Non-Contiguous Memory Allocation Schemes

(Practice Questions)

Paging (1 to 15):

1. In a system, a logical address space of 64 pages, each of 512 Bytes mapped into physical memory of 1024 frames. Compute: lengths (in bits) of p, d, f, logical address and physical address.
2. Logical address space is 36-bit, physical address space is 32-bit and page size is 4KB. Determine lengths of p, f and d, format of logical and physical addresses, maximum number of pages per process and maximum number of frames in the system.
3. Logical address space is 48-bit, physical address space is 32-bit and page size is 4KB. Determine the lengths of p, f and d, format of logical and physical addresses, maximum number of pages per process, maximum number of frames in the system, page table entry size (PTES) and the size of page table.
4. Consider a system that allows maximum 2M pages per process with 2KB page size. Determine the length of the logical address.
5. In a system with 24-bit physical address space supports the frame size of exactly 512 bytes. Calculate the page table entry size (PTES) and the length of physical address.
6. Consider a system with 48-bit logical address space and 64 GB physical memory. It allows 4KB frame size. If the PTES is 4 bytes then what will be the size of page table? Would there be a hierarchical page table? If yes then find the number of levels? Also determine the format of logical and physical addresses.
7. Consider a system that supports 2KB page size and a single level page table. Let the Page Table Base Register (PTBR) contains 14000 and PTES is 4 bytes. What will be the starting address of the last entry of the page table? Also determine the maximum number of pages per process supported by the system. What will be the format of the logical address?
8. Consider a system that supports 2KB page size and a 2-level page table. Let the Page Table Base Register (PTBR) contains 14000 and PTES is 4 bytes. What will be the starting address of the last entry of the outer page table? Also determine the maximum number of pages per process supported by the system. What will be the format of the logical address? Can you calculate the number of frames with the provided information?
9. Consider a system with 36-bit logical address space that supports 4KB page size. Available physical memory is 64MB. Calculate p, d, f, PTES, maximum number of levels of the page table per process, format of logical and physical addresses.
10. Consider a system with 36-bit logical address space that supports 4KB page size. Available physical memory is 64GB. Operating system running on this system supports 32-bit process IDs. Calculate p, d, f, PTES in an inverted page table, size of the inverted page table, and the format of logical and physical addresses.
11. Let 14000 is a logical address, in which page does it exist if the page size is 1KB?

12. In a system, with 2048 MB RAM and a 4 KB page size, how many entries will there be in an inverted page table?
13. In a system, with 34-bit logical and 32-bit physical address and a 16 KB page size, how many entries will there be in the page table?
14. In a System, each 34-bit logical address is viewed as a 22-bit page identifier (p) and 12 bit offset (d). Size of physical address is 32-bit. Compute page size, PTES, number of pages, size of page table and size of (page table if we use an) inverted page table
15. Consider a virtual address 40808. Compute the virtual page number and offset for a 4 KB page.

Segmentation (16, 17):

16. Use the following segment table and compute the physical addresses for the given logical addresses:

Logical Address
0,0
2,120
6,10
3,444
5,3399
4,1200
0,99

Segment #	Length	Base
0	100	12000
1	1200	12100
2	190	13300
3	444	15500
4	19308	18008
5	3400	5000

17. Consider the segment table above, if Segment Table Base Register (STBR) contains 36500 and segment table entry size (STES) is 64-bit then what will be the size of segment table? Also compute the address of the last entry.

Paged Segmentation (18 to 20):

18. How many page tables will be constructed for the process whose segments are shown in the above segment table?
19. Consider the segment table above, if the system implements paged-segmentation with the page size of exactly 2KB then compute the page number and offset for the logical address of <4, 12765>. Also compute the number of pages in segment 4 and the address of the 3rd entry in the page table (suppose PTES is 4B).
20. Consider a logical address in paged segmentation <16, 7865>, where 16 is segment number and 7865 is offset. Suppose segment 16 has length of 15690 bytes. Page size is 1KB. Compute:
    a. No of pages in segment no. 16.
    b. In which page the above said offset will reside?
    c. How would you represent the above address in <s, p, d`> format?
    d. Let the page p is stored at frame 30, what would be the physical address?

21. If the hit ratio to a TLB is 80%, and it takes 15 nanoseconds to search the TLB and 150 nanoseconds to access the main memory, then what must be the Effective Memory Access Time in nanoseconds?
22. If the hit ratio to register is 30% and hit ratio to TLB is 50%, and it takes 1 and 10 nanoseconds to search the register and the TLB respectively and 150 nanoseconds to access the main memory, then what must be the Effective Memory Access Time in nanoseconds?
23. A machine has a 32 bit address space and an 16 KB page. The page table is entirely in the memory, with one 32 bit word per entry. When a process starts, the page table is copied to the cache from memory, at one word every 100 nsec. If each process runs for 100 msec (including the time to load the page table), what fraction of the CPU time is devoted to loading the page tables?
24. A swapping system eliminates holes by compaction. Assuming a random distribution of many holes and many data segments and a time to read or write a 32 bit memory word of 10 nsec, about how long does it take to compact 128 MB? For simplicity, assume that word 0 is part of a hole and that the highest word in memory contains valid data.
25. Consider a system with Memory Access Time of 100 nanosecond. Average Page fault service time is 10 ms. Compute Effective Access Time and discuss how much the system will be slowed down if one out of 10000 accesses causes a page fault. Also compute the page fault rate (p), if we want a slow down by less than 10%.