

C502 – Operating Systems Tutorial *

File Systems – Solutions

Note: The solution notes below only briefly list (some of) the key points that should be included in an answer. They are by no means complete. In an exam, you are expected to spell out the solution more fully and include a detailed explanation of your reasoning.

1. Explain what it means to “defragment a file system”? Are there file systems that do not require defragmentation? How can a file system reduce the amount of fragmentation?

Answer: Defragmenting a file system means that blocks belonging to the same file are moved around to ensure that they occupy consecutive disk blocks. This improves the read/write performance because it avoids unnecessary disk seeks.

In general, any file systems can become fragmented to a certain extent. However, some file systems may suffer less from fragmentation when they ensure that blocks for files are allocated in nearby or consecutive locations on the disk. For example, block allocations may be done according to zones.

2. Assume that we are using i-nodes. Consider a file system that maintains a unique i-node for each file in the system. Each i-node includes 8 direct pointers, a single indirect pointer, and a double indirect pointer. The file system block size is 1024 (2^{10}) bytes, and a block pointer occupies 4 bytes.

How many disk operations will be required if a process reads data from the N^{th} block of a file? Assume that the file is already open, the buffer cache is empty, and each disk operation reads a single file block. Your answer should be given in terms of N .

Answer:

if $0 \leq N < 8$, then one operation is required,

if $8 \leq N < 256 + 8$, then two operations are required,

if $256 + 8 \leq N < 2^3 + 2^8 + 2^{16}$, then three operations are required.

3. Consider a file system that uses i-nodes with single-indirect and double-indirect blocks and a block size of 1024 (2^{10}) bytes. If the block size of the file system is doubled, by approximately what factor does the maximum possible file size increase? Your answer should be an integer.

*Answer: For very large files, almost all data blocks are accessed through the double indirect pointer, so consider only those blocks. For a block size of B and a pointer size of P , the total size of the double-indirect data blocks is: $(B/P)^2 * B = B^3/P^2$*

So, if B is doubled, the maximum file size will increase by approximately a factor of 2^3 , or 8.

4. Assuming that we are using i-nodes to maintain file information. Why is the information maintained in main memory for an open file larger than that for a closed file. What does this extra information represent? (Exam question in 2016-17).

Answer: Additional information contained - location of the original i-node by maintaining the disk device number and i-node device number in order that the i-node can itself be written back to disk, if it had been modified. Additionally contains information about the number of process that have opened this file.

*with thanks to Roman Kolcun