Jeffrey Everett
CSCI 3753
5/3/2017

Problem Set 5

Question 1:
Storage of FAT file system: 2000 disk blocks * 4 bytes per disk block = **8000 bytes**

Storage of UNIX-style file system: (15 entries in first index block + 15 entries in singly indirect index block + 15 * 15 entries in index blocks pointed to by singly indirect index block + 15 entries in doubly indirect index block + 8 * 15 entries in singly indirect index blocks pointed to by doubly indirect index blocks + 118 * 15 entries in index blocks pointed to by singly indirect index blocks pointed to by doubly indirect index blocks) * 4 bytes per entry in index block = **8640 bytes**

Operations of FAT file system: start at first block and then progress in the linked list 1098 times to reach the 1099th block, so **1099 search operations** (assuming accessing the first block counts)

Operations of UNIX-style file system: index into first index block to get doubly indirect index block, move to this index block, index into this to get singly indirect index block, move to this block, index into this block to get direct index block, move to this block, index into this block to get answer, so **7 search operations**

Question 2:
Bitmap used: 1 bit for each page, so 1024 bits, or **128 bytes**
Linked list used: 2 bytes for each page, so **2048 bytes**

A linked list would be more memory-efficient is most of the pages were allocated.

Question 3:
FCFS scheduling: 97->84->155->103->96->197, total seek: **244 cylinders**
SCAN scheduling: 97->103->155->197->199->96->84, total seek: **217 cylinders**
LOOK scheduling: 97->103->155->197->96->84, total seek: **213 cylinders**

Question 4:
SSH works as follows: first, the client contacts the server and requests a public key. The server then responds with the public key. The client uses this public key to encrypt a symmetric key he chooses and sends this data to the server. The server decrypts this message using its private key and then stores this symmetric key. For all future transactions, the client and the server will encrypt data sent over the network with this key and the receiving machine will decrypt it upon arrival.

When a user supplies their login password, it is encrypted by a symmetric key. This is because all transactions done after the initial key exchange are done using symmetric key encryption because symmetric key encryption is much faster than public key encryption.

SSH is resilient to eavesdropping attacks because all useful data is encrypted. The request for the public key and the public key itself are not important. When the client machine sends its symmetric key, this is encrypted in a way that only those that possess the private key corresponding to the public key will be able to decrypt it. Eavesdroppers should not have this key. All data afterwards is encrypted using the symmetric key, so eavesdroppers cannot decrypt it.

SSH is vulnerable to man-in-the-middle attacks because the initial public key is not certified.

SSH is not vulnerable to replay attacks because the symmetric key changes at each login.