CSCI 3753 **Liam Kolber**
Problem Set 5 **Spring 2018, CU-Boulder**

RECITATION: 104
NUMBER OF HOURS TO COMPLETE PS: 4

1. *(10 pts) Consider a file system that uses inodes to represent files. Disk blocks are 4 KB in size, and a pointer to a disk block requires 4 bytes. This file system has 16 direct disk blocks, as well as single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system? What percentage of the required storage is overhead if the file is 100K bytes? 1000K bytes?*

$$(16 \cdot 4 \text{ KB}) + (1024 \cdot 4 \text{ KB}) + (1024 \cdot 1024 \cdot 4 \text{ KB}) + (1024 \cdot 1024 \cdot 1024 \cdot 4 \text{ KB})$$

$$= 64 + 4096 + 4194304 + 4294967296 \text{ KB}$$

$$= 4299165760 \text{ KB}$$

$$\approx 4 \text{ TB}$$

2. *(20 pts) Describe how a log based transactional file system (journaling) is beneficial for networked file systems.*

   Journaling speeds up read/write times by writing to RAM and then moving that data to the HDD when most convenient. The OS can read/write to files more quickly when they are in RAM than if they needed to be swapped out. Networks already have their own inherent delays, and by decreasing the slow down from the significantly slower HDD speeds, networks can respond much more quickly and be more efficient.

3. *(10 pts) Show pseudocode for a function to return the outer and inner index positions for the block indicated by file_offset . Assume that BLOCK_SIZE is the constant for the number of bytes per block and N_INDEX is the constant for number of addresses stored in a block.*

```
FindBlockAddress(addr file_offset, addr *outer_index, addr *inner_index) {
  *inner_index = file_offset //file_offset is address of start of block
  //N_INDEX * BLOCKSIZE = total block size
  //total_blocksize + offset = address at end of block
  *outer_index = file_offset + (N_INDEX * BLOCKSIZE)
}
```

4. *(20 pts) Based on the types of threats and security breaches we have covered, what type of security problems are Spectre and Meltdown?*

   - Spectre: allowed for the speculative operations that would not normally be authorizedby the OS during normal operation. This again bypasses any security the OS couldprovide.Both of these are basically like putting a bank vault inside the prison and expectingthe vault to be safe when you need to access it. It?s not.
   - Meltdown: can bypass security since it operates below the OS level. Once a drive has been unlocked by an encryption password, an unauthorized user can read any bit of memory they'd like due to out-of-order memory.

5. *(20 pts) Describe two advantages of encrypting data stored in the computer system?*

   - Encrypted data is always encrypted until it is decrypted. In other words, even in states where the information could be vulnerable (i.e. data transfer between users), the information is next to impossible to decypher.
   - It also allows only authorized users to access sensitive or private information without the fear of others finding loopholes or back doors to bypass any kinds of locks.

6. *(20 pts) Is it always crucial to know that the message you have sent has arrived at its destination safely? If your answer is "yes," explain why. If your answer is "no," give appropriate examples.*

   There are devices (i.e. radio transmitters) whose purpose is to do no more than simply broadcast information out for any device that wants to receive it. The radio doesn't need to know that others received it properly. Should anything ever go wrong with the system, those on the receiving end would know because the device broadcasting data has gone silent and can act accordingly. However, this system isn't foolproof as it places the responsibility on the receiver to monitor the broadcaster. In systems where an arrival signal is used, the broadcaster can determine on its own whether there are any issues and begin troubleshooting without the need of outside interference.