Özgür Taylan ÖZCAN / 54284 08.06.2018

COMP 304 - ASSIGNMENT #3

Problem 1:

Contiguous allocation results in external fragmentation. As new processes added and removed, holes occur between allocations. Since these holes aren't contiguous, new requests may not be satisfied although total memory space is enough. Code sharing is not possible in external fragmentation, Because all processes have their own contiguous allocations.

Segmentation also suffers from external fragmentation. Just like in contiguous allocation, holes may occur between segments. Code sharing is possible in this method, because processes may share the same segments in the memory.

In paging, there is no external fragmentation because all memory space is divided into fixed sized frames. Code sharing is possible since processes may share pages.

Problem 2:

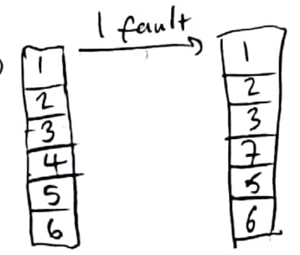
P1	P2	offset
12-bit	16-bit	12-bit

- page size = 1 byte $\times 2^{12} = 4$ KB
- there are 2^{12} entries in table 1 and $2^{12} \cdot 2^{16} = 2^{28}$ entries in table 2.

So; there are 2^{28} pages in the address space.

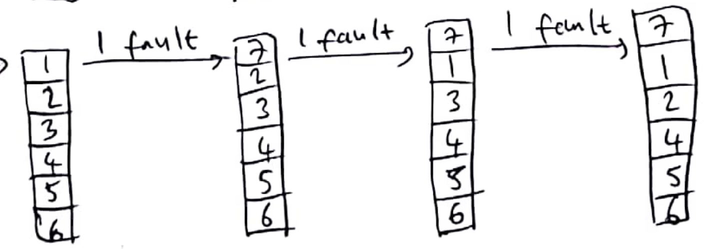
Problem 3:

- LRU: 6 faults →



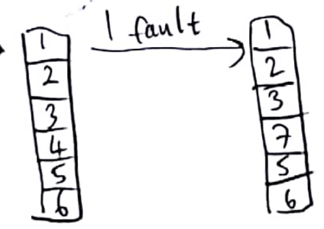
total 7 faults

- FIFO: 6 faults →



total 9 faults

- Optimal: 6 faults →



total 7 faults

Scanned by CamScanner

Problem 4:

a) file1's and file2's inodes are the same. It is 286781. They have the same contents.

b) After file2 is edited, they have the same contents. Inodes are also the same.

c) After file1's deletion, file2 still exists.

d) unlink() system call is used.

e) They are not the same. They are unique.

f) file3's been altered as well.

g) file4 is not editable. It is read only.

h) Hard link creates a mirror of the original file. They have the same inodes. Changes on the original will reflect in the other.

Soft link creates different inodes. It contains the path for the original file, not the contents.

Hard links are useful when the existence of the files are not needed to be tied.

Soft links can be used for shortcuts.