COSC 6360 FINAL EXAMINATION MAY 12, 2004

This exam is **closed book**. You can have **one page** of notes. UH expels cheaters.

1. Answer the following questions in one or two sentences. (6×5 pts).

a) How does Elephant select ***landmark versions*** of a file?

b) What is the function of the ***i-map*** in a log-structured file system?

c) What is the main advantage of a ***stateless file server***?

d) What characterizes a ***Byzantine failure***?

e) Why did the authors of NFS add a ***client-side attribute cache*** to their original design?

f) What is the main advantage of ***optimistic replication***? (***Hint:*** *There are two possible answers.*)

Higher availability of the replicated data especially in the presence of network partitions.

2. What actions should be taken by a Kerberos system after it has detected one of the following security breaches and all the systems have been made secure again? (2×5 pts) How would these actions affect the end users of the system? (2×5 pts)

a) The security of the ticket granting server has been compromised.

Kerberos should __invalidate all tickets issued by its TGS and change all the keys it shares with other services ($K_{s,\ tgs}$)

____

The end user will have to __login again to obtain a new ticket __________________

b) The security of the Kerberos system has been compromised.

Kerberos should __shut down and require users to get new accounts with new passwords ____

The end user will have to __ get a new account with a new password _____________

____

3. What does ***close-to-open consistency*** guarantee**?** (4 points) Cite three systems implementing it. (3×2 pts)

4. A RAID-5 array consist of five disk drives, each of which containing 20% of the parity blocks. What is the most efficient way of updating two data blocks in the same stripe, say, b_0 and b_1., assuming that we do not know the previous values of these two blocks?

Step 1: Read __the two other blocks in the stripe (say b_2 and b_3) _________________
Step 2: Compute __the new parity block $p = b_0$ XOR b_1 XOR b_2 XOR b_3 _____________
Step 3: Write __b_0, b_1 and p ____

My solution would require a total of __2__ disk reads and __**3**__ disk writes. (2×5 pts)

5. What does ***close-to-open consistency*** guarantee**?** (4 points) Cite three systems implementing it. (3×2 pts)

Close-to-open consistency "means that if you write and then close a file on one client, and then open and read that same file on another client, the data on the second client is guaranteed to be up-to-date." [D. Hitz and A. Watson, The Evolution of NFS]

AFS, CODA and recent versions of NFS implement it.

6. What are the main advantage and the main disadvantage of ***buffering log writes*** in a journaling file system? (2×5 pts)

Buffering log writes makes the file system more efficient as it reduces the number of writes to the log. Unfortunately, it also means that the updates that were not yet written on the log when the system crashes will never be recovered.

7. How do ***leases*** differ from ***callback***s? (10 pts)

First, leases have a finite lifetime. This lifetime is rather short, which means that they will always expire by the time the server reboots after a crash. (This means that the server can store its list of leases in volatile memory.)

Second, the server cannot break a lease without contacting the client that had the lease. If it cannot do it, it will have to wait until the lease expires (which is another reason for having short lease lifetimes). This is not the case for callbacks: a server can always break a callback even if it cannot contact the client holding the callback.

8. How does LBFS partition a file into chunks? (5 pts) What is the main advantage of this rather complicated approach? (5 pts)

Chunk boundaries are defined by the chuck contents and not by a specific offset. This means that chunks containing the same data will always be recognized as so even if they start at different offsets. As a result, LBFS can recognize similarities between two versions of a file even if new contents were inserted.