CSCI 2310 – Address Translation Exercises

Consider a machine with a total memory size of 256 bytes, a page size of 16 bytes, and the (partial) page table shown below:

Page	Frame
0	2
1	6
2	11
3	9
4	5
5	0
6	4
…	…

1. How many possible pages can a process have?

2. How many bits are in a virtual address?

3. How many bits of the virtual address are for **p** (page number) and for **d** (page offset)?

4. Translate virtual address 24 to its physical memory address. Do this in two ways: using the computationally expensive non-binary method, and then using the efficient binary method.

So far we've assumed that you can address every byte of memory individually. In general, this isn't true — computers address chunks of memory, called **words**, instead of individual bytes. A word of memory is like a single logical piece of data from the CPU's perspective, and on modern machines typically consists of either 32 bits (4 bytes) or 64 bits (8 bytes). When you say that you have a '64-bit architecture', or just a 64-bit machine, this means that your machine addresses 64-bit words of memory. A word is the smallest addressable quantity of memory.

5. Assume we have the same memory system described above, but are now using a 32-bit architecture (i.e., 4 byte words) instead of an 8-bit architecture (i.e., 1 byte words). Repeat questions 1 through 4 given this memory architecture. Remember that addresses are in terms of words, not bytes. What set of bytes in physical memory does virtual address 24 reference?

6. Translate virtual address 13 to its physical memory address (again using both methods), and give the set of physical memory bytes that this corresponds to.