1 Explain the difference between internal and external fragmentation.

2. Given five memory partitions of 100 KB, 500 KB, 200 KB, 300 KB, and 600 KB (in order), how would each of the first-fit, best-fit, and worst-fit algorithms place processes of 212 KB, 417 KB, 112 KB, and 426 KB (in order)? Which algorithm makes the most efficient use of memory?

3. Most systems allow programs to allocate more memory to its address space during execution. Data allocated in the heap segments of programs are an example of such allocated memory. What is required to support dynamic memory allocation in the following schemes:

a. contiguous-memory allocation
b. pure segmentation
c. pure paging

4. Compare the main memory organization schemes of contiguous memory allocation, pure segmentation, and pure paging with respect to the following issues:

a. external fragmentation
b. internal fragmentation
c. ability to share code across processes

5. On a system with paging, a process cannot access memory that it does not own; why? How could the operating system allow access to other memory? Why should it or should it not?

6 Assuming a 1 KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):

a. 2375
b. 19366
c. 30000
d. 256
e. 16385

7. Consider a logical address space of 32 pages with 1024 words per page; mapped onto a physical memory of 16 frames.

a. How many bits are required in the logical address?
b. How many bits are required in the physical address?

8 Consider a computer system with a 32-bit logical address and 4-KB page size. The system supports up to 512 MB of physical memory. How many entries are there in each of the following?

a. A conventional single-level page table?
b. An inverted page table?

9. Consider a paging system with the page table stored in memory.

a. If a memory reference takes 200 nanoseconds, how long does a paged memory reference take?
b. If we add TLBs, and 75 percent of all page-table references are found in the TLBs, what is the effective memory reference time? (Assume that finding a page-table entry in the TLBs takes zero time, if the entry is there.) **Answer:**

a. 400 nanoseconds: 200 nanoseconds to access the page table and 200 nanoseconds to access the word in memory.

b. Effective access time = 0.75 × (200 nanoseconds) + 0.25 × (400 nanoseconds) = 250 nanoseconds.

10. Why are segmentation and paging sometimes combined into one scheme?

11 What is the purpose of paging the page tables?

12 Consider a system that uses pure demand paging:

a. When a process first starts execution, how would you characterize the page fault rate?
b. Once the working set for a process is loaded into memory, how would you characterize the page fault rate?
c. Assume a process changes its locality and the size of the new working set is too large to be stored into available free memory. Identify some options system designers could choose from to handle this situation?

13.What is the copy-on-write feature, and under what circumstances is it beneficial? What hardware support is required to implement this feature?

14.A certain computer provides its users with a virtual-memory space of 2^{32} bytes. The computer has 2^{18} bytes of physical memory. The virtual memory is implemented by paging, and the page size is 4096 bytes. A user process generates the virtual address 11123456. Explain how the system establishes the corresponding physical location. Distinguish between software and hardware operations.

15.Assume we have a demand-paged memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty page is available or the replaced page is not modified, and 20 milliseconds if the replaced page is modified. Memory access time is 100 nanoseconds.

Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

16.When a page fault occurs, the process requesting the page must block while waiting for the page to be brought from disk into physical memory. Assume that there exists a process with five user-level threads and that the mapping of user threads to kernel threads is many to one. If one user thread incurs a page fault while accessing its stack, will the other user threads belonging to the same process also be affected by the page fault—that is, will they also have to wait for the faulting page to be brought into memory? Explain.

17.Consider the following page reference string:

7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0 , 1.

18 Assume that you are monitoring the rate at which the pointer in the clock algorithm (which indicates the candidate page for replacement) moves. What can you say about the system if you notice the following behavior:

a. pointer is moving fast
b. pointer is moving slow

19. Discuss situations in which the LFU page-replacement algorithm generates fewer page faults than the LRU page-replacement algorithm. Also discuss under what circumstances the opposite holds.

20.Discuss situations in which the MFU page-replacement algorithm generates fewer page faults than the LRU page-replacement algorithm. Also discuss under what circumstances the opposite holds.

21.Suppose that a machine provides instructions that can access memory locations using the one-level indirect addressing scheme. What is the sequence of page faults incurred when all of the pages of a program are currently nonresident and the first instruction of the program is an indirect memory load operation?

22.What happens when the operating system is using a per-process frame allocation technique and only two pages are allocated to this process?

23.Suppose that your replacement policy (in a paged system) is to examine each page regularly and to discard that page if it has not been used since the last examination. What would you gain and what would you lose by using this policy rather than LRU or second-chance replacement?

24.A page-replacement algorithm should minimize the number of page faults. We can achieve this minimization by distributing heavily used pages evenly over all of memory, rather than having them compete for a small number of page frames. We can associate with each page frame a counter of the number of pages associated with that frame. Then, to replace a page, we can search for the page frame with the smallest counter.

a. Define a page-replacement algorithm using this basic idea. Specifically address these problems:
    i. What the initial value of the counters
    is ii. When counters are increased iii.
    When counters are decreased iv. How the
    page to be replaced is selected
b. How many page faults occur for your algorithm for the following reference string, for four page frames?

    1, 2, 3, 4, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 9, 5, 4, 5, 4, 2.

c. What is the minimum number of page faults for an optimal page replacement strategy for the reference string in part b with four page frames?

25.What is the cause of thrashing? How does the system detect thrashing? Once it detects thrashing, what can the system do to eliminate this problem?