File System

Monday, December 7, 2015 2:17 PM

File System Interface

1. Linux ln is described as follows in the manual

```
LN(1)                    User Commands                    LN(1)

NAME
       ln - make links between files

SYNOPSIS
       ln [OPTION]... [-T] TARGET LINK_NAME   (1st form)
       ln [OPTION]... TARGET                  (2nd form)
       ln [OPTION]... TARGET... DIRECTORY     (3rd form)
       ln [OPTION]... -t DIRECTORY TARGET...  (4th form)

DESCRIPTION
       In the 1st form, create a link to TARGET with the name LINK_NAME.  In the 2nd form,
       create a link to TARGET in the current directory.  In the 3rd and 4th forms, create
       links to each TARGET in DIRECTORY.  Create hard links by default, symbolic links with
       --symbolic.  When creating hard links, each TARGET must exist.

       -d, -F, --directory
              allow the superuser to attempt to hard link directories (note: will probably
              fail due to system restrictions, even for the superuser)
```

   Sloppy the super user has created a cyclic hard link, what happens when executing the following two commands on the directory where the link is located?

   a. `rm`
   b. `find`

2. Problems from the textbook:
   a. Solve Exercise 11.4 ("Could you simulate a multilevel...")
   b. Solve Exercise 11.8 ("Researchers have suggested that...")
   c. Solve Exercise 11.9 ("Consider a file system in which...")
   d. Solve Exercise 11.13 ("Some systems automatically...")
   e. Solve Exercise 11.15 ("Give an example...")

File System Implementation

Part I

1. List the 6 key items contained within a volume control block?
2. What is another name for a File Control Block (FCB)?
3. What are the 2 major problems associated with linked allocation of disk blocks?

Part II

1. Generally speaking, where should a file seek pointer be implemented, e.g., in the user file descriptor table, file structure table, or an inode structure? (There is more than one answer! Justify your explanation, write down assumptions)
   a. Which one and why?
   b. Consider the scenario where a parent and a child process share the same file at the same time, then, another unrelated/independent process also reads the same file. Where should the file seek pointer be implemented?

2. What are the similarities and differences in file allocation between Linux and Windows NTFS? (*yes, you'll have to research this one*)

3. Solve problem 11.15(8th edition) / 12.16 (9th edition) from your book

   ("Consider a file system that uses inodes to represent files...")

Part III

1. Consider a system where free space is kept in a free-space list. (Again, you may have to do some extra-thinking and research)
   a. Suppose that the pointer to the free-space list is lost. Can the system reconstruct the free-space list? Explain your answer.

   b. Consider a file system similar to the one used by Linux with indexed allocation. How many disk I/O operations might be required to read the contents of a small local file at /a/b/c? Assume that none of the disk blocks is currently being cached.

   c. Suggest a scheme to ensure that the pointer is never lost as a result of memory failure.

2. Consider a file system where part of the iNode is redefined as follows:
   - 10 pointers to direct bock
   - One pointer to single indirect block
   - One pointer to double indirect block

   (Pointer size is 2-byte) Assume that the block size is 512B
   a. What is the maximum size of the hard drive that the file system can support? Express your answer in number of blocks.
   b. Without taking into account the storage space needed for the iNode, how many disk blocks are required to store 4000 bytes of data?

3. Consider a file system on a disk that has both logical and physical block sizes of 512 bytes. You can assume that the information about each file is already in memory. If contiguous allocation strategy is used:

   a. How does the logical-to-physical address mapping work in this system? (You can assume that a file always occupies less than 512 blocks).

   b. If the last logical block accessed was block 10, and the next block to access is logical block 4, how many physical blocks must be read from the disk?

   c. Answer a,b above for the case when indexed allocation strategy is used

4. Consider a file system using a File Allocation Table (FAT). Each FAT entry is 7-bits long, and each Disk Block size is 4KB. What is the maximum hard drive size that this file system can support?

5. Problems from the textbook (may be Ch11 in some editions)
   a. 12.12 ("Consider a system where free space...")
   b. 12.16 Consider a file system that uses inodes to represent files. Disk blocks are 8 KB in size, and a pointer to a disk block requires 4 bytes. This file system has 12 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?