# University of Newcastle
# School of Electrical Engineering and Computing

## COMP2240 - Operating Systems
## Workshop 09
## Topics: File Management

**1.** In a hashed file organisation, the division method is used to compute the hash address of a record. This method can be stated as follows:

> Choose a large prime number *m* which is close to the number of keys *n*. Define the hash function ***h**(k) = k (**mod** m) +c*, where *c* is the lower limit of address.

If a set of records needs to be stored in 100 locations, starting from the address 7865, compute the address for the records having IDs 1234, 2345, 3333, and 4433.

**2.** For a B-Tree (degree of the B-Tree is *d=3*) in Figure 1, show the result of inserting the key 97.

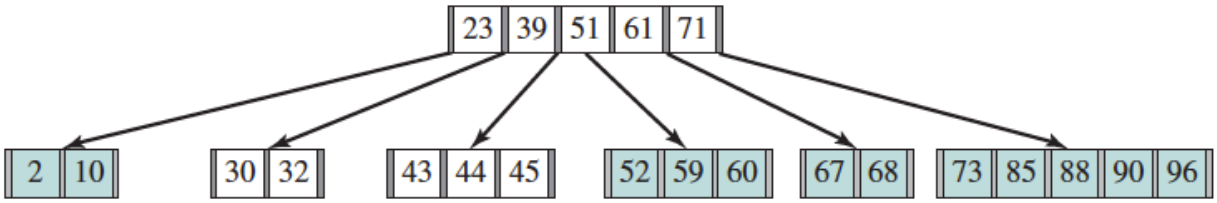


Figure 1: An Instance of B-Tree

**3.** One way to use contiguous allocation of the disk and not suffer from holes is to compact the disk every time a file is removed. Since all files are contiguous, copying a file requires a seek and rotational delay to read the file, followed by the transfer at full speed. Writing the file back requires the same work.

a) Assuming a seek time of 5 msec, a rotational delay of 4 msec, a transfer rate of 8 MB/sec, and an average file size of 8KB, how long does it take to read a file into main memory and then write it back to the disk at a new location?

b) Using these numbers, how long would it take to compact half of a 16-GB disk?

**4.** A sequential file is stored in a disk occupying 100 contiguous disk blocks. The disk has an average rotational delay of 2.5 ms. The time taken to seek the head of the drive to the required cylinder is 25 ms and the time taken to read a block is 0.25 ms. Find the minimum, maximum, and average time to search for a record using a linear search process.

**5.** A sequential file has 10 million records. How does efficiency in access improve by using a two-level index? Assume 100 entries in a higher-level index and 10,000 entries in a lower-level index.

**Supplementary problems:**

**S1.** An alternative algorithm for insertion into a B-tree is the following: As the insertion algorithm travels down the tree, each full node that is encountered is immediately split, even though it may turn out that the split was unnecessary.

a) What is the advantage of this technique?
b) What are the disadvantages?

**S2.** Consider the organization of a UNIX file as represented by the inode shown in Figure 2.

Assume that there are 12 direct block pointers, and a singly, doubly, and triply indirect pointer in each inode. Further, assume that the system block size and the disk sector size are both 8K. If the disk block pointer is 32 bits, with 8 bits to identify the physical disk and 24 bits to identify the physical block, then

a) What is the maximum file size supported by this system?
b) What is the maximum file system partition supported by this system?
c) Assuming no information other than that the file inode is already in main memory, how many disk accesses are required to access the byte in position 13,423,956?

