Superpages

(1) What are superpages? Why are they useful? What are the constraints on their sizes, placement, and page attributes (protection, reference, and dirty bits)?

(2) Superpages
   A. What are the advantages and disadvantages of superpages?
   B. How many TLB entries and page table entries are there for each superpage?
   C. What are the restrictions on superpage size, allocation, and placement?

(3) If a superpage has a size equal to 4 base pages, how many TLB entries are occupied by the superpage? How many page table entries are occupied by the same superpage? Explain why.

(4) How do superpages affect (increase/decrease/don't change) internal and external memory fragmentation? Why?

(5) In the paper, superpage allocations are performed in two steps: preemptible reservations and incremental promotions. Why do we need this two-step mechanism? If we already know how much to reserve, why not create the entire superpage in one shot the first time any of its base pages is accessed?

(6) Why is it that using a mix of multiple superpage sizes tends to yield better application speedups than using superpages of only one size?

(7) In the superpage paper, explain how ***preemptible reservations***, ***incremental promotions*** and ***speculative demotions*** work.

OR

In the "superpages paper" (Navaro et. al.), when and how are (a) reservation, (b) promotion, and (c) demotions carried out? (d) Why are reservations called "pre-emptible"?

(8) Why do superpages make both internal and external fragmentation worse?

OR

Which fragmentation (Internal/external) is made worse by superpages? Why?

(9) In the superpage paper, how does the system proposed avoid the need to page-out an entire superpage to the disk upon memory pressure? What is the source of performance overhead in this mechanism?

(10) What is the new contribution made by the superpage paper by Navarro et. al? Remember that both software and hardware support for superpages had already been proposed in earlier papers before this work was published.

(11) Use of superpages (as they are currently designed) requires the use of contiguity restoration techniques in the operating systems. To avoid the need for contiguity restoration, one could get rid of the requirement that the base physical pages of a superpage be contiguous - in other words, allow the base pages in physical memory not to be next to each other. If one does so, describe how you would redesign (if at all) the TLB, page-tables, and the mechanism for translating virtual to physical addresses? You can assume either architected page-tables or architected TLB. Remember to list any assumption you make, justifying why they are reasonable.