CS323 – Exercises
Week 9
16 May 2019

Exercise 1: (File system on-disk data structures)

What is the size in bytes of the biggest file that can be represented with an inode that contains PD direct pointers (disk addresses), PI pointers to an indirect block, and PDI pointers to double-indirect blocks? Assume that a disk block is S bytes and P pointers (disk addresses) fit into a disk block.

Answer:
S* (PD + PI*P + PDI*P*P)

Exercise 2: (File System Data Structures)

Consider the following file system very much like the one discussed in class. It maintains on disk a device directory consisting of num_inodes inodes at a fixed disk address inode_start. Each inode contains a reference count, a length in bytes, and a number PD of disk addresses for the sectors with data belonging to that inode. For simplicity, assume that there are no indirect or double-indirect entries in the inode. The bitmap of free disk sectors is kept only in memory (i.e., not on disk). After a machine crash that information is obviously lost. Write pseudocode for a program that rebuilds the free list on crash recovery.

Answer:
Assumptions:

- sector == block, total N blocks
- inode is of fixed size inode_size
- value 0 in the free block bitmap means "free"
- value 0 for a pointer means that the pointer is not used

Helper function: addr_to_blk(addr) - returns the block number of an address

Pseudocode:

```
bit freelist[N];
build_free_list() {
  // boot sector
  freelist[0] = 1;
  // device directory
  start_block = addr_to_blk(inode_start);
  end_block = addr_to_blk(inode_start + num_inodes * inode_size);
  for i in start_block..end_block
    freelist[i] = 1;
  // user space
  foreach inode i in device_directory
    if (i.refcount > 0)
      foreach pointer p in i.PD
        if (p != 0)
          freelist[p] = 1;
}
```

Exercise 3: (RAID)

Consider a 4-disks, 200GB-per-disks RAID array. What is the available data storage capacity for each of the RAID levels 0, 1, 3, 4, and 5?

Answer:
Number of disks: 4
Size of each disk: 200 GB
RAID 0: Usable disk size = 4 * 200 GB = 800 GB
RAID 1: Usable disk size = 1 * 200 GB = 200 GB
RAID 2, 3, 4, 5: Usable disk size = (4 – 1) * 200 GB = 600 GB

Exercise 4: (RAID)

RAID level 3 stripes data at byte-level and is able to correct single-bit errors using only one parity drive. In contrast, RAID level 2 stripes data at bit-level, and uses Hamming error-correction codes (ECCs) stored across at least 2 additional drives.
What is the point of RAID level 2? After all, it also can only correct one error and takes more drives to do so.

Answer:
RAID level 2 can not only recover from crashed drives, but also from undetected transient errors. If one drive delivers a single bad bit, RAID level 2 will correct this using the Hamming ECCs, but RAID level 3 will not.