CSE 421/521 – Operating Systems **Sassi Kesavan**
Fall 2018 - Homework Assignment #4 **UBIT: sassikes**

Problem 1:

Consider a file system on a disk that has both logical and physical block sizes of 512 bytes. Assume that the information about each file is already in memory. For each of the three allocation strategies (contiguous, linked, and indexed), answer these questions:

A) How is the logical-to-physical address mapping accomplished in this system? (For the indexed allocation, assume that a file is always less than 512 blocks long.)

Let Z be the starting file address (block number),

- Contiguous. Divide the logical address by 512 with X and Y the resulting quotient and remainder respectively. Add X to Z to obtain the physical block number. Y is the displacement into that block.

- Linked. Divide the logical physical address by 511 with X and Y the resulting quotient and remainder respectively. Chase down the linked list (getting X + 1 blocks). Y + 1 is the displacement into the last physical block.

- Indexed. Divide the logical address by 512 with X and Y the resulting quotient and remainder respectively. Get the index block into memory. Physical block address is contained in the index block at location X. Y is the displacement into the desired physical block

B) If we are currently at logical block 10 (the last block accessed was block 10) and want to access logical block 4, how many physical blocks must be read from the disk?

- Contiguous- 1
- Linked- 4
- Indexed- 2

Problem 2:

Consider a RAID Level 5 organization comprising five disks, with the parity for sets of four blocks on four disks stored on the fifth disk. How many blocks are accessed in order to perform the following?
a. A write of one block of data
b. A write of seven continuous blocks of data

a. A write of one block of data requires the following: read of the parity block, read of the old data stored in the target block, computation of the new parity based on the differences between the new and old contents of the target block, and write of the parity block and the target block.

b. Assume that the seven contiguous blocks begin at a four-block boundary. A write of seven contiguous blocks of data could be performed by writing the seven contiguous blocks, writing the parity block of the first four blocks, reading the eight block, computing the parity for the next set of four blocks and writing the corresponding parity block onto disk.

Problem 3:

Consider a distributed system with two sites, A and B. Consider whether site A can distinguish among the following:
a. B goes down.
b. The link between A and B goes down.
c. B is extremely overloaded and its response time is 100 times longer than normal.
What implications does your answer have for recovery in distributed systems?

One technique would be for B to periodically send a message to A indicating it is still alive. If A does not receive a message, it can assume either B—or the network link—is down. Note that a message does not allow A to distinguish between each type of failure. One technique that allows A better to determine if the network is down is to send a message to B using an alternate route. If it receives a reply, it can determine that indeed the network link is down and that B is up. If we assume that A knows B is up and is reachable and that A has some value N that indicates a normal response time, A could monitor the response time from B and compare values to N, allowing A to determine if B is overloaded or not. The implications of both of these techniques are that A could choose another host—say C—in the system if B is either down, unreachable, or overloaded.

Problem 4:

Consider the centralized and the fully distributed approaches to deadlock detection. Compare the two algorithms in terms of message complexity.

The centralized algorithm for deadlock detection requires individual processors or sites to report its local waits-for graph to a centralized manager. The edges in the waits-for graph are combined and a cycle detection algorithm is performed in the centralized manager. The cost of this algorithm is the cost of communicating the various waits-for graph to the centralized server. In the distributed approach, each site builds its own local waits-for graph and predicts whether there is a possibility of a cycle based on local observations. If indeed there is a possibility of a cycle, a message is sent along the various sites that might constitute a cyclic dependency. Multiple sites that are potentially involved in the cyclic dependency could initiate this operation simultaneously. Therefore, the distributed algorithm is sometimes better than the centralized algorithm and is sometimes worse than the centralized algorithm in terms of message complexity. It is better than the centralized algorithm since it does not communicate the entire local waits-for graph to the centralized server. However, the distributed algorithm could incur more messages when multiple sites are simultaneously exploring the existence of a cyclic dependency.