Operating System Assignment

Part 1 : Virtual Memory Questions

By Shubham Upadhyay (2010CS1036)

Ans 1 :

Virtual memory space = 2^{32} bytes
Page size = 2^{12} bytes
Number of pages = 2^{20}
Number of indexes in the Page table = 2^{20}
Since page size is equal to frame size,
Frame size = 2^{12} bytes
Size of physical memory = 2^{18} bytes
No. of frames = $2^{18}/2^{12} = 2^{6} = 64$ frames
As the number of virtual pages (2^{20}) is greater than number of frames (2^{6}), the two-level paging system is taken into account.
Virtual Address = 11123456 (base 16)
Virtual address in binary form = 0001 0001 0001 0010 0011 0100 0101 0110.

The low order 12 bits are used as page offset. The remaining first 20 bits are used as page number in the page table. The offset bits are added to the page numbers from the page table to get to the final physical address.

When the CPU ask for a 32 bits memory address, the hardware checks for a page number using the high order 20 bits of the virtual address. If there is a hit in the page table, then it would the page offset and return the physical address or else if the page is not in memory, the hardware traps to the operating system. Now the operating system which is the software, looks for a free frame in the physical memory.

The operating system then schedules a disk I/O operation to read the desired page into the free frame in physical memory or replaces an existing page if there is no free frame. The page table is then modified to indicate that the requested page is now in memory. The state of the interrupted process is reset and the instruction is re-executed.

Ans 2 :

We converting all the values to one time unit i.e. nanoseconds.
A = Effective access time
Ma = Memory access time
Fault = Page fault time
P = Probability that a page fault occurs (Page fault rate)

Therefore, A = (1-P)*Ma + P*F ault

Given :

Memory access time = 100 nanoseconds.

It is given that 70% of the time page to be replaced is modified .

If page is modified :
Servicing page fault takes 8 milliseconds i.e. 20*106 nanoseconds.

If page is not modified :
Servicing page fault time takes 8*106 nanoseconds.

Fault = 0.7*20*106 + 0.3*8*106 =16.4 * 106 nanoseconds

Therefore,
A = (1-P) * 100 + P * 16.4 * 106
=> 200 >= 100 - 100P + 16.4 * 106 P
=> 100 >= -100P + 16.4 * 106 P
=> P <= 100/(16.4 * 106)
= 6.09 * 10^{-6} (approx) .
Hence the maximum acceptable page fault rate is approximately 6*10^{-6}.