Some of the answers to these questions can be found using the lecture slides, the recommended textbooks or other sources (question marked with *). Some questions may have more than one possible answer, or be more or less open for discussion. Note that no answers (solutions) will be given to these questions, but if help is needed the assistants will be available to answer questions. The concepts marked with yellow are important and should be fully understood.

Memory Management

1. What is the difference between a physical address and a virtual address?

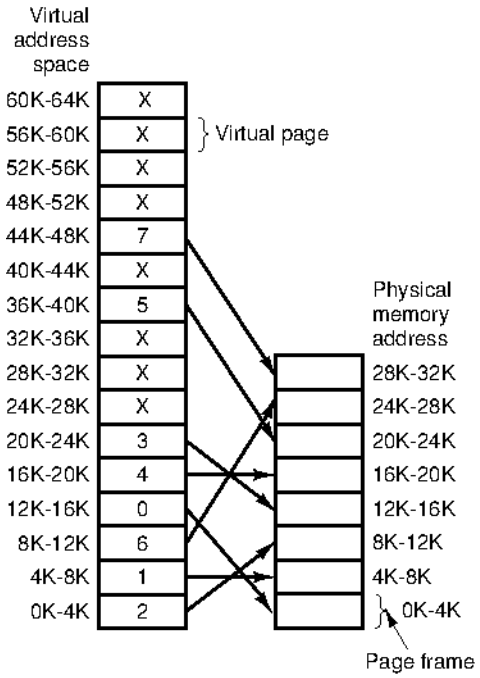

Figure 1: The relation between virtual addresses and physical memory addresses is given by the page table

2. Using the page table of Fig. 1, give the physical address corresponding to each of the following virtual addresses:
   (a) 20
   (b) 4100
   (c) 8300
3. Multi-level page tables:

(a) Give an example for a two-level page table scheme to be used with 32 bit virtual addresses. Specifiy the amount and sizes of page tables and the page sizes.

(b) For 64 bit virtual addresses so called inverted page tables are used instead of the previously discussed multi-level page tables. Why is that? Give an example for such a scheme using inverted page tables!

(c) Describe how the lookup times of inverted page tables can be improved using a hashed page table design.

4. A computer's main memory is divided into four page frames. The time of loading, time of last access, and the R and M bits for each page are shown below (the times are in clock ticks):

Page	Loaded	Last referenced	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

Table 1: Four page frames in a simple computer

(a) Which page will **NRU** replace?

(b) Which page will **FIFO** replace?

(c) Which page will **LRU** replace?

(d) Which page will **second chance** replace?

5. A computer provides each process with 65536 bytes of address space divided into pages of 4096 bytes. A particular program has a text size of 32768 bytes, a data size of 16386 bytes and a stack size of 15870 bytes. Will this program fit in the address space? What if the page size is 512 bytes? We assume that a page may not contain parts of two different segments. (Why is that a valid assumption?)

6. Segmentation is an addressing scheme that provides programmers with many independent address spaces. Describe the segmentation concept and its benefits over a one-dimensional address space.

7. Segmentation and paging are sometimes combined into one scheme. What are the expected benefits?

8. Consider the segment table described in the Table 2. What are the physical addresses for the following logical addresses (given as $< segment, address >$ pairs)?

(a) $< 0, 430 >$

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

Table 2: A segment table

(b) $< 1, 10 >$

(c) $< 2, 500 >$

(d) $< 3, 400 >$

(e) $< 4, 112 >$

9. Discuss the differences between paging and segmentation regarding the aspects listed below!

   (a) Awareness to programmer?

   (b) Number of address spaces?

   (c) Can total address space exceed the size of memory?

   (d) Can procedures and data be distinguished and separately protected?

   (e) Can tables whose size fluctuates be accommodated easily?

   (f) Is sharing of procedures between users facilitated?

   (g) Why was this technique invented?