OS Tutorial

1. Consider an x86 computer with the following memory architecture
   • 32-bit virtual address space
   • 32-bit physical address space
   • 4 KiB page size
   • Two-level page table
   • 32-bit page table entry (PTE) size

   How many bits of the 32 bit virtual address are used for (a) index of outer page table (b) index of inner page table (c) offset in the page. Why? (d) total size of the page table

abc) 4KiB page size -> 2^{12} bytes -> 12 bits required for offset - (c)
32-bit PTE size -> 4KiB/32 bit -> 1024 entries possible in a single page
The 2nd level page table is present in a single page and hence has 1024 entries. We need 10 bits to uniquely identify entries in the 2nd level page table. Index of inner page table requires 10 bits (b)
Remaining bits are used for index of outer page table 32 - 12 -10 = 10 (a)

d) 1 outer page table + 1024 inner page tables each of 4KiB size
Total size = 1025 * 4KiB

2. What is the Effective Access Time of 1/2 level page table with TLB?

a) 1 level page table

Let ‘a’ be the %TLB hit
TLB_time : time to go through TLB
m_time : time to access memory

e = a(TLB_time + m_time) + 1-a(TLB_time + m_time + m_time) [extra m_time in 2nd term is for accessing the page table which is in memory]

b) 2 level page table

Let ‘a’ be the %TLB hit
TLB_time : time to go through TLB
m_time : time to access memory

e = a(TLB_time + m_time) + 1-a(3m_time + TLB_time)
[extra m_time in 2nd term is for accessing the 1st level page table and 2nd level page table which are in memory]

3. Prove that optimal page replacement algorithm is stack page replacement policy

OPT assigns a priority to a page that is independent of the number of frames allocated. If the number of frames increases, a lower priority page is also included in the allocated frames, but an already existing page in a lower number of frames allocation will not be removed as it has higher priority.

4. Simulate second chance algorithm on the following reference string:
7,0,1,2,0,3,0,4,2,3,0,3,0,3,2,1,2,0,1,7,0
There can be 3 pages in memory at a time per process. Assume that swapping of pages is non preemptive.

7

Frame	R
7	1

0

Frame	R
7	0
0	1

1

Frame	R
7	0
0	0
1	1

2

Frame	R
2	1
0	0
1	1

0

Frame	R
2	1
0	1
1	1

3

Frame	R
3	1
0	0
1	0

0

Frame	R
3	1
0	1
1	0

4

Frame	R
3	0
0	0
4	1

2

Frame	R
2	1
0	0
4	1

3

Frame	R
2	0
3	1
4	1

0

Frame	R
0	1
3	1
4	1

3

Frame	R
0	1
3	1
4	1

0

Frame	R
0	1
3	1
4	1

3

Frame	R
0	1
3	1
4	1

2

Frame	R
2	1
3	0
4	0

1

Frame	R
2	0
1	1
4	0

2

Frame	R
2	1
1	1
4	0

0

Frame	R
2	0
1	0
0	1

1

Frame	R
2	0
1	1
0	1

7

Frame	R
7	1
1	1
0	1

0

Frame	R
7	1
1	1
0	1

5. What is the maximum number of addressable blocks with a 4n byte block sizes and 4 byte pointer sizes for the following file system architectures?
a) FAT
b) 2 level indexing
c) Combination scheme with 3 single indirect, 2 double indirect, 1 triple indirect and rest direct pointers

a) Number of pointers in a block = 4n/4 = n
Addressable blocks in FAT = n

b) Number of pointers in 1st level = n
Number of pointers in 2nd level = n*n = n^2
Addressable blocks in 2 level indexing = n^2

c) Number of direct pointers = n - 3 - 2 - 1 = n - 6
Addressable blocks by single indirect = n
Addressable blocks by double indirect = n*n = n^2
Addressable blocks by triple indirect = n*n*n = n^3
Total addressable blocks = (n-6) + 3n + $2n^2$ + n^3

6. Answer True/False with reason
a) FAT file can have soft links
b) FAT file can have hard links

a) False
FAT doesn't support soft links. However, support could be added to allow soft links if the file system marks the file as a link using a bit.

b) False
FAT doesn't support hard links. Hard links pretty much rely on having the file name in a different place from the rest of the metadata. On FAT, the directory entry contains the name *and* all other metadata, so there's no one place to keep the metadata of a file with more than one hard link to it.