Virtual Memory Management in the VAX/VMS Operating System
Levy, H. and Lipman, P.,
IEEE Computer, Vol. 15, No. 3, March 1982, pp. 35-41.

Background

1. What challenges did VAX/VMS need to contend with?
2. What were the VAX/VMS designers particularly concerned with for a paging system? Why were these issues a particular concern in their environment?
3. What were their solutions?

Hardware

4. With the VAX-11 hardware, how is the 32-bit virtual address allocated? (DRAW.) With a 512 byte page, how many bits are part of the offset? How many bits for the virtual page number (vpn)?
5. What does the virtual address space look like? Why is the first page of P0 region reserved?
6. What does a page-table-entry (PTE) look like?
7. Imagine that P0 and P1 PTBR (page table base register) pointed to a physical address. How would you do an address translation? What are the drawbacks of keeping the user page tables in physical memory?
8. With the VAX/VMS approach, the user page tables are kept in the systems virtual address space. Thus, the P0 and P1 PTBR contain virtual addresses in the system space. And the system PTBR (SPTBR) contains a physical address. What are the steps for VAX/VMS to translate a virtual address in user space to its physical address?
9. How many extra memory accesses are required for each memory access performed by an application? How is this overhead reduced? Does anything special need to happen on a context switch?

Solution: Use process-local page replacement
Solution: Use simple FIFO replacement with global free and modified lists

10. How does per-process local replacement of pages work?
11. FIFO is not a good replacement policy because it does not handle locality in access patterns. Why didn't VAX/VMS use LRU? How does VAX/VMS get around FIFO's limitations? What happens on a page fault?
12. What are the advantages of using a modified page list?

Solution: Cluster multiple reads together

13. What are the requirements to perform a cluster read?
14. When is clustering most effective? Why is clustering not always useful?
15. Can you create a synthetic workload that could perform worse with clustering than without?

Solution: Employ a swapper to bring in entire resident set of process

16. What is the purpose of the swapper? How does the swapper work?

Conclusion?