CS323 – Exercises
Week 6
28 March 2019

Problem 1:
A computer system has a 36-bit virtual address space with a page size of 8 KB, 4 bytes per page table entry and uses single-level paging.

1. How many pages are in the virtual address space?
2. What is the maximum size of addressable physical memory in this system?

Answer:

1. A 36-bit address can address 2^{36} bytes in a byte addressable machine. Since the size of a page 8K bytes (2^{13}), the number of addressable pages is $2^{36}/ 2^{13} = 2^{23}$
2. With 4-byte entries in the page table we can reference 2^{32} pages. Since each page is 2^{13} B long, the maximum addressable physical memory size is $2^{32} * 2^{13} = 2^{45}$ B (assuming no protection bits are used).

Problem 2:
An x86-64 computer has a page table size of 4096 bytes, its page size is the same. A page table entry takes 8 bytes. Its CPU implements 48 bits of virtual address space. A multi-level page table is used because each table must be contained within a page. How many levels are required?

Answer:
Since page table must fit in a page, page table size is 4096 bytes and each entry is 8 bytes thus a table holds 4096/8=512 entries. To address these 512 entries, it requires 9 bits.
The total number of bits available to encode the entry for each page level is:
bits of virtual address space – page offset bits = $48 - \log_2(4096) = 48 - 12 = 36$ bits.
Thus, the total number of levels required is 36/9=4.

Problem 3:
We have a 32-bit address space, 4 KB page sizes and 2 levels of page tables with the bits arranged in the following way: d = 12, p1 = 10, p2 = 10. Only the upper 10 MB and lower 2 MB are used. How many page table entries are required?

Answer:
We have 2-level page table with p1 = 10 and p2 = 10

- For the first level we have 2^{10} = 1K PTEs
- For the second level: **3**$*2^{10}$ (upper) + **1**$*2^{10}$ (lower) = $4*2^{10}$ = 4K PTEs

Explanation:
For the upper 10MB:
$10 \text{ MB} / 4\text{KB} = (10 * 2^{20})/ (4 * 2^{10})= 2.5 *2^{10} = 2.5 \text{ K}$

So, we would need **3 K** to fit 2.5 K.

For the lower 2MB:

2MB / 4KB = $2^{20} / 2^{12} = 2^{8} = 256$

So, we would need only **1 K** (1024) to fit 2^{8} (256)

So, the answer is a total of 5K PTEs.

Problem 4:

The hardware for computer Dumbo was badly designed. The hardware designers included a valid bit in each page table entry, with the usual meaning, but they forgot to include a reference bit. Can you think of a way to implement a reference bit in software? You may assume that each page table entry has an unused bit field that you can read and write.

Answer:

For each entry:

- Set valid bit to 0
- Access will cause page fault
- Set software reference bit to 1
- Set valid bit to 1

Problem 5:

A TLB has a hit rate of 95% and the TLB penalty T_{miss} = 150 cycles. Assume that, on a TLB hit, address translation takes $T_{hit} = 0$. Compute the average time in cycles required for an address translation.

Answer:

Avg. time for address translation = $T_{hit} * P_{hit} + T_{miss} * P_{miss} = 0 * 0.95 + 150*0.05 = 7.5$ cycles

We get a reduction of 20X over the translation time without the TLB.