Chapter 4 exercises

1. Name two differences between logical and physical addresses.

2. Consider a system in which a program can be separated into two parts: code and data. The CPU knows whether it wants an instruction (instruction fetch) or data (data fetch or store). Therefore, two base – limit register pairs are provided: one for instructions and one for data. The instruction base –limit register pair is automatically read-only, so programs can be shared among different users. Discuss the advantages and disadvantages of this scheme.

3. Why are page sizes always powers of 2?

4. Consider a logical address space of 64 pages of 1,024 words each, mapped onto a physical memory of 32 frames.
a. How many bits are there in the logical address?
b. How many bits are there in the physical address?

5. What is the effect of allowing two entries in a page table to point to the same page frame in memory? Explain how this effect could be used to decrease the amount of time needed to copy a large amount of memory from one place to another. What effect would updating some byte on the one page have on the other page?

6. Describe a mechanism by which one segment could belong to the address space of two different processes.

7. Sharing segments among processes without requiring that they have the same segment number is possible in a dynamically linked segmentation system.
a. Define a system that allows static linking and sharing of segments without requiring that the segment numbers be the same.
b. Describe a paging scheme that allows pages to be shared without requiring that the page numbers be the same.

8. Explain the difference between internal and external fragmentation.

9. Given six memory partitions of 300 KB , 600 KB , 350 KB , 200 KB , 750 KB , and 125 KB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115 KB , 500 KB , 358 KB , 200 KB , and 375 KB (in order)? Rank the algorithms in terms of how efficiently they use memory.

10. Most systems allow a program to allocate more memory to its address space during execution. Allocation of data in the heap segments of programs is 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

11. Compare the 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

12. 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?

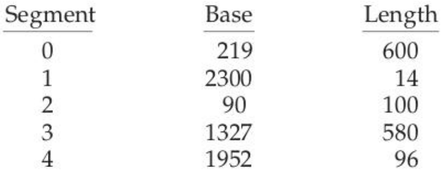
13. Compare paging with segmentation with respect to how much memory the address translation structures require to convert virtual addresses to physical addresses.

14. Assuming a 1- KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):
a. 3085
b. 42095
c. 215201
d. 650000
e. 2000001

15. The BTV operating system has a 21-bit virtual address, yet on certain embedded devices, it has only a 16-bit physical address. It also has a 2- KB page size. How many entries are there in each of the following?
a. A conventional, single-level page table
b. An inverted page table

16. Consider a logical address space of 256 pages with a 4-KB page size, mapped onto a physical memory of 64 frames.
a. How many bits are required in the logical address?
b. How many bits are required in the physical address?

17. Consider the following segment table:



Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

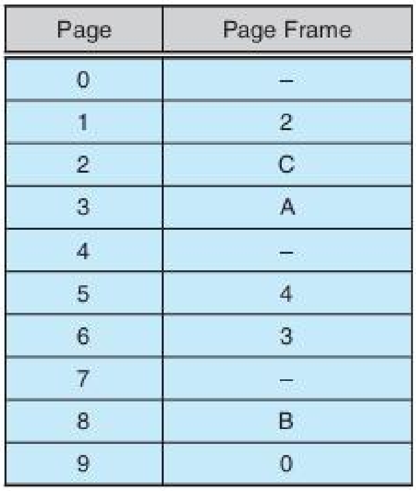
What are the physical addresses for the following logical addresses?
a. 0,430
b. 1,10
c. 2,500
d. 3,400
e. 4,112

18. What is the purpose of paging the page tables?

19. Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs.

20. Assume that you have a page-reference string for a process with m frames (initially all empty). The page-reference string has length p, and n distinct page numbers occur in it. Answer these questions for any page-replacement algorithms:
a. What is a lower bound on the number of page faults?
b. What is an upper bound on the number of page faults?

21. Consider the page table shown in the following figure for a system with 12-bit virtual and physical addresses and with 256-byte pages. The list of free page frames is D, E, F (that is, D is at the head of the list, E is second, and F is last).



Page	Page Frame
0	–
1	2
2	C
3	A
4	–
5	4
6	3
7	–
8	B
9	0

Convert the following virtual addresses to their equivalent physical addresses in hexadecimal. All numbers are given in hexadecimal. (A dash for a page frame indicates that the page is not in memory.)

- 9EF
- 111
- 700
- 0FF

22. Consider the following page-replacement algorithms. Rank these algorithms on a five-point scale from “bad” to “perfect” according to their page-fault rate. Separate those algorithms that suffer from Belady’s anomaly from those that do not.

a. LRU replacement
b. FIFO replacement
c. Optimal replacement
d. Second-chance replacement

23. Consider the two-dimensional array A:

```
int A[][] = new int[100][100];
```

where A[0][0] is at location 200 in a paged memory system with pages of size 200. A small process that manipulates the matrix resides in page 0 (locations 0 to 199). Thus, every instruction fetch will be from page 0. For three page frames, how many page faults are generated by the following array-initialization loops? Use LRU replacement, and assume that page frame 1 contains the process and the other two are initially empty.

```
    a. for (int j = 0; j < 100; j++)
            for (int i = 0; i < 100; i++)
               A[i][j] = 0;
    b. for (int i = 0; i < 100; i++)
            for (int j = 0; j < 100; j++)
               A[i][j] = 0;
```

24. Consider the following page reference string:
1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.
How many page faults would occur for the following replacement algorithms, assuming one, two, three, four, five, six, and seven frames? Remember that all frames are initially empty, so your first unique pages will cost one fault each.

- LRU replacement
- FIFO replacement
- Optimal replacement

25. Consider a demand-paged computer system where the degree of mul-tiprogramming is currently fixed at four. The system was recently measured to determine utilization of the CPU and the paging disk. Three alternative results are shown below. For each case, what is happening? Can the degree of multi-programming be increased to increase the CPU utilization? Is the paging helping?
a. CPU utilization 13 percent; disk utilization 97 percent
b. CPU utilization 87 percent; disk utilization 3 percent
c. CPU utilization 13 percent; disk utilization 3 percent

26. A simplified view of thread states is Ready, Running, and Blocked, where a thread is either ready and waiting to be scheduled, is running on the processor, or is blocked (for example, waiting for I/O). Assuming a thread is in the Running state, answer the following questions, and explain your answer:
a. Will the thread change state if it incurs a page fault? If so, to what state will it change?
b. Will the thread change state if it generates a TLB miss that is resolved in the page table? If so, to what state will it change?
c. Will the thread change state if an address reference is resolved in the page table? If so, to what state will it change?

27. A certain computer provides its users with a virtual memory space of 2^32 bytes. The computer has 2^22 bytes of physical memory. The virtual memory is implemented by paging, and the page size is 4,096 bytes. A user process generates the virtual address 11123456. Explain how the system establishes the corresponding physical location. Distinguish between software and hardware operations.

28. Assume that you are monitoring the rate at which the pointer in the clock algorithm moves. (The pointer indicates the candidate page for replacement.) What can you say about the system if you notice the following behavior:
a. Pointer is moving fast.
b. Pointer is moving slow.

29. 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?