Benjamín Aage B. Birgisson

Exercise 8.2.2:

(a)

An ***absolute path name*** of a file, uniquely identified by an internal ID, f, is the concatenation of the directory and file names leading from the root to the file f.

- Shortest absolute path for file f4: **/x/z/n**
- Shortest absolute path for file f8: **/w/e** (or **/y/p**)

(b)

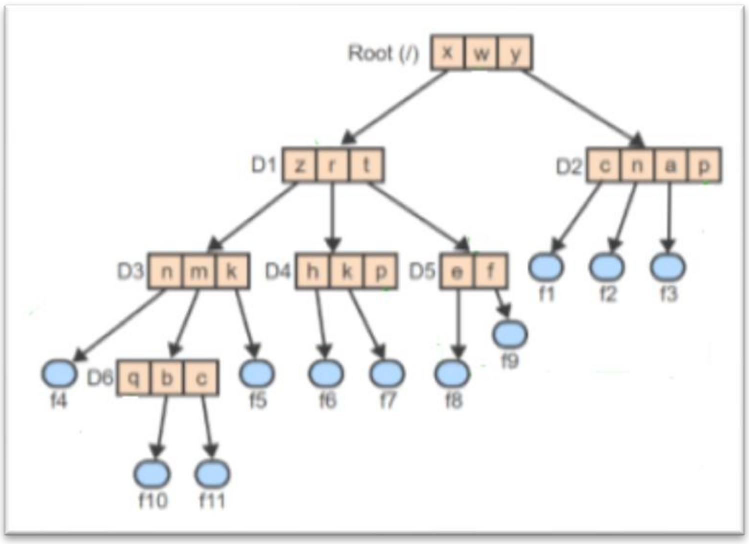
A ***relative path name*** is a concatenation of file names starting with the current directory.

Current directory D2:
- Shortest relative path for file f4: **/../x/z/n**
- Shortest relative path for file f8: **/p**

(c)

Current directory D6:
- Shortest relative path for file f4: **/../n**
- Shortest relative path for file f8: **/r/p** (or **/t/e**)

(d)



1) Determine the shortest absolute path for the files f4 and f8.
   - Shortest absolute path for file f4: **/x/z/n**
   - Shortest absolute path for file f8: **/x/t/e**

2) Determine the shortest relative path for the files f4 and f8 when the current directory is D2.

   Current directory D2:
   - Shortest relative path for file f4: **/../x/z/n**
   - Shortest relative path for file f8: **/../x/t/e**

3) Determine the shortest relative path for the files f4 and f8 when the current directory is D6.

   Current directory D6:
   - Shortest relative path for file f4: **/../n**
   - Shortest relative path for file f8: **/../../t/e**

(e)

move /y /w:
Shortest absolute path for file f1: **/w/y/c**

(f)

delete /x/z/m/q
files and directories deleted:
q

delete /x
files and directories deleted:
x, z, r, t, n, m, k, h, k, p, f4, q, b, c, f10, f11, f5, f6, f7, e, f, f8, and f9

Exercise 8.3.1:

(a)

50 (directory file entries) * 10 (file name length) * 10 (attribute length) = 5000 pages / 1000 (page size) = 5 pages

(b)

50 (directory file entries) * 10 (file name length) * 20 (pointer + attribute length) = 10000 pages / 1000 (page size) = 10 pages

(c)

50 (directory file entries) * 10 (file name length) * 110 (attribute length) = 55000 pages / 1000 (page size) = 55 pages

(d)

50 (directory file entries) * 10 (file name length) * 20 (pointer + attribute length) = 10000 pages / 1000 (page size) = 10 pages

(e)

50 (directory file entries) * 10 (file name length * X (attribute length) = 10

➔ 500X = 10

➔ X = 10 / 500 = 0.02

Exercise 8.4.1:

(a)

- The file system verifies first that the process has the necessary access rights. If so, the file system allocates a free open file table entry. Here, it would for example be entry number 7 in 'open file table'.
- Current position for entry number 7 is 0 (zero). The file length is 280 (L).
- The file system then copies the pointer to the FCB into the file table entry, thereby making the directory no longer needed.
- To allow future r/w operations to access the file data, the file system copies the first block of the file's data from the disk to the r/w buffer.

Changes: Open file table – Descriptor index (7) -> 19
Open file table – Current position -> 0

(b)

- A seek operation moves the current position of an open file to a new specified position (in this case, position 60).The current position is 0, and the block size is 256 bytes. The r/w buffer contains block 0 with an offset of 60. The current position is updated to 60, and as the new position is within the same block, the new offset is 60.

Changes: Open file table – Descriptor index 7 – Current position -> 60

(c)

- First entry in current directory is 'free', and becomes 'new'.
  Descriptor index -> 15
- File descriptor 15 – File length -> 0
  Disk block # -> NULL

(d)

- Open file table (7) – Descriptor index -> 15
  Current position -> 0

(e)

- File descriptor (7) – File length -> 0
  File descriptor (7) – Disk block -> 0
- Open file table (7) -> Current position -> 30

(f)

- File descriptor (7) – File length -> 30
- Open file table (7) – Current position -> None
  Open file table (7) - free

(g)

- File descriptor (15) – File length -> free
  Disk block -> None
- Current directory (new) – Descriptor index -> None
  Symbolic name (new) -> free

Exercise 8.5.3:

(a)

Contiguous:

In a ***contiguous block allocation scheme***, every file is mapped into a contiguous sequence of disk blocks. The FCB points to the first disk block. The file length, also maintained in the FCB, determines how many blocks are occupied by the file.

Read block D: D
Read block F: F
Delete block E: E
Insert block 'C: C D (check if the spot is 'free', and if not, allocate a new larger area on disk)

Linked:

With a ***linked block allocation scheme***, the blocks containing a file may be scattered throughout the disk. The FCB points to the first block and each block points to the logically next block.

Read block D: E F G H A B C D
Read block F: E F
Delete block E: E
Insert block 'C: E F G H A B C

FAT:

A ***File allocation table*** (***FAT***) is an array where each entry corresponds to a disk block. The FAT keeps track of which disk blocks belong to a file by linking the blocks in a chain of indices. A file's FCB contains the index of the first file block, which in turn contains the index of the next block, etc.

Read block D: E D
Read block F: E D F
Delete block E: E D
Insert block ‘C: E D F C

Indexed:

With an ***indexed block allocation scheme***, file blocks may reside anywhere on the disk. An index table is provided for each file, which keeps track of the blocks belonging to the file.

Read block D: D
Read block F: F
Delete block E: E
Insert block ‘C: C

Exercise 8.5.4:

(a)

22 -> 30 -> 17

(b)

31 -> 29 -> 18 -> 25

Exercise 8.5.5:

(a)

512 bytes per block / 4 byte integer = 128 bytes

(b)

512 bytes per block / 4 byte integer = 128 bytes * 100,000 blocks = 12,800,000 bytes

(c)

512 bytes for 1 disk block, as a block has 512 bytes each.

Exercise 8.5.7:

(a)

512 bytes, as it is still in 1-level index structure (1 * 512 bytes).

(b)

It would double, from 512 bytes to 1024 bytes.

Exercise 8.5.11:

(a)

66.67% or 2/3 must be free, such that the bitmap and the linked list occupy the same amount of disk space, as the bitmap needs 1 bit for each block and the linked list needs 2 integers for each free block.

(b)

Then 6.67% of blocks (or 2/30) would need to be freed, when the average group size is 10 blocks.
