Exercises on memory management

1. Assume the base and limit registers contain the same value, 16,384. Is this just an accident, or are they always the same? It is just an accident, why are they the same in this case?

2. A system implements a paged virtual address space for each process using a one-level page table. The maximum size of an address space is 16 megabytes. The page table for the running process includes the following entries:

0	4
1	8
2	16
3	17
4	9

   The page size is 4KB and the maximum physical memory size of the machine is 2 megabytes.

   (a) How many bits are required for each page table entry?

   (b) What is the maximum number of entries in a page table?

   (c) How many bits are there in a virtual address?

   (d) To which physical address will the virtual address 1524 translate to?

   (e) Assuming that frame 0 starts at the physical address 0, that consecutive frame numbers have consecutive physical addresses (the first address of frame $i + 1$ follows immediately the last address of frame i), then, which virtual address will translate to physical address 65536?

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

   (a) If a memory reference takes 400ns, how long does it takes to fetch data through the page table?

   (b) If we add a TLB, and 85% of all page-table references are found in the TLB (i.e. hit rate is 85%), what is the effective memory reference time? (Assume that finding a page-table entry in the associative registers takes zero time, if the entry is there.)

4. Consider a paging system with the page table stored in memory. Assume the overhead to read an entry in the page table is 5ns (access to the page table is 5 ns). To reduce this overhead, a TLB is installed, where a look up in the TLB cost 1ns. What hit rate is needed to reduce the average overhead to access the page table to 2ns?

5. A system implements a paged virtual address space using a two-level page table. The maximum size of a process address space is 2^{27} bytes (about 134 megabytes). The page table for a running process includes the following entries:

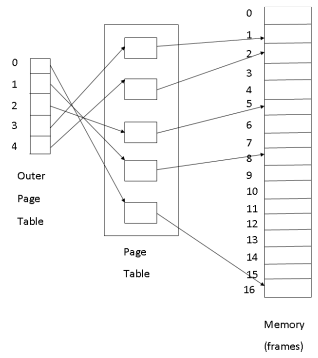

Figure 1: Two levels page table

The frame size is 2^{10} bytes (1KB) and the physical memory size is 2^{16} bytes (65 KBs).

(a) How many bits are required for each page table entry?

(b) What is the maximum number of entries in the inner page table?

(c) How many bits there is in virtual addresses generated by the CPU?

(d) What is the maximum number of entries of the inner page table that can be stored in one frame?

(e) How many frames are needed to store a maximum size process?

(f) How many entries there will be in the outer page table?

(g) To which physical address will the virtual address 9812 translate to?

(h) Assuming that frame 0 starts at the physical address 0, that consecutive frame numbers have consecutive physical addresses (the first address of frame $i + 1$ follows immediately the last address of frame i), then, which virtual address will translate to physical address 33768?

6. Assume the physical memory is 64 bits addressable. Each frame is 16 KB or 2^{14} bytes. We have a process of 2^{64} bytes. The total number of pages in the process is $2^{64}/2^{14} = 2^{50}$. Rather than using a multilevel page table, we will use a hashed page table. The hashed page table has 2^{14} entries.

(a) What is the size of the hashed table?

(b) How many bits of a virtual address are used to address in the hashed page table?

(c) Assume the bits of a virtual address that are used to index into the hashed table convert into the integer 337,656,234. What is the corresponding entry in the hash table?