CSE 378 Winter 2010 Homework 4

Due Date: Saturday, 3/13 at 11 pm. No late days.

You can either drop it off during Friday's lab hours, or submit the file via the Catalyst drop box.

Change 3/10: Question 2(d) is now optional.

1. As we saw in class, page tables require fairly large amounts of memory, even if most of the entries are invalid. One solution is to use a hierarchy of page tables. The virtual page number can be broken up into two pieces, a "page table number" and a "page table offset," which is described in the figure below.

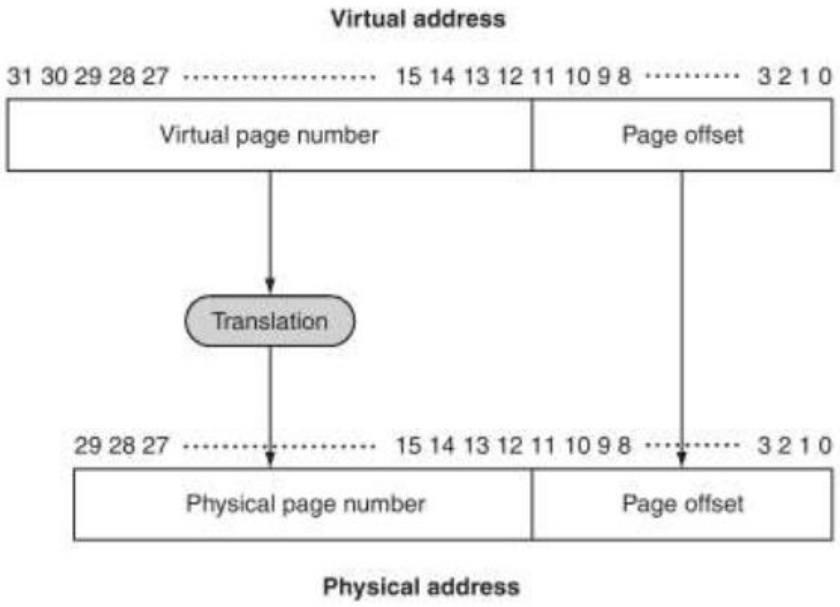

FIGURE 7.20 Mapping from a virtual to a physical address. The page size is 2^{12} = 4 KB. The number of physical pages allowed in memory is 2^{18}, since the physical page number has 18 bits in it. Thus, main memory can have at most 1 GB, while the virtual address space is 4 GB.

(From Hennessy and Patterson's **Computer Organization and Design, 3rd Edition**, pg 513)

The page table number can be used to index a first-level page table that provides a physical address for a second-level page table. The page table offset is used to index into the second-level page table to retrieve the physical page number.
One way to arrange such a scheme is to have the second-level page tables occupy exactly one page of memory.

Assuming a 32-bit virtual address space with 4KB pages and 4 bytes per page table entry, how many bytes will each program need to use to store the first-level page table (which must always be in memory)? Provide information on how to decode the address (i.e. what bits for the page table number, pagetable offset and physical page offset). Explain your solution.

2. **Virtual Memory and TLBs**:

   a. Given the following stream of 32-bit virtual addresses, update the TLB and Page Table shown below. Assume 4 KB pages, a four-entry fully-associative TLB, and true LRU replacement. If pages must be brought in from disk, increment to the next largest page number.

Addresses: 4095, 31272, 15789, 15000, 7193, 4096, 8912

TLB

Valid	Tag	Physical Page
1	11	12
1	7	4
1	3	6
0	4	9

Page Table

Valid	Physical page or in Disk
1	5
0	Disk
0	Disk
1	6
1	9
1	11
0	Disk
1	4
0	Disk
0	Disk
1	3
1	12
0	Disk
1	3
1	7
0	Disk

   b. Repeat the question, but this time assume that pages are 16 KB (instead of 4 KB).

   c. What would be an advantage of having a larger page size? Are there any disadvantages?

   d. (optional) Show the final contents of the TLB if it is two-way set associative (and pages are 4 KB).

3. **I/O:** Given the following two applications 1) A database for storing large image files, 2) a video game, answer these questions about appropriate disk usage:

   a. Is decreasing the sector size likely to help or hurt this application? (explain)

   b. Would increasing disk rotation speed help or hurt this application?(explain)

4. **I/O**: Assume a hard disk has the following specifications:
   - An average seek time of 11 ms
   - A 7200 RPM rotational speed
   - A 30MB/s average transfer rate
   - 3 ms of overheads for making a request

   a. Given that sectors are 1024 bytes, what is the average time to read or write a 1024-byte sector?

   b. Now assume that sectors are 2048 bytes. What is number of random 2048-byte sectors that can be read in one second?

5. **Performance**: Our two favorite programs have the following distribution of instructions of type A, B, C, and D, and our current processor has the following CPI for those instructions:

	A	B	C	D
Fraction of program 1	20%	15%	55%	10%
Fraction of program 2	30%	30%	25%	15%
CPI	5	4	3	2

   a. We have heard there is a new processor that implements Type B instructions blazingly fast – each one now executes in only one cycle. What is maximum performance improvement we might expect from this new processor for program 1 and program 2?

   b. What would be our performance improvement if instead we just made our original processor run twice as fast? For program 1? For program 2?