Course Number: EEL5721

Principles of Computer System Design

Homework 2

Fall 2019

Name: Md Jubaer Hossain Pantho

Student ID: 95484861

A-1) Textbook exercise: Exercise 2.3

(a) The problem with synonyms is that if two names refer to the same object, and one is in the cache, and someone refers to the other one, chances are that the cache will not realize that the object is already in the cache. This will result in an unnecessary look-up and space in the cache. Moreover, if the system allows modifying objects by name. A modification to the copy of object associated with one name can propagate to the other name on the cache.

(b) One solution to this problem can be to to UID on the cache along with the cache. In this case, whenever there is a modification, it is possible to search the cache for entries that have the identical UID. Then the system can either modify or invalidate those copies.

A-2) Textbook exercise: Exercise 2.4

The claim that the user will detect no difference, except for faster name resolution is not correct. The idea of adding a *referenced object table* (ROT) can only be effective if the context does not change during a session for a user. However, this is not the case. As we know, the name resolver must resolve the name represented by the context reference before it can proceed with the original name resolution. This may result in incorrect name resolution if the context changes.

For example, for incompletely qualified file name, the file system uses the working directory as a default context reference. This working directory may change during a session. Moreover, renaming can create added problems to the ROT scheme. All these make the use of ROT an incorrect design choice.

A-3)

i) Suppose blocks are 8KBytes (8192 Bytes) in size, and there are 8 inodes per block. Assume an inode uses 8 Bytes to store type and reference count; the rest of the inode space is used to store block numbers. Suppose the disk has 2^32 blocks, and 2048 blocks are used to store inodes.

a. What is the largest number of files that can be stored?

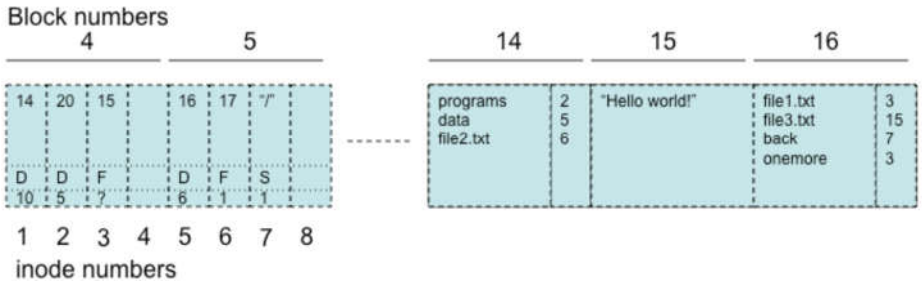
- Assuming we can only make as many files as the number of inodes, here,
  Number of blocks used to store inodes are: 2048
  Inodes per block: 8
  Largest number of files that can be stored: (2048 * 8) = 16384 (Answer)

b. What is the largest file size that can be stored (in KBytes)?

- The disk has 2^32 block. This means it will require 4 Bytes to address a block.
  Since there are 8 inodes per block
  Inode size = 8KB / 8 = 1KByte = 1024 Byte
  The inode uses 8 Bytes to store meta data
  This means space available to store block information is,
  (1024 – 8) = 1016 Bytes

Since there is no mention about double indirect pointers, we will ignore this case. Block indexes that can be stored in 1016 Byte is,

1016/4 = 254 Blocks
Largest file size = (254 * 8) KB
= 2032 KB (Answer)



ii) *Consider the LOOKUP resolution procedure as discussed in class.*
a. What is returned from LOOKUP("data",14)?

- This will result in a failure. Since in the given image there is no inode number with index 14.

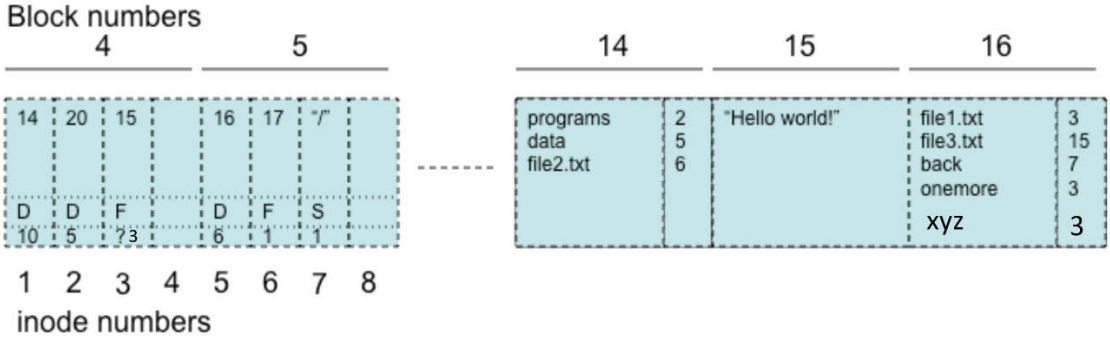
b. What is returned from LOOKUP("file1.txt",5)?

- This will return 3.

c. What is returned from LOOKUP("programs",5)?

- This will return failure. Since there is no file name with the given name.

iii) *Suppose you create a new hard link "xyz" in directory "/data" that links to "/data/file1.txt". Describe (in words, or use a drawing based on the figure above) the new state of all blocks that may have changed due to this operation*



The reference count will be three referreing to file1.txt, onemore and xyz.

iv) *Assume the sequence of events for two different processes (A and B) running in the same computer where this file system is mounted:*

Time T1: A issues fdA=open("/data/file1.txt"), and obtains fdA=3
Time T2: B issues fdB=open("/data/file1.txt") and obtains fdB=3
Time T3: thread A issues read(fdA,bufA,n=5)
Time T4: thread A issues close(fdA)
Time T5: thread B issues read(fdB,bufB,n=5)
Where fdA, fdB are file descriptors, bufA and bufB are memory buffers, and n is the number of characters to read. What values (if any) are returned in the buffer bufA of thread A at time T3, and bufB in thread B at time T5?

- Since They are differenct process, the cursor will be different for each one of them. That means,
  bufA = "Hello"
  bufB = "Hello"

v) *Suppose now that the system has a second hard disk, with a second file system as depicted in the figure below. Suppose you mount this second file system under directory "/mnt" of the root file system. Is it possible to create a hard link "file4.txt" under "/data" that links to "file4.txt" in the second file system? Is it possible to create a symbolic link "file4.txt" under "/data" that links to "file4.txt" in the second file system? Describe (in words, or use a drawing based on the figure above) the new state of all blocks that may have changed due to these operations, if they are possible.*

Since this is a separate disk, it will not be possible to create a hard link for file4.txt. However, it will be possible to create a symbolic link (soft link) that can link to the file4.txt in the second file system. Since they are linked by name. In the inode table the *type* field will be a symlink. And the data in the array *blocks* will actually contain the characters of the path name rather than a set of inode numbers.