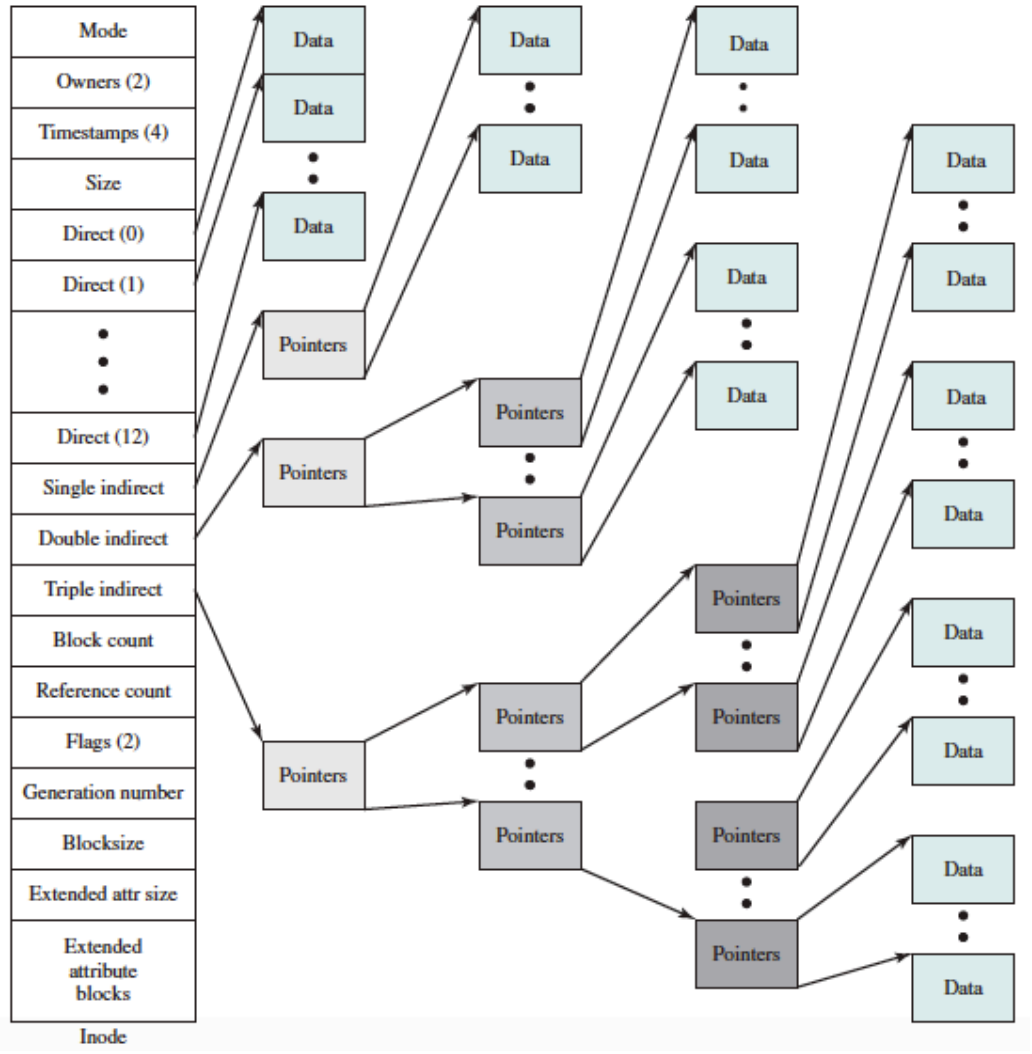


Figure 2: Structure of FreeBSD Inode and File

**S3.** A i-node in Linux system is 128 bytes long and describes exactly one file. An i-node contains accounting information (including all the information returned by stat, which simply takes it from the i-node), as well as enough information to locate all the disk blocks that hold the file's data
A Linux i-node has 12 disk addresses for data blocks, as well as the addresses of single, double, and triple indirect blocks. If each of these holds 256 disk addresses, what is the size of the largest file that can be handled, assuming that a disk block is 1 KB?

**S4.** After a system crash and reboot, a recovery program is usually run. Suppose this program discovers that the link count in a disk i-node is 2, but only one directory entry references the inode. Can it fix the problem, and if so, how?

**S5.** Both the search and the insertion time for a B-tree are a function of the height of the tree. We would like to develop a measure of the worst-case search or insertion time. Consider a B-tree of degree d that contains a total of n keys. Develop an inequality that shows an upper bound on the height *h* of the tree as a function of *d* and *n*.