

Lab 4

Part 1

For this lab, you can use whatever resource (but do not copy from your peers, please work individually) and calculator, please answer the following questions:

1. How many values can one 2-base digit take? List all of them.
2. How many values can one 10-base digit take? List all of them.
3. How many values can 7 bits represent? DO NOT list all of them.
4. Recall, look up or come up with a formula for a generic case: how many values can n m-valued bits represent?
5. If our machine is byte-addressable (i.e., each byte must have its own address), how many addresses do we need to have a table with pointers to every byte in 1 Megabyte?
6. What if our machine was bit-addressable?
7. Consider you are using an array of pointers, each entry of which is sixty-four bits (or 8 bytes) to have a pointer to each individual byte from the question 5, how much memory would this array occupy?
8. Consider you are using an array of pointers, each entry of which is sixty-four bits (or 8 bytes) to have a pointer to each individual bit from the question 6, how much memory would this array occupy?

Part 2

1. Write a short C program (or work out on paper with calculator, for all the entries in input.txt) that would translate logical addresses (in a 32-bit machine) into physical addresses (using the paging mechanism we discussed in class), with 12-bit page offset and 20-bit page number. If the page was not loaded into the physical memory, you are to use the next free frame, however you must reuse the pages that already have frames assigned to them before. For this lab, consider you have all $2^{(\text{WORDSIZE}-\text{OFFSETBITS})}$ frames available. Refer to the starter code provided with the lab.
2. How many megabytes are required to store 2^{20} integers?

Submission instructions

If you have the C code for Part 2, embed the answers to Part 1 and all physical addresses for each logical address in Part II, as well as the answer to 2.2, as comments into the C file.

If you do not have the C code submit a text document with all the answers to Part I and all physical addresses for each logical address in Part II as well as the answer to 2.2.