MAC. That is, the secret key should be shared by both the sender and the receiver.
Digital signatures are very useful in that they enable anyone (knowing the public key of the sender) to verify the authenticity of the message. The sender, however, keeps the secret key secretly.

7. **2 points**
Consider a time-consuming authentication scenario where a database records all secret keys of a large number of users. When the system authenticates a user, it first issues a challenge message to the user. The user then uses his/her key to encrypt the challenge and then returns the encrypted challenge to the system. The system then encrypts the challenge using one key in the database after another and compares the result with the received encryption. Once a match is found, the system accepts the user. Otherwise, the user is denied. This authentication protocol surely takes a lot of time and computation.

Design a possible solution to speed up the authentication process.
[Grading Rubric: Open question, again! Time to convince the TA.
If you consider such questions interesting, join a research lab.
Finally, the last question of the last assignment of CMPT 300. Enjoy.]

Open question, to be discussed in class.
The performance improvement by the proposed solution need be justified.