HCMC University Of Technology
Faculty of Computer Science & Engineering

Lab 10 File System
Course: Operating Systems

Lecturer: Thanh Le-Hai Hoang
Email: thanhhoang@hcmut.edu.vn

June 1, 2021

Goal: The objective of this lab is file system consolidation.

Content In detail, this lab requires student to:

- Analyze the Contiguous file allocation strategy
- Analyze the Linked-List file allocation strategy
- Analyze the FAT file allocation strategy
- Consider how block size affects read system performance

Result After doing this lab, student can understand the principle of file system.

Requirement Student need to review the theory of file system.

1 Exercise

1. Consider a file currently consisting of 100 blocks. Assume that the file control block (and the index block, in the case of indexed allocation) is already in memory. Calculate how many disk I/O operations are required for contiguous, linked, and indexed (single-level) allocation strategies, if, for one block, the following conditions hold. In the contiguous-allocation case, assume that there is no room to grow at the beginning but there is room to grow at the end. Also assume that the block information to be added is stored in memory. **Please explain in detail how you calculate the result**
   a) The block is added at the beginning.
   b) The block is added in the middle.
   c) The block is added at the end.
   d) The block is removed from the beginning.
   e) The block is removed from the middle.
   f) The block is removed from the end.

2. Consider a system that supports the strategies of contiguous, linked, and indexed allocation. What criteria should be used in deciding which strategy is best utilized for a particular file? For each strategy, please provide the context where it can be used for that type of file.

3. Suppose you have a 10GB drive (2^{30} * 10 bytes). How many entries would a FAT need if the block size is 1KB (2^{10}), 4KB, and 8KB?? Please explain your caculation.

4. Consider a system where free space is kept in a free-space list.
   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 UNIX 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.

Submission: Write your answer into a single pdf file with the following format `<ID>.pdf`, then submit this pdf file to BKEL without any compression.

All other formats or any late submissions will be rejected